МИНИСТЕРСТВО ОБРАЗОВАНИЯ РОСТОВСКОЙ ОБЛАСТИ

ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ПРОФЕССТОНАЛЬНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ РОСТОВСКОЙ ОБЛАСТИ «РОСТОВСКИЙ-НА-ДОНУ КОЛЛЕДЖ СВЯЗИ И ИНФОРМАТИКИ»

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

по учебной практике

по специальности
09.02.13 Интеграция решений с применением технологий искусственного интеллекта
(базовой подготовки)

г. Ростов-на-Дону, 2026 г.

РАССМОТРЕНО
на заседании цикловой комиссии
Программирование
Протокол № 1 от 31.08.2026 года
Председатель ЦК Сулавко С.Н.

УТВЕРЖДАЮ:
Зам. директора по УМР
И.В. Подцатова
«16» ноября 2026 г.

Комплект контрольно-оценочных средств по учебной практике разработан в соответствии с рабочей программой по учебной практике, разработанной в 2026 году по специальности 09.02.13 Интеграция решений с применением технологий искусственного интеллекта.

.

Организация-разработчик: государственное бюджетное профессиональное образовательное учреждение Ростовской области «Ростовский-на-Дону колледж связи и информатики»

Разработчики:

Кошкина А.А., Черевко А.Д., Болгов Д.Р., Сулавко С.Н. преподаватели государственного бюджетного профессионального образовательного учреждения Ростовской области «Ростовский-на-Дону колледж связи и информатики»

Рецензент от работодателя:

СОДЕРЖАНИЕ

1.Общие положения ……....**Ошибка! Закладка не определена.**

2. Результаты освоения модуля, подлежащие проверке……..…4

3. Оценка освоения учебной практики……..……..19

4. Контроль приобретения практического опыта. Оценка по учебной практике……....226

5. Контрольно-оценочные материалы для учебной практики……...……..240

1. Общие положения

Результатом освоения учебной практики является готовность обучающегося к выполнению видов профессиональной деятельности:

- Разработка кода для искусственного интеллекта
- Администрирование баз данных
- Обучение готовых моделей искусственного интеллекта

и сформированность профессиональных и общих компетенций.

Формой аттестации по учебным практикам является зачет.

1.Формы промежуточной аттестации по профессиональному модулю

Таблица 1

Код УП	**Объем, ак.ч.**	**Форма проведения учебной практики** (концентрированно/ рассредоточено)	**Курс / семестр**	**Форма промежуточной аттестации**
УП.01	108	концентрировано	3/6	зачет
УП.02	144	концентрировано	3/5	зачет
УП.03	108	концентрировано	4/8	зачет
Всего УП	360	X	X	X

2. Результаты освоения, подлежащие проверке.

2.1. Профессиональные и общие компетенции

В результате контроля и оценки по учебной практике осуществляется комплексная проверка следующих профессиональных и общих компетенций:

Таблица 1

Код ОК / ПК	**Наименование ОК / ПК**
ОК 01	Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам
ОК 02	Использовать современные средства поиска, анализа и интерпретации информации и информационные технологии для выполнения задач профессиональной деятельности

ОК 05	Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста
ОК 09	Пользоваться профессиональной документацией на государственном и иностранном языках
ПК 1.1	Формировать алгоритмы разработки программных модулей в соответствии с техническим заданием
ПК 1.2	Разрабатывать программные модули в соответствии с техническим заданием
ПК 1.3	Оформлять программный код в соответствии с техническим заданием
ПК 1.4	Использовать систему контроля версий программного кода с учетом обеспечения возможности организации групповой разработки
ПК 1.5	Выполнять отладку программных модулей с использованием специализированных программных средств
ПК 1.6	Выполнять тестирование программного кода
ПК 1.7	Составлять тестовые сценарии
ПК 1.8	Разрабатывать программные модули с использованием алгоритмов и методов искусственного интеллекта
ПК 1.9	Разрабатывать программный код мобильных приложений в соответствии с техническим заданием и утвержденными спецификациями
ПК 1.10	Осуществлять оптимизацию программного кода мобильных приложений в соответствии с техническим заданием, оценивая трудоемкость и сроки выполнения работ
ПК 1.11	Документировать логику и поведение интерфейса: описывать взаимосвязи, взаимодействия и возможные состояния элементов графического пользовательского интерфейса
ПК 1.12	Использовать специализированные инструменты для прототипирования и проектирования графических пользовательских интерфейсов
ПК 1.13	Выполнять разработку программных модулей
ПК 1.14	Выполнять тестирование программных модулей и ИИ-решений
ПК 1.15	Осуществлять интеграцию программных модулей и компонентов программного обеспечения
ПК 1.16	Разрабатывать программные интерфейсы и процедуры сборки и развертывания программного обеспечения
ПК 1.17	Осуществлять миграцию, преобразование и конвертацию данных программных систем
ПК 1.18	Разрабатывать и интегрировать серверные программные модули для ИИ-решений
ПК 1.19	Разрабатывать и настраивать программные интерфейсы взаимодействия с ИИ-компонентами и внешними системами
ПК 1.20	Выполнять развертывание, тестирование и обеспечение работоспособности серверных компонентов ИИ-решений
ПК 1.21	Осуществлять защиту данных в мобильных приложениях
ПК 1.22	Интегрировать информационную систему с существующими информационными системами заказчика
ПК 1.23	Выполнять непрерывную интеграцию и непрерывное развертывание программного обеспечения в процессе разработки
ПК 1.24	Управлять конфигурациями и инфраструктурой
ПК 1.25	Управлять версиями и кодом
ПК 2.1	Выявлять проблемы, возникающие в процессе эксплуатации баз данных
ПК 2.2	Осуществлять процедуры администрирования баз данных

ПК 2.3	Проводить аудит систем безопасности баз данных с использованием регламентов по защите информации
ПК 2.4	Формировать требования хранилищ банка данных для обучения
ПК 2.5	Подготавливать данные для базы знаний
ПК 2.6	Осуществлять управление базами данных
ПК 2.7	Осуществлять автоматизацию баз данных
ПК 2.8	Разрабатывать объекты баз данных NoSQL
ПК 2.9	Осуществлять конвейерную обработку данных для систем ИИ
ПК 2.10	Осуществлять мониторинг и оркестрацию конвейеров обработки данных
ПК 2.11	Осуществлять контроль качества данных в конвейерах обработки
ПК 3.1	Осуществлять выбор готовых моделей искусственного интеллекта
ПК 3.2	Формировать сценарии обучения готовых моделей искусственного интеллекта
ПК 3.3	Проводить обучение и последующую калибровку готовых моделей искусственного интеллекта
ПК 3.4	Контролировать результат обучения
ПК 3.5	Оформлять результат проведения процедуры обучения
ПК 3.6	Формировать запросы для работы с искусственным интеллектом с целью визуализации данных
ПК 3.7	Формировать сценарии обучения готовых моделей искусственного интеллекта
ПК 3.8	Интегрировать готовые модели искусственного интеллекта в информационные системы с использованием программных интерфейсов
ПК 3.9	Создавать контент программного продукта средствами нейросетей
ПК 3.10	Разрабатывать блоки программного кода с использованием нейросетей
ПК 3.11	Создавать и управлять концепцией ИТ-продукта
ПК 3.12	Планировать управление требованиями заказчика в проектах любого уровня сложности в области ИТ

2.2 Практический опыт (сформированные умения)

В результате прохождения учебной практики по видам деятельности, предусмотренным ФГОС СПО и запросам работодателей осуществляется комплексная проверка полученного практического опыта (сформированных умений):

Таблица 2

Наименование вида деятельности	**Практический опыт / умения**
Разработка кода для искусственного интеллекта	**Практический опыт:** - Разработки, оптимизации и оценки сложности алгоритмов для ИИ-программ. - Использования библиотек и инструментов для работы с алгоритмами и данными (например: Pandas, NumPy, Scikit-learn). - Применения структур данных (деревья, графы, списки) для реализации алгоритмов. - Разработки модульных ИИ-систем, соответствующих требованиям

	производительности и безопасности. - Внедрения разработанных ИИ-модулей в комплексные программные системы. - Оптимизации кода и работы с интерфейсами для взаимодействия между модулями. - Оформления, документирования и структурирования кода для последующей поддержки. - Использования инструментов статического анализа кода для выявления ошибок и улучшения качества. - Работы с системами документирования кода (например, Doxygen, Sphinx). - Управления проектами с использованием систем контроля версий для организации командной работы. - Разрешения конфликтов при слиянии веток и использования pull request для рецензирования кода. - Настройки процессов CI/CD для автоматического тестирования и развертывания кода. - Отладки программных модулей с использованием пошаговой проверки. - Применения методов логирования и профилирования производительности. - Использования специальных средств для отладки многопоточных программ. - Выполнения статического тестирования программного кода на предмет выявления ошибок/дефектов алгоритмов, в том числе – на наличие обработки исключений - Выполнения тестирования программных модулей в соответствии в тест-планом - Генерирования тестовых данных - Выполнения интеграционного тестирования в соответствии с заданием - Выполнения регрессионного тестирования в соответствии с заданием. - Работы с CI/CD пайплайнами для автоматизации тестирования. - Разработки тестовых сценариев в соответствии с тестовым планом (тестирование производительности, надежности, UI-тестирование), в том числе с применением средств автоматизации проектирования. - Разработки тестовых пакетов и заданий на выполнение тестирования. - Оценки тестовых данных на предмет покрытия строк и покрытия ветвей, выполнения валидации данных. - Автоматизации создания и выполнения тестовых сценариев. - Составления формализованных описаний решений поставленных задач в соответствии с требованиями технического задания или внутренних документов организации - Разработки алгоритмов решения поставленных задач в соответствии с требованиями технического задания или внутренних документов организации - Проверки корректности алгоритмов решения поставленных задач - Создание программного кода в соответствии с техническим заданием (готовыми спецификациями) - Оптимизация программного кода с использованием специализированных программных средств - Оценка и согласование сроков выполнения поставленных задач - Создания программного кода в строгом соответствии с техническим заданием и готовыми спецификациями. - Оптимизации программного кода с использованием профильных инструментов и средств.

	- Оценки и согласования реалистичных сроков выполнения поставленных задач. - Разработки и проверки интерфейсных текстов с учетом глоссария терминов. - Детального описания логики работы, взаимосвязей, взаимодействия и возможных состояний элементов интерфейса. - Разработки и оформления проектной документации на графический интерфейс. - Проектирования графического пользовательского интерфейса согласно утвержденной концепции или готовому образцу. - Создания интерактивных прототипов графического интерфейса. - Эскизирования (набросков) графических пользовательских интерфейсов. - Работы с программами и инструментами прототипирования интерфейсов. - Выявление типов тестовых данных для выполнения тестирования ПО - Выделение тестовых данных из имеющихся машинных файлов - Выделение тестовых данных из внемашинных документов - Генерирование тестовых данных (при необходимости) - Формирование и представление отчетности о подготовке тестовых данных в соответствии с установленными регламентами - Проверка компонентов инструментария и тестируемого ПО на корректное начальное состояние для начала тестирования - Выполнение тестовых процедур на тестовых данных - Сравнение фактического и ожидаемого результатов выполнения тестовых процедур - Формирование и представление отчетности о выполнении процесса тестирования ПО в соответствии с установленными регламентами - Анализа и выявления проблем сопряжения неоднородных модулей и компонентов компьютерного программного обеспечения. - Разработки и документирования программных интерфейсов. - Разработки процедур сборки модулей и компонентов компьютерного программного обеспечения. - Разработки процедур развертывания и обновления компьютерного программного обеспечения. - Разработки процедур миграции и преобразования (конвертации) данных. - Сборки программных модулей и компонентов в программный продукт. - Внесения изменений в процедуры сборки модулей и компонентов компьютерного программного обеспечения, развертывания компьютерного программного обеспечения, миграции и преобразования данных - Подключения программного продукта к компонентам внешней среды - Проверка работоспособности выпусков программного продукта - Разработки безопасных методов аутентификации и авторизации пользователей - Обработки и хранения конфиденциальных данных - Отслеживания и обработки уязвимостей безопасности - Использования шифрования для защиты данных в покое и в движении - Использование шифрования данных для защиты конфиденциальной информации, такой как пароли, персональные данные пользователей и другие чувствительные данные. - Реализация механизмов аутентификации и авторизации для обеспечения доступа только авторизованным пользователям. - Применение механизмов хеширования для защиты паролей

	пользователей от несанкционированного доступа. - Обеспечение безопасности передачи данных между клиентскими устройствами и серверами с использованием протоколов шифрования, таких как SSL/TLS. - Разработка механизмов контроля доступа к данным, чтобы предотвратить несанкционированное чтение, изменение или удаление данных. - Проектирование и реализация систем резервного копирования и восстановления данных для обеспечения их сохранности в случае сбоев или потери устройства. - Тестирование приложений на уязвимости безопасности, такие как SQL-инъекции, межсайтовые сценарии и другие уязвимости, и принятие мер по их устранению. - Соблюдение законодательства и регуляций в области защиты данных - Интеграция информационной системы с существующими системами заказчика - Разработка API для интеграции информационной системы - Тестирование и отладка интеграции информационной системы - Проектирования интерфейсов обмена данными в соответствии с трудовым заданием - Разработки интерфейсов обмена данными в соответствии с трудовым заданием - Создания и настройки сборочных и развертывающих пайплайнов (pipelines) - Автоматизации тестирования, сборки и доставки приложений - Использования инструментов для автоматизации CI/CD - Использования контейнеров для изоляции этапов выполнения pipelines - Разработка и настройка CI/CD пайплайнов для автоматической сборки, тестирования и развертывания веб-приложений. - Использование инструментов для автоматического тестирования кода - Разработка и настройка систем контроля версий, таких как Git, для управления исходным кодом и версионирования приложений. - Настройка инфраструктуры для развертывания веб-приложений, включая сервера приложений, базы данных и другие необходимые компоненты. - настройки и управления конфигурацией инфраструктуры через код - использования инструментов для автоматизации конфигурации - создания и поддержания сценариев управления конфигурациями - разработка и настройка конфигурационных файлов для различных компонентов веб-приложений, таких как серверы приложений, базы данных и другие сервисы. - автоматизация процесса развертывания и конфигурации инфраструктуры с использованием инструментов - управление версиями конфигурационных файлов и инфраструктуры с использованием систем контроля версий - использования систем контроля версий - работы с репозиториями кода и ветками разработки - разрешения конфликтов и объединения кода - использование систем контроля версий для управления и отслеживания изменений в коде приложений - разработка и поддержка процессов работы с Git, включая создание веток, слияние изменений и управление конфликтами - настройка инфраструктуры для хранения и управления кодом приложений с использованием репозиториев Git **Умения:**

	- Анализировать технические задания и выявлять требования к алгоритмам. - Применять методы алгоритмизации для решения задач программирования. - Разрабатывать оптимальные алгоритмы для решения задач в области ИИ. - Реализовывать программные модули на основе требований технического задания. - Соблюдать при разработке принципы «чистого кода». - Использовать стандартные библиотеки и фреймворки для ускорения разработки.- - Оформлять код в соответствии с принятыми стандартами и требованиями. - Документировать разработанный программный код. - Соблюдать соглашения о наименованиях переменных, функций и классов (например, PEP8 для Python). - Работать с системами контроля версий для управления проектами. - Организовывать совместную работу над проектом через ветки разработки и слияние изменений. - Разрешать конфликты при слиянии кода.- - Использовать инструменты для отладки программного кода. - Идентифицировать и исправлять ошибки в программе. - Применять методы логирования для анализа выполнения программ. - Проводить различные виды тестирования (юнит-тестирование, интеграционное тестирование). - Выполнять настройки окружения т подготовку тестовых данных - Фиксировать результаты выполнения тестов и подготавливать отчеты о результатах тестов. - Определять уровень критичности дефектов. - Разрабатывать автоматизированные тесты для тестирования модулей и/или отдельных функций - Восстанавливать окружение и тесты после сбоя - Проектировать тестовые сценарии на основе тестовых планов. - Разрабатывать тестовые пакеты и задания на выполнение тестирования. - Использовать шаблоны для написания тест-кейсов. - Оценивать риски при отборе тестов для регрессионного тестирования. - Оценивать тесты на соответствие целям тестирования. - Использовать методы и приемы формализации поставленных задач - Использовать методы и приемы алгоритмизации поставленных задач - Использовать программное обеспечение для графического отображения алгоритмов - Применять алгоритмы решения типовых задач в соответствующих областях - Применять выбранные языки программирования для разработки программных модулей - Использовать выбранную среду программирования и средства системы управления базами данных - Использовать возможности имеющейся технической и/или программной архитектуры для написания программного кода - Применять выбранные языки программирования для эффективного написания программного кода. - Использовать возможности существующей технической и программной архитектуры для написания интеграционного и адаптивного кода.

	- Использовать выбранную интегрированную среду разработки и инструменты для работы с базами данных. - Создавать интерфейсные тексты, понятные пользователю, и корректировать глоссарий терминов - Применять стандарты и руководства по проектированию для конкретных платформ и операционных систем (Human Interface Guidelines, Material Design). - Следовать стандартам и принципам эргономики взаимодействия «человек — система». - Использовать основы верстки (языки разметки и стилей) и сценарных языков для понимания технических ограничений и возможностей. - Учитывать основы маркетинга при проектировании пользовательских интерфейсов. - Работать с инструментами подготовки тестовых данных - Выбирать необходимые генераторы тестовых данных - Применять генераторы тестовых данных - Получать и использовать информацию, необходимую для выполнения задания на тестирование ПО - Составлять отчет о подготовке тестовых данных - Выполнять модульные тесты с использованием инструментов тестирования, в том числе автоматизированного тестирования - Использовать системы контроля дефектов ПО - Составлять отчет о выполнении тестирования ПО - Работать в команде со специалистами по тестированию ПО и разработчиками - Писать программный код процедур интеграции программных модулей. - Использовать выбранную среду программирования для разработки процедур интеграции программных модулей. - Применять методы и средства сборки модулей и компонентов компьютерного программного обеспечения, разработки процедур для развертывания компьютерного программного обеспечения, миграции и преобразования данных, создания программных интерфейсов. - Применять методы и средства логирования работы интегрированных программных модулей. - Выполнять процедуры сборки программных модулей и компонентов в программный продукт. - Производить настройки параметров программного продукта и осуществлять запуск процедур сборки. - Использовать командные средства разработки компьютерного программного обеспечения. - Проводить проверку работоспособности программного продукта. - Создавать резервные копии программ и данных, выполнять восстановление, обеспечивать целостность программного продукта и данных. - Разрабатывать и реализовывать меры безопасности - Реализовывать хэширование паролей, сессионные токены и двухфакторную аутентификацию - Осуществлять валидацию данных, поступающих от пользователей - Разрабатывать политику доступа и права пользователей к данным и функциональности приложения - Реализовывать меры контроля доступа и аудита для отслеживания действий пользователей и обнаружения несанкционированных действий - Работать в команде над интеграцией модулей в информационную систему

	- Выполнять интеграцию программный модулей в программный продукт - Кодировать на языках программирования - Находить и анализировать ключевые понятия и термины в сторонней документации для интеграции, а также разбираться в их контексте и использовании в рамках проекта - разрабатывать и настраивать пайплайны для непрерывной интеграции и непрерывной доставки - создавать скрипты автоматизации для тестирования и развертывания приложений - управлять и мониторить автоматизированными процессами - способность разрабатывать и настраивать CI/CD пайплайны с использованием различных инструментов и технологий - навыки написания автоматических тестов для проверки работоспособности и качества кода веб-приложений. - умение настраивать системы контроля версий для эффективной работы над проектами в команде - методологии разработки программного обеспечения, таких как Agile и DevOps, и умение применять их в практике. - создавать и изменять конфигурацию инфраструктуры через код - управлять и обновлять инфраструктуру автоматически - обеспечивать согласованность и надежность инфраструктуры - способность разрабатывать и настраивать конфигурационные файлы для различных компонентов веб-приложений. - навыки автоматизации процесса развертывания и конфигурации инфраструктуры с использованием инструментов управления конфигурацией. - умение использовать системы контроля версий для управления и версионирования конфигурационных файлов и инфраструктуры - эффективно использовать системы контроля версий для управления кодом - работать с ветками и выполнять слияния кода - разрешать конфликты и отслеживать историю изменений - способность эффективно использовать системы контроля версий для управления и отслеживания изменений в коде приложений. - навыки работы с Git, включая создание веток, слияние изменений и разрешение конфликтов. - умение настраивать инфраструктуру для хранения и управления кодом приложений с использованием репозиториев Git.
Администрирование баз данных	**Практический опыт:** - Идентификации проблем, связанных с нормальным функционированием базы данных; - Восстановления системы. - Администрирования сервера баз данных; - Участия в администрировании отдельных компонент серверов; - Документирования результатов аудита безопасности информации; - Использования процедуры резервного копирования баз данных; - Использования процедуры восстановления баз данных - Подготовки документации по формированию требований хранилищ банка данных - Проектирования, разработки и эксплуатации баз данных

	- Планирование процедур резервного копирования данных - Запуск процедуры резервного копирования данных - Мониторинг выполнения процедур резервного копирования данных - Контроль завершения процедуры резервного копирования данных - Проведение повторной процедуры резервного копирования данных в случае ее нештатного завершения - Хранение резервных копий БД - Назначение приоритета запросов к БД - Оптимизация выполнения статистически значимых запросов к БД - Создание программного кода в соответствии с техническим заданием (готовыми спецификациями) - Оптимизация программного кода с использованием специализированных программных средств - Определение источников больших данных для анализа, идентификация внешних и внутренних источников данных для проведения аналитических работ - Получение и фильтрация больших объемов данных из гетерогенных источников - Извлечение, проверка и очистка больших объемов данных из гетерогенных источников - Агрегация и разработка представления больших объемов данных из гетерогенных источников - Оценка соответствия набора данных предметной области и задачам аналитических работ - Возможности имеющейся у исполнителя методологической и технологической инфраструктуры анализа больших данных - Предметная область анализа - Теоретические и прикладные основы анализа больших данных - Современные методы и инструментальные средства анализа больших данных - Современный опыт использования анализа больших данных - Осуществление стратегического и оперативного руководства кадровыми и материальными ресурсами в процессах получения, размещения, хранения, распределения, миграции, архивирования и удаления данных - Осуществление стратегического и оперативного руководства материально-техническими ресурсами и элементами технологической инфраструктуры больших данных - Разработка моделей данных, адаптированных к технологиям больших данных - Интеграция больших данных с системами хранения данных организации - Создание и интеграция облачных репозиториев с системами хранения данных организации - Создание параллельных систем хранения и обработки информации - Разработка методов и регламентов реализации различных

	режимов обработки больших данных, в том числе режима реального времени - Мониторинг и оценка производительности обработки данных в организации, разработка предложений по повышению производительности обработки больших данных - Разработка методов и регламентов преобразования данных - Разработка и согласование стратегии формирования информационных активов организации - Разработка методов и регламентов оценки информационных активов организации - Формирование состава и координация деятельности персонала, отвечающего за приобретение и получение больших данных - Оценка эффективности системы хранения и обработки данных организации - Разработка предложений по развитию и совершенствованию системы получения, хранения, передачи, обработки больших данных - Формирование целей, приоритетов и ограничений управления качеством данных, адаптация целей к изменениям внутренних и внешних условий, потребностей и возможностей, в том числе технологических - Разработка стратегии управления качеством больших данных - Выделение ресурсов для управления качеством больших данных - Реализация стратегии обеспечения качества больших данных - Анализ реализации стратегии обеспечения качества больших данных - Проведение оценки качества больших данных - Разрешение проблем и инцидентов с качеством больших данных - Оценка рисков использования больших данных **Умения:** Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам Использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации Пользоваться профессиональной документацией на государственном и иностранном языках - Производить идентификацию проблем, связанных с нормальным функционированием базы данных; - Принимать решения по локализации проблем, связанных с нормальным функционированием базы данных; - Документировать внештатные ситуации связанные с нормальным функционированием базы данных; - Осуществлять основные функции по администрированию

	баз данных; - Настраивать политики безопасности при работе с сервером баз данных - Дать независимую оценку уровня безопасности - Производить регламентное обновление программного обеспечения - Разрабатывать перечень рекомендаций по дальнейшей эксплуатации БД с максимальной защитой хранящейся информации. - Производить формирование требований к обработке данных и их извлечению; - Добавлять, удалять и изменять данные в базе данных; - Производить операции по импорту и экспорту данных в различных форматах - Создавать расписание резервного копирования данных - Вычислять размер полной резервной копии БД - Читать техническую документацию на БД - Работать с устройствами резервного копирования данных и носителями резервных копий - Выполнять регламентные процедуры по резервированию данных - Проверять восстановимость резервной копии данных - Выбирать критерии оптимизации выполнения запросов к БД - Производить статистический анализ запросов к БД - Классифицировать запросы к БД по видам и по результатам статистического анализа - Применять языки и системы программирования БД для оптимизации выполнения запросов - Осуществлять коммуникации с заинтересованными сторонами - Использовать выбранную среду программирования и средства системы управления базами данных - Применять выбранные языки программирования для написания программного кода - Определять требования к поставщикам данных из гетерогенных источников - Осуществлять взаимодействие с внутренними и внешними поставщиками данных из гетерогенных источников - Разрабатывать и оценивать модели больших данных - Использовать инструментальные средства для извлечения, преобразования, хранения и обработки данных из разнородных источников, в том числе в режиме реального времени - Производить очистку данных для проведения аналитических работ - Проводить интеграцию и преобразование больших объемов данных - Оценивать соответствие наборов данных задачам анализа больших данных - Оценивать стоимость данных для проведения аналитических работ - Взаимодействовать с пользователями информации и

	поставщиками больших данных, инструментария и сервисов - Пользоваться методами оценки эффективности системы хранения и обработки данных организации - Разрабатывать системы хранения и обработки данных - Проводить интеграцию систем хранения и обработки данных - Разрабатывать модели данных, адаптированных к технологиям больших данных - Пользоваться методами и инструментами получения, хранения, передачи, обработки больших данных - Определять происхождение данных и оценивать источники больших данных - Определять и устранять проблемные зоны при обеспечении качества больших данных - Измерять и оценивать качество больших данных - Проводить оценку и устранение рисков использования больших данных
Обучение готовых моделей искусственного интеллекта	**Практический опыт:** - подбора и настройки готовых моделей ИИ с учетом поставленных задач, анализа результатов их применения. - создания сценариев обучения, подготовки данных для обучения, настройки гиперпараметров для достижения оптимального результата. - реализации процесса обучения моделей на подготовленных данных, применения методов калибровки для улучшения точности моделей. - оценки эффективности обученных моделей, корректировка обучения при необходимости, анализ ошибок и улучшение модели. - создания отчетов по обучению моделей, использование инструментов для визуализации (Matplotlib, Seaborn) для наглядного представления данных. - формирования запросов для получения и анализа данных, построение графиков и диаграмм для визуализации результатов работы ИИ. - определения источников больших данных для анализа, идентификация внешних и внутренних источников данных для проведения аналитических работ - получения и фильтрации больших объемов данных из гетерогенных источников - извлечения, проверки и очистки больших объемов данных из гетерогенных источников - агрегации и разработки представления больших объемов данных из гетерогенных источников - оценки соответствия набора данных предметной области и задачам аналитических работ - выбора методов и инструментальных средств анализа больших данных для проведения аналитических работ - разработки, проверки и оценки используемых моделей больших данных - адаптации и развертывания моделей больших данных в предметной среде

	- выбора средств представления результатов аналитики больших данных - подготовки отчета по результатам аналитических работ - разработки кода ИС и баз данных ИС в рамках выполнения работ по созданию (модификации) и сопровождению ИС - верификации кода ИС и баз данных ИС относительно дизайна ИС и структуры баз данных ИС в рамках выполнения работ по созданию (модификации) и сопровождению ИС - устранения обнаруженных несоответствий в коде ИС и базах данных ИС относительно дизайна ИС и структуры баз данных ИС в рамках выполнения работ по созданию (модификации) и сопровождению ИС - разработки интерфейсов обмена данными между ИС заказчика в рамках выполнения работ по созданию (модификации) и сопровождению ИС - разработки форматов обмена данными между ИС заказчика в рамках выполнения работ по созданию (модификации) и сопровождению ИС - разработки технологий обмена данными между ИС заказчика в соответствии с трудовым заданием в рамках выполнения работ по созданию (модификации) и сопровождению ИС - переработки текстов различной тематики (рерайт) - поисковой оптимизации и адаптации текстовых материалов - создания программного кода в соответствии с техническим заданием (готовыми спецификациями) - оптимизации программного кода с использованием специализированных программных средств - сбора идей о требованиях к ИТ-продукту из различных источников - разработки функциональных требований к ИТ-продукту - создания макетов интерфейса ИТ-продукта - разработки нефункциональных требований к ИТ-продукту - организации разработки и разработки плана управления требованиями заказчика в проектах в области ИТ любого уровня сложности - согласования плана управления требованиями в проектах в области ИТ любого уровня сложности с заинтересованными лицами - утверждения плана управления требованиями заказчика в проектах в области ИТ любого уровня сложности **Умения:** - анализировать задачи для выбора подходящих готовых моделей ИИ, учитывать их ограничения и возможности. - разрабатывать сценарии обучения, определять параметры обучения для различных типов моделей ИИ. - настраивать процесс обучения, выбирать подходящие датасеты и корректировать параметры обучения для калибровки. - осуществлять мониторинг качества обучения моделей, выявлять отклонения и проблемы в результатах работы. - подготавливать отчёты и документировать результаты работы с

	моделями ИИ, используя стандарты и требования к оформлению. - формировать запросы для получения данных из моделей ИИ, представлять результаты в виде графиков и таблиц. - определять требования к поставщикам данных из гетерогенных источников - осуществлять взаимодействие с внутренними и внешними поставщиками данных из гетерогенных источников - разрабатывать и оценивать модели больших данных - использовать инструментальные средства для извлечения, преобразования, хранения и обработки данных из разнородных источников, в том числе в режиме реального времени - производить очистку, интеграцию и преобразование данных для проведения аналитических работ - планировать аналитические работы с использованием технологий больших данных - проводить сравнительный анализ методов и инструментальных средств анализа больших данных - проводить аналитические работы с использованием технологий больших данных - программировать на языках высокого уровня, ориентированных на работу с большими данными - решать задачи классификации, кластеризации, регрессии, прогнозирования, снижения размерности и ранжирования данных - решать проблемы переобучения и недообучения алгоритма - кодировать на языках программирования в рамках выполнения работ по созданию (модификации) и сопровождению ИС - тестировать результаты кодирования ИС в рамках выполнения работ по созданию (модификации) и сопровождению ИС - анализировать и структурировать входные данные в рамках выполнения работ по созданию (модификации) и сопровождению ИС - кодировать на языках программирования ИС в рамках выполнения работ по созданию (модификации) и сопровождению ИС - тестировать результаты разработки ИС в рамках выполнения работ по созданию (модификации) и сопровождению ИС - реферировать, аннотировать и модифицировать тексты - вводить и редактировать данные в текстовом редакторе - использовать выбранную среду программирования и средства системы управления базами данных - использовать возможности имеющейся технической и/или программной архитектуры для написания программного кода - расставлять приоритеты в области требований к ИТ-продукту - планировать работы в проектах в области ИТ любого уровня сложности - осуществлять коммуникации в проектах в области ИТ любого уровня сложности - - проводить переговоры с заинтересованными сторонами проекта в области ИТ любого уровня сложности

3. Оценка освоения учебной практики

3.1 Задания для оценки освоения УП.01.01

1. Задания для оценки освоения УП.01

Раздел 1. Работа с данными для генеративных ИИ-моделей

Тема 1.1. Сбор и анализ данных для LLM.

Практическое занятие №1. Формирование технического задания.

Проверяемые результаты обучения:

- ПК 1.1 Формировать алгоритмы разработки программных модулей в соответствии с техническим заданием
- ПК 1.4 Использовать систему контроля версий программного кода с учетом обеспечения возможности организации групповой разработки
- ПК 1.11 Документировать логику и поведение интерфейса: описывать взаимосвязи, взаимодействия и возможные состояния элементов графического пользовательского интерфейса.
- ПК 1.12 Использовать специализированные инструменты для прототипирования и проектирования графических пользовательских интерфейсов.

1. Цель практической работы.

Развить навыки разработки ИИ-приложений на основе больших языковых моделей (LLM), освоить процессы планирования, проектирования и документирования проекта, а также закрепить знания в области системного анализа и разработки технического задания к ИИ-проекту.

2. Задачи практической работы.

– Выбрать кейс (тему проекта) для разработки ИИ-приложения;
– Сформулировать уточняющие вопросы по кейсу для формирования четкого ТЗ;
– Разработать техническое задание на создание ИИ-приложения;
– Определить требования к функциональности, интерфейсу и данным;
– Согласовать этапы и сроки выполнения работ

3. Этапы выполнения:

- Выбор кейса (темы проекта).
  Выберите один из предлагаемых вариантов ИИ-приложения для разработки:

– Чат-бот с RAG-системой для работы с пользовательскими документами;
– Агент для преобразования текстовых запросов в SQL-запросы;
– Приложение для анализа изображений с помощью мультимодальной LLM;
– Консольное приложение для диалога с LLM с настраиваемыми параметрами генерации.

- Формулировка уточняющих вопросов по кейсу.
Для формирования четкого технического задания подготовьте вопросы, например:
– Какова основная цель разрабатываемого ИИ-приложения?
– Какие функции должно выполнять приложение?
– Какие типы данных будут использоваться (текст, изображения, структурированные данные)?
– Какие источники данных будут задействованы?
– Какие параметры генерации должны настраиваться пользователем?
– Какие требования к точности и скорости работы системы?
– Какие требования к безопасности и конфиденциальности данных?
– Есть ли предпочтения по используемым технологиям или фреймворкам?
– Какой интерфейс должен быть реализован (консольный, веб-интерфейс)?
- Создание технического задания (ТЗ).
На основе полученных ответов сформируйте ТЗ, в котором укажите:
– Название проекта и его краткое описание;
– Общие сведения;
– Требования к функциональности и интерфейсу;
– Технические требования (выбор LLM, векторной базы данных, фреймворков);
– Требования к данным (форматы, объемы, источники);
– Требования к безопасности и конфиденциальности (если есть);
– Этапы и сроки выполнения.

4. Теоретический материал

Жизненный цикл разработки ИИ-приложений – это процесс от принятия решения о создании продукта до его эксплуатации. В нём можно выделить пять главных этапов:

- Анализ требований. Формирование цели проекта, определение перечня задач, сферы применения будущего ИИ-приложения, условий, сроков и бюджета проекта. На этом этапе определяются требования к данным, моделям и инфраструктуре.
- Этап проектирования. Определение архитектуры ИИ-приложения, функций, требований к функционалу и интерфейсу. Выбор LLM, векторной базы данных, фреймворков. Проектирование пайплайнов обработки данных и RAG-системы.
- Написание кода. Создание прототипа готового продукта и его компонентов. Разработка скриптов для взаимодействия с LLM, создание векторной базы

данных и индексация документов, реализация RAG-пайплайна и агентов.
- Тестирование и отладка. Проверка соответствия требованиям к продукту, устранение обнаруженных ошибок. Тестирование качества ответов LLM, точности семантического поиска, производительности системы.
- Эксплуатация и сопровождение. Развертывание приложения, обучение пользователей, документирование, адаптация приложения к новым требованиям.

Техническое задание (ТЗ) – это документ, который подробно описывает требования и условия выполнения проекта или разработки ИИ-приложения. Он служит основой для согласования ожиданий между заказчиком и исполнителем.
Основные функции технического задания:
– Уточнение целей и задач проекта;
– Описание требований к функциональности и качеству продукта;
– Определение технических стандартов и ограничений;
– Указание сроков выполнения и этапов работ.

5. Результат: готовое техническое задание.

Пример ТЗ:

ТЕХНИЧЕСКОЕ ЗАДАНИЕ

на разработку ИИ-приложения «RAG-бот "TechSupport"»

1. Общие сведения

- Наименование проекта: Консольное приложение «RAG-бот "TechSupport"».
- Краткое описание: Приложение предназначено для ответов на вопросы пользователей на основе загруженной базы знаний (PDF-инструкций) с использованием технологии RAG (Retrieval-Augmented Generation) и локальной большой языковой модели (LLM).

Заказчик: Учебная практика по ПМ.01.
Исполнитель: Студент группы [Номер группы] [ФИО].
Основание для разработки: Задание на практическое занятие №1.

2. Назначение и цели создания системы

Назначение: Автоматизация процесса поиска ответов на технические вопросы сотрудников на основе внутренней документации.
Цель: Сокращение времени поиска информации и повышение точности ответов за счет использования семантического поиска по документам.

3. Требования к функциональности

Приложение должно выполнять следующие функции:
Загрузка документов: Чтение PDF-файлов из указанной директории.
Обработка данных: Извлечение текста, очистка от шумов (колонтитулы, номера страниц) и разбиение на фрагменты (чанки).
Индексация: Создание векторных представлений (эмбеддингов) для каждого фрагмента и сохранение их в векторной базе данных.
Диалог: Ожидание ввода вопроса от пользователя в консольном режиме.
Поиск и генерация: Поиск наиболее релевантных фрагментов в базе данных и генерация ответа с помощью LLM на основе найденного контекста.
Вывод ответа: Отображение ответа в консоли с указанием источника (имя файла).
Управление: Поддержка команд /exit (выход) и /clear (очистка истории диалога).

4. Требования к интерфейсу

Тип интерфейса: Консольный (CLI).
Формат вывода: Текстовый вывод в стандартный поток вывода (stdout).
Язык интерфейса: Русский.

5. Технические требования

Язык программирования: Python 3.10+.
LLM (Large Language Model): Локальная модель Llama 3 (8B) через фреймворк Ollama.
Фреймворк для LLM-приложений: LangChain.
Векторная база данных: ChromaDB (локальное развертывание).
Модель эмбеддингов: all-MiniLM-L6-v2 (библиотека sentence-transformers).
Библиотеки для работы с PDF: pypdf или PyMuPDF.

6. Требования к данным

Источники данных: PDF-файлы с технической документацией (инструкции, руководства пользователя).
Форматы: .pdf.
Объем: До 100 страниц текста.
Язык данных: Русский.
Структура данных для индексации:

- id (уникальный идентификатор фрагмента);
- vector (эмбеддинг текста);
- text (оригинальный текст фрагмента);
- metadata (источник: имя файла, номер страницы).

7. Требования к безопасности и конфиденциальности

Приложение работает локально, без передачи данных в облачные сервисы. Исходные документы и векторная база данных хранятся на локальном диске. Персональные данные пользователей не собираются и не обрабатываются.

8. Этапы и сроки выполнения

Этап	**Содержание работ**	**Срок**
1	Сбор и анализ данных, очистка, чанкинг	1 ак. час
2	Генерация эмбеддингов, создание векторной БД	1 ак. час
3	Настройка LLM, написание скрипта RAG-пайплайна	1 ак. час
4	Тестирование, отладка, документирование	1 ак. час

Критерии оценки.

«5» «отлично»

- студент продемонстрировал готовность и способность к самостоятельному выполнению видов работ в стандартных ситуациях;
- все работы выполнил правильно и уверенно;
- проявил активность в освоении новых практических умений;
- содержание отчета по практическим работам полноценно отражает информацию по выполненным видам работ, установленных программой практики;
- оформление отчета по практическим работам полностью соответствует предъявляемым требованиям;
- при выполнении практических работ студент уверенно продемонстрировал обладание необходимыми компонентами общих и профессиональных компетенций;

«4» «хорошо»

- студент продемонстрировал готовность и способность к самостоятельному выполнению видов работ в стандартных ситуациях;
- все работы выполнил правильно и уверенно;
- содержание отчета по практическим работам в целом правильно отражает информацию по выполненным видам работ;
- оформление отчета по практическим работам соответствует предъявляемым требованиям, однако имеются замечания (некоторая неаккуратность, неполное описание проделанной работы и освоенных умений);
- при выполнении практических работ студент продемонстрировал обладание компонентами общих и профессиональных компетенций;
- характеристики студента положительные, имеются отдельные замечания и рекомендации;

«3» «удовлетворительно»

- при выполнении видов работ студент нуждался во внешнем сопровождении и контроле;
- в отчете по практическим работам содержится описание видов выполняемых работ согласно программе дисциплины, однако, записи неполные и неточные;
- отчет по практическим работам оформлен с отклонениями от установленных требований;
- при выполнении практических работ студент продемонстрировал обладание отдельными компонентами общих и профессиональных компетенций;

«**2» «неудовлетворительно»**

– студент не справился с запланированными видами работ;
– при выполнении практических работ студент продемонстрировал слабое обладание отдельными компонентами общих и профессиональных компетенций;
– отчет студентом не подготовлен.

Практическое занятие №2. Анализ качества и структуры данных.

Проверяемые результаты обучения:

- ПК 1.1 Формировать алгоритмы разработки программных модулей в соответствии с техническим заданием
- ПК 1.4 Использовать систему контроля версий программного кода с учетом обеспечения возможности организации групповой разработки
- ПК 1.11 Документировать логику и поведение интерфейса: описывать взаимосвязи, взаимодействия и возможные состояния элементов графического пользовательского интерфейса.
- ПК 1.12 Использовать специализированные инструменты для прототипирования и проектирования графических пользовательских интерфейсов.

1. Цель практической работы

Научиться проводить анализ структуры и качества собранных текстовых данных, выявлять шумы и аномалии для последующей очистки.

2. Задачи практической работы

– Провести анализ структуры собранных данных;
– Выявить и классифицировать шумы и аномалии в данных;
– Оценить объем и полноту собранных данных;
– Определить требования к предварительной обработке данных.

3. Этапы выполнения

1. Проведите анализ структуры каждого документа (заголовки, параграфы, списки, таблицы, изображения и т.д.);
2. Выявите шумы в данных – лишние символы, нечитаемые фрагменты, служебную информацию;
3. Выявите аномалии – нестандартные структуры, ошибки, неполные данные;
4. Составьте классификацию выявленных проблем;
5. Определите требования к очистке данных.

4. Теоретический материал

Анализ качества данных – это процесс оценки данных на предмет их пригодности для использования в ИИ-системах. Качество данных определяется их полнотой, точностью, актуальностью и структурной целостностью.
Типичные проблемы с данными:

- Шум – нерелевантная информация, лишние символы, ошибки;
- Пропуски – отсутствие значимых частей данных;
- Дубликаты – повторяющиеся фрагменты информации;
- Неоднородность – различные форматы и структуры;
- Устаревшие данные – информация, потерявшая актуальность.

Критерии оценки качества данных:

- Полнота данных;
- Точность и достоверность информации;
- Актуальность информации;
- Структурная целостность;
- Единообразие форматов.

Критерии оценки.

«5» «отлично»

- студент продемонстрировал готовность и способность к самостоятельному выполнению видов работ в стандартных ситуациях;

- все работы выполнил правильно и уверенно;
- проявил активность в освоении новых практических умений;
- содержание отчета по практическим работам полноценно отражает информацию по выполненным видам работ, установленных программой практики;
- оформление отчета по практическим работам полностью соответствует предъявляемым требованиям;
- при выполнении практических работ студент уверенно продемонстрировал обладание необходимыми компонентами общих и профессиональных компетенций;

«4» «хорошо»

- студент продемонстрировал готовность и способность к самостоятельному выполнению видов работ в стандартных ситуациях;
- все работы выполнил правильно и уверенно;
- содержание отчета по практическим работам в целом правильно отражает информацию по выполненным видам работ;
- оформление отчета по практическим работам соответствует предъявляемым требованиям, однако имеются замечания (некоторая неаккуратность, неполное описание проделанной работы и освоенных умений);
- при выполнении практических работ студент продемонстрировал обладание компонентами общих и профессиональных компетенций;
- характеристики студента положительные, имеются отдельные замечания и рекомендации;

«3» «удовлетворительно»

- при выполнении видов работ студент нуждался во внешнем сопровождении и контроле;
- в отчете по практическим работам содержится описание видов выполняемых работ согласно программе дисциплины, однако, записи неполные и неточные;
- отчет по практическим работам оформлен с отклонениями от установленных требований;
- при выполнении практических работ студент продемонстрировал обладание отдельными компонентами общих и профессиональных компетенций;

«**2» «неудовлетворительно»**

– студент не справился с запланированными видами работ;

– при выполнении практических работ студент продемонстрировал слабое обладание отдельными компонентами общих и профессиональных компетенций;

– отчет студентом не подготовлен.

Практическое занятие №3. Очистка и предварительная обработка данных

Проверяемые результаты обучения:

- ПК 1.1 Формировать алгоритмы разработки программных модулей в соответствии с техническим заданием
- ПК 1.4 Использовать систему контроля версий программного кода с учетом обеспечения возможности организации групповой разработки
- ПК 1.11 Документировать логику и поведение интерфейса: описывать взаимосвязи, взаимодействия и возможные состояния элементов графического пользовательского интерфейса.
- ПК 1.12 Использовать специализированные инструменты для прототипирования и проектирования графических пользовательских интерфейсов.

1. Цель практической работы

Освоить навыки очистки и предварительной обработки текстовых данных для использования в LLM и RAG-системах.

2. Задачи практической работы

– Удалить шумы и нерелевантную информацию из данных;
– Провести нормализацию и стандартизацию текстовых данных;
– Выполнить токенизацию и лемматизацию текстов;
– Сохранить обработанные данные в требуемом формате.

3. Этапы выполнения

– Напишите скрипт для загрузки и чтения собранных документов;
– Реализуйте очистку текста: удаление спецсимволов, лишних пробелов, нормализацию регистра;
– Выполните токенизацию текстов;
– Проведите лемматизацию токенов (при необходимости);
– Сохраните очищенные данные в структурированном формате.

4. Теоретический материал

Очистка данных – процесс удаления или исправления нерелевантной, поврежденной или некорректной информации из набора данных. Чистые данные – основа качественной работы любой ИИ-системы.

Основные методы очистки текстовых данных:

Метод	Описание	Применение

Удаление шумов	Удаление лишних символов, тегов, специальных знаков	Первичная очистка
Нормализация	Приведение текста к единому формату (регистр, Unicode)	Унификация данных
Удаление стоп-слов	Исключение слов, не несущих смысловой нагрузки	Уменьшение объема данных
Токенизация	Разбиение текста на отдельные слова или токены	Подготовка к анализу
Лемматизация	Приведение слов к нормальной форме	Унификация словоформ

Результат: очищенные данные.

Пример кода на python:

```
import re
from natasha import Segmenter, MorphVocab, NewsEmbedding,
NewsMorphTagger, Doc

# Пример очистки текста
text = "  Это   пример текста!  \n\n  С   лишними пробелами и переносами.  "
cleaned_text = re.sub(r'\s+', ' ', text).strip()
# cleaned_text = "Это пример текста! С лишними пробелами и переносами."

# Токенизация и лемматизация
segmenter = Segmenter()
```

```
morph_vocab = MorphVocab()
emb = NewsEmbedding()
morph_tagger = NewsMorphTagger(emb)
doc = Doc(cleaned_text)
doc.segment(segmenter)
doc.tag_morph(morph_tagger)
for token in doc.tokens:
    token.lemmatize(morph_vocab)
# Леммы: 'это', 'пример', 'текст', 'с', 'лишний', 'пробел', 'и', 'перенос'
```

Критерии оценки.

«5» «отлично»

- студент продемонстрировал готовность и способность к самостоятельному выполнению видов работ в стандартных ситуациях;
- все работы выполнил правильно и уверенно;
- проявил активность в освоении новых практических умений;
- содержание отчета по практическим работам полноценно отражает информацию по выполненным видам работ, установленных программой практики;
- оформление отчета по практическим работам полностью соответствует предъявляемым требованиям;
- при выполнении практических работ студент уверенно продемонстрировал обладание необходимыми компонентами общих и профессиональных компетенций;

«4» «хорошо»

- студент продемонстрировал готовность и способность к самостоятельному выполнению видов работ в стандартных ситуациях;
- все работы выполнил правильно и уверенно;
- содержание отчета по практическим работам в целом правильно отражает информацию по выполненным видам работ;
- оформление отчета по практическим работам соответствует предъявляемым требованиям, однако имеются замечания (некоторая неаккуратность, неполное описание проделанной работы и освоенных умений);
- при выполнении практических работ студент продемонстрировал обладание компонентами общих и профессиональных компетенций;
- характеристики студента положительные, имеются отдельные замечания и рекомендации;

«3» «удовлетворительно»

- при выполнении видов работ студент нуждался во внешнем сопровождении и контроле;
- в отчете по практическим работам содержится описание видов выполняемых работ согласно программе дисциплины, однако, записи неполные и неточные;

- отчет по практическим работам оформлен с отклонениями от установленных требований;
- при выполнении практических работ студент продемонстрировал обладание отдельными компонентами общих и профессиональных компетенций;

«2» «неудовлетворительно»

– студент не справился с запланированными видами работ;

– при выполнении практических работ студент продемонстрировал слабое обладание отдельными компонентами общих и профессиональных компетенций;

– отчет студентом не подготовлен.

Тема 1.2. Подготовка данных для RAG-систем и семантического поиска.

Практическое занятие №4. Разбиение текстов на фрагменты для индексации.

Проверяемые результаты обучения:

- ПК 1.1 Формировать алгоритмы разработки программных модулей в соответствии с техническим заданием
- ПК 1.4 Использовать систему контроля версий программного кода с учетом обеспечения возможности организации групповой разработки
- ПК 1.11 Документировать логику и поведение интерфейса: описывать взаимосвязи, взаимодействия и возможные состояния элементов графического пользовательского интерфейса.
- ПК 1.12 Использовать специализированные инструменты для прототипирования и проектирования графических пользовательских интерфейсов.

1. Цель практической работы

Научиться разбивать тексты на фрагменты (чанки) для последующей индексации и использования в RAG-системах.

2. Задачи практической работы

– Изучить стратегии разбиения текстов на фрагменты;
– Реализовать скрипты для разбиения текстов на фрагменты;
– Экспериментировать с различными размерами фрагментов;
– Оценить влияние размера фрагментов на качество поиска.

3. Этапы выполнения

1. Изучите стратегии чанкинга;
2. Реализуйте скрипт для разбиения текстов на фрагменты с

настраиваемыми параметрами (размер, перекрытие);
3. Проведите эксперименты с разными размерами фрагментов;
4. Сравните результаты и выберите оптимальные параметры;
5. Сохраните полученные фрагменты в структурированном формате.

4. Теоретический материал

Чанкинг – это процесс разбиения больших текстов на небольшие смысловые фрагменты для их последующей индексации и семантического поиска. Правильный выбор размера и стратегии чанкинга критически влияет на качество RAG-системы.

Стратегии чанкинга:



5. Результат: Файл с программным кодом чанкинга, описание выбранной стратегии, результаты экспериментов.

Пример кода на Python:

```
from langchain.text_splitter import RecursiveCharacterTextSplitter

text = "Длинный текст документа..."
text_splitter = RecursiveCharacterTextSplitter(
    chunk_size=500,
```

```
    chunk_overlap=50,
    length_function=len,
)
chunks = text_splitter.split_text(text)
# Результат: список фрагментов по ~500 символов с перекрытием 50 символов.
```

Критерии оценки.

«5» «отлично»

- студент продемонстрировал готовность и способность к самостоятельному выполнению видов работ в стандартных ситуациях;
- все работы выполнил правильно и уверенно;
- проявил активность в освоении новых практических умений;
- содержание отчета по практическим работам полноценно отражает информацию по выполненным видам работ, установленных программой практики;
- оформление отчета по практическим работам полностью соответствует предъявляемым требованиям;
- при выполнении практических работ студент уверенно продемонстрировал обладание необходимыми компонентами общих и профессиональных компетенций;

«4» «хорошо»

- студент продемонстрировал готовность и способность к самостоятельному выполнению видов работ в стандартных ситуациях;
- все работы выполнил правильно и уверенно;
- содержание отчета по практическим работам в целом правильно отражает информацию по выполненным видам работ;
- оформление отчета по практическим работам соответствует предъявляемым требованиям, однако имеются замечания (некоторая неаккуратность, неполное описание проделанной работы и освоенных умений);
- при выполнении практических работ студент продемонстрировал обладание компонентами общих и профессиональных компетенций;
- характеристики студента положительные, имеются отдельные замечания и рекомендации;

«3» «удовлетворительно»

- при выполнении видов работ студент нуждался во внешнем сопровождении и контроле;
- в отчете по практическим работам содержится описание видов выполняемых работ согласно программе дисциплины, однако, записи неполные и неточные;
- отчет по практическим работам оформлен с отклонениями от установленных требований;
- при выполнении практических работ студент продемонстрировал обладание отдельными компонентами общих и профессиональных компетенций;

«**2» «неудовлетворительно»**

– студент не справился с запланированными видами работ;
– при выполнении практических работ студент продемонстрировал слабое обладание отдельными компонентами общих и профессиональных компетенций;
– отчет студентом не подготовлен.

Практическое занятие №5. Генерация эмбеддингов текстов

Проверяемые результаты обучения:

- ПК 1.1 Формировать алгоритмы разработки программных модулей в соответствии с техническим заданием
- ПК 1.4 Использовать систему контроля версий программного кода с учетом обеспечения возможности организации групповой разработки
- ПК 1.11 Документировать логику и поведение интерфейса: описывать взаимосвязи, взаимодействия и возможные состояния элементов графического пользовательского интерфейса.
- ПК 1.12 Использовать специализированные инструменты для прототипирования и проектирования графических пользовательских интерфейсов.

1. Цель практической работы

Освоить навыки генерации эмбеддингов текстов с использованием моделей преобразования текста в векторы.

2. Задачи практической работы

– Загрузить модели для преобразования текста в векторы;
– Сгенерировать эмбеддинги для подготовленных текстовых данных;
– Сравнить качество эмбеддингов различных моделей;
– Визуализировать и проанализировать полученные эмбеддинги.

3. Этапы выполнения

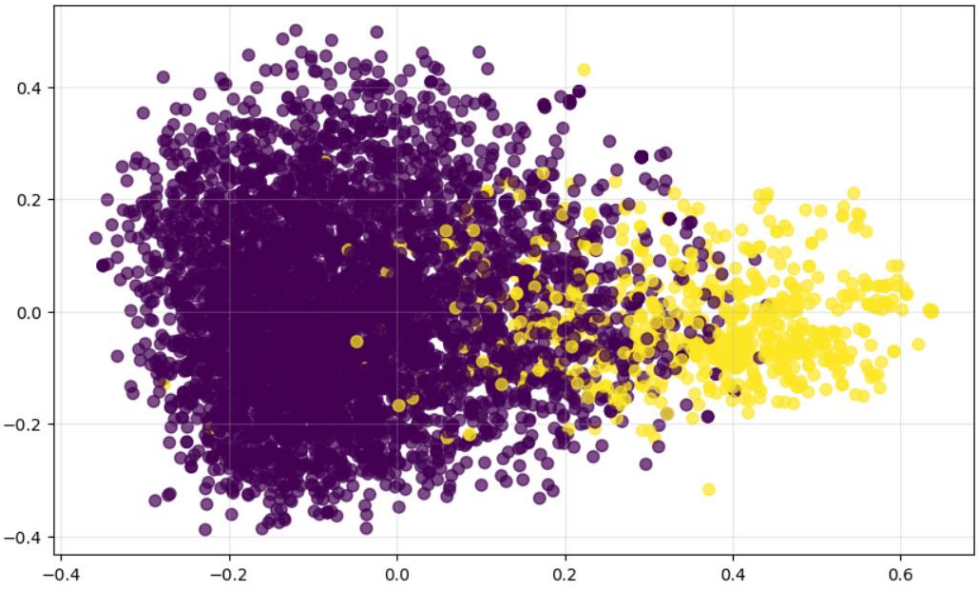
1. Загрузите модель для генерации эмбеддингов;
2. Разработайте скрипт для генерации эмбеддингов всех подготовленных фрагментов текста;
3. Сгенерируйте эмбеддинги и сохраните их;
4. Проведите визуализацию эмбеддингов для проверки качества;
5. Сравните качество эмбеддингов разных моделей (по желанию).

4. Теоретический материал

Эмбеддинги – это числовые векторные представления текстов, сохраняющие их смысловое значение. Качество эмбеддингов напрямую влияет на эффективность семантического поиска и RAG-систем.

Популярные модели для генерации эмбеддингов:

Модель	Размерность	Описание	Применение
text-embedding-ada-002	1536	OpenAI, высокое качество	Универсальные задачи
BERT	768	Архитектура трансформера	Семантический поиск
Sentence-BERT	384/768	Специализированная для предложений	RAG-системы
E5	768	Мультиязычная поддержка	Мультиязычные задачи



5. Результат: в ходе выполнения практической работы должны создать код генерации эмбадинга.

Пример кода на Python:

```
from sentence_transformers import SentenceTransformer
import numpy as np

model = SentenceTransformer('all-MiniLM-L6-v2')
embeddings = model.encode(chunks, show_progress_bar=True)
print(f"Размерность эмбеддингов: {embeddings.shape}")  # (кол-во_чанков, 384)

# Визуализация (пример)
from sklearn.manifold import TSNE
import matplotlib.pyplot as plt

tsne = TSNE(n_components=2, random_state=42)
vis_data = tsne.fit_transform(embeddings)
plt.scatter(vis_data[:, 0], vis_data[:, 1], alpha=0.5)
plt.title("Визуализация эмбеддингов")
plt.show()
```

Критерии оценки.

«5» «отлично»

- студент продемонстрировал готовность и способность к самостоятельному выполнению видов работ в стандартных ситуациях;
- все работы выполнил правильно и уверенно;
- проявил активность в освоении новых практических умений;
- содержание отчета по практическим работам полноценно отражает информацию по выполненным видам работ, установленных программой практики;
- оформление отчета по практическим работам полностью соответствует предъявляемым требованиям;
- при выполнении практических работ студент уверенно продемонстрировал обладание необходимыми компонентами общих и профессиональных компетенций;

«4» «хорошо»

- студент продемонстрировал готовность и способность к самостоятельному выполнению видов работ в стандартных ситуациях;
- все работы выполнил правильно и уверенно;
- содержание отчета по практическим работам в целом правильно отражает информацию по выполненным видам работ;
- оформление отчета по практическим работам соответствует предъявляемым требованиям, однако имеются замечания (некоторая неаккуратность, неполное описание проделанной работы и освоенных умений);
- при выполнении практических работ студент продемонстрировал обладание компонентами общих и профессиональных компетенций;

- характеристики студента положительные, имеются отдельные замечания и рекомендации;

«3» «удовлетворительно»

- при выполнении видов работ студент нуждался во внешнем сопровождении и контроле;
- в отчете по практическим работам содержится описание видов выполняемых работ согласно программе дисциплины, однако, записи неполные и неточные;
- отчет по практическим работам оформлен с отклонениями от установленных требований;
- при выполнении практических работ студент продемонстрировал обладание отдельными компонентами общих и профессиональных компетенций;

«**2» «неудовлетворительно»**

– студент не справился с запланированными видами работ;
– при выполнении практических работ студент продемонстрировал слабое обладание отдельными компонентами общих и профессиональных компетенций;
– отчет студентом не подготовлен.

Практическое занятие №6. Создание и настройка векторной базы данных

Проверяемые результаты обучения:

- ПК 1.1 Формировать алгоритмы разработки программных модулей в соответствии с техническим заданием
- ПК 1.4 Использовать систему контроля версий программного кода с учетом обеспечения возможности организации групповой разработки
- ПК 1.11 Документировать логику и поведение интерфейса: описывать взаимосвязи, взаимодействия и возможные состояния элементов графического пользовательского интерфейса.
- ПК 1.12 Использовать специализированные инструменты для прототипирования и проектирования графических пользовательских интерфейсов.

1. Цель практической работы

Научиться создавать и настраивать векторные базы данных для хранения и поиска эмбеддингов.

2. Задачи практической работы

– Выбрать подходящую векторную базу данных;
– Установить и настроить векторную базу данных;
– Создать коллекции для хранения эмбеддингов;
– Настроить параметры индексации и поиска;

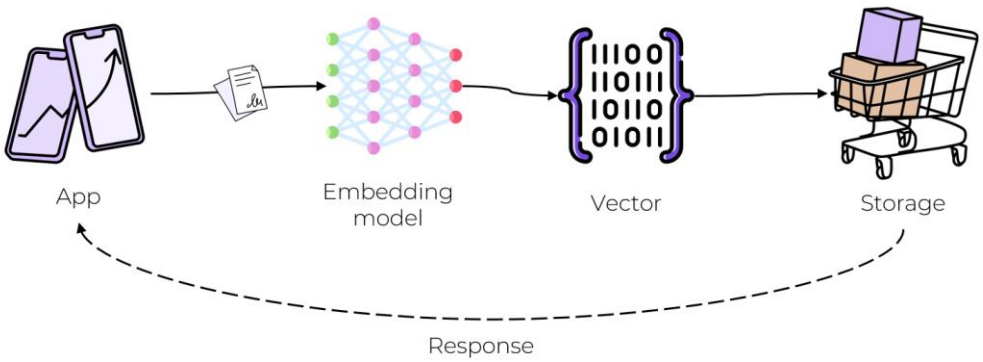
– Протестировать производительность векторной базы данных.

3. Этапы выполнения

1. Выберите и установите векторную базу данных;
2. Создайте коллекцию с подходящими параметрами;
3. Настройте параметры индексации и поиска;
4. Протестируйте производительность базы данных;
5. Задокументируйте настройки и результаты тестирования

4. Теоретический материал

Векторные базы данных – это специализированные системы, предназначенные для хранения, управления и эффективного поиска векторных представлений (эмбеддингов). Они обеспечивают быстрый семантический поиск по сходству.



Типы векторных баз данных:

База данных	Тип	Особенности	Применение
Chroma	Локальная	Простота, Python-интеграция	Прототипирование
FAISS	Библиотека	Высокая производительность	Исследования

Pinecone	Облачная	Управляемая, масштабируемая	Промышленное использование
Weaviate	Гибридная	Поддержка различных типов данных	Универсальные задачи
Qdrant	Локальная/Облачная	Высокая производительность	RAG-системы

5. Результат: в ходе выполнения работы выбрали векторную базу данных, и сравнили с другими вариантами.

Пример кода на Python:

```
import chromadb

client = chromadb.Client()
collection = client.create_collection(
   name="tech_support_docs",
   metadata={"hnsw:space": "cosine"}  # Косинусное сходство
)
print(f"Коллекция создана: {collection.name}")
print(f"Количество записей: {collection.count()}")
```

Критерии оценки.

«5» «отлично»

- студент продемонстрировал готовность и способность к самостоятельному выполнению видов работ в стандартных ситуациях;
- все работы выполнил правильно и уверенно;
- проявил активность в освоении новых практических умений;
- содержание отчета по практическим работам полноценно отражает информацию по выполненным видам работ, установленных программой практики;
- оформление отчета по практическим работам полностью соответствует предъявляемым требованиям;
- при выполнении практических работ студент уверенно продемонстрировал обладание необходимыми компонентами общих и профессиональных компетенций;

«4» «хорошо»

- студент продемонстрировал готовность и способность к самостоятельному выполнению видов работ в стандартных ситуациях;
- все работы выполнил правильно и уверенно;
- содержание отчета по практическим работам в целом правильно отражает информацию по выполненным видам работ;
- оформление отчета по практическим работам соответствует предъявляемым требованиям, однако имеются замечания (некоторая неаккуратность, неполное описание проделанной работы и освоенных умений);
- при выполнении практических работ студент продемонстрировал обладание компонентами общих и профессиональных компетенций;
- характеристики студента положительные, имеются отдельные замечания и рекомендации;

«3» «удовлетворительно»

- при выполнении видов работ студент нуждался во внешнем сопровождении и контроле;
- в отчете по практическим работам содержится описание видов выполняемых работ согласно программе дисциплины, однако, записи неполные и неточные;
- отчет по практическим работам оформлен с отклонениями от установленных требований;
- при выполнении практических работ студент продемонстрировал обладание отдельными компонентами общих и профессиональных компетенций;

«**2» «неудовлетворительно»**

– студент не справился с запланированными видами работ;

– при выполнении практических работ студент продемонстрировал слабое обладание отдельными компонентами общих и профессиональных компетенций;

– отчет студентом не подготовлен.

Практическое занятие №7. Индексация данных в векторной базе данных

Проверяемые результаты обучения:

- ПК 1.1 Формировать алгоритмы разработки программных модулей в соответствии с техническим заданием
- ПК 1.4 Использовать систему контроля версий программного кода с учетом обеспечения возможности организации групповой разработки
- ПК 1.11 Документировать логику и поведение интерфейса: описывать взаимосвязи, взаимодействия и возможные состояния элементов графического пользовательского интерфейса.
- ПК 1.12 Использовать специализированные инструменты для прототипирования и проектирования графических пользовательских интерфейсов.

1. Цель практической работы

Освоить навыки индексации подготовленных данных в векторной базе данных для последующего семантического поиска.

2. Задачи практической работы

– Разработать пайплайн индексации документов;
– Сгенерировать эмбеддинги для фрагментов текстов;
– Загрузить эмбеддинги в векторную базу данных;
– Проверить корректность индексации и выполнить тестовые поиски.

3. Этапы выполнения

1. Разработайте пайплайн индексации документов;
2. Сгенерируйте эмбеддинги для подготовленных фрагментов текстов;
3. Загрузите эмбеддинги в векторную базу данных вместе с текстами и метаданными;
4. Проверьте корректность индексации, выполнив несколько тестовых запросов;
5. Задокументируйте процесс индексации.

4. Теоретический материал

Индексация данных – это процесс загрузки векторных представлений (эмбеддингов) в векторную базу данных для их последующего быстрого поиска.
Этапы индексации в RAG-системе:

1. Подготовка и обработка текстовых документов;
2. Разбиение текстов на фрагменты;
3. Генерация эмбеддингов для всех фрагментов;
4. Загрузка эмбеддингов в векторную базу данных;
5. Проверка корректности индексации.

Структура данных для индексации:

- ID (уникальный идентификатор документа или фрагмента);
- Вектор (эмбеддинг текста);
- Текст (оригинальный текст или метаданные);
- Метаданные (источник, дата, тема, раздел).

5. Результат: в ходе выполнения практической работы должен быть написан файл с программным кодом индексации, описание результатов выполнения и примеры поисковых запросов.

Пример кода на Python:

```
# Добавление эмбеддингов, текстов и метаданных в коллекцию
collection.add(
    embeddings=embeddings.tolist(),
    documents=chunks,
    metadatas=[{"source": "manual.pdf", "page": i // 10} for i in range(len(chunks))],
    ids=[f"id_{i}" for i in range(len(chunks))]
)
print(f"Проиндексировано: {collection.count()} фрагментов.")

# Тестовый поиск
results = collection.query(query_texts=["Как настроить систему?"], n_results=3)
for doc, meta in zip(results['documents'][0], results['metadatas'][0]):
    print(f"Источник: {meta['source']}, стр. {meta['page']}")
    print(f"Текст: {doc[:200]}...\n")
```

Критерии оценки.

«5» «отлично»

- студент продемонстрировал готовность и способность к самостоятельному выполнению видов работ в стандартных ситуациях;
- все работы выполнил правильно и уверенно;
- проявил активность в освоении новых практических умений;
- содержание отчета по практическим работам полноценно отражает информацию по выполненным видам работ, установленных программой практики;
- оформление отчета по практическим работам полностью соответствует предъявляемым требованиям;
- при выполнении практических работ студент уверенно продемонстрировал обладание необходимыми компонентами общих и профессиональных компетенций;

«4» «хорошо»

- студент продемонстрировал готовность и способность к самостоятельному выполнению видов работ в стандартных ситуациях;
- все работы выполнил правильно и уверенно;
- содержание отчета по практическим работам в целом правильно отражает информацию по выполненным видам работ;
- оформление отчета по практическим работам соответствует предъявляемым требованиям, однако имеются замечания (некоторая неаккуратность, неполное описание проделанной работы и освоенных умений);
- при выполнении практических работ студент продемонстрировал обладание компонентами общих и профессиональных компетенций;
- характеристики студента положительные, имеются отдельные замечания и рекомендации;

«3» «удовлетворительно»

- при выполнении видов работ студент нуждался во внешнем сопровождении и контроле;
- в отчете по практическим работам содержится описание видов выполняемых работ согласно программе дисциплины, однако, записи неполные и неточные;
- отчет по практическим работам оформлен с отклонениями от установленных требований;
- при выполнении практических работ студент продемонстрировал обладание отдельными компонентами общих и профессиональных компетенций;

«**2» «неудовлетворительно»**

– студент не справился с запланированными видами работ;

– при выполнении практических работ студент продемонстрировал слабое обладание отдельными компонентами общих и профессиональных компетенций;

– отчет студентом не подготовлен.

Раздел 2. Настройка среды разработки и основы работы с LLM

Тема 2.1 Развертывание локального окружения для LLM.

Практическое занятие №8. Установка и настройка локальной среды выполнения LLM.

Проверяемые результаты обучения:

- ПК 1.8 Разрабатывать программные модули с использованием алгоритмов и методов искусственного интеллекта
- ПК 1.15 Осуществлять интеграцию программных модулей и компонентов программного обеспечения.
- ПК 1.16 Разрабатывать программные интерфейсы и процедуры сборки и развертывания программного обеспечения
- ПК 1.17 Осуществлять миграцию, преобразование и конвертацию данных программных систем
- ПК 1.18 Разрабатывать и интегрировать серверные программные модули для ИИ-решений
- ПК 1.19 Разрабатывать и настраивать программные интерфейсы взаимодействия с ИИ-компонентами и внешними системами
- ПК 1.20 Выполнять развертывание, тестирование и обеспечение работоспособности серверных компонентов ИИ-решений.

1. Цель практической работы

Освоить навыки установки и настройки локальной среды выполнения LLM для разработки ИИ-приложений.

2. Задачи практической работы

– Установить и настроить локальную среду выполнения LLM;
– Настроить переменные окружения и конфигурационные файлы;
– Проверить работоспособность установленного окружения;
– Задокументировать настройки.

3. Этапы выполнения

1. Установите локальную среду выполнения LLM (Ollama или аналоги);
2. Настройте переменные окружения и конфигурационные файлы;
3. Проверьте доступные системные ресурсы (RAM, GPU);
4. Запустите среду и убедитесь в её работоспособности;
5. Задокументируйте настройки и результаты проверки.

4. Теоретический материал

Локальная среда выполнения LLM – это программное обеспечение, позволяющее запускать и использовать большие языковые модели на локальном компьютере без обращения к облачным API.

Популярные средства для запуска локальных LLM:

Средство	Описание	Преимущества	Ограничения
Ollama	Управление и запуск LLM	Простота, поддержка многих моделей	Требует ресурсов
LM Studio	Десктопное приложение	Удобный интерфейс	Windows/macOS
Hugging Face Transformers	Библиотека для работы с LLM	Гибкость, полный контроль	Требует программирования

llamacpp	Легковесная реализация	Высокая производительность	Интерфейс командной строки

5. Результат: в ходе выполнения практической работы создали и настроили среду для разработки искусственного интеллекта.

Критерии оценки.

«5» «отлично»

- студент продемонстрировал готовность и способность к самостоятельному выполнению видов работ в стандартных ситуациях;
- все работы выполнил правильно и уверенно;
- проявил активность в освоении новых практических умений;
- содержание отчета по практическим работам полноценно отражает информацию по выполненным видам работ, установленных программой практики;
- оформление отчета по практическим работам полностью соответствует предъявляемым требованиям;
- при выполнении практических работ студент уверенно продемонстрировал обладание необходимыми компонентами общих и профессиональных компетенций;

«4» «хорошо»

- студент продемонстрировал готовность и способность к самостоятельному выполнению видов работ в стандартных ситуациях;
- все работы выполнил правильно и уверенно;
- содержание отчета по практическим работам в целом правильно отражает информацию по выполненным видам работ;
- оформление отчета по практическим работам соответствует предъявляемым требованиям, однако имеются замечания (некоторая неаккуратность, неполное описание проделанной работы и освоенных умений);
- при выполнении практических работ студент продемонстрировал обладание компонентами общих и профессиональных компетенций;
- характеристики студента положительные, имеются отдельные замечания и рекомендации;

«3» «удовлетворительно»

- при выполнении видов работ студент нуждался во внешнем сопровождении и контроле;
- в отчете по практическим работам содержится описание видов выполняемых работ согласно программе дисциплины, однако, записи неполные и неточные;

- отчет по практическим работам оформлен с отклонениями от установленных требований;
- при выполнении практических работ студент продемонстрировал обладание отдельными компонентами общих и профессиональных компетенций;

«2» «неудовлетворительно»

– студент не справился с запланированными видами работ;
– при выполнении практических работ студент продемонстрировал слабое обладание отдельными компонентами общих и профессиональных компетенций;
– отчет студентом не подготовлен.

Практическое занятие №9. Настройка виртуального окружения и среды разработки .

Проверяемые результаты обучения:

- ПК 1.8 Разрабатывать программные модули с использованием алгоритмов и методов искусственного интеллекта
- ПК 1.15 Осуществлять интеграцию программных модулей и компонентов программного обеспечения.
- ПК 1.16 Разрабатывать программные интерфейсы и процедуры сборки и развертывания программного обеспечения
- ПК 1.17 Осуществлять миграцию, преобразование и конвертацию данных программных систем
- ПК 1.18 Разрабатывать и интегрировать серверные программные модули для ИИ-решений
- ПК 1.19 Разрабатывать и настраивать программные интерфейсы взаимодействия с ИИ-компонентами и внешними системами
- ПК 1.20 Выполнять развертывание, тестирование и обеспечение работоспособности серверных компонентов ИИ-решений.

1. Цель практической работы

Освоить навыки настройки виртуального окружения и интегрированной среды разработки для проектов ИИ.

2. Задачи практической работы

– Создать и настроить виртуальное окружение;
– Установить необходимые библиотеки для работы с LLM;
– Настроить интегрированную среду разработки для ИИ-проектов;
– Проверить корректность установки всех компонентов.

3. Этапы выполнения

1. Создайте виртуальное окружение Python для проекта;
2. Установите необходимые библиотеки;
3. Настройте среду разработки для работы с ИИ-проектами;
4. Проверьте корректность установки всех компонентов;
5. Задокументируйте настройки окружения.

4. Теоретический материал

Виртуальное окружение– это изолированная среда для разработки проектов, позволяющая управлять зависимостями и версиями библиотек без конфликтов между проектами.

Инструменты для создания виртуальных окружений:

Инструмент	**Описание**	**Преимущества**
venv	Встроенный модуль Python	Стандартный, простой
conda	Менеджер пакетов и окружений	Поддержка разных языков
pipenv	Интеграция pip и virtualenv	Файл с зависимостями
poetry	Современный менеджер зависимостей	Активное развитие

Библиотеки для работы с LLM:

- transformers (Hugging Face) – работа с моделями;
- torch (PyTorch) – работа с нейросетями;
- sentence-transformers – генерация эмбеддингов;
- langchain – фреймворк для LLM-приложений;
- chromadb / faiss – векторные базы данных.

5. Результат: В ходе выполнения практической работы, создали виртуального окружение и скачали необходимые библиотеки.

Код для терминала:

```
python -m venv venv
source venv/bin/activate  # Linux/macOS
# или venv\Scripts\activate на Windows

pip install langchain langchain-community chromadb sentence-transformers pypdf ollama
pip freeze > requirements.txt
```

Критерии оценки.

«5» «отлично»

- студент продемонстрировал готовность и способность к самостоятельному выполнению видов работ в стандартных ситуациях;
- все работы выполнил правильно и уверенно;
- проявил активность в освоении новых практических умений;
- содержание отчета по практическим работам полноценно отражает информацию по выполненным видам работ, установленных программой практики;
- оформление отчета по практическим работам полностью соответствует предъявляемым требованиям;
- при выполнении практических работ студент уверенно продемонстрировал обладание необходимыми компонентами общих и профессиональных компетенций;

«4» «хорошо»

- студент продемонстрировал готовность и способность к самостоятельному выполнению видов работ в стандартных ситуациях;
- все работы выполнил правильно и уверенно;
- содержание отчета по практическим работам в целом правильно отражает информацию по выполненным видам работ;
- оформление отчета по практическим работам соответствует предъявляемым требованиям, однако имеются замечания (некоторая неаккуратность, неполное описание проделанной работы и освоенных умений);
- при выполнении практических работ студент продемонстрировал обладание компонентами общих и профессиональных компетенций;
- характеристики студента положительные, имеются отдельные замечания и рекомендации;

«3» «удовлетворительно»

- при выполнении видов работ студент нуждался во внешнем сопровождении и контроле;
- в отчете по практическим работам содержится описание видов выполняемых работ согласно программе дисциплины, однако, записи неполные и неточные;
- отчет по практическим работам оформлен с отклонениями от установленных требований;

- при выполнении практических работ студент продемонстрировал обладание отдельными компонентами общих и профессиональных компетенций;

«2» «неудовлетворительно»

– студент не справился с запланированными видами работ;
– при выполнении практических работ студент продемонстрировал слабое обладание отдельными компонентами общих и профессиональных компетенций;
– отчет студентом не подготовлен.

Практическое занятие №10. Загрузка и запуск предварительно обученных моделей.

Проверяемые результаты обучения:

- ПК 1.8 Разрабатывать программные модули с использованием алгоритмов и методов искусственного интеллекта
- ПК 1.15 Осуществлять интеграцию программных модулей и компонентов программного обеспечения.
- ПК 1.16 Разрабатывать программные интерфейсы и процедуры сборки и развертывания программного обеспечения
- ПК 1.17 Осуществлять миграцию, преобразование и конвертацию данных программных систем
- ПК 1.18 Разрабатывать и интегрировать серверные программные модули для ИИ-решений
- ПК 1.19 Разрабатывать и настраивать программные интерфейсы взаимодействия с ИИ-компонентами и внешними системами
- ПК 1.20 Выполнять развертывание, тестирование и обеспечение работоспособности серверных компонентов ИИ-решений.

1. Цель практической работы

Научиться загружать и запускать предварительно обученные модели для использования в ИИ-приложениях.

2. Задачи практической работы

– Загрузить предварительно обученные модели с платформ для обмена моделями;
– Запустить модели в локальной среде выполнения;
– Протестировать работоспособность загруженных моделей;
– Выполнить отладку типовых ошибок при загрузке и запуске.

3. Этапы выполнения

1. Выберите подходящую модель для вашего проекта;

2. Загрузите модель с выбранной платформы;
3. Запустите модель в локальной среде выполнения;
4. Проверьте работоспособность модели на тестовых запросах;
5. Задокументируйте процесс загрузки и результаты тестирования.

4. Теоретический материал

Платформы для обмена моделями:

Платформа	Описание	Особенности
Hugging Face	Крупнейший репозиторий моделей	Тысячи моделей, открытый доступ
Ollama Library	Репозиторий для Ollama	Оптимизированные сборки
PyTorch Hub	Официальный репозиторий PyTorch	Стандартные модели

Типы моделей для загрузки:
- Базовые LLM (Llama, Mistral, Gemma);
- Модели для эмбеддингов (all-MiniLM-L6-v2, E5);
- Мультимодальные модели (LLaVA);
- Специализированные модели (для кода, медицины и др.).

Требования к модели:
- Размер модели (количество параметров);
- Объем памяти для запуска;
- Квалификация (базовая/дообученная);
- Языковая поддержка.

5. Результат: в ходе выполнения практической работы должна быть установлена и протестирована модель.

Пример кода на Python:
```
from langchain_community.llms import Ollama

llm = Ollama(model="llama3", temperature=0.1)
response = llm.invoke("Какая сегодня погода?")
print(f"Ответ модели: {response}")
```

```
# Тест с другой моделью (если доступна)
# llm_mistral = Ollama(model="mistral")
# print(llm_mistral.invoke("Привет!"))
```

Критерии оценки.

«5» «отлично»

- студент продемонстрировал готовность и способность к самостоятельному выполнению видов работ в стандартных ситуациях;
- все работы выполнил правильно и уверенно;
- проявил активность в освоении новых практических умений;
- содержание отчета по практическим работам полноценно отражает информацию по выполненным видам работ, установленных программой практики;
- оформление отчета по практическим работам полностью соответствует предъявляемым требованиям;
- при выполнении практических работ студент уверенно продемонстрировал обладание необходимыми компонентами общих и профессиональных компетенций;

«4» «хорошо»

- студент продемонстрировал готовность и способность к самостоятельному выполнению видов работ в стандартных ситуациях;
- все работы выполнил правильно и уверенно;
- содержание отчета по практическим работам в целом правильно отражает информацию по выполненным видам работ;
- оформление отчета по практическим работам соответствует предъявляемым требованиям, однако имеются замечания (некоторая неаккуратность, неполное описание проделанной работы и освоенных умений);
- при выполнении практических работ студент продемонстрировал обладание компонентами общих и профессиональных компетенций;
- характеристики студента положительные, имеются отдельные замечания и рекомендации;

«3» «удовлетворительно»

- при выполнении видов работ студент нуждался во внешнем сопровождении и контроле;
- в отчете по практическим работам содержится описание видов выполняемых работ согласно программе дисциплины, однако, записи неполные и неточные;
- отчет по практическим работам оформлен с отклонениями от установленных требований;
- при выполнении практических работ студент продемонстрировал обладание отдельными компонентами общих и профессиональных компетенций;

«**2» «неудовлетворительно»**

– студент не справился с запланированными видами работ;

– при выполнении практических работ студент продемонстрировал слабое обладание отдельными компонентами общих и профессиональных компетенций;
– отчет студентом не подготовлен.

Практическое занятие №11. Настройка инфраструктуры и проверка окружения.

Проверяемые результаты обучения:

- ПК 1.8 Разрабатывать программные модули с использованием алгоритмов и методов искусственного интеллекта
- ПК 1.15 Осуществлять интеграцию программных модулей и компонентов программного обеспечения.
- ПК 1.16 Разрабатывать программные интерфейсы и процедуры сборки и развертывания программного обеспечения
- ПК 1.17 Осуществлять миграцию, преобразование и конвертацию данных программных систем
- ПК 1.18 Разрабатывать и интегрировать серверные программные модули для ИИ-решений
- ПК 1.19 Разрабатывать и настраивать программные интерфейсы взаимодействия с ИИ-компонентами и внешними системами
- ПК 1.20 Выполнять развертывание, тестирование и обеспечение работоспособности серверных компонентов ИИ-решений.

1. Цель практической работы

Освоить навыки настройки инфраструктуры и комплексной проверки работоспособности окружения для разработки ИИ-приложений.

2. Задачи практической работы

– Настроить инфраструктуру для работы с данными и моделями;
– Интегрировать среду разработки с локальной средой выполнения LLM;
– Провести комплексное тестирование работоспособности окружения;
– Задокументировать настроенную инфраструктуру.

3. Этапы выполнения

1. Настройте инфраструктуру для работы с данными и моделями;
2. Обеспечьте интеграцию всех компонентов системы;
3. Проведите комплексное тестирование всей системы;
4. Задокументируйте архитектуру и настройки инфраструктуры.

4. Теоретический материал

Инфраструктура разработки ИИ-приложений включает:

- Среду выполнения LLM;
- Среду разработки Python (IDE + виртуальное окружение);
- Векторную базу данных;
- Инструменты для управления данными;
- Систему контроля версий.

Компоненты инфраструктуры RAG-системы:
Источник данных → Обработка → Чанкинг → Эмбеддинги → Векторная БД → Поиск → LLM → Ответ.

5. Результат: в ходе выполнения практической работы должна быть выбрана и создана структура проекта.

Пример структуры проекта:

```
project/
├── data/
│   ├── raw/          # Исходные PDF
│   ├── processed/    # Очищенные тексты
│   └── indexed/      # Векторная БД
├── configs/          # Конфигурации
├── src/              # Исходный код
│   ├── data_loader.py
│   ├── indexer.py
│   └── rag_bot.py
└── requirements.txt
```

Критерии оценки.

«5» «отлично»

- студент продемонстрировал готовность и способность к самостоятельному выполнению видов работ в стандартных ситуациях;
- все работы выполнил правильно и уверенно;
- проявил активность в освоении новых практических умений;
- содержание отчета по практическим работам полноценно отражает информацию по выполненным видам работ, установленных программой практики;
- оформление отчета по практическим работам полностью соответствует предъявляемым требованиям;

- при выполнении практических работ студент уверенно продемонстрировал обладание необходимыми компонентами общих и профессиональных компетенций;

«4» «хорошо»

- студент продемонстрировал готовность и способность к самостоятельному выполнению видов работ в стандартных ситуациях;
- все работы выполнил правильно и уверенно;
- содержание отчета по практическим работам в целом правильно отражает информацию по выполненным видам работ;
- оформление отчета по практическим работам соответствует предъявляемым требованиям, однако имеются замечания (некоторая неаккуратность, неполное описание проделанной работы и освоенных умений);
- при выполнении практических работ студент продемонстрировал обладание компонентами общих и профессиональных компетенций;
- характеристики студента положительные, имеются отдельные замечания и рекомендации;

«3» «удовлетворительно»

- при выполнении видов работ студент нуждался во внешнем сопровождении и контроле;
- в отчете по практическим работам содержится описание видов выполняемых работ согласно программе дисциплины, однако, записи неполные и неточные;
- отчет по практическим работам оформлен с отклонениями от установленных требований;
- при выполнении практических работ студент продемонстрировал обладание отдельными компонентами общих и профессиональных компетенций;

«**2» «неудовлетворительно»**

– студент не справился с запланированными видами работ;

– при выполнении практических работ студент продемонстрировал слабое обладание отдельными компонентами общих и профессиональных компетенций;

– отчет студентом не подготовлен.

Тема 2.2 Работа с данными в среде LLM.

Практическое занятие №12. Загрузка данных в среду выполнения LLM

Проверяемые результаты обучения:

- ПК 1.8 Разрабатывать программные модули с использованием алгоритмов и методов искусственного интеллекта
- ПК 1.15 Осуществлять интеграцию программных модулей и компонентов программного обеспечения.
- ПК 1.16 Разрабатывать программные интерфейсы и процедуры сборки и развертывания программного обеспечения

- ПК 1.17 Осуществлять миграцию, преобразование и конвертацию данных программных систем
- ПК 1.18 Разрабатывать и интегрировать серверные программные модули для ИИ-решений
- ПК 1.19 Разрабатывать и настраивать программные интерфейсы взаимодействия с ИИ-компонентами и внешними системами
- ПК 1.20 Выполнять развертывание, тестирование и обеспечение работоспособности серверных компонентов ИИ-решений.

1. Цель практической работы

Освоить навыки загрузки подготовленных данных в среду выполнения LLM и проверки их корректности для использования в ИИ-приложениях.

2. Задачи практической работы

– Разработать скрипты для загрузки данных в среду выполнения LLM;
– Загрузить подготовленные данные (текстовые фрагменты и эмбеддинги) в среду выполнения;
– Проверить корректность загрузки данных;
– Обработать возможные ошибки при загрузке данных.

3. Этапы выполнения

1. Создайте структуру папок для данных в проекте, если она еще не создана (папки raw, processed, indexed, configs).
2. Разработайте скрипт загрузки данных, который должен:
    1. Проверять наличие всех необходимых файлов в папках проекта;
    2. Загружать подготовленные текстовые фрагменты и эмбеддинги из папки processed;
    3. Загружать метаданные документов (источник, дата, тема);
    4. Загружать конфигурационные файлы с параметрами модели и векторной БД.
3. Реализуйте в скрипте проверку корректности загрузки:
    1. Проверьте, что все ожидаемые файлы присутствуют;
    2. Проверьте, что данные имеют правильную структуру (количество полей, типы данных);
    3. Выведите статистику по загруженным данным (количество файлов, фрагментов, размер данных).
4. Реализуйте обработку ошибок при загрузке данных:
    1. Обработка ситуации, когда файл отсутствует;
    2. Обработка ситуации, когда данные повреждены или имеют неправильный формат;
    3. Вывод информативных сообщений об ошибках для

пользователя.

5. Выполните загрузку подготовленных данных из предыдущих работ в среду выполнения LLM.
6. Проверьте, что данные успешно загружены и доступны для использования в LLM. Выполните тестовый запрос к LLM с использованием загруженных данных.

4. Теоретический материал

Загрузка данных в среду выполнения LLM – это процесс передачи подготовленных данных (текстовых фрагментов, эмбеддингов, метаданных) в окружение, где работает большая языковая модель. Это важный этап, обеспечивающий доступ модели к данным для генерации ответов.

Типы данных для загрузки:

Тип данных	**Формат**	**Способ загрузки**	**Назначение**
Текстовые фрагменты	JSON, CSV, TXT	Чтение из файлов	Контекст для RAG
Эмбеддинги	Numpy, Parquet	Загрузка в векторную БД	Семантический поиск
Метаданные	JSON, CSV	Чтение из файлов	Фильтрация и сортировка
Конфигурации	YAML, JSON	Парсинг конфигов	Настройки модели

Способы загрузки данных:

1. Чтение из файловой системы – загрузка данных из локальных файлов;
2. Через API – загрузка данных из удаленных источников;
3. Из векторной базы данных – загрузка уже индексированных данных;
4. Из облачных хранилищ – загрузка из S3, Google Cloud Storage и др.

Проверка корректности загрузки:

- Проверка целостности данных (все ли файлы загружены);
- Проверка формата данных (соответствие ожидаемой структуре);
- Проверка объема данных (количество загруженных записей);
- Проверка доступности данных для модели

5. Результат: в ходе работы должны быть разработаны скрипты для загрузки данных в среду выполнения LLM, выполнена загрузка подготовленных текстовых фрагментов, эмбеддингов и метаданных, проведена проверка корректности загрузки. В отчете должны быть представлены: листинг разработанных скриптов с комментариями, статистика по загруженным данным, описание ошибок и способов их решения, результаты тестового запроса к LLM.

Пример кода на Python:

```
import json
import os

def load_processed_data(path='data/processed/chunks.json'):
    if not os.path.exists(path):
        raise FileNotFoundError(f"Файл не найден: {path}")

    with open(path, 'r', encoding='utf-8') as f:
        data = json.load(f)

    # Проверка структуры
    required_keys = ['id', 'text', 'metadata']
    for item in data:
        for key in required_keys:
            if key not in item:
                raise ValueError(f"Отсутствует ключ {key} в элементе {item.get('id')}")

    print(f"Загружено {len(data)} фрагментов.")
    print(f"Пример: {data[0]['text'][:100]}...")
    return data

data = load_processed_data()
```

Критерии оценки.

«5» «отлично»

- студент продемонстрировал готовность и способность к самостоятельному выполнению видов работ в стандартных ситуациях;
- все работы выполнил правильно и уверенно;
- проявил активность в освоении новых практических умений;
- содержание отчета по практическим работам полноценно отражает информацию по выполненным видам работ, установленных программой практики;

- оформление отчета по практическим работам полностью соответствует предъявляемым требованиям;
- при выполнении практических работ студент уверенно продемонстрировал обладание необходимыми компонентами общих и профессиональных компетенций;

«4» «хорошо»

- студент продемонстрировал готовность и способность к самостоятельному выполнению видов работ в стандартных ситуациях;
- все работы выполнил правильно и уверенно;
- содержание отчета по практическим работам в целом правильно отражает информацию по выполненным видам работ;
- оформление отчета по практическим работам соответствует предъявляемым требованиям, однако имеются замечания (некоторая неаккуратность, неполное описание проделанной работы и освоенных умений);
- при выполнении практических работ студент продемонстрировал обладание компонентами общих и профессиональных компетенций;
- характеристики студента положительные, имеются отдельные замечания и рекомендации;

«3» «удовлетворительно»

- при выполнении видов работ студент нуждался во внешнем сопровождении и контроле;
- в отчете по практическим работам содержится описание видов выполняемых работ согласно программе дисциплины, однако, записи неполные и неточные;
- отчет по практическим работам оформлен с отклонениями от установленных требований;
- при выполнении практических работ студент продемонстрировал обладание отдельными компонентами общих и профессиональных компетенций;

«**2» «неудовлетворительно»**

– студент не справился с запланированными видами работ;

– при выполнении практических работ студент продемонстрировал слабое обладание отдельными компонентами общих и профессиональных компетенций;

– отчет студентом не подготовлен.

Практическое занятие №13. Преобразование данных для обработки LLM

Проверяемые результаты обучения:

- ПК 1.8 Разрабатывать программные модули с использованием алгоритмов и методов искусственного интеллекта
- ПК 1.15 Осуществлять интеграцию программных модулей и компонентов программного обеспечения.
- ПК 1.16 Разрабатывать программные интерфейсы и процедуры сборки и развертывания программного обеспечения

- ПК 1.17 Осуществлять миграцию, преобразование и конвертацию данных программных систем
- ПК 1.18 Разрабатывать и интегрировать серверные программные модули для ИИ-решений
- ПК 1.19 Разрабатывать и настраивать программные интерфейсы взаимодействия с ИИ-компонентами и внешними системами
- ПК 1.20 Выполнять развертывание, тестирование и обеспечение работоспособности серверных компонентов ИИ-решений.

Цель практической работы

Освоить навыки преобразования подготовленных данных в формат, пригодный для непосредственной обработки LLM, включая формирование промптов с контекстом.

Задачи практической работы

– Разработать скрипты для преобразования данных в формат, пригодный для LLM;
– Преобразовать подготовленные данные в требуемый формат;
– Протестировать преобразованные данные на LLM;
– Оптимизировать процесс преобразования данных.

Этапы выполнения

1. Проанализируйте структуру данных, которые вы загрузили в среду выполнения LLM (текстовые фрагменты, эмбеддинги, метаданные).
2. Разработайте шаблон промпта для вашего ИИ-приложения. Учитывайте:
    1. Системную инструкцию (роль модели, правила поведения);
    2. Структуру вставки контекстных данных;
    3. Формат вопроса пользователя;
    4. Дополнительные требования к ответу.
3. Разработайте скрипт преобразования данных, который должен:
    1. Загружать данные из папки processed;
    2. Преобразовывать их в формат, соответствующий разработанному шаблону;
    3. Сохранять преобразованные данные в отдельную папку (например, formatted).
4. Протестируйте преобразованные данные, отправив несколько запросов к LLM:
    1. Проверьте корректность формирования промпта;
    2. Убедитесь, что контекст правильно вставляется в промпт;

3. Проанализируйте качество ответов LLM.

5. Оптимизируйте процесс преобразования данных:
    1. Оцените время преобразования большого объема данных;
    2. Улучшите структуру промпта для повышения качества ответов;
    3. Добавьте обработку крайних случаев (пустые данные, длинные тексты).
6. Задокументируйте процесс преобразования данных.

4. Теоретический материал

Преобразование данных для LLM – это процесс приведения данных к формату, который может быть передан в модель. Для LLM это обычно текстовый промпт, содержащий инструкцию и контекстные данные.

Форматы данных для LLM:

Формат	**Описание**	**Пример использования**
Простой текст	Неструктурированный текст	Базовые запросы
Шаблон промпта	Структурированный промпт с переменными	RAG-системы
JSON	Структурированный формат	API-взаимодействие
Инструкция-ответ	Пара для обучения/дообучения	Файн-тюнинг

5. Результат: в ходе работы должны быть разработаны шаблоны промптов и скрипты преобразования данных, выполнено преобразование подготовленных данных в формат, пригодный для LLM, проведено тестирование преобразованных данных.

Пример кода на Python:

```
def format_prompt(context, question):
```

```
    template = """Ты — ассистент технической поддержки. Ответь на вопрос пользователя, используя только следующий контекст.
Если в контексте нет ответа, скажи "Я не знаю".

Контекст:
{context}

Вопрос: {question}
Ответ:"""
    return template.format(context=context, question=question)

# Пример
context = "\n".join([item['text'] for item in data[:3]])
prompt = format_prompt(context, "Как настроить систему?")
print(prompt[:500])
```

Критерии оценки.

«5» «отлично»

- студент продемонстрировал готовность и способность к самостоятельному выполнению видов работ в стандартных ситуациях;
- все работы выполнил правильно и уверенно;
- проявил активность в освоении новых практических умений;
- содержание отчета по практическим работам полноценно отражает информацию по выполненным видам работ, установленных программой практики;
- оформление отчета по практическим работам полностью соответствует предъявляемым требованиям;
- при выполнении практических работ студент уверенно продемонстрировал обладание необходимыми компонентами общих и профессиональных компетенций;

«4» «хорошо»

- студент продемонстрировал готовность и способность к самостоятельному выполнению видов работ в стандартных ситуациях;
- все работы выполнил правильно и уверенно;
- содержание отчета по практическим работам в целом правильно отражает информацию по выполненным видам работ;
- оформление отчета по практическим работам соответствует предъявляемым требованиям, однако имеются замечания (некоторая неаккуратность, неполное описание проделанной работы и освоенных умений);
- при выполнении практических работ студент продемонстрировал обладание компонентами общих и профессиональных компетенций;
- характеристики студента положительные, имеются отдельные замечания и рекомендации;

«3» «удовлетворительно»

- при выполнении видов работ студент нуждался во внешнем сопровождении и контроле;
- в отчете по практическим работам содержится описание видов выполняемых работ согласно программе дисциплины, однако, записи неполные и неточные;
- отчет по практическим работам оформлен с отклонениями от установленных требований;
- при выполнении практических работ студент продемонстрировал обладание отдельными компонентами общих и профессиональных компетенций;

«**2» «неудовлетворительно»**

– студент не справился с запланированными видами работ;

– при выполнении практических работ студент продемонстрировал слабое обладание отдельными компонентами общих и профессиональных компетенций;

– отчет студентом не подготовлен.

Практическое занятие №14. Управление контекстом и историей диалога .

Проверяемые результаты обучения:

- ПК 1.8 Разрабатывать программные модули с использованием алгоритмов и методов искусственного интеллекта
- ПК 1.15 Осуществлять интеграцию программных модулей и компонентов программного обеспечения.
- ПК 1.16 Разрабатывать программные интерфейсы и процедуры сборки и развертывания программного обеспечения
- ПК 1.17 Осуществлять миграцию, преобразование и конвертацию данных программных систем
- ПК 1.18 Разрабатывать и интегрировать серверные программные модули для ИИ-решений
- ПК 1.19 Разрабатывать и настраивать программные интерфейсы взаимодействия с ИИ-компонентами и внешними системами
- ПК 1.20 Выполнять развертывание, тестирование и обеспечение работоспособности серверных компонентов ИИ-решений.

1. Цель практической работы

Научиться управлять контекстом и историей диалога в диалоговых системах на основе LLM, обеспечивая качественное и связное взаимодействие с пользователем..

2. Задачи практической работы

– Реализовать управление контекстом диалога;

– Настроить хранение и управление историей диалога;

– Протестировать работу с контекстом на различных сценариях;
– Оценить влияние контекста на качество ответов.

3. Этапы выполнения

1. Спроектируйте структуру хранения истории диалога для вашего приложения. Определите:
    1. Какие данные будут храниться (роль пользователя/ассистента, текст, время);
    2. Сколько сообщений будет храниться (ограничение по токенам);
    3. Будет ли использоваться сжатие длинных диалогов.
2. Реализуйте класс или модуль для управления памятью диалога, который должен:
    1. Добавлять новые сообщения в историю;
    2. Удалять старые сообщения при превышении лимита;
    3. Формировать контекст для передачи в LLM (с учетом ролей).
3. Реализуйте интеграцию управления памятью с вашим ИИ-приложением:
    1. Добавьте хранение истории для каждого пользователя/сессии;
    2. Обеспечьте передачу истории в промпт при каждом запросе;
    3. Реализуйте возможность очистки истории.
4. Проведите тестирование работы с контекстом на различных сценариях:
    1. Короткий диалог на 2-3 вопроса;
    2. Длинный диалог на 10+ вопросов;
    3. Смена темы разговора;
    4. Уточняющие вопросы с обращением к предыдущим ответам.
5. Проведите оценку влияния контекста на качество ответов:
    1. Сравните ответы LLM с контекстом и без контекста;
    2. Оцените качество ответов при различных размерах окна контекста;
    3. Выявите оптимальные параметры для вашего сценария.

4. Теоретический материал

Управление контекстом в диалоговых системах – это процесс поддержания и использования информации о предыдущих взаимодействиях с пользователем для обеспечения связности и релевантности ответов.

Компоненты управления контекстом:

Компонент	**Описание**	**Назначение**

История диалога	Хранение всех сообщений в беседе	Сохранение контекста
Окно контекста	Ограничение количества хранимых сообщений	Управление памятью
Сжатие контекста	Суммирование длинных диалогов	Экономия токенов
Извлечение релевантного	Выбор важных сообщений из истории	Улучшение качества

Стратегии управления контекстом:

1. Сохранять всю историю – подходит для коротких диалогов, но быстро заполняет контекстное окно;
2. Ограниченное окно – хранение последних N сообщений, подходит для большинства сценариев;
3. Сжатие – использование LLM для суммаризации длинных диалогов, экономия токенов;
4. Извлечение – поиск релевантных сообщений из истории по запросу.

5. Результат: в ходе работы должны быть реализован модуль управления памятью диалога, интегрирован в ИИ-приложение, проведено тестирование работы с контекстом на различных сценариях.

Пример кода на Python:

```
from langchain.memory import ConversationBufferMemory

memory = ConversationBufferMemory(memory_key="chat_history", return_messages=True)

def chat(user_input):
    memory.chat_memory.add_user_message(user_input)
    # Здесь должен быть вызов LLM с учетом истории
    response = "Пример ответа"
    memory.chat_memory.add_ai_message(response)
```

```
    return response

chat("Привет!")
chat("Как настроить систему?")
print(memory.load_memory_variables({}))
# Результат: {'chat_history': [HumanMessage(content='Привет!'),
AIMessage(content='Пример ответа'), ...]}
```

Критерии оценки.

«5» «отлично»

- студент продемонстрировал готовность и способность к самостоятельному выполнению видов работ в стандартных ситуациях;
- все работы выполнил правильно и уверенно;
- проявил активность в освоении новых практических умений;
- содержание отчета по практическим работам полноценно отражает информацию по выполненным видам работ, установленных программой практики;
- оформление отчета по практическим работам полностью соответствует предъявляемым требованиям;
- при выполнении практических работ студент уверенно продемонстрировал обладание необходимыми компонентами общих и профессиональных компетенций;

«4» «хорошо»

- студент продемонстрировал готовность и способность к самостоятельному выполнению видов работ в стандартных ситуациях;
- все работы выполнил правильно и уверенно;
- содержание отчета по практическим работам в целом правильно отражает информацию по выполненным видам работ;
- оформление отчета по практическим работам соответствует предъявляемым требованиям, однако имеются замечания (некоторая неаккуратность, неполное описание проделанной работы и освоенных умений);
- при выполнении практических работ студент продемонстрировал обладание компонентами общих и профессиональных компетенций;
- характеристики студента положительные, имеются отдельные замечания и рекомендации;

«3» «удовлетворительно»

- при выполнении видов работ студент нуждался во внешнем сопровождении и контроле;
- в отчете по практическим работам содержится описание видов выполняемых работ согласно программе дисциплины, однако, записи неполные и неточные;
- отчет по практическим работам оформлен с отклонениями от установленных требований;

- при выполнении практических работ студент продемонстрировал обладание отдельными компонентами общих и профессиональных компетенций;

«2» «неудовлетворительно»

– студент не справился с запланированными видами работ;
– при выполнении практических работ студент продемонстрировал слабое обладание отдельными компонентами общих и профессиональных компетенций;
– отчет студентом не подготовлен.

Практическое занятие №15. Экспериментирование с параметрами генерации

Проверяемые результаты обучения:

- ПК 1.8 Разрабатывать программные модули с использованием алгоритмов и методов искусственного интеллекта
- ПК 1.15 Осуществлять интеграцию программных модулей и компонентов программного обеспечения.
- ПК 1.16 Разрабатывать программные интерфейсы и процедуры сборки и развертывания программного обеспечения
- ПК 1.17 Осуществлять миграцию, преобразование и конвертацию данных программных систем
- ПК 1.18 Разрабатывать и интегрировать серверные программные модули для ИИ-решений
- ПК 1.19 Разрабатывать и настраивать программные интерфейсы взаимодействия с ИИ-компонентами и внешними системами
- ПК 1.20 Выполнять развертывание, тестирование и обеспечение работоспособности серверных компонентов ИИ-решений.

1. Цель практической работы

Изучить влияние параметров генерации на качество ответов LLM и научиться подбирать оптимальные параметры для различных сценариев использования.

2. Задачи практической работы

– Изучить основные параметры генерации LLM;
– Провести эксперименты с различными значениями параметров;
– Сравнить результаты при различных настройках;
– Выбрать оптимальные параметры для различных сценариев использования;
– Документировать результаты экспериментов.

3. Этапы выполнения

1. Создайте скрипт для тестирования параметров генерации, который позволяет:
    1. Задавать все основные параметры (температура, top_p, top_k, max_tokens);
    2. Отправлять один и тот же запрос с разными параметрами;
    3. Сохранять ответы для сравнения.
2. Подготовьте набор тестовых запросов для экспериментов:
    1. Фактический вопрос (требуется точный ответ);
    2. Творческий запрос (написание текста);
    3. Вопрос с несколькими вариантами ответа;
    4. Запрос на генерацию кода.
3. Проведите эксперименты, изменяя параметры:
    1. Эксперимент 1: изменение температуры (0.0, 0.3, 0.7, 1.0, 1.5);
    2. Эксперимент 2: изменение top-p (0.5, 0.7, 0.9, 1.0);
    3. Эксперимент 3: изменение max tokens (50, 100, 200, 500);
    4. Эксперимент 4: комбинации оптимальных параметров для разных сценариев.
4. Проведите анализ результатов экспериментов:
    1. Сравните ответы для каждого тестового запроса;
    2. Оцените качество, точность и креативность ответов;
    3. Выявите закономерности влияния параметров.
5. Определите оптимальные параметры для каждого сценария использования вашего приложения.

4. Теоретический материал

Параметры генерации LLM – это настройки, определяющие, как модель формирует ответ. От правильного выбора параметров зависит качество, креативность и структура сгенерированного текста.

Основные параметры генерации:

Параметр	**Описание**	**Диапазон**	**Влияние**
Температура (temperature)	Степень случайности	0.0 - 2.0	Низкие – детерминированные, высокие – креативные
Top-p (nucleus sampling)	Вероятностное отсечение	0.0 - 1.0	Ограничивает выбор токенов по вероятности

Top-k	Ограничение количества токенов	1 - бесконечность	Выбор из k наиболее вероятных токенов
Max tokens	Максимальная длина ответа	1 - контекстное окно	Ограничение объема ответа
Frequency penalty	Штраф за повторение слов	-2.0 - 2.0	Снижает повторение
Presence penalty	Штраф за появление новых тем	-2.0 - 2.0	Поощряет разнообразие

Рекомендации по выбору параметров:

Сценарий	**Температура**	**Top-p**	**Max tokens**	**Описание**
Фактические ответы	0.0 - 0.3	0.9	Короткие	Точность, минимум творчества
Творческие задачи	0.7 - 0.9	0.9 - 1.0	Средние	Баланс точности и креативности
Генерация кода	0.1 - 0.3	0.8	Средние/длинные	Структурированные ответы
Беседа	0.5 - 0.7	0.9	Короткие/средние	Естественный диалог
Суммаризация	0.2 - 0.4	0.9	Короткие	Сжатие информации

5. Результат: в ходе работы должен быть разработан скрипт для тестирования параметров генерации, проведены эксперименты с различными значениями параметров на нескольких типах запросов, выполнен анализ результатов.

Пример кода на Python:

```
profiles = {
   "factual": {"temperature": 0.1, "top_p": 0.9, "max_tokens": 200},
   "creative": {"temperature": 0.9, "top_p": 1.0, "max_tokens": 500},
   "code": {"temperature": 0.2, "top_p": 0.8, "max_tokens": 1000}
}

def ask_with_profile(prompt, profile="factual"):
   params = profiles.get(profile, profiles["factual"])
   llm = Ollama(model="llama3", temperature=params["temperature"],
            top_p=params["top_p"], num_predict=params["max_tokens"])
   return llm.invoke(prompt)

print(ask_with_profile("Сколько будет 2+2?", "factual"))
print(ask_with_profile("Напиши стих про осень", "creative"))
```

Критерии оценки.

«5» «отлично»

- студент продемонстрировал готовность и способность к самостоятельному выполнению видов работ в стандартных ситуациях;
- все работы выполнил правильно и уверенно;
- проявил активность в освоении новых практических умений;
- содержание отчета по практическим работам полноценно отражает информацию по выполненным видам работ, установленных программой практики;
- оформление отчета по практическим работам полностью соответствует предъявляемым требованиям;
- при выполнении практических работ студент уверенно продемонстрировал обладание необходимыми компонентами общих и профессиональных компетенций;

«4» «хорошо»

- студент продемонстрировал готовность и способность к самостоятельному выполнению видов работ в стандартных ситуациях;
- все работы выполнил правильно и уверенно;
- содержание отчета по практическим работам в целом правильно отражает информацию по выполненным видам работ;

- оформление отчета по практическим работам соответствует предъявляемым требованиям, однако имеются замечания (некоторая неаккуратность, неполное описание проделанной работы и освоенных умений);
- при выполнении практических работ студент продемонстрировал обладание компонентами общих и профессиональных компетенций;
- характеристики студента положительные, имеются отдельные замечания и рекомендации;

«3» «удовлетворительно»

- при выполнении видов работ студент нуждался во внешнем сопровождении и контроле;
- в отчете по практическим работам содержится описание видов выполняемых работ согласно программе дисциплины, однако, записи неполные и неточные;
- отчет по практическим работам оформлен с отклонениями от установленных требований;
- при выполнении практических работ студент продемонстрировал обладание отдельными компонентами общих и профессиональных компетенций;

«**2» «неудовлетворительно»**

– студент не справился с запланированными видами работ;

– при выполнении практических работ студент продемонстрировал слабое обладание отдельными компонентами общих и профессиональных компетенций;

– отчет студентом не подготовлен.

Раздел 3. Разработка базовых скриптов взаимодействия с LLM

Тема 3.1. Разработка скриптов взаимодействия с LLM.

Практическое занятие №16. Создание базовых скриптов для взаимодействия с LLM

Проверяемые результаты обучения:

- ПК 1.2 Разрабатывать программные модули в соответствии с техническим заданием
- ПК 1.3 Оформлять программный код в соответствии с техническим заданием
- ПК 1.5 Выполнять отладку программных модулей с использованием специализированных программных средств
- ПК 1.6 Выполнять тестирование программного кода
- ПК 1.7 Составлять тестовые сценарии
- ПК 1.9 Разрабатывать программный код мобильных приложений в соответствии с техническим заданием и утвержденными спецификациями

- ПК 1.10 Осуществлять оптимизацию программного кода мобильных приложений в соответствии с техническим заданием, оценивая трудоемкость и сроки выполнения работ.
- ПК 1.13 Выполнять разработку программных модулей
- ПК 1.14 Выполнять тестирование программных модулей и ИИ-решений
- ПК 1.21 Осуществлять защиту данных в мобильных приложениях
- ПК 1.25 Управлять версиями и кодом

1. Цель практической работы

Освоить навыки создания базовых скриптов для взаимодействия с LLM, отправки запросов и обработки ответов

2. Задачи практической работы

– Создать скрипты для отправки текстовых запросов к LLM;
– Реализовать обработку и вывод ответов от LLM;
– Реализовать базовые шаблоны промптов;
– Протестировать взаимодействие с различными моделями.

3. Этапы выполнения

1. Настройте подключение к выбранной LLM (локальной или через API):
    1. Определите способ взаимодействия (библиотека, API, CLI);
    2. Настройте параметры подключения (адрес, ключи, модель).
2. Разработайте функцию отправки запросов к LLM, которая должна:
    1. Принимать текст запроса и параметры генерации;
    2. Отправлять запрос к модели;
    3. Получать и обрабатывать ответ;
    4. Обрабатывать ошибки.
3. Реализуйте базовые шаблоны промптов для различных задач:
    1. Шаблон для фактического вопроса;
    2. Шаблон для творческой задачи;
    3. Шаблон для анализа текста;
    4. Шаблон для генерации кода.
4. Создайте скрипт командной строки для интерактивного взаимодействия с LLM:
    1. Принимает ввод от пользователя;
    2. Отправляет запрос к LLM;
    3. Выводит ответ в консоль.
5. Протестируйте взаимодействие с различными моделями (если доступно несколько моделей):
    1. Сравните ответы разных моделей на один запрос;

2. Оцените скорость и качество ответов.

4. Теоретический материал

Взаимодействие с LLM – это процесс отправки запросов (промптов) к модели и получения ответов. Базовые скрипты взаимодействия являются основой для всех ИИ-приложений.

Способы взаимодействия с LLM:

Способ	**Описание**	**Применение**
Через API	HTTP-запросы к удаленному сервису	Облачные модели
Локальный запуск	Запуск модели на локальной машине	Конфиденциальные данные
Через библиотеку	Использование библиотек (transformers)	Научные исследования
Через CLI	Командная строка	Быстрые тесты

Базовые функции скрипта:

1. Загрузка конфигурации (модель, параметры);
2. Отправка запроса к модели;
3. Обработка ответа;
4. Вывод результата пользователю.

5. Результат: В ходе работы должны быть созданы базовые скрипты для взаимодействия с LLM, разработаны шаблоны промптов для различных задач, проведено тестирование взаимодействия с моделями.

Пример кода на Python:

```
from langchain_community.llms import Ollama

class LLMClient:
    def __init__(self, model="llama3"):
```

```
        self.llm = Ollama(model=model)

    def ask(self, prompt, **kwargs):
        try:
            return self.llm.invoke(prompt, **kwargs)
        except Exception as e:
            return f"Ошибка: {e}"

client = LLMClient()
print(client.ask("Привет! Как дела?"))
```

Критерии оценки.

«5» «отлично»

- студент продемонстрировал готовность и способность к самостоятельному выполнению видов работ в стандартных ситуациях;
- все работы выполнил правильно и уверенно;
- проявил активность в освоении новых практических умений;
- содержание отчета по практическим работам полноценно отражает информацию по выполненным видам работ, установленных программой практики;
- оформление отчета по практическим работам полностью соответствует предъявляемым требованиям;
- при выполнении практических работ студент уверенно продемонстрировал обладание необходимыми компонентами общих и профессиональных компетенций;

«4» «хорошо»

- студент продемонстрировал готовность и способность к самостоятельному выполнению видов работ в стандартных ситуациях;
- все работы выполнил правильно и уверенно;
- содержание отчета по практическим работам в целом правильно отражает информацию по выполненным видам работ;
- оформление отчета по практическим работам соответствует предъявляемым требованиям, однако имеются замечания (некоторая неаккуратность, неполное описание проделанной работы и освоенных умений);
- при выполнении практических работ студент продемонстрировал обладание компонентами общих и профессиональных компетенций;
- характеристики студента положительные, имеются отдельные замечания и рекомендации;

«3» «удовлетворительно»

- при выполнении видов работ студент нуждался во внешнем сопровождении и контроле;

- в отчете по практическим работам содержится описание видов выполняемых работ согласно программе дисциплины, однако, записи неполные и неточные;
- отчет по практическим работам оформлен с отклонениями от установленных требований;
- при выполнении практических работ студент продемонстрировал обладание отдельными компонентами общих и профессиональных компетенций;

«2» «неудовлетворительно»

– студент не справился с запланированными видами работ;

– при выполнении практических работ студент продемонстрировал слабое обладание отдельными компонентами общих и профессиональных компетенций;

– отчет студентом не подготовлен.

Практическое занятие №17. Настройка параметров генерации в коде

Проверяемые результаты обучения:

- ПК 1.2 Разрабатывать программные модули в соответствии с техническим заданием
- ПК 1.3 Оформлять программный код в соответствии с техническим заданием
- ПК 1.5 Выполнять отладку программных модулей с использованием специализированных программных средств
- ПК 1.6 Выполнять тестирование программного кода
- ПК 1.7 Составлять тестовые сценарии
- ПК 1.9 Разрабатывать программный код мобильных приложений в соответствии с техническим заданием и утвержденными спецификациями
- ПК 1.10 Осуществлять оптимизацию программного кода мобильных приложений в соответствии с техническим заданием, оценивая трудоемкость и сроки выполнения работ.
- ПК 1.13 Выполнять разработку программных модулей
- ПК 1.14 Выполнять тестирование программных модулей и ИИ-решений
- ПК 1.21 Осуществлять защиту данных в мобильных приложениях
- ПК 1.25 Управлять версиями и кодом

1. Цель практической работы

Освоить навыки программной настройки параметров генерации в коде для различных сценариев использования.

2. Задачи практической работы

– Реализовать настройку параметров генерации в скриптах;

– Создать конфигурации для различных сценариев использования;
– Протестировать различные комбинации параметров;
– Оценить влияние параметров на качество ответов.

3. Этапы выполнения

1. Создайте конфигурационный файл (в формате JSON, YAML или Python-словарь) с наборами параметров для различных сценариев:
    a. Сценарий "factual" – для фактических ответов (низкая температура);
    b. Сценарий "creative" – для творческих задач (высокая температура);
    c. Сценарий "code" – для генерации кода (средние параметры);
    d. Сценарий "chat" – для диалога (сбалансированные параметры).
2. Реализуйте функцию, которая загружает конфигурацию параметров из файла и применяет их при отправке запроса:
    a. Функция должна принимать название сценария в качестве аргумента;
    b. Подставлять параметры из соответствующего профиля.
3. Модифицируйте ранее разработанный скрипт взаимодействия с LLM для поддержки профилей:
    a. Добавьте возможность выбора профиля при вызове функции;
    b. Реализуйте переключение между профилями в интерактивном режиме.
4. Проведите тестирование различных профилей на нескольких типах запросов:
    a. Оцените качество ответов для каждого профиля;
    b. Проверьте соответствие выбранного профиля типу задачи.
5. Добавьте возможность динамической настройки отдельных параметров поверх профиля:
    a. Возможность вручную указать температуру, top_p и другие параметры;
    b. Приоритет ручной настройки над профилем.

4. Теоретический материал

Программная настройка параметров генерации – это возможность управлять поведением LLM через код, что позволяет адаптировать модель к различным задачам без ручной настройки каждый раз.

Способы настройки параметров:

Способ	**Описание**	**Преимущества**

Конфигурационный файл	Параметры в отдельном файле (JSON, YAML)	Гибкость, разделение кода и настроек
Параметры в коде	Прямая передача параметров в функции	Простота, контроль
Профили	Предустановленные наборы параметров	Быстрый выбор сценария
Динамическая настройка	Изменение параметров на основе запроса	Адаптивность

5. Результат: в ходе работы должны быть созданы конфигурационные профили параметров для различных сценариев, реализована функция загрузки и применения профилей, проведено тестирование различных профилей на запросах.

Пример кода на Python:

```
def ask_llm_with_params(prompt, profile="factual"):
    profiles = {
        "factual": {"temperature": 0.1, "top_p": 0.9},
        "creative": {"temperature": 0.9, "top_p": 1.0},
        "code": {"temperature": 0.2, "top_p": 0.8}
    }
    params = profiles.get(profile, profiles["factual"])
    llm = Ollama(model="llama3", **params)
    return llm.invoke(prompt)

# Тест
print("Фактический:", ask_llm_with_params("Столица России?", "factual"))
print("Творческий:", ask_llm_with_params("Придумай имя для кота", "creative"))
```

Критерии оценки.

«5» «отлично»

- студент продемонстрировал готовность и способность к самостоятельному выполнению видов работ в стандартных ситуациях;
- все работы выполнил правильно и уверенно;
- проявил активность в освоении новых практических умений;
- содержание отчета по практическим работам полноценно отражает информацию по выполненным видам работ, установленных программой практики;
- оформление отчета по практическим работам полностью соответствует предъявляемым требованиям;
- при выполнении практических работ студент уверенно продемонстрировал обладание необходимыми компонентами общих и профессиональных компетенций;

«4» «хорошо»

- студент продемонстрировал готовность и способность к самостоятельному выполнению видов работ в стандартных ситуациях;
- все работы выполнил правильно и уверенно;
- содержание отчета по практическим работам в целом правильно отражает информацию по выполненным видам работ;
- оформление отчета по практическим работам соответствует предъявляемым требованиям, однако имеются замечания (некоторая неаккуратность, неполное описание проделанной работы и освоенных умений);
- при выполнении практических работ студент продемонстрировал обладание компонентами общих и профессиональных компетенций;
- характеристики студента положительные, имеются отдельные замечания и рекомендации;

«3» «удовлетворительно»

- при выполнении видов работ студент нуждался во внешнем сопровождении и контроле;
- в отчете по практическим работам содержится описание видов выполняемых работ согласно программе дисциплины, однако, записи неполные и неточные;
- отчет по практическим работам оформлен с отклонениями от установленных требований;
- при выполнении практических работ студент продемонстрировал обладание отдельными компонентами общих и профессиональных компетенций;

«**2» «неудовлетворительно»**

– студент не справился с запланированными видами работ;

– при выполнении практических работ студент продемонстрировал слабое обладание отдельными компонентами общих и профессиональных компетенций;

– отчет студентом не подготовлен.

Практическое занятие №18. Обработка и структурирование ответов

Проверяемые результаты обучения:

- ПК 1.2 Разрабатывать программные модули в соответствии с техническим заданием
- ПК 1.3 Оформлять программный код в соответствии с техническим заданием
- ПК 1.5 Выполнять отладку программных модулей с использованием специализированных программных средств
- ПК 1.6 Выполнять тестирование программного кода
- ПК 1.7 Составлять тестовые сценарии
- ПК 1.9 Разрабатывать программный код мобильных приложений в соответствии с техническим заданием и утвержденными спецификациями
- ПК 1.10 Осуществлять оптимизацию программного кода мобильных приложений в соответствии с техническим заданием, оценивая трудоемкость и сроки выполнения работ.
- ПК 1.13 Выполнять разработку программных модулей
- ПК 1.14 Выполнять тестирование программных модулей и ИИ-решений
- ПК 1.21 Осуществлять защиту данных в мобильных приложениях
- ПК 1.25 Управлять версиями и кодом

1. Цель практической работы

Освоить навыки обработки и структурирования ответов от LLM для использования в приложениях.

2. Задачи практической работы

– Разработать скрипты для обработки и структурирования ответов;
– Реализовать извлечение необходимой информации из ответов LLM;
– Преобразовать ответы в требуемый формат;
– Реализовать обработку ошибок и некорректных ответов.

3. Этапы выполнения

1. Проанализируйте типы ответов, которые возвращает LLM на различные запросы:
    a. Определите структуру ответов для разных сценариев;
    b. Выявите возможные проблемы с ответами (неполные, некорректные).
2. Разработайте функции для обработки ответов, которые должны:
    a. Очищать ответ от лишних пробелов и символов;
    b. Извлекать JSON-данные из текстового ответа (если возможно);
    c. Структурировать текстовые ответы (разбивать на части, выделять ключевые моменты).
3. Реализуйте обработку ошибок в ответах:

a. Проверка на пустой ответ;
b. Проверка на обрыв ответа (если достигнут max_tokens);
c. Проверка на некорректный JSON (с повторной попыткой);
d. Повторные запросы при некорректных ответах.

4. Интегрируйте обработку ответов с вашим скриптом взаимодействия с LLM:
   a. Модифицируйте функцию получения ответа для автоматической обработки;
   b. Добавьте логирование ошибок.
5. Протестируйте обработку ответов на различных типах запросов:
   a. Запрос с ожиданием JSON-ответа;
   b. Запрос с ожиданием структурированного текста;
   c. Запрос с возможным обрывом ответа.

4. Теоретический материал

Обработка ответов LLM – это процесс преобразования сырого текстового ответа модели в структурированный формат, пригодный для использования в приложении.

Методы обработки ответов:

Метод	Описание	Применение
Парсинг	Извлечение данных с помощью регулярных выражений	Структурированные ответы
JSON-декодирование	Преобразование JSON-строки в объект	API-интеграция
Семантический анализ	Извлечение смысловых компонентов	Анализ текста
Пост-обработка	Очистка, форматирование	Улучшение качества

5. Результат: В ходе работы должны быть разработаны функции обработки и структурирования ответов от LLM, реализована обработка ошибок и некорректных ответов, проведено тестирование на различных типах

запросов.

Пример кода на Python:

```
import json
import re

def parse_json_response(response):
    try:
        # Поиск JSON в ответе
        start = response.find('{')
        end = response.rfind('}') + 1
        if start == -1 or end == 0:
            return {"error": "JSON не найден"}
        return json.loads(response[start:end])
    except json.JSONDecodeError as e:
        return {"error": f"Ошибка парсинга: {e}"}

# Тест
raw_response = 'Вот ваш ответ: {"name": "Иван", "age": 25}'
print(parse_json_response(raw_response))
```

Критерии оценки.

«5» «отлично»

- студент продемонстрировал готовность и способность к самостоятельному выполнению видов работ в стандартных ситуациях;
- все работы выполнил правильно и уверенно;
- проявил активность в освоении новых практических умений;
- содержание отчета по практическим работам полноценно отражает информацию по выполненным видам работ, установленных программой практики;
- оформление отчета по практическим работам полностью соответствует предъявляемым требованиям;
- при выполнении практических работ студент уверенно продемонстрировал обладание необходимыми компонентами общих и профессиональных компетенций;

«4» «хорошо»

- студент продемонстрировал готовность и способность к самостоятельному выполнению видов работ в стандартных ситуациях;
- все работы выполнил правильно и уверенно;
- содержание отчета по практическим работам в целом правильно отражает информацию по выполненным видам работ;

- оформление отчета по практическим работам соответствует предъявляемым требованиям, однако имеются замечания (некоторая неаккуратность, неполное описание проделанной работы и освоенных умений);
- при выполнении практических работ студент продемонстрировал обладание компонентами общих и профессиональных компетенций;
- характеристики студента положительные, имеются отдельные замечания и рекомендации;

«3» «удовлетворительно»

- при выполнении видов работ студент нуждался во внешнем сопровождении и контроле;
- в отчете по практическим работам содержится описание видов выполняемых работ согласно программе дисциплины, однако, записи неполные и неточные;
- отчет по практическим работам оформлен с отклонениями от установленных требований;
- при выполнении практических работ студент продемонстрировал обладание отдельными компонентами общих и профессиональных компетенций;

«**2» «неудовлетворительно»**

– студент не справился с запланированными видами работ;

– при выполнении практических работ студент продемонстрировал слабое обладание отдельными компонентами общих и профессиональных компетенций;

– отчет студентом не подготовлен.

Практическое занятие №19. Разработка модулей для повторного использования

Проверяемые результаты обучения:

- ПК 1.2 Разрабатывать программные модули в соответствии с техническим заданием
- ПК 1.3 Оформлять программный код в соответствии с техническим заданием
- ПК 1.5 Выполнять отладку программных модулей с использованием специализированных программных средств
- ПК 1.6 Выполнять тестирование программного кода
- ПК 1.7 Составлять тестовые сценарии
- ПК 1.9 Разрабатывать программный код мобильных приложений в соответствии с техническим заданием и утвержденными спецификациями
- ПК 1.10 Осуществлять оптимизацию программного кода мобильных приложений в соответствии с техническим заданием, оценивая трудоемкость и сроки выполнения работ.
- ПК 1.13 Выполнять разработку программных модулей
- ПК 1.14 Выполнять тестирование программных модулей и ИИ-решений
- ПК 1.21 Осуществлять защиту данных в мобильных приложениях

- ПК 1.25 Управлять версиями и кодом

6. Цель практической работы.

Освоить навыки создания модулей для взаимодействия с LLM, пригодных для повторного использования в различных проектах.

7. Задачи практической работы.

– Создать модуль для взаимодействия с LLM;
– Инкапсулировать логику взаимодействия в функции и классы;
– Задокументировать разработанный модуль;
– Протестировать модуль на различных сценариях.

8. Этапы выполнения.

1. Проанализируйте функциональность, которую вы реализовали в предыдущих работах:
    1. Отправка запросов к LLM;
    2. Настройка параметров генерации;
    3. Обработка ответов;
    4. Управление контекстом.
2. Разработайте структуру модуля для взаимодействия с LLM:
    1. Определите основные компоненты модуля (файлы и их назначение);
    2. Спроектируйте API модуля (функции и классы для использования).
3. Реализуйте класс или набор функций для взаимодействия с LLM, инкапсулируя:
    1. Подключение к модели (локальной или через API);
    2. Отправку запросов с параметрами;
    3. Обработку ответов;
    4. Обработку ошибок.
4. Добавьте в модуль поддержку различных моделей:
    1. Возможность переключения между моделями;
    2. Адаптацию промптов под разные модели.
5. Задокументируйте разработанный модуль:
    1. Создайте документацию по установке и использованию;
    2. Добавьте примеры использования.
6. Протестируйте модуль на различных сценариях:
    1. Разные типы запросов;
    2. Разные модели (если доступны);
    3. Обработка ошибок и крайних случаев.

9. Теоретический материал.

Модули повторного использования – это компоненты программного обеспечения, которые могут быть применены в различных проектах без необходимости их переписывания.

Принципы разработки модулей:

Принцип	Описание
Единая ответственность	Модуль выполняет одну функцию
Инкапсуляция	Скрытие внутренней логики
Конфигурируемость	Настройка через параметры
Документированность	Четкое описание API

10. Результат: в ходе работы должен быть разработан модуль для взаимодействия с LLM, пригодный для повторного использования, оформлена документация, проведено тестирование модуля.

Пример кода на Python:

```
# module_llm.py
from langchain_community.llms import Ollama

class LLMClient:
    """Клиент для взаимодействия с LLM."""
    def __init__(self, model_name="llama3", temperature=0.1):
        self.model_name = model_name
        self.llm = Ollama(model=model_name, temperature=temperature)

    def ask(self, prompt):
        """Отправка запроса к LLM."""
        return self.llm.invoke(prompt)

    def ask_with_context(self, context, question):
```

```
    """Запрос с контекстом."""
    prompt = f"Контекст:\n{context}\n\nВопрос: {question}\nОтвет:"
    return self.ask(prompt)
```

Критерии оценки.

«5» «отлично»

- студент продемонстрировал готовность и способность к самостоятельному выполнению видов работ в стандартных ситуациях;
- все работы выполнил правильно и уверенно;
- проявил активность в освоении новых практических умений;
- содержание отчета по практическим работам полноценно отражает информацию по выполненным видам работ, установленных программой практики;
- оформление отчета по практическим работам полностью соответствует предъявляемым требованиям;
- при выполнении практических работ студент уверенно продемонстрировал обладание необходимыми компонентами общих и профессиональных компетенций;

«4» «хорошо»

- студент продемонстрировал готовность и способность к самостоятельному выполнению видов работ в стандартных ситуациях;
- все работы выполнил правильно и уверенно;
- содержание отчета по практическим работам в целом правильно отражает информацию по выполненным видам работ;
- оформление отчета по практическим работам соответствует предъявляемым требованиям, однако имеются замечания (некоторая неаккуратность, неполное описание проделанной работы и освоенных умений);
- при выполнении практических работ студент продемонстрировал обладание компонентами общих и профессиональных компетенций;
- характеристики студента положительные, имеются отдельные замечания и рекомендации;

«3» «удовлетворительно»

- при выполнении видов работ студент нуждался во внешнем сопровождении и контроле;
- в отчете по практическим работам содержится описание видов выполняемых работ согласно программе дисциплины, однако, записи неполные и неточные;
- отчет по практическим работам оформлен с отклонениями от установленных требований;
- при выполнении практических работ студент продемонстрировал обладание отдельными компонентами общих и профессиональных компетенций;

«**2» «неудовлетворительно»**

– студент не справился с запланированными видами работ;

– при выполнении практических работ студент продемонстрировал слабое обладание отдельными компонентами общих и профессиональных компетенций;
– отчет студентом не подготовлен.

Тема 3.2. Разработка приложений на основе LLM.

Практическое занятие №20. Разработка консольного чат-приложения

Проверяемые результаты обучения:

- ПК 1.2 Разрабатывать программные модули в соответствии с техническим заданием
- ПК 1.3 Оформлять программный код в соответствии с техническим заданием
- ПК 1.5 Выполнять отладку программных модулей с использованием специализированных программных средств
- ПК 1.6 Выполнять тестирование программного кода
- ПК 1.7 Составлять тестовые сценарии
- ПК 1.9 Разрабатывать программный код мобильных приложений в соответствии с техническим заданием и утвержденными спецификациями
- ПК 1.10 Осуществлять оптимизацию программного кода мобильных приложений в соответствии с техническим заданием, оценивая трудоемкость и сроки выполнения работ.
- ПК 1.13 Выполнять разработку программных модулей
- ПК 1.14 Выполнять тестирование программных модулей и ИИ-решений
- ПК 1.21 Осуществлять защиту данных в мобильных приложениях
- ПК 1.25 Управлять версиями и кодом

1. Цель практической работы.

Разработать консольное чат-приложение с поддержкой диалога с LLM, интегрирующее разработанные ранее модули и компоненты.

2. Задачи практической работы.

– Разработать консольное приложение с поддержкой диалога;
– Реализовать цикл взаимодействия с пользователем;
– Интегрировать разработанные модули в приложение;
– Протестировать работу приложения.

3. Этапы выполнения.

1. Спроектируйте структуру консольного чат-приложения:
    1. Определите основной цикл работы приложения;
    2. Определите набор поддерживаемых команд;
    3. Определите формат отображения диалога.
2. Разработайте приложение, используя созданные ранее модули:
    1. Подключите модуль для взаимодействия с LLM;
    2. Подключите модуль управления памятью диалога;
    3. Подключите модуль обработки ответов.
3. Реализуйте основные функции приложения:
    1. Инициализация приложения (загрузка модели, конфигурации);
    2. Основной цикл взаимодействия;
    3. Обработка пользовательского ввода (текст и команды);
    4. Вывод ответов LLM с форматированием.
4. Добавьте поддержку команд:
    1. /help – справка по командам;
    2. /exit – выход из приложения;
    3. /clear – очистка истории диалога;
    4. /model – смена модели;
    5. /config – просмотр и изменение параметров.
5. Проведите тестирование приложения:
    1. Проверьте работу всех команд;
    2. Проверьте сохранение истории диалога;
    3. Проверьте переключение между моделями (если поддерживается);
    4. Проверьте обработку ошибок.

4. Теоретический материал.

Консольное чат-приложение – это приложение с интерфейсом командной строки, позволяющее пользователю вести диалог с LLM в интерактивном режиме.

Компоненты консольного чат-приложения:

Компонент	**Описание**
Цикл ввода-вывода	Бесконечный цикл чтения ввода пользователя и вывода ответов
Обработка команд	Обработка специальных команд (/help, /exit, /clear)

Управление контекстом	Хранение и передача истории диалога
Настройка параметров	Возможность изменения параметров генерации на лету
Форматирование вывода	Красивое оформление ответов

5. Результат: в ходе работы должно быть разработано консольное чат-приложение для диалога с LLM, интегрированы ранее разработанные модули, реализованы команды управления.

Пример кода на Python:

```
from module_llm import LLMClient

def main():
    client = LLMClient()
    print("Чат-бот запущен. Введите /exit для выхода.")
    history = []

    while True:
        user_input = input("Вы: ").strip()
        if not user_input:
            continue
        if user_input.lower() == "/exit":
            print("До свидания!")
            break
        if user_input.lower() == "/clear":
            history = []
            print("История очищена.")
            continue

        history.append(f"User: {user_input}")
        context = "\n".join(history[-5:])
        response = client.ask(context)
        history.append(f"Assistant: {response}")
        print(f"Бот: {response}")
```

```
if __name__ == "__main__":
    main()
```

Критерии оценки.

«5» «отлично»

- студент продемонстрировал готовность и способность к самостоятельному выполнению видов работ в стандартных ситуациях;
- все работы выполнил правильно и уверенно;
- проявил активность в освоении новых практических умений;
- содержание отчета по практическим работам полноценно отражает информацию по выполненным видам работ, установленных программой практики;
- оформление отчета по практическим работам полностью соответствует предъявляемым требованиям;
- при выполнении практических работ студент уверенно продемонстрировал обладание необходимыми компонентами общих и профессиональных компетенций;

«4» «хорошо»

- студент продемонстрировал готовность и способность к самостоятельному выполнению видов работ в стандартных ситуациях;
- все работы выполнил правильно и уверенно;
- содержание отчета по практическим работам в целом правильно отражает информацию по выполненным видам работ;
- оформление отчета по практическим работам соответствует предъявляемым требованиям, однако имеются замечания (некоторая неаккуратность, неполное описание проделанной работы и освоенных умений);
- при выполнении практических работ студент продемонстрировал обладание компонентами общих и профессиональных компетенций;
- характеристики студента положительные, имеются отдельные замечания и рекомендации;

«3» «удовлетворительно»

- при выполнении видов работ студент нуждался во внешнем сопровождении и контроле;
- в отчете по практическим работам содержится описание видов выполняемых работ согласно программе дисциплины, однако, записи неполные и неточные;
- отчет по практическим работам оформлен с отклонениями от установленных требований;
- при выполнении практических работ студент продемонстрировал обладание отдельными компонентами общих и профессиональных компетенций;

«**2» «неудовлетворительно»**

– студент не справился с запланированными видами работ;

– при выполнении практических работ студент продемонстрировал слабое обладание отдельными компонентами общих и профессиональных компетенций;
– отчет студентом не подготовлен.

Практическое занятие №21. Реализация истории диалога и управления контекстом

Проверяемые результаты обучения:

- ПК 1.2 Разрабатывать программные модули в соответствии с техническим заданием
- ПК 1.3 Оформлять программный код в соответствии с техническим заданием
- ПК 1.5 Выполнять отладку программных модулей с использованием специализированных программных средств
- ПК 1.6 Выполнять тестирование программного кода
- ПК 1.7 Составлять тестовые сценарии
- ПК 1.9 Разрабатывать программный код мобильных приложений в соответствии с техническим заданием и утвержденными спецификациями
- ПК 1.10 Осуществлять оптимизацию программного кода мобильных приложений в соответствии с техническим заданием, оценивая трудоемкость и сроки выполнения работ.
- ПК 1.13 Выполнять разработку программных модулей
- ПК 1.14 Выполнять тестирование программных модулей и ИИ-решений
- ПК 1.21 Осуществлять защиту данных в мобильных приложениях
- ПК 1.25 Управлять версиями и кодом

1. Цель практической работы.

Реализовать полноценное управление историей диалога и контекстом в разработанном чат-приложении, обеспечивая связность и качество диалога.

2. Задачи практической работы.

– Реализовать сохранение истории диалога в разработанном приложении;
– Реализовать управление размером контекста (ограничение по токенам);
– Реализовать функцию сжатия/суммаризации длинных диалогов;
– Оценить качество диалога с управлением контекстом.

3. Этапы выполнения.

1. Усовершенствуйте управление памятью в вашем приложении:
    1. Добавьте подсчет токенов в истории диалога;

    2. Реализуйте автоматическое обрезание истории при превышении лимита;
    3. Сохраняйте историю с указанием ролей (user, assistant, system).
2. Реализуйте функцию сжатия (суммаризации) длинных диалогов:
    1. При достижении определенного количества сообщений или токенов;
    2. Используйте LLM для создания краткой сводки предыдущего диалога;
    3. Заменяйте длинную историю на сводку для экономии места в контексте.
3. Интегрируйте расширенное управление контекстом в ваше чат-приложение:
    1. Модифицируйте отправку запросов для использования оптимизированного контекста;
    2. Добавьте отображение текущего размера контекста в режиме реального времени.
4. Проведите сравнительное тестирование качества диалога:
    1. Протестируйте диалог с полной историей (если позволяет модель);
    2. Протестируйте диалог с ограниченным окном (последние N сообщений);
    3. Протестируйте диалог со сжатием истории.
5. Сравните результаты и выберите оптимальную стратегию для вашего приложения.

4. Теоретический материал.

Управление историей диалога – это процесс сохранения и эффективного использования предыдущих сообщений для обеспечения связности диалога.

Стратегии управления историей:

Стратегия	**Описание**	**Преимущества**	**Недостатки**
Полная история	Хранение всех сообщений	Максимальный контекст	Переполнение токенов
Окно сдвига	Хранение последних N сообщений	Простота	Потеря раннего контекста

Сжатие	Суммаризация длинной истории	Экономия токенов	Потеря деталей
Извлечение	Выбор релевантных сообщений	Точный контекст	Сложность реализации

5. Результат: в ходе работы должна быть реализована расширенная система управления историей диалога и контекстом в чат-приложении, проведено сравнительное тестирование различных стратегий управления.

Пример кода на Python:

```
class DialogManager:
    def __init__(self, max_history=10):
        self.history = []
        self.max_history = max_history

    def add_message(self, role, text):
        self.history.append({"role": role, "text": text})
        if len(self.history) > self.max_history:
            self.history.pop(0)

    def get_context(self):
        return "\n".join([f"{m['role']}: {m['text']}" for m in self.history])

    def clear(self):
        self.history = []

# Использование
dm = DialogManager()
dm.add_message("User", "Привет!")
dm.add_message("Assistant", "Здравствуйте!")
print(dm.get_context())
```

Критерии оценки.

«5» «отлично»

- студент продемонстрировал готовность и способность к самостоятельному выполнению видов работ в стандартных ситуациях;
- все работы выполнил правильно и уверенно;
- проявил активность в освоении новых практических умений;

- содержание отчета по практическим работам полноценно отражает информацию по выполненным видам работ, установленных программой практики;
- оформление отчета по практическим работам полностью соответствует предъявляемым требованиям;
- при выполнении практических работ студент уверенно продемонстрировал обладание необходимыми компонентами общих и профессиональных компетенций;

«4» «хорошо»

- студент продемонстрировал готовность и способность к самостоятельному выполнению видов работ в стандартных ситуациях;
- все работы выполнил правильно и уверенно;
- содержание отчета по практическим работам в целом правильно отражает информацию по выполненным видам работ;
- оформление отчета по практическим работам соответствует предъявляемым требованиям, однако имеются замечания (некоторая неаккуратность, неполное описание проделанной работы и освоенных умений);
- при выполнении практических работ студент продемонстрировал обладание компонентами общих и профессиональных компетенций;
- характеристики студента положительные, имеются отдельные замечания и рекомендации;

«3» «удовлетворительно»

- при выполнении видов работ студент нуждался во внешнем сопровождении и контроле;
- в отчете по практическим работам содержится описание видов выполняемых работ согласно программе дисциплины, однако, записи неполные и неточные;
- отчет по практическим работам оформлен с отклонениями от установленных требований;
- при выполнении практических работ студент продемонстрировал обладание отдельными компонентами общих и профессиональных компетенций;

«2» «неудовлетворительно»

– студент не справился с запланированными видами работ;

– при выполнении практических работ студент продемонстрировал слабое обладание отдельными компонентами общих и профессиональных компетенций;

– отчет студентом не подготовлен.

Практическое занятие №22. Промпт-инжиниринг для различных задач

Проверяемые результаты обучения:

- ПК 1.2 Разрабатывать программные модули в соответствии с техническим заданием

- ПК 1.3 Оформлять программный код в соответствии с техническим заданием
- ПК 1.5 Выполнять отладку программных модулей с использованием специализированных программных средств
- ПК 1.6 Выполнять тестирование программного кода
- ПК 1.7 Составлять тестовые сценарии
- ПК 1.9 Разрабатывать программный код мобильных приложений в соответствии с техническим заданием и утвержденными спецификациями
- ПК 1.10 Осуществлять оптимизацию программного кода мобильных приложений в соответствии с техническим заданием, оценивая трудоемкость и сроки выполнения работ.
- ПК 1.13 Выполнять разработку программных модулей
- ПК 1.14 Выполнять тестирование программных модулей и ИИ-решений
- ПК 1.21 Осуществлять защиту данных в мобильных приложениях
- ПК 1.25 Управлять версиями и кодом

1. Цель практической работы.

Освоить навыки промпт-инжиниринга для различных типов задач и сценариев использования, создавая эффективные шаблоны промптов.

2. Задачи практической работы.

– Разработать промпты для решения различных задач;
– Отработать техники промпт-инжиниринга;
– Сравнить качество ответов при различных подходах;
– Создать библиотеку шаблонов промптов.

3. Этапы выполнения.

1. Проанализируйте различные типы задач, которые будет решать ваше ИИ-приложение:
    1. Фактические вопросы (требуют точных ответов);
    2. Творческие задачи (генерация текста, идей);
    3. Аналитические задачи (анализ данных, сравнение);
    4. Структурированные задачи (генерация JSON, кода).
2. Для каждого типа задачи разработайте несколько вариантов промптов:
    1. Разные стили (формальный, неформальный);
    2. Разные техники (zero-shot, few-shot, chain-of-thought);
    3. Разный уровень детализации.
3. Протестируйте разработанные промпты на реальных запросах:
    1. Оцените качество и точность ответов;
    2. Проверьте соответствие формату (если задан);

3. Оцените время генерации.
4. Сравните различные подходы для каждого типа задачи и выявите наиболее эффективные.
5. Создайте библиотеку шаблонов промптов для вашего приложения:
   Оформите шаблоны в отдельном файле или классе;
   Добавьте возможность динамической подстановки параметров.

4. Теоретический материал.

Промпт-инжиниринг – это практика разработки эффективных промптов для получения качественных ответов от LLM. Правильно составленный промпт – ключ к качественной работе модели.

Основные техники промпт-инжиниринга:

Техника	**Описание**	**Пример**
Zero-shot	Простая инструкция без примеров	"Напиши письмо клиенту"
Few-shot	Несколько примеров в промпте	"Пример 1: ... Пример 2: ... Теперь выполни"
Chain-of-Thought	Пошаговое мышление	"Шаг 1: ... Шаг 2: ..."
Role-playing	Задание роли модели	"Ты – эксперт по Python"
Format constraint	Ограничение формата ответа	"Ответь в формате JSON"

Структура эффективного промпта:
1. Роль – кем должна быть модель;
2. Инструкция – что нужно сделать;
3. Контекст – дополнительная информация;
4. Ограничения – требования к формату и содержанию;
5. Примеры – демонстрация желаемого результата.

5. Результат: В ходе работы должны быть разработаны и протестированы промпты для различных типов задач, создана библиотека шаблонов промптов.

Пример промтов:

```
# Few-shot пример
few_shot_prompt = """
Пример 1: Вопрос: "Как дела?" Ответ: "Отлично!"
Пример 2: Вопрос: "Что нового?" Ответ: "Всё хорошо!"
Теперь ответь на вопрос: "Как настроение?"
"""

# Chain-of-Thought
cot_prompt = """
Вопрос: У Ивана было 5 яблок, он отдал 2. Сколько осталось?
Давай подумаем пошагово:
1. Изначально было 5 яблок.
2. Он отдал 2 яблока.
3. Осталось: 5 - 2 = 3.
Ответ: 3 яблока.
"""

# Role-playing
role_prompt = "Ты — эксперт по Python. Объясни, что такое декораторы."
```

Критерии оценки.

«5» «отлично»

- студент продемонстрировал готовность и способность к самостоятельному выполнению видов работ в стандартных ситуациях;
- все работы выполнил правильно и уверенно;
- проявил активность в освоении новых практических умений;
- содержание отчета по практическим работам полноценно отражает информацию по выполненным видам работ, установленных программой практики;
- оформление отчета по практическим работам полностью соответствует предъявляемым требованиям;
- при выполнении практических работ студент уверенно продемонстрировал обладание необходимыми компонентами общих и профессиональных компетенций;

«4» «хорошо»

- студент продемонстрировал готовность и способность к самостоятельному выполнению видов работ в стандартных ситуациях;
- все работы выполнил правильно и уверенно;

- содержание отчета по практическим работам в целом правильно отражает информацию по выполненным видам работ;
- оформление отчета по практическим работам соответствует предъявляемым требованиям, однако имеются замечания (некоторая неаккуратность, неполное описание проделанной работы и освоенных умений);
- при выполнении практических работ студент продемонстрировал обладание компонентами общих и профессиональных компетенций;
- характеристики студента положительные, имеются отдельные замечания и рекомендации;

«3» «удовлетворительно»

- при выполнении видов работ студент нуждался во внешнем сопровождении и контроле;
- в отчете по практическим работам содержится описание видов выполняемых работ согласно программе дисциплины, однако, записи неполные и неточные;
- отчет по практическим работам оформлен с отклонениями от установленных требований;
- при выполнении практических работ студент продемонстрировал обладание отдельными компонентами общих и профессиональных компетенций;

«**2» «неудовлетворительно»**

– студент не справился с запланированными видами работ;

– при выполнении практических работ студент продемонстрировал слабое обладание отдельными компонентами общих и профессиональных компетенций;

– отчет студентом не подготовлен.

Практическое занятие №23. Тестирование и отладка приложений

Проверяемые результаты обучения:

- ПК 1.2 Разрабатывать программные модули в соответствии с техническим заданием
- ПК 1.3 Оформлять программный код в соответствии с техническим заданием
- ПК 1.5 Выполнять отладку программных модулей с использованием специализированных программных средств
- ПК 1.6 Выполнять тестирование программного кода
- ПК 1.7 Составлять тестовые сценарии
- ПК 1.9 Разрабатывать программный код мобильных приложений в соответствии с техническим заданием и утвержденными спецификациями
- ПК 1.10 Осуществлять оптимизацию программного кода мобильных приложений в соответствии с техническим заданием, оценивая трудоемкость и сроки выполнения работ.
- ПК 1.13 Выполнять разработку программных модулей

- ПК 1.14 Выполнять тестирование программных модулей и ИИ-решений
- ПК 1.21 Осуществлять защиту данных в мобильных приложениях
- ПК 1.25 Управлять версиями и кодом

1. Цель практической работы.

Освоить навыки тестирования и отладки ИИ-приложений, выявления и устранения ошибок при работе с LLM.

2. Задачи практической работы.

– Провести комплексное тестирование разработанного приложения;
– Выявить и устранить ошибки;
– Оптимизировать производительность приложения;
– Задокументировать результаты тестирования.

3. Этапы выполнения.

1. Подготовьте план тестирования разработанного приложения:
    1. Определите тестируемые функции и сценарии;
    2. Подготовьте тестовые данные и запросы;
    3. Определите критерии успешного тестирования.
2. Проведите функциональное тестирование:
    1. Проверьте все команды и функции приложения;
    2. Проверьте работу всех модулей;
    3. Проверьте обработку ошибок.
3. Проведите тестирование качества ответов:
    1. Проверьте точность ответов на фактических вопросах;
    2. Проверьте креативность и релевантность на творческих запросах;
    3. Проверьте корректность структурированных ответов.
4. Проведите тестирование производительности:
    1. Измерьте время ответа на различные запросы;
    2. Проверьте потребление памяти и ресурсов;
    3. Оптимизируйте "узкие места".
5. Устраните обнаруженные ошибки и задокументируйте их:
    1. Создайте список ошибок и способов их решения;
    2. Внесите улучшения в код;
    3. Обновите документацию.

4. Теоретический материал.

Тестирование ИИ-приложений – это процесс проверки работоспособности и качества приложения, выявления и устранения ошибок и

проблем.

Типы тестирования ИИ-приложений:

Тип	**Описание**	**Что проверяется**
Функциональное	Проверка работы функций	Соответствие требованиям
Интеграционное	Проверка взаимодействия компонентов	Совместная работа модулей
Производительности	Проверка скорости работы	Время ответа, потребление ресурсов
Качества ответов	Проверка качества генерации	Точность, релевантность
Нагрузочное	Проверка при большой нагрузке	Стабильность, масштабируемость

Типичные ошибки при работе с LLM:

Ошибка	**Причина**	**Решение**
Долгий ответ	Сложный промпт, большая модель	Оптимизация промпта
Некорректный формат	Неправильная инструкция	Уточнение формата
Галлюцинации	Недостаток контекста	Добавление контекста
Обрыв ответа	Достигнут max_tokens	Увеличение лимита

5. Результат: В ходе работы должно быть проведено комплексное тестирование разработанного ИИ-приложения, выявлены и устранены ошибки, проведена оптимизация производительности. В отчете должны быть представлены: план тестирования, результаты всех видов тестирования, список обнаруженных ошибок и способов их решения, внесенные улучшения, примеры работы приложения до и после оптимизации.

Критерии оценки.

«5» «отлично»

- студент продемонстрировал готовность и способность к самостоятельному выполнению видов работ в стандартных ситуациях;
- все работы выполнил правильно и уверенно;
- проявил активность в освоении новых практических умений;
- содержание отчета по практическим работам полноценно отражает информацию по выполненным видам работ, установленных программой практики;
- оформление отчета по практическим работам полностью соответствует предъявляемым требованиям;
- при выполнении практических работ студент уверенно продемонстрировал обладание необходимыми компонентами общих и профессиональных компетенций;

«4» «хорошо»

- студент продемонстрировал готовность и способность к самостоятельному выполнению видов работ в стандартных ситуациях;
- все работы выполнил правильно и уверенно;
- содержание отчета по практическим работам в целом правильно отражает информацию по выполненным видам работ;
- оформление отчета по практическим работам соответствует предъявляемым требованиям, однако имеются замечания (некоторая неаккуратность, неполное описание проделанной работы и освоенных умений);
- при выполнении практических работ студент продемонстрировал обладание компонентами общих и профессиональных компетенций;
- характеристики студента положительные, имеются отдельные замечания и рекомендации;

«3» «удовлетворительно»

- при выполнении видов работ студент нуждался во внешнем сопровождении и контроле;
- в отчете по практическим работам содержится описание видов выполняемых работ согласно программе дисциплины, однако, записи неполные и неточные;
- отчет по практическим работам оформлен с отклонениями от установленных требований;

- при выполнении практических работ студент продемонстрировал обладание отдельными компонентами общих и профессиональных компетенций;

«2» «неудовлетворительно»

– студент не справился с запланированными видами работ;

– при выполнении практических работ студент продемонстрировал слабое обладание отдельными компонентами общих и профессиональных компетенций;

– отчет студентом не подготовлен.

3.2 Задания для оценки освоения УП.02.01

Раздел 1. Управление и автоматизация баз данных

Тема 1.1. Установка и настройка программного обеспечения для администрирования баз данных

Практическое занятие №1. Установка СУБД и проверка системных требований

Продолжительность: 4 академических часа(ов).

Проверяемые компетенции:

ПК 2.1 — Выявлять проблемы, возникающие в процессе эксплуатации баз данных.

ПК 2.2 — Осуществлять процедуры администрирования баз данных.

ПК 2.3 — Проводить аудит систем безопасности баз данных с использованием регламентов по защите информации.

ПК 2.5 — Подготавливать данные для базы знаний.

ПК 2.6 — Осуществлять управление базами данных.

ПК 2.7 — Осуществлять автоматизацию баз данных.

Цель занятия: Проверка компьютера и установка PostgreSQL.

Теоретические сведения и объяснение: Для учебного стенда используется PostgreSQL 16 или более новая совместимая версия. Перед установкой необходимо узнать версию ОС, объём ОЗУ, свободное место и наличие прав администратора. СУБД — серверная программа, которая хранит данные и обрабатывает SQL-запросы.

Оборудование и исходные материалы: учебный ПК, терминал, PostgreSQL/pgAdmin, тренировочная БД и текстовый редактор.

Порядок выполнения и примеры

Этап 1. Проверка исходной конфигурации.

Откройте свойства системы и определите ОС, разрядность, ОЗУ и свободное место на диске. Заполните таблицу «Параметр / Фактическое значение / Требование установщика / Соответствует». Требования берите из документации именно выбранной версии PostgreSQL.

Пояснение и пример: Не следует подменять фактическую проверку фразой «компьютер подходит». В отчёте должны быть конкретные значения.

Этап 2. Выбор способа установки.

Выберите официальный установщик для ОС либо пакетный менеджер. Запишите версию сервера, каталог данных, номер порта (обычно 5432) и имя служебной учётной записи. Сформулируйте, какие параметры можно менять при установке.

Пояснение и пример: Для Windows допустим графический установщик; для Linux пример: sudo apt install postgresql (доступность пакета зависит от дистрибутива).

Этап 3. Установка сервера.

Запустите установку, создайте надёжный пароль администратора, убедитесь, что служба PostgreSQL запускается. Не размещайте пароль и секреты в скриншотах отчёта.

Пояснение и пример: Если порт 5432 занят, зафиксируйте причину и выберите согласованный свободный порт.

Этап 4. Проверка подключения.
Выполните psql --version, затем подключитесь локально командой psql -h localhost -p 5432 -U postgres -d postgres. В SQL-консоли выполните SELECT version(); и SELECT current_database(), current_user;.
Пояснение и пример: Результатом должен быть ответ сервера и отображение текущей БД и роли; ошибка подключения требует диагностики службы, порта и пароля.
Этап 5. Создание пробной базы.
Создайте учебную БД командой CREATE DATABASE training_db; и подключитесь к ней. Запишите в отчёт выполненные команды и вывод.
Пояснение и пример: Успех: новая база доступна в списке через psql -l или средство администрирования.

Что должно получиться

Снимок параметров ПК, версия СУБД, журнал установки, два SQL-результата и вывод о соответствии требованиям.

Самопроверка и контрольные вопросы

1. Какие действия в работе №1 подтверждают достижение её цели?
2. Какая проверка показала, что получен правильный результат, и что именно она измеряет?
3. Какую типичную ошибку можно допустить в этой работе и как её обнаружить?
4. Какие данные, настройки или права нельзя переносить из учебной среды в рабочую без согласования?

Критерии оценивания

Оценка	**Требования к выполнению**
5 «отлично»	Выполнены все этапы; результат воспроизводим, проверки (включая негативные, если предусмотрены) подтверждены; примеры адаптированы к заданию; отчёт содержит команды/артефакты и самостоятельные объяснения.
4 «хорошо»	Основные этапы и контрольные результаты выполнены; допущены единичные неточности в объяснениях или оформлении, не влияющие на корректность и безопасность результата.
3 «удовлетворительно»	Получен минимальный рабочий результат, но часть проверок или пояснений отсутствует; есть ошибки, устранённые с подсказкой; отчёт неполон.
2 «неудовлетворительно»	Основной результат отсутствует либо ошибки не устранены; ключевые этапы или доказательства проверки не представлены; допущены критические нарушения безопасности учебного стенда.

Оценивание проводится по фактически продемонстрированному результату, качеству объяснения шагов и полноте отчёта.

Практическое занятие №2. Настройка параметров серверной службы
Продолжительность: 6 академических часа(ов).
Проверяемые компетенции:
ПК 2.1 — Выявлять проблемы, возникающие в процессе эксплуатации баз данных.

ПК 2.2 — Осуществлять процедуры администрирования баз данных.
ПК 2.3 — Проводить аудит систем безопасности баз данных с использованием регламентов по защите информации.
ПК 2.5 — Подготавливать данные для базы знаний.
ПК 2.6 — Осуществлять управление базами данных.
ПК 2.7 — Осуществлять автоматизацию баз данных.

Цель занятия: Изменение настроек тестового сервера.

Теоретические сведения и объяснение: Файл postgresql.conf задаёт конфигурацию экземпляра; некоторые параметры вступают в силу после перезагрузки настроек, другие — после перезапуска службы. Применяйте изменения только на учебном экземпляре.

Оборудование и исходные материалы: учебный ПК, терминал, PostgreSQL/pgAdmin, тренировочная БД и текстовый редактор.

Порядок выполнения и примеры

Этап 1. Фиксация начального состояния.

Подключитесь к PostgreSQL и выполните SHOW port; SHOW max_connections; SHOW shared_buffers; SHOW log_min_duration_statement;. Сохраните текущие значения и путь к конфигурации: SHOW config_file;.

Пояснение и пример: Исходные значения нужны, чтобы доказать влияние настройки и при необходимости откатить её.

Этап 2. План изменения.

Для лабораторного стенда выберите безопасную настройку: log_min_duration_statement = 500 (миллисекунд). Определите по документации, достаточно ли перечитать конфигурацию.

Пояснение и пример: Настройка регистрирует в журнале запросы длительнее порога; это не ограничение времени исполнения.

Этап 3. Изменение параметров.

При наличии прав суперпользователя выполните ALTER SYSTEM SET log_min_duration_statement = 500; затем SELECT pg_reload_conf();. Проверьте SHOW log_min_duration_statement;.

Пояснение и пример: Альтернатива: отредактируйте postgresql.conf и перечитайте конфигурацию. Не редактируйте production без согласования.

Этап 4. Эксперимент.

Выполните SELECT pg_sleep(0.7);, после чего найдите запись о длительности запроса в журнале сервера. Зафиксируйте время и содержание сообщения.

Пояснение и пример: Если запись не появилась, проверьте параметры logging_collector, log_destination и расположение журнала.

Этап 5. Откат и объяснение.

При необходимости отмените изменение: ALTER SYSTEM RESET log_min_duration_statement; SELECT pg_reload_conf();. Сравните значения «до/после», объясните разницу между reload и restart.

Пояснение и пример: Отчёт должен содержать не только команды, но и интерпретацию их результата.

Что должно получиться

Таблица параметров до и после, выбранный способ настройки, фрагмент журнала и объяснение проверки.

Самопроверка и контрольные вопросы

1. Какие действия в работе №2 подтверждают достижение её цели?
2. Какая проверка показала, что получен правильный результат, и что именно она измеряет?
3. Какую типичную ошибку можно допустить в этой работе и как её обнаружить?
4. Какие данные, настройки или права нельзя переносить из учебной среды в рабочую без согласования?

Критерии оценивания

Оценка	**Требования к выполнению**

5 «отлично»	Выполнены все этапы; результат воспроизводим, проверки (включая негативные, если предусмотрены) подтверждены; примеры адаптированы к заданию; отчёт содержит команды/артефакты и самостоятельные объяснения.
4 «хорошо»	Основные этапы и контрольные результаты выполнены; допущены единичные неточности в объяснениях или оформлении, не влияющие на корректность и безопасность результата.
3 «удовлетворительно»	Получен минимальный рабочий результат, но часть проверок или пояснений отсутствует; есть ошибки, устранённые с подсказкой; отчёт неполон.
2 «неудовлетворительно»	Основной результат отсутствует либо ошибки не устранены; ключевые этапы или доказательства проверки не представлены; допущены критические нарушения безопасности учебного стенда.

Оценивание проводится по фактически продемонстрированному результату, качеству объяснения шагов и полноте отчёта.

Практическое занятие №3. Подключение и проверка средств администрирования

Продолжительность: 6 академических часа(ов).

Проверяемые компетенции:

ПК 2.1 — Выявлять проблемы, возникающие в процессе эксплуатации баз данных.
ПК 2.2 — Осуществлять процедуры администрирования баз данных.
ПК 2.3 — Проводить аудит систем безопасности баз данных с использованием регламентов по защите информации.
ПК 2.5 — Подготавливать данные для базы знаний.
ПК 2.6 — Осуществлять управление базами данных.
ПК 2.7 — Осуществлять автоматизацию баз данных.

Цель занятия: Подключение pgAdmin и диагностика запросов.

Теоретические сведения и объяснение: Средство администрирования помогает видеть базы, роли, схемы и план выполнения запросов. SQL остаётся основным способом проверить, что подключение действительно работает.

Оборудование и исходные материалы: учебный ПК, терминал, PostgreSQL/pgAdmin, тренировочная БД и текстовый редактор.

Порядок выполнения и примеры

Этап 1. Создание соединения.

Запустите pgAdmin, добавьте сервер: Host = localhost, Port = 5432, Maintenance database = postgres, Username = postgres. Пароль вводите в защищённое поле.

Пояснение и пример: Сохранение пароля в публичном отчёте недопустимо.

Этап 2. Проверка объектов.

Откройте список databases, найдите training_db; в Query Tool выполните SELECT current_user, current_database(), inet_server_port();. Сопоставьте результат с настройками подключения.

Пояснение и пример: Это доказывает, к какому экземпляру и от чьего имени подключён студент.

Этап 3. Создание тестовых данных.

В учебной БД выполните CREATE TABLE IF NOT EXISTS demo_orders(id integer PRIMARY KEY, amount numeric(10,2)); INSERT INTO demo_orders VALUES (1,100),(2,250),(3,70) ON CONFLICT (id) DO NOTHING;.

Пояснение и пример: Таблица создаётся в учебной схеме и не затрагивает рабочие данные.
Этап 4. Анализ SQL.
Выполните EXPLAIN ANALYZE SELECT * FROM demo_orders WHERE amount > 100;. Найдите узлы плана, фактическое число строк и время исполнения.
Пояснение и пример: На маленькой таблице последовательное чтение нормально; наличие Seq Scan само по себе не означает ошибку.
Этап 5. Диагностика ошибки.
Умышленно выполните SELECT * FROM demo_order; (неправильное имя). Зафиксируйте сообщение и исправьте запрос.
Пояснение и пример: Опишите, как отличить ошибки SQL от ошибок сети, имени сервера и аутентификации.

Что должно получиться

Скриншот дерева объектов без секретов, SQL-код, план выполнения и классификация обнаруженной ошибки.

Самопроверка и контрольные вопросы

1. Какие действия в работе №3 подтверждают достижение её цели?
2. Какая проверка показала, что получен правильный результат, и что именно она измеряет?
3. Какую типичную ошибку можно допустить в этой работе и как её обнаружить?
4. Какие данные, настройки или права нельзя переносить из учебной среды в рабочую без согласования?

Критерии оценивания

Оценка	**Требования к выполнению**
5 «отлично»	Выполнены все этапы; результат воспроизводим, проверки (включая негативные, если предусмотрены) подтверждены; примеры адаптированы к заданию; отчёт содержит команды/артефакты и самостоятельные объяснения.
4 «хорошо»	Основные этапы и контрольные результаты выполнены; допущены единичные неточности в объяснениях или оформлении, не влияющие на корректность и безопасность результата.
3 «удовлетворительно»	Получен минимальный рабочий результат, но часть проверок или пояснений отсутствует; есть ошибки, устранённые с подсказкой; отчёт неполон.
2 «неудовлетворительно»	Основной результат отсутствует либо ошибки не устранены; ключевые этапы или доказательства проверки не представлены; допущены критические нарушения безопасности учебного стенда.

Оценивание проводится по фактически продемонстрированному результату, качеству объяснения шагов и полноте отчёта.

Тема 1.2. Управление доступом и обеспечение безопасности баз данных

Практическое занятие №4. Создание учетных записей и ролей
Продолжительность: 4 академических часа(ов).
Проверяемые компетенции:
ПК 2.1 — Выявлять проблемы, возникающие в процессе эксплуатации баз данных.
ПК 2.2 — Осуществлять процедуры администрирования баз данных.
ПК 2.3 — Проводить аудит систем безопасности баз данных с использованием регламентов по защите информации.
ПК 2.5 — Подготавливать данные для базы знаний.
ПК 2.6 — Осуществлять управление базами данных.
ПК 2.7 — Осуществлять автоматизацию баз данных.
Цель занятия: Создание учебных пользователей и ролей.
Теоретические сведения и объяснение: В PostgreSQL роли используются и как учётные записи для подключения, и как наборы прав. Атрибут LOGIN разрешает подключение, NOLOGIN — удобен для групповых ролей.
Оборудование и исходные материалы: учебный ПК, терминал, PostgreSQL/pgAdmin, тренировочная БД и текстовый редактор.

Порядок выполнения и примеры

Этап 1. Проектирование ролей.
Опишите трёх участников: администратор, оператор чтения, оператор редактирования. Заполните таблицу «Роль / Назначение / LOGIN / Ожидаемые права».
Пояснение и пример: Роль и реальный человек — разные сущности: несколько пользователей могут входить в одну групповую роль.
Этап 2. Создание ролей.
На учебном сервере от имени администратора выполните CREATE ROLE reader_group NOLOGIN; CREATE ROLE editor_group NOLOGIN; CREATE ROLE student_reader LOGIN PASSWORD 'Use-a-unique-lab-password!';.
Пояснение и пример: Пароль в примере — заглушка; замените собственным секретом и не включайте его в отчёт.
Этап 3. Назначение членства.
Выполните GRANT reader_group TO student_reader;. Проверьте членство через SELECT r.rolname FROM pg_roles r WHERE r.rolname LIKE '%group' OR r.rolname = 'student_reader';.
Пояснение и пример: Создание группы само по себе не наделяет её доступом к таблицам.
Этап 4. Проверка входа.
Откройте новое соединение с логином student_reader и убедитесь, что оно проходит. Затем попробуйте подключиться именем reader_group и объясните отказ.
Пояснение и пример: У групповой роли NOLOGIN прямое подключение запрещено.
Этап 5. Анализ результатов.
Отразите различия между пользователем, ролью и членством. Укажите, кто может назначать привилегии и почему запрещено использовать администратора для обычных операций.
Пояснение и пример: Запишите конкретное поведение системы для каждой роли.

Что должно получиться

Матрица ролей, команды создания без настоящих паролей, проверка входа и пояснение принципа наименьших привилегий.

Самопроверка и контрольные вопросы

1. Какие действия в работе №4 подтверждают достижение её цели?
2. Какая проверка показала, что получен правильный результат, и что именно она измеряет?
3. Какую типичную ошибку можно допустить в этой работе и как её обнаружить?
4. Какие данные, настройки или права нельзя переносить из учебной среды в рабочую без согласования?

Критерии оценивания

Оценка	Требования к выполнению
5 «отлично»	Выполнены все этапы; результат воспроизводим, проверки (включая негативные, если предусмотрены) подтверждены; примеры адаптированы к заданию; отчёт содержит команды/артефакты и самостоятельные объяснения.
4 «хорошо»	Основные этапы и контрольные результаты выполнены; допущены единичные неточности в объяснениях или оформлении, не влияющие на корректность и безопасность результата.
3 «удовлетворительно»	Получен минимальный рабочий результат, но часть проверок или пояснений отсутствует; есть ошибки, устранённые с подсказкой; отчёт неполон.
2 «неудовлетворительно»	Основной результат отсутствует либо ошибки не устранены; ключевые этапы или доказательства проверки не представлены; допущены критические нарушения безопасности учебного стенда.

Оценивание проводится по фактически продемонстрированному результату, качеству объяснения шагов и полноте отчёта.

Практическое занятие №5. Назначение прав доступа и ограничений

Продолжительность: 6 академических часа(ов).

Проверяемые компетенции:

ПК 2.1 — Выявлять проблемы, возникающие в процессе эксплуатации баз данных.
ПК 2.2 — Осуществлять процедуры администрирования баз данных.
ПК 2.3 — Проводить аудит систем безопасности баз данных с использованием регламентов по защите информации.
ПК 2.5 — Подготавливать данные для базы знаний.
ПК 2.6 — Осуществлять управление базами данных.
ПК 2.7 — Осуществлять автоматизацию баз данных.

Цель занятия: Разграничение прав на таблицы.

Теоретические сведения и объяснение: GRANT и REVOKE регулируют полномочия; права на схему, базу и таблицы проверяются отдельно. Для проверки важны успешный и запрещённый сценарии.

Оборудование и исходные материалы: учебный ПК, терминал, PostgreSQL/pgAdmin, тренировочная БД и текстовый редактор.

Порядок выполнения и примеры

Этап 1. Подготовка объектов.

Под администратором создайте схему lab и таблицу lab.clients(id int PRIMARY KEY, name text NOT NULL); добавьте две строки.

Пояснение и пример: Используйте учебные, а не персональные данные.

Этап 2. Установка минимальных прав.

Выполните REVOKE ALL ON SCHEMA lab FROM PUBLIC; GRANT USAGE ON SCHEMA lab TO reader_group; GRANT SELECT ON lab.clients TO reader_group; GRANT USAGE ON SCHEMA lab TO editor_group; GRANT SELECT, INSERT, UPDATE ON lab.clients TO editor_group;.

Пояснение и пример: Для доступа к объектам в схеме требуется USAGE на схему; это частая причина отказа.

Этап 3. Тест чтения.
Подключитесь как student_reader и выполните SELECT * FROM lab.clients;. Запишите фактический результат, а затем попытайтесь INSERT INTO lab.clients VALUES (3,'Тест');.
Пояснение и пример: Ожидается успешный SELECT и отказ на INSERT.
Этап 4. Тест изменения.
Создайте учебную роль student_editor с LOGIN и включите её в editor_group; проверьте INSERT и UPDATE. Попробуйте DELETE и зафиксируйте отказ.
Пояснение и пример: Фиксируйте и положительные, и отрицательные тесты; удаление данных не должно быть разрешено.
Этап 5. Отзыв полномочий.
Отзовите SELECT у reader_group: REVOKE SELECT ON lab.clients FROM reader_group; повторите SELECT от имени student_reader и объясните изменение.
Пояснение и пример: Если у пользователя есть прямые или наследуемые права от других ролей, результат может отличаться: исследуйте их.

Что должно получиться

Матрица «роль × действие», команды GRANT/REVOKE, доказательства минимум трёх разрешённых/запрещённых проверок.

Самопроверка и контрольные вопросы

1. Какие действия в работе №5 подтверждают достижение её цели?
2. Какая проверка показала, что получен правильный результат, и что именно она измеряет?
3. Какую типичную ошибку можно допустить в этой работе и как её обнаружить?
4. Какие данные, настройки или права нельзя переносить из учебной среды в рабочую без согласования?

Критерии оценивания

Оценка	**Требования к выполнению**
5 «отлично»	Выполнены все этапы; результат воспроизводим, проверки (включая негативные, если предусмотрены) подтверждены; примеры адаптированы к заданию; отчёт содержит команды/артефакты и самостоятельные объяснения.
4 «хорошо»	Основные этапы и контрольные результаты выполнены; допущены единичные неточности в объяснениях или оформлении, не влияющие на корректность и безопасность результата.
3 «удовлетворительно»	Получен минимальный рабочий результат, но часть проверок или пояснений отсутствует; есть ошибки, устранённые с подсказкой; отчёт неполон.
2 «неудовлетворительно»	Основной результат отсутствует либо ошибки не устранены; ключевые этапы или доказательства проверки не представлены; допущены критические нарушения безопасности учебного стенда.

Оценивание проводится по фактически продемонстрированному результату, качеству объяснения шагов и полноте отчёта.

Практическое занятие №6. Настройка аудита действий пользователей
Продолжительность: 6 академических часа(ов).
Проверяемые компетенции:
ПК 2.1 — Выявлять проблемы, возникающие в процессе эксплуатации баз данных.
ПК 2.2 — Осуществлять процедуры администрирования баз данных.
ПК 2.3 — Проводить аудит систем безопасности баз данных с использованием регламентов по защите информации.
ПК 2.5 — Подготавливать данные для базы знаний.
ПК 2.6 — Осуществлять управление базами данных.
ПК 2.7 — Осуществлять автоматизацию баз данных.
Цель занятия: Сбор и анализ событий доступа.
Теоретические сведения и объяснение: Аудит нужен для восстановления последовательности действий. В PostgreSQL базовый журнал позволяет фиксировать подключения и отдельные SQL-команды; расширение pgAudit возможно при наличии в стенде.
Оборудование и исходные материалы: учебный ПК, терминал, PostgreSQL/pgAdmin, тренировочная БД и текстовый редактор.

Порядок выполнения и примеры

Этап 1. Формулировка требований.
Определите события аудита: вход, неудачная попытка входа, изменения таблиц и ролей. Для каждого события запишите «зачем фиксируется / кто проверяет / как долго хранится».
Пояснение и пример: Не требуйте записывать пароли и содержимое секретов в журнал.
Этап 2. Настройка логирования.
На учебном сервере уточните SHOW log_connections; SHOW log_disconnections; SHOW log_statement;. При наличии прав включите log_connections=on и log_disconnections=on; для демонстрации изменения данных временно задайте log_statement='mod' и примените конфигурацию.
Пояснение и пример: Параметр log_statement=all способен создать большой объём журнала и показать чувствительные параметры; для лабораторной работы предпочтителен mod.
Этап 3. Генерация событий.
Создайте два успешных сеанса с разными ролями, выполните тестовый INSERT и UPDATE. Отдельно выполните заведомо запрещённую операцию и отметьте её ошибку.
Пояснение и пример: Ошибки прав могут оказаться в журнале вне зависимости от типа выбранной команды.
Этап 4. Исследование журнала.
Найдите записи по времени события. Сопоставьте пользователя, источник соединения, команду, статус и временную метку; соберите таблицу из минимум четырёх событий.
Пояснение и пример: Если поля пользователя отсутствуют, проверьте формат log_line_prefix.
Этап 5. Разбор инцидента.
Составьте небольшой вывод: какая роль попыталась изменить данные без полномочий и чем это подтверждается. Верните исходные настройки, если они были временными.
Пояснение и пример: Не путайте системное логирование PostgreSQL с полноценным неизменяемым журналом аудита.

Что должно получиться

Таблица событий с временными метками, параметры логирования, пример расследования и рекомендации по безопасному хранению журналов.

Самопроверка и контрольные вопросы

1. Какие действия в работе №6 подтверждают достижение её цели?
2. Какая проверка показала, что получен правильный результат, и что именно она измеряет?
3. Какую типичную ошибку можно допустить в этой работе и как её обнаружить?
4. Какие данные, настройки или права нельзя переносить из учебной среды в рабочую без согласования?

Критерии оценивания

Оценка	Требования к выполнению
5 «отлично»	Выполнены все этапы; результат воспроизводим, проверки (включая негативные, если предусмотрены) подтверждены; примеры адаптированы к заданию; отчёт содержит команды/артефакты и самостоятельные объяснения.
4 «хорошо»	Основные этапы и контрольные результаты выполнены; допущены единичные неточности в объяснениях или оформлении, не влияющие на корректность и безопасность результата.
3 «удовлетворительно»	Получен минимальный рабочий результат, но часть проверок или пояснений отсутствует; есть ошибки, устранённые с подсказкой; отчёт неполон.
2 «неудовлетворительно»	Основной результат отсутствует либо ошибки не устранены; ключевые этапы или доказательства проверки не представлены; допущены критические нарушения безопасности учебного стенда.

Оценивание проводится по фактически продемонстрированному результату, качеству объяснения шагов и полноте отчёта.

Тема 1.3. Резервное копирование и восстановление баз данных

Практическое занятие №7. Планирование резервного копирования

Продолжительность: 4 академических часа(ов).

Проверяемые компетенции:

ПК 2.1 — Выявлять проблемы, возникающие в процессе эксплуатации баз данных.
ПК 2.2 — Осуществлять процедуры администрирования баз данных.
ПК 2.3 — Проводить аудит систем безопасности баз данных с использованием регламентов по защите информации.
ПК 2.5 — Подготавливать данные для базы знаний.
ПК 2.6 — Осуществлять управление базами данных.
ПК 2.7 — Осуществлять автоматизацию баз данных.

Цель занятия: План резервного копирования и восстановления.

Теоретические сведения и объяснение: Резервная копия должна позволять восстановить требуемые данные за приемлемое время. RPO описывает допустимую потерю данных, RTO — допустимое время восстановления.

Оборудование и исходные материалы: учебный ПК, терминал, PostgreSQL/pgAdmin, тренировочная БД и текстовый редактор.

Порядок выполнения и примеры

Этап 1. Анализ учебного сценария.

Предположите, что БД учебного магазина изменяется ежедневно, а потерять разрешено не более суток данных. Сформулируйте RPO = 24 ч и примерный RTO = 2 ч как учебные требования.

Пояснение и пример: Эти цифры — условия учебного кейса, а не универсальные нормы.

Этап 2. Выбор способа.

Сравните pg_dump (логическая копия отдельной БД), pg_dumpall (общая логическая копия кластера) и физическое резервирование (например, pg_basebackup). Опишите ограничения каждого.

Пояснение и пример: pg_dump не резервирует автоматически все глобальные роли и параметры экземпляра.

Этап 3. Разработка расписания.

Для кейса предложите ежедневную копию в 20:00, еженедельную тестовую проверку восстановления и срок хранения 7 дней. Составьте расписание «операция / периодичность / ответственный / место хранения».

Пояснение и пример: Реальная политика хранения определяется требованиями организации.

Этап 4. Модель рисков.

Опишите отказ диска, ошибочное удаление таблицы, утрату пароля к хранилищу резервных копий. Для каждого укажите меру предотвращения и проверки.

Пояснение и пример: Не храните единственную копию на том же диске, что и основную базу.

Этап 5. Критерии проверки.

Создайте чек-лист: файл создан, размер больше нуля, контрольная сумма записана, архив защищён, пробное восстановление завершилось.

Пояснение и пример: Наличие файла без проверки восстановления недостаточно.

Что должно получиться

План резервирования с RPO/RTO, расписание, оценка рисков и чек-лист испытания.

Самопроверка и контрольные вопросы

1. Какие действия в работе №7 подтверждают достижение её цели?
2. Какая проверка показала, что получен правильный результат, и что именно она измеряет?
3. Какую типичную ошибку можно допустить в этой работе и как её обнаружить?
4. Какие данные, настройки или права нельзя переносить из учебной среды в рабочую без согласования?

Критерии оценивания

Оценка	**Требования к выполнению**
5 «отлично»	Выполнены все этапы; результат воспроизводим, проверки (включая негативные, если предусмотрены) подтверждены; примеры адаптированы к заданию; отчёт содержит команды/артефакты и самостоятельные объяснения.
4 «хорошо»	Основные этапы и контрольные результаты выполнены; допущены единичные неточности в объяснениях или оформлении, не влияющие на корректность и безопасность результата.
3 «удовлетворительно»	Получен минимальный рабочий результат, но часть проверок или пояснений отсутствует; есть ошибки, устранённые с подсказкой; отчёт неполон.
2 «неудовлетворительно»	Основной результат отсутствует либо ошибки не устранены; ключевые этапы или доказательства проверки не представлены; допущены критические нарушения безопасности учебного стенда.

Оценивание проводится по фактически продемонстрированному результату, качеству объяснения шагов и полноте отчёта.

Практическое занятие №8. Выполнение и мониторинг резервного копирования
Продолжительность: 6 академических часа(ов).
Проверяемые компетенции:
ПК 2.1 — Выявлять проблемы, возникающие в процессе эксплуатации баз данных.
ПК 2.2 — Осуществлять процедуры администрирования баз данных.
ПК 2.3 — Проводить аудит систем безопасности баз данных с использованием регламентов по защите информации.
ПК 2.5 — Подготавливать данные для базы знаний.
ПК 2.6 — Осуществлять управление базами данных.
ПК 2.7 — Осуществлять автоматизацию баз данных.
Цель занятия: Создание резервной копии и контроль её состояния.
Теоретические сведения и объяснение: pg_dump позволяет экспортировать логическую копию базы. Формат -Fc удобен для pg_restore и выборочного восстановления. Команды выполняйте на учебной базе.
Оборудование и исходные материалы: учебный ПК, терминал, PostgreSQL/pgAdmin, тренировочная БД и текстовый редактор.

Порядок выполнения и примеры

Этап 1. Подготовка контрольных данных.
В training_db создайте таблицу backup_probe(id int PRIMARY KEY, value text); вставьте строки (1,'A'), (2,'B'), (3,'C'). Посчитайте SELECT count(*) FROM backup_probe;.
Пояснение и пример: Контрольное число строк позволяет сравнить исходную и восстановленную версии.
Этап 2. Создание архива.
В терминале выполните pg_dump -h localhost -U postgres -d training_db -Fc -f training_db.backup. Выберите каталог, доступный для записи; пароль передавайте интерактивно или через защищённый механизм.
Пояснение и пример: Параметр -Fc задаёт custom-формат, -f — путь к файлу.
Этап 3. Проверка статуса.
Проверьте код завершения утилиты, существование файла и его размер. Выполните pg_restore -l training_db.backup и найдите в списке backup_probe.
Пояснение и пример: Вывод списка объектов подтверждает читаемость оглавления архива, но не заменяет полного восстановления.
Этап 4. Контроль целостности файла.
Рассчитайте SHA-256 архива доступной системной командой (например, sha256sum training_db.backup на Linux). Перенесите копию в отдельный учебный каталог и убедитесь, что хеш совпадает.
Пояснение и пример: Фиксируйте дату, размер, хеш и расположение, без публикации секретных путей.
Этап 5. Диагностика ошибки.
Смоделируйте неверный путь назначения или отсутствие прав на каталог и объясните полученное сообщение. После устранения причины выполните резервирование повторно.
Пояснение и пример: Изменять права файлов на «доступно всем» ради устранения ошибки запрещено.

Что должно получиться

Файл копии, перечень объектов, хеш и журнал с успешной и исправленной ошибкой.

Самопроверка и контрольные вопросы

1. Какие действия в работе №8 подтверждают достижение её цели?
2. Какая проверка показала, что получен правильный результат, и что именно она измеряет?
3. Какую типичную ошибку можно допустить в этой работе и как её обнаружить?
4. Какие данные, настройки или права нельзя переносить из учебной среды в рабочую без согласования?

Критерии оценивания

Оценка	Требования к выполнению
5 «отлично»	Выполнены все этапы; результат воспроизводим, проверки (включая негативные, если предусмотрены) подтверждены; примеры адаптированы к заданию; отчёт содержит команды/артефакты и самостоятельные объяснения.
4 «хорошо»	Основные этапы и контрольные результаты выполнены; допущены единичные неточности в объяснениях или оформлении, не влияющие на корректность и безопасность результата.
3 «удовлетворительно»	Получен минимальный рабочий результат, но часть проверок или пояснений отсутствует; есть ошибки, устранённые с подсказкой; отчёт неполон.
2 «неудовлетворительно»	Основной результат отсутствует либо ошибки не устранены; ключевые этапы или доказательства проверки не представлены; допущены критические нарушения безопасности учебного стенда.

Оценивание проводится по фактически продемонстрированному результату, качеству объяснения шагов и полноте отчёта.

Практическое занятие №9. Восстановление БД и проверка целостности

Продолжительность: 6 академических часа(ов).

Проверяемые компетенции:

ПК 2.1 — Выявлять проблемы, возникающие в процессе эксплуатации баз данных.
ПК 2.2 — Осуществлять процедуры администрирования баз данных.
ПК 2.3 — Проводить аудит систем безопасности баз данных с использованием регламентов по защите информации.
ПК 2.5 — Подготавливать данные для базы знаний.
ПК 2.6 — Осуществлять управление базами данных.
ПК 2.7 — Осуществлять автоматизацию баз данных.

Цель занятия: Восстановление в отдельную базу и проверка результата.

Теоретические сведения и объяснение: Восстановление выполняется не поверх рабочего экземпляра, а сначала на отдельном тестовом объекте. После него нужна проверка данных, ограничений и доступности запросов.

Оборудование и исходные материалы: учебный ПК, терминал, PostgreSQL/pgAdmin, тренировочная БД и текстовый редактор.

Порядок выполнения и примеры

Этап 1. Создание базы назначения.

Создайте пустую базу training_restore: createdb -h localhost -U postgres training_restore или CREATE DATABASE training_restore;. Убедитесь, что используете нужное подключение.

Пояснение и пример: Не восстанавливайте учебный архив в production.

Этап 2. Импорт архива.

Выполните pg_restore -h localhost -U postgres -d training_restore --no-owner training_db.backup. Зафиксируйте код завершения и предупреждения.

Пояснение и пример: Для сложных баз воссоздание расширений и прав может требовать отдельной настройки.

Этап 3. Сопоставление данных.
Подключитесь к обеим базам и выполните SELECT count(*) FROM backup_probe; SELECT * FROM backup_probe ORDER BY id;. Оформите таблицу «Проверка / Исходная БД / Восстановленная БД / Итог».
Пояснение и пример: Ожидается одинаковое количество и содержание трёх строк.
Этап 4. Проверка ограничений.
В восстановленной базе попробуйте повторно INSERT INTO backup_probe VALUES (1,'duplicate');. Ожидайте ошибку уникального ключа. Затем выполните корректный SELECT.
Пояснение и пример: Проверка ограничения показывает, что восстановилась не только информация, но и структура.
Этап 5. Протокол испытания.
Запишите длительность восстановления, выявленные проблемы и план действий в случае неуспеха. Сравните длительность с RTO из занятия №7.
Пояснение и пример: Тест считается завершённым только после документирования фактических проверок.

Что должно получиться

Протокол восстановления, сравнение строк, результаты негативного теста и заключение о достижении RTO.

Самопроверка и контрольные вопросы

1. Какие действия в работе №9 подтверждают достижение её цели?
2. Какая проверка показала, что получен правильный результат, и что именно она измеряет?
3. Какую типичную ошибку можно допустить в этой работе и как её обнаружить?
4. Какие данные, настройки или права нельзя переносить из учебной среды в рабочую без согласования?

Критерии оценивания

Оценка	**Требования к выполнению**
5 «отлично»	Выполнены все этапы; результат воспроизводим, проверки (включая негативные, если предусмотрены) подтверждены; примеры адаптированы к заданию; отчёт содержит команды/артефакты и самостоятельные объяснения.
4 «хорошо»	Основные этапы и контрольные результаты выполнены; допущены единичные неточности в объяснениях или оформлении, не влияющие на корректность и безопасность результата.
3 «удовлетворительно»	Получен минимальный рабочий результат, но часть проверок или пояснений отсутствует; есть ошибки, устранённые с подсказкой; отчёт неполон.
2 «неудовлетворительно»	Основной результат отсутствует либо ошибки не устранены; ключевые этапы или доказательства проверки не представлены; допущены критические нарушения безопасности учебного стенда.

Оценивание проводится по фактически продемонстрированному результату, качеству объяснения шагов и полноте отчёта.

Раздел 2. Технология разработки и защиты баз данных

Тема 2.1. Проектирование реляционной базы данных

Практическое занятие №10. Анализ предметной области и требований
Продолжительность: 4 академических часа(ов).
Проверяемые компетенции:
ПК 2.1 — Выявлять проблемы, возникающие в процессе эксплуатации баз данных.
ПК 2.2 — Осуществлять процедуры администрирования баз данных.
ПК 2.3 — Проводить аудит систем безопасности баз данных с использованием регламентов по защите информации.
ПК 2.4 — Формировать требования хранилищ банка данных для обучения.
ПК 2.5 — Подготавливать данные для базы знаний.
ПК 2.8 — Разрабатывать объекты баз данных NoSQL.
Цель занятия: Сбор требований для базы учебного магазина.
Теоретические сведения и объяснение: Предметная область — магазин учебных товаров: покупатели оформляют заказы, каждый заказ включает товары и количество. Требования должны описывать сущности, действия и ограничения без привязки к конкретной СУБД.
Оборудование и исходные материалы: учебный ПК, терминал, PostgreSQL/pgAdmin, тренировочная БД и текстовый редактор.

Порядок выполнения и примеры

Этап 1. Выделение участников.
Назовите роли: покупатель, менеджер и администратор. Для каждой составьте минимум три действия с данными (просмотр, создание, корректировка, анализ).
Пояснение и пример: Например: менеджер должен найти заказы покупателя за выбранный период.
Этап 2. Описание бизнес-событий.
Запишите сценарий «оформление заказа»: выбор покупателя, добавление позиции, указание количества, расчёт суммы и подтверждение. Обозначьте входные и выходные сведения.
Пояснение и пример: Сумма позиции = количество × цена на момент продажи; текущая цена товара может измениться позже.
Этап 3. Составление словаря данных.
Составьте таблицу «Поле / Смысл / Тип значения / Обязательность / Пример»: email покупателя, наименование товара, дата заказа, количество, цена на момент заказа.
Пояснение и пример: Пример: quantity — положительное целое, обязательно, значение 2.
Этап 4. Фиксация правил.
Запишите не менее шести правил: у заказа один покупатель; у заказа не менее одной позиции; quantity > 0; цена неотрицательна; email покупателя уникален при наличии; дата заказа обязательна.
Пояснение и пример: Укажите, какие правила обеспечивает БД, а какие — приложение.
Этап 5. Проверка требований.
Составьте пять вопросов заказчику и два тестовых сценария, позволяющих подтвердить полноту требований.
Пояснение и пример: Например: можно ли отменять заказ и требуется ли история изменения цены?

Что должно получиться

Документ требований: роли, пять полей словаря или более, правила, основные сценарии и открытые вопросы.

Самопроверка и контрольные вопросы

1. Какие действия в работе №10 подтверждают достижение её цели?
2. Какая проверка показала, что получен правильный результат, и что именно она измеряет?
3. Какую типичную ошибку можно допустить в этой работе и как её обнаружить?
4. Какие данные, настройки или права нельзя переносить из учебной среды в рабочую без согласования?

Критерии оценивания

Оценка	Требования к выполнению
5 «отлично»	Выполнены все этапы; результат воспроизводим, проверки (включая негативные, если предусмотрены) подтверждены; примеры адаптированы к заданию; отчёт содержит команды/артефакты и самостоятельные объяснения.
4 «хорошо»	Основные этапы и контрольные результаты выполнены; допущены единичные неточности в объяснениях или оформлении, не влияющие на корректность и безопасность результата.
3 «удовлетворительно»	Получен минимальный рабочий результат, но часть проверок или пояснений отсутствует; есть ошибки, устранённые с подсказкой; отчёт неполон.
2 «неудовлетворительно»	Основной результат отсутствует либо ошибки не устранены; ключевые этапы или доказательства проверки не представлены; допущены критические нарушения безопасности учебного стенда.

Оценивание проводится по фактически продемонстрированному результату, качеству объяснения шагов и полноте отчёта.

Практическое занятие №11. Построение ER-модели базы данных

Продолжительность: 6 академических часа(ов).

Проверяемые компетенции:

ПК 2.1 — Выявлять проблемы, возникающие в процессе эксплуатации баз данных.
ПК 2.2 — Осуществлять процедуры администрирования баз данных.
ПК 2.3 — Проводить аудит систем безопасности баз данных с использованием регламентов по защите информации.
ПК 2.4 — Формировать требования хранилищ банка данных для обучения.
ПК 2.5 — Подготавливать данные для базы знаний.
ПК 2.8 — Разрабатывать объекты баз данных NoSQL.

Цель занятия: Построение ER-диаграммы.

Теоретические сведения и объяснение: ER-модель показывает сущности и связи, не подменяя их перечнем SQL-таблиц. Для магазина подходят Customer, Product, Order и OrderItem.

Оборудование и исходные материалы: учебный ПК, терминал, PostgreSQL/pgAdmin, тренировочная БД и текстовый редактор.

Порядок выполнения и примеры

Этап 1. Выделение сущностей.

На основе требований занятия №10 определите 4 сущности: Покупатель, Товар, Заказ, Позиция заказа. Для каждой задайте идентификатор и 3–5 атрибутов.

Пояснение и пример: Позиция заказа нужна, чтобы корректно представить связь «заказ содержит много товаров».

Этап 2. Определение отношений.

Нарисуйте связи: Покупатель 1—N Заказ; Заказ 1—N Позиция; Товар 1—N Позиция. Возле концов связей укажите обязательность участия.

Пояснение и пример: Для каждой позиции обязателен ровно один заказ и один товар.

Этап 3. Разбор кардинальности.
Объясните, почему прямая связь Заказ—Товар относится к M—N и как сущность Позиция заказа превращает её в две связи 1—N.
Пояснение и пример: Пример: один заказ содержит два товара, один товар может входить в разные заказы.
Этап 4. Создание диаграммы.
С помощью diagrams.net, pgAdmin ERD или другого редактора нарисуйте сущности прямоугольниками, ключевые поля отметьте PK, связи подпишите. Экспортируйте диаграмму в PNG.
Пояснение и пример: Имена сущностей и атрибутов должны совпадать со словарём требований.
Этап 5. Проверка модели.
Проиграйте два кейса: один покупатель оформил два заказа; один товар продан в трёх заказах. Убедитесь, что модель не требует дублировать покупателя или товар.
Пояснение и пример: Опишите выявленные неоднозначности и решение.

Что должно получиться

ER-диаграмма, список сущностей и объяснение кардинальностей и двух проверочных ситуаций.

Самопроверка и контрольные вопросы

1. Какие действия в работе №11 подтверждают достижение её цели?
2. Какая проверка показала, что получен правильный результат, и что именно она измеряет?
3. Какую типичную ошибку можно допустить в этой работе и как её обнаружить?
4. Какие данные, настройки или права нельзя переносить из учебной среды в рабочую без согласования?

Критерии оценивания

Оценка	**Требования к выполнению**
5 «отлично»	Выполнены все этапы; результат воспроизводим, проверки (включая негативные, если предусмотрены) подтверждены; примеры адаптированы к заданию; отчёт содержит команды/артефакты и самостоятельные объяснения.
4 «хорошо»	Основные этапы и контрольные результаты выполнены; допущены единичные неточности в объяснениях или оформлении, не влияющие на корректность и безопасность результата.
3 «удовлетворительно»	Получен минимальный рабочий результат, но часть проверок или пояснений отсутствует; есть ошибки, устранённые с подсказкой; отчёт неполон.
2 «неудовлетворительно»	Основной результат отсутствует либо ошибки не устранены; ключевые этапы или доказательства проверки не представлены; допущены критические нарушения безопасности учебного стенда.

Оценивание проводится по фактически продемонстрированному результату, качеству объяснения шагов и полноте отчёта.
Практическое занятие №12. Разработка логической модели базы данных
Продолжительность: 6 академических часа(ов).
Проверяемые компетенции:

ПК 2.1 — Выявлять проблемы, возникающие в процессе эксплуатации баз данных.
ПК 2.2 — Осуществлять процедуры администрирования баз данных.
ПК 2.3 — Проводить аудит систем безопасности баз данных с использованием регламентов по защите информации.
ПК 2.4 — Формировать требования хранилищ банка данных для обучения.
ПК 2.5 — Подготавливать данные для базы знаний.
ПК 2.8 — Разрабатывать объекты баз данных NoSQL.

Цель занятия: Перевод ER-модели в логическую структуру.
Теоретические сведения и объяснение: Логическая модель переводит сущности в отношения и ключи; нормализация устраняет повторяющиеся группы и зависимости от неключевых атрибутов.
Оборудование и исходные материалы: учебный ПК, терминал, PostgreSQL/pgAdmin, тренировочная БД и текстовый редактор.

Порядок выполнения и примеры

Этап 1. Определение отношений.
Запишите схему: customers(customer_id PK, name, email); products(product_id PK, title, current_price); orders(order_id PK, customer_id FK, created_at); order_items(order_id FK, product_id FK, quantity, price_at_sale).
Пояснение и пример: Для order_items составной ключ (order_id, product_id) подходит, если один товар появляется в заказе только одной строкой.
Этап 2. Проверка первой нормальной формы.
Уберите поля вида product1, product2 и списки товаров через запятую. Объясните, почему каждое поле должно хранить одно значение заданного домена.
Пояснение и пример: Вместо products_list = «1,2,3» используются отдельные строки order_items.
Этап 3. Проверка второй/третьей НФ.
Объясните, почему имя покупателя не должно храниться в orders, а название товара — в order_items. Перечислите функциональные зависимости.
Пояснение и пример: customer_id → name,email; product_id → title,current_price.
Этап 4. Разработка ограничений.
Назначьте NOT NULL, UNIQUE(email), CHECK(quantity > 0), CHECK(price_at_sale >= 0), FK-связи и правила удаления. Обоснуйте поведение ON DELETE RESTRICT/CASCADE в разных таблицах.
Пояснение и пример: Удаление товара с историей заказов обычно следует ограничивать либо заменять архивированием.
Этап 5. Составление словаря модели.
Оформите таблицу полей с типами PostgreSQL, PK/FK, допустимостью NULL и примерами значений.
Пояснение и пример: Итог должен быть пригоден для последующего создания DDL.

Что должно получиться

Схема четырёх отношений, зависимости, объяснение нормализации и подробный словарь полей.

Самопроверка и контрольные вопросы

1. Какие действия в работе №12 подтверждают достижение её цели?
2. Какая проверка показала, что получен правильный результат, и что именно она измеряет?
3. Какую типичную ошибку можно допустить в этой работе и как её обнаружить?
4. Какие данные, настройки или права нельзя переносить из учебной среды в рабочую без согласования?

Критерии оценивания

Оценка	**Требования к выполнению**
5 «отлично»	Выполнены все этапы; результат воспроизводим, проверки (включая негативные, если предусмотрены) подтверждены; примеры адаптированы к

	заданию; отчёт содержит команды/артефакты и самостоятельные объяснения.
4 «хорошо»	Основные этапы и контрольные результаты выполнены; допущены единичные неточности в объяснениях или оформлении, не влияющие на корректность и безопасность результата.
3 «удовлетворительно»	Получен минимальный рабочий результат, но часть проверок или пояснений отсутствует; есть ошибки, устранённые с подсказкой; отчёт неполон.
2 «неудовлетворительно»	Основной результат отсутствует либо ошибки не устранены; ключевые этапы или доказательства проверки не представлены; допущены критические нарушения безопасности учебного стенда.

Оценивание проводится по фактически продемонстрированному результату, качеству объяснения шагов и полноте отчёта.

Тема 2.2. Реализация и оптимизация реляционной базы данных

Практическое занятие №13. Создание таблиц и ограничений SQL
Продолжительность: 4 академических часа(ов).

Проверяемые компетенции:

ПК 2.1 — Выявлять проблемы, возникающие в процессе эксплуатации баз данных.
ПК 2.2 — Осуществлять процедуры администрирования баз данных.
ПК 2.3 — Проводить аудит систем безопасности баз данных с использованием регламентов по защите информации.
ПК 2.4 — Формировать требования хранилищ банка данных для обучения.
ПК 2.5 — Подготавливать данные для базы знаний.
ПК 2.8 — Разрабатывать объекты баз данных NoSQL.

Цель занятия: Создание реляционных таблиц и тестирование ограничений.
Теоретические сведения и объяснение: SQL DDL описывает структуру и ограничения: PRIMARY KEY, FOREIGN KEY, UNIQUE, NOT NULL и CHECK. Ошибки ограничений — ожидаемый результат негативных тестов.
Оборудование и исходные материалы: учебный ПК, терминал, PostgreSQL/pgAdmin, тренировочная БД и текстовый редактор.

Порядок выполнения и примеры

Этап 1. Создание схемы.
В training_db создайте схему shop. Выполните CREATE TABLE shop.customers(customer_id integer GENERATED ALWAYS AS IDENTITY PRIMARY KEY, name text NOT NULL, email text UNIQUE);.
Пояснение и пример: Использование GENERATED ALWAYS AS IDENTITY позволяет не задавать ключ вручную.

Этап 2. Создание товаров.
Создайте shop.products(product_id integer GENERATED ALWAYS AS IDENTITY PRIMARY KEY, title text NOT NULL, current_price numeric(10,2) NOT NULL CHECK(current_price >= 0));.
Пояснение и пример: NUMERIC применён для цены, чтобы избегать ошибок округления бинарных float.

Этап 3. Создание заказов.
Создайте shop.orders(order_id integer GENERATED ALWAYS AS IDENTITY PRIMARY KEY, customer_id integer NOT NULL REFERENCES shop.customers(customer_id), created_at timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP);.

Пояснение и пример: Внешний ключ гарантирует, что заказ не ссылается на отсутствующего покупателя.

Этап 4. Создание позиций.

Создайте shop.order_items(order_id integer REFERENCES shop.orders(order_id), product_id integer REFERENCES shop.products(product_id), quantity integer NOT NULL CHECK(quantity > 0), price_at_sale numeric(10,2) NOT NULL CHECK(price_at_sale >= 0), PRIMARY KEY(order_id,product_id));.

Пояснение и пример: Составной PK исключает дублирование одного товара в том же заказе.

Этап 5. Негативные проверки.

Попробуйте вставить товар с отрицательной ценой, заказ с несуществующим customer_id, позицию с quantity=0. Для каждой ошибки укажите нарушенное ограничение.

Пояснение и пример: После ошибочной команды в явной транзакции выполните ROLLBACK перед дальнейшей работой.

Что должно получиться

DDL всех четырёх таблиц, описание ограничений и журнал трёх отрицательных тестов.

Самопроверка и контрольные вопросы

1. Какие действия в работе №13 подтверждают достижение её цели?
2. Какая проверка показала, что получен правильный результат, и что именно она измеряет?
3. Какую типичную ошибку можно допустить в этой работе и как её обнаружить?
4. Какие данные, настройки или права нельзя переносить из учебной среды в рабочую без согласования?

Критерии оценивания

Оценка	**Требования к выполнению**
5 «отлично»	Выполнены все этапы; результат воспроизводим, проверки (включая негативные, если предусмотрены) подтверждены; примеры адаптированы к заданию; отчёт содержит команды/артефакты и самостоятельные объяснения.
4 «хорошо»	Основные этапы и контрольные результаты выполнены; допущены единичные неточности в объяснениях или оформлении, не влияющие на корректность и безопасность результата.
3 «удовлетворительно»	Получен минимальный рабочий результат, но часть проверок или пояснений отсутствует; есть ошибки, устранённые с подсказкой; отчёт неполон.
2 «неудовлетворительно»	Основной результат отсутствует либо ошибки не устранены; ключевые этапы или доказательства проверки не представлены; допущены критические нарушения безопасности учебного стенда.

Оценивание проводится по фактически продемонстрированному результату, качеству объяснения шагов и полноте отчёта.

Практическое занятие №14. Импорт и заполнение данных

Продолжительность: 6 академических часа(ов).

Проверяемые компетенции:

ПК 2.1 — Выявлять проблемы, возникающие в процессе эксплуатации баз данных.
ПК 2.2 — Осуществлять процедуры администрирования баз данных.
ПК 2.3 — Проводить аудит систем безопасности баз данных с использованием регламентов по защите информации.
ПК 2.4 — Формировать требования хранилищ банка данных для обучения.
ПК 2.5 — Подготавливать данные для базы знаний.
ПК 2.8 — Разрабатывать объекты баз данных NoSQL.

Цель занятия: Подготовка CSV и импорт данных.
Теоретические сведения и объяснение: Импорт требует согласования разделителя, кодировки, заголовков и типов. Ошибочные строки лучше выявлять до записи в целевые таблицы.
Оборудование и исходные материалы: учебный ПК, терминал, PostgreSQL/pgAdmin, тренировочная БД и текстовый редактор.

Порядок выполнения и примеры

Этап 1. Создание учебного набора.
Подготовьте customers.csv с заголовками name,email и 5 строками, products.csv с title,current_price и 8 строками. Включите в отдельную испорченную копию товара цену -50 или текст вместо числа.
Пояснение и пример: Учебный CSV в UTF-8, разделитель — запятая; если значение содержит запятую, заключайте поле в кавычки.
Этап 2. Разведка структуры.
Откройте файлы в текстовом редакторе, проверьте число строк, кодировку и порядок столбцов. Составьте соответствие «поле CSV → столбец таблицы».
Пояснение и пример: Не импортируйте поля identity с заранее выдуманными номерами без необходимости.
Этап 3. Загрузка данных.
В psql выполните \copy shop.customers(name,email) FROM 'customers.csv' WITH (FORMAT csv, HEADER true, ENCODING 'UTF8');; аналогично загрузите products.
Пояснение и пример: Команда \copy читает локальный файл клиента; SQL COPY на сервере имеет другой контекст доступа.
Этап 4. Проверка и очистка.
Выполните SELECT count(*) для каждой таблицы, проверьте NULL и дубликаты email. Для импортируемой ошибочной строки зафиксируйте сообщение и исправьте файл.
Пояснение и пример: Если импорт прервался, проверьте, были ли изменения применены; используйте транзакцию там, где она подходит.
Этап 5. Создание заказов.
Создайте минимум 3 заказа и 6 позиций через INSERT...SELECT или явные ключи, полученные SELECT. Рассчитайте итог каждого заказа запросом SUM(quantity * price_at_sale).
Пояснение и пример: Не подставляйте текущую цену товара вместо сохранённой цены продажи в историческом отчёте.

Что должно получиться

Два корректных CSV, журнал загрузки, запросы проверки, 3 заказа и таблица расчёта итогов.

Самопроверка и контрольные вопросы

1. Какие действия в работе №14 подтверждают достижение её цели?
2. Какая проверка показала, что получен правильный результат, и что именно она измеряет?
3. Какую типичную ошибку можно допустить в этой работе и как её обнаружить?
4. Какие данные, настройки или права нельзя переносить из учебной среды в рабочую без согласования?

Критерии оценивания

Оценка	**Требования к выполнению**
5 «отлично»	Выполнены все этапы; результат воспроизводим, проверки (включая

	негативные, если предусмотрены) подтверждены; примеры адаптированы к заданию; отчёт содержит команды/артефакты и самостоятельные объяснения.
4 «хорошо»	Основные этапы и контрольные результаты выполнены; допущены единичные неточности в объяснениях или оформлении, не влияющие на корректность и безопасность результата.
3 «удовлетворительно»	Получен минимальный рабочий результат, но часть проверок или пояснений отсутствует; есть ошибки, устранённые с подсказкой; отчёт неполон.
2 «неудовлетворительно»	Основной результат отсутствует либо ошибки не устранены; ключевые этапы или доказательства проверки не представлены; допущены критические нарушения безопасности учебного стенда.

Оценивание проводится по фактически продемонстрированному результату, качеству объяснения шагов и полноте отчёта.

Практическое занятие №15. Разработка запросов и настройка индексов

Продолжительность: 6 академических часа(ов).

Проверяемые компетенции:

ПК 2.1 — Выявлять проблемы, возникающие в процессе эксплуатации баз данных.
ПК 2.2 — Осуществлять процедуры администрирования баз данных.
ПК 2.3 — Проводить аудит систем безопасности баз данных с использованием регламентов по защите информации.
ПК 2.4 — Формировать требования хранилищ банка данных для обучения.
ПК 2.5 — Подготавливать данные для базы знаний.
ПК 2.8 — Разрабатывать объекты баз данных NoSQL.

Цель занятия: Запросы JOIN, агрегаты и проверка индексов.

Теоретические сведения и объяснение: Индексы ускоряют отдельные виды поиска ценой хранения и стоимости обновления. План EXPLAIN позволяет проверить, используется ли выбранный индекс.

Оборудование и исходные материалы: учебный ПК, терминал, PostgreSQL/pgAdmin, тренировочная БД и текстовый редактор.

Порядок выполнения и примеры

Этап 1. Базовые выборки.

Составьте SELECT с фильтром по цене и сортировкой, затем JOIN заказов и покупателей. Добавьте запрос на общую стоимость каждого заказа с GROUP BY.

Пояснение и пример: Пример суммы: SELECT order_id, SUM(quantity*price_at_sale) AS total FROM shop.order_items GROUP BY order_id;.

Этап 2. Усложнение отчёта.

Найдите три самых продаваемых товара по суммарному количеству и покупателей с более чем одним заказом. Опишите, какие соединения и агрегаты применены.

Пояснение и пример: Проверьте, не пропадают ли покупатели без заказов из INNER JOIN; при необходимости используйте LEFT JOIN.

Этап 3. Исходный план.

Выполните EXPLAIN (ANALYZE, BUFFERS) SELECT * FROM shop.orders WHERE customer_id = 1;. Сохраните план и оценку фактического времени.

Пояснение и пример: Для маленькой таблицы план может предпочесть Seq Scan; это ожидаемое поведение.

Этап 4. Создание индекса.
Выполните CREATE INDEX idx_orders_customer ON shop.orders(customer_id); затем ANALYZE shop.orders; и повторите план. Сравните cost, фактическое время и число прочитанных строк.
Пояснение и пример: Индекс не гарантирует ускорения на малом объёме; не делайте ложный вывод только по одному запуску.
Этап 5. Оценка целесообразности.
Опишите, почему индекс на внешнем ключе orders.customer_id может быть полезен, а индексирование всех столбцов обычно вредно. Предложите тест на большем наборе.
Пояснение и пример: Вывод должен отражать фактическое наблюдение, а не обещать ускорение при любых условиях.

Что должно получиться

Тексты минимум четырёх запросов, их результаты, планы до/после и вывод об индексации.

Самопроверка и контрольные вопросы

1. Какие действия в работе №15 подтверждают достижение её цели?
2. Какая проверка показала, что получен правильный результат, и что именно она измеряет?
3. Какую типичную ошибку можно допустить в этой работе и как её обнаружить?
4. Какие данные, настройки или права нельзя переносить из учебной среды в рабочую без согласования?

Критерии оценивания

Оценка	**Требования к выполнению**
5 «отлично»	Выполнены все этапы; результат воспроизводим, проверки (включая негативные, если предусмотрены) подтверждены; примеры адаптированы к заданию; отчёт содержит команды/артефакты и самостоятельные объяснения.
4 «хорошо»	Основные этапы и контрольные результаты выполнены; допущены единичные неточности в объяснениях или оформлении, не влияющие на корректность и безопасность результата.
3 «удовлетворительно»	Получен минимальный рабочий результат, но часть проверок или пояснений отсутствует; есть ошибки, устранённые с подсказкой; отчёт неполон.
2 «неудовлетворительно»	Основной результат отсутствует либо ошибки не устранены; ключевые этапы или доказательства проверки не представлены; допущены критические нарушения безопасности учебного стенда.

Оценивание проводится по фактически продемонстрированному результату, качеству объяснения шагов и полноте отчёта.

Тема 2.3. Защита и развитие баз данных

Практическое занятие №16. Настройка ролевой модели безопасности
Продолжительность: 4 академических часа(ов).
Проверяемые компетенции:
ПК 2.1 — Выявлять проблемы, возникающие в процессе эксплуатации баз данных.

ПК 2.2 — Осуществлять процедуры администрирования баз данных.
ПК 2.3 — Проводить аудит систем безопасности баз данных с использованием регламентов по защите информации.
ПК 2.4 — Формировать требования хранилищ банка данных для обучения.
ПК 2.5 — Подготавливать данные для базы знаний.
ПК 2.8 — Разрабатывать объекты баз данных NoSQL.
Цель занятия: Ролевая модель магазина.
Теоретические сведения и объяснение: Ролевой доступ строится по должностным функциям, а не путём выдачи каждому пользователю всех разрешений. Права на объекты лучше выдавать групповым ролям.
Оборудование и исходные материалы: учебный ПК, терминал, PostgreSQL/pgAdmin, тренировочная БД и текстовый редактор.

Порядок выполнения и примеры

Этап 1. Матрица полномочий.
Задайте роли shop_viewer (чтение), shop_sales (чтение и добавление заказов), shop_admin (изменение структуры по согласованию). Для таблиц customers/products/orders/order_items составьте таблицу SELECT/INSERT/UPDATE/DELETE.
Пояснение и пример: Менеджеру продаж обычно не требуется DROP TABLE и создание ролей.
Этап 2. Создание групп.
Выполните CREATE ROLE shop_viewer NOLOGIN; CREATE ROLE shop_sales NOLOGIN;. Выдайте USAGE ON SCHEMA shop группам; SELECT ON ALL TABLES IN SCHEMA shop роли shop_viewer.
Пояснение и пример: Избегайте выдачи ALL PRIVILEGES без конкретного обоснования.
Этап 3. Назначение рабочих прав.
Выдайте shop_sales SELECT на нужные справочники и INSERT на orders и order_items. Для identity/sequence прав проверьте корректность вставки под выбранной ролью.
Пояснение и пример: Если вставка не проходит из-за sequence, определите нужные права отдельно, вместо подключения как superuser.
Этап 4. Проверка матрицы.
Создайте тестовые логины, подключитесь каждым и выполните разрешённый SELECT, разрешённый INSERT (при необходимости) и запрещённый DROP TABLE.
Пояснение и пример: Для каждого действия фиксируйте учётную запись, SQL и фактический результат.
Этап 5. Исправление избыточных прав.
Искусственно предоставьте лишнее DELETE на одну таблицу, обнаружьте его через тест и затем отзовите REVOKE.
Пояснение и пример: После REVOKE повторно проверьте отрицательный сценарий.

Что должно получиться

Матрица прав, SQL ролей, протокол проверок и объяснение минимально необходимых полномочий.

Самопроверка и контрольные вопросы

1. Какие действия в работе №16 подтверждают достижение её цели?
2. Какая проверка показала, что получен правильный результат, и что именно она измеряет?
3. Какую типичную ошибку можно допустить в этой работе и как её обнаружить?
4. Какие данные, настройки или права нельзя переносить из учебной среды в рабочую без согласования?

Критерии оценивания

Оценка	**Требования к выполнению**
5 «отлично»	Выполнены все этапы; результат воспроизводим, проверки (включая негативные, если предусмотрены) подтверждены; примеры адаптированы к

	заданию; отчёт содержит команды/артефакты и самостоятельные объяснения.
4 «хорошо»	Основные этапы и контрольные результаты выполнены; допущены единичные неточности в объяснениях или оформлении, не влияющие на корректность и безопасность результата.
3 «удовлетворительно»	Получен минимальный рабочий результат, но часть проверок или пояснений отсутствует; есть ошибки, устранённые с подсказкой; отчёт неполон.
2 «неудовлетворительно»	Основной результат отсутствует либо ошибки не устранены; ключевые этапы или доказательства проверки не представлены; допущены критические нарушения безопасности учебного стенда.

Оценивание проводится по фактически продемонстрированному результату, качеству объяснения шагов и полноте отчёта.

Практическое занятие №17. Настройка аудита и защиты данных

Продолжительность: 6 академических часа(ов).

Проверяемые компетенции:

ПК 2.1 — Выявлять проблемы, возникающие в процессе эксплуатации баз данных.
ПК 2.2 — Осуществлять процедуры администрирования баз данных.
ПК 2.3 — Проводить аудит систем безопасности баз данных с использованием регламентов по защите информации.
ПК 2.4 — Формировать требования хранилищ банка данных для обучения.
ПК 2.5 — Подготавливать данные для базы знаний.
ПК 2.8 — Разрабатывать объекты баз данных NoSQL.

Цель занятия: Контроль доступа к чувствительным полям и журналирование.

Теоретические сведения и объяснение: Защита данных объединяет выборочный доступ, безопасные представления, аудит и процедуры проверки. Учебная работа не использует реальные персональные данные.

Оборудование и исходные материалы: учебный ПК, терминал, PostgreSQL/pgAdmin, тренировочная БД и текстовый редактор.

Порядок выполнения и примеры

Этап 1. Разделение публичных и закрытых полей.

В shop.customers выделите поле email как чувствительное для учебного кейса. Создайте VIEW shop.customer_public AS SELECT customer_id, name FROM shop.customers;.

Пояснение и пример: Представление исключает email из ответа, но не заменяет запрет прямого чтения исходной таблицы.

Этап 2. Проверка прав.

Запретите вспомогательной роли viewer_email прямой SELECT на shop.customers, разрешите SELECT на shop.customer_public и USAGE на shop. Проверьте обе операции под ролью.

Пояснение и пример: Если другие роли или PUBLIC предоставляют право на таблицу, диагностируйте наследование.

Этап 3. Логирование изменений.

Сконфигурируйте на учебном сервере журналирование команд изменения данных по аналогии с занятием №6; выполните тестовый UPDATE имени покупателя. Найдите событие по времени и роли.

Пояснение и пример: Не включайте в отчёт реальные email и пароли.

Этап 4. Оценка угроз.

Составьте таблицу «Угроза / Возможное последствие / Мера защиты / Способ проверки»:

избыточные права, утечка резервной копии, несанкционированное изменение.
Пояснение и пример: Для копий упомяните контроль доступа и шифрование в утверждённом хранилище.
Этап 5. Составление акта проверки.
Сформируйте акт: дата, используемые роли, результат проверки доступа, найденные события, остаточные риски и необходимые меры.
Пояснение и пример: Различайте подтверждённые факты лаборатории и рекомендуемые меры на практике.

Что должно получиться

Представление без чувствительного поля, тест разрешённого/запрещённого чтения, журнал изменения и акт аудита.

Самопроверка и контрольные вопросы

1. Какие действия в работе №17 подтверждают достижение её цели?
2. Какая проверка показала, что получен правильный результат, и что именно она измеряет?
3. Какую типичную ошибку можно допустить в этой работе и как её обнаружить?
4. Какие данные, настройки или права нельзя переносить из учебной среды в рабочую без согласования?

Критерии оценивания

Оценка	**Требования к выполнению**
5 «отлично»	Выполнены все этапы; результат воспроизводим, проверки (включая негативные, если предусмотрены) подтверждены; примеры адаптированы к заданию; отчёт содержит команды/артефакты и самостоятельные объяснения.
4 «хорошо»	Основные этапы и контрольные результаты выполнены; допущены единичные неточности в объяснениях или оформлении, не влияющие на корректность и безопасность результата.
3 «удовлетворительно»	Получен минимальный рабочий результат, но часть проверок или пояснений отсутствует; есть ошибки, устранённые с подсказкой; отчёт неполон.
2 «неудовлетворительно»	Основной результат отсутствует либо ошибки не устранены; ключевые этапы или доказательства проверки не представлены; допущены критические нарушения безопасности учебного стенда.

Оценивание проводится по фактически продемонстрированному результату, качеству объяснения шагов и полноте отчёта.
Практическое занятие №18. Разработка кода доступа к NoSQL-базе
Продолжительность: 6 академических часа(ов).
Проверяемые компетенции:
ПК 2.1 — Выявлять проблемы, возникающие в процессе эксплуатации баз данных.
ПК 2.2 — Осуществлять процедуры администрирования баз данных.
ПК 2.3 — Проводить аудит систем безопасности баз данных с использованием регламентов по защите информации.

ПК 2.4 — Формировать требования хранилищ банка данных для обучения.
ПК 2.5 — Подготавливать данные для базы знаний.
ПК 2.8 — Разрабатывать объекты баз данных NoSQL.
Цель занятия: CRUD-операции MongoDB из Python.
Теоретические сведения и объяснение: Документная модель NoSQL хранит JSON-подобные документы. Для учебной базы каталога удобна коллекция products с вложенными tags и характеристиками.
Оборудование и исходные материалы: учебный ПК, Python и инструменты соответствующего занятия (MongoDB/PySpark/Airflow/Docker/средство визуализации).

Порядок выполнения и примеры

Этап 1. Подготовка среды.
Разверните учебный MongoDB и установите pymongo в отдельном Python-окружении: python -m pip install pymongo. Настройте строку подключения через переменную окружения MONGO_URI.
Пояснение и пример: Не записывайте учётные данные MongoDB в код и не открывайте сетевой порт для публичного доступа.
Этап 2. Создание документов.
В Python создайте MongoClient(os.environ['MONGO_URI']), db=client['lab_catalog'], col=db['products']; выполните insert_many для трёх документов с полями sku, title, price, tags.
Пояснение и пример: Пример: {"sku":"P100","title":"Тетрадь","price":45,"tags":["канцтовары"]}.
Этап 3. Чтение и обновление.
Найдите позиции дороже 50 через find({"price":{"$gt":50}}), затем измените цену одного SKU через update_one({"sku":"P100"},{"$set":{"price":49}}).
Пояснение и пример: Проверяйте matched_count и modified_count, а не только отсутствие исключений.
Этап 4. Индекс и удаление.
Создайте уникальный индекс col.create_index("sku", unique=True). Попробуйте вставить документ с повторяющимся sku, поймайте ошибку DuplicateKeyError. Удалите только тестовый документ через delete_one по sku.
Пояснение и пример: Не используйте delete_many({}) без ограничений; это удалит всю коллекцию.
Этап 5. Оформление сценария.
Напишите функцию get_product_by_sku(sku), которая возвращает документ без служебного _id или None. Продемонстрируйте 2 теста.
Пояснение и пример: Код должен закрывать соединение и сообщать об ошибках подключения понятным образом.

Что должно получиться

Python-скрипт с CRUD, созданный уникальный индекс, два теста функции и пояснение структуры документа.

Самопроверка и контрольные вопросы

1. Какие действия в работе №18 подтверждают достижение её цели?
2. Какая проверка показала, что получен правильный результат, и что именно она измеряет?
3. Какую типичную ошибку можно допустить в этой работе и как её обнаружить?
4. Какие данные, настройки или права нельзя переносить из учебной среды в рабочую без согласования?

Критерии оценивания

Оценка	**Требования к выполнению**
5 «отлично»	Выполнены все этапы; результат воспроизводим, проверки (включая негативные, если предусмотрены) подтверждены; примеры адаптированы к заданию; отчёт содержит

	команды/артефакты и самостоятельные объяснения.
4 «хорошо»	Основные этапы и контрольные результаты выполнены; допущены единичные неточности в объяснениях или оформлении, не влияющие на корректность и безопасность результата.
3 «удовлетворительно»	Получен минимальный рабочий результат, но часть проверок или пояснений отсутствует; есть ошибки, устранённые с подсказкой; отчёт неполон.
2 «неудовлетворительно»	Основной результат отсутствует либо ошибки не устранены; ключевые этапы или доказательства проверки не представлены; допущены критические нарушения безопасности учебного стенда.

Оценивание проводится по фактически продемонстрированному результату, качеству объяснения шагов и полноте отчёта.

Раздел 3. Конвейеры обработки и контроль качества данных для систем ИИ

Тема 3.1. Подготовка и контроль качества данных для систем ИИ

Практическое занятие №19. Профилирование и классификация источников данных
Продолжительность: 4 академических часа(ов).
Проверяемые компетенции:
ПК 2.9 — Осуществлять конвейерную обработку данных для систем ИИ.
ПК 2.10 — Осуществлять мониторинг и оркестрацию конвейеров обработки данных.
ПК 2.11 — Осуществлять контроль качества данных в конвейерах обработки.
Цель занятия: Исследование входных источников данных для ИИ.
Теоретические сведения и объяснение: Качество модели ИИ зависит от происхождения данных и семантики признаков. Профилирование фиксирует типы, пропуски, повторения и диапазоны ещё до очистки.
Оборудование и исходные материалы: учебный ПК, Python, pandas, синтетические CSV/JSON, среда Jupyter или редактор кода.

Порядок выполнения и примеры

Этап 1. Определение источников.
Рассмотрите три учебных источника: CSV с заказами, таблицу PostgreSQL с товарами и JSON с категориями. Для каждого опишите владельца, схему, период обновления, формат и возможные ограничения доступа.
Пояснение и пример: Используйте синтетический набор заказов, а не реальные сведения покупателей.
Этап 2. Создание профиля CSV.
С помощью pandas прочитайте orders_raw.csv, выведите df.shape, df.dtypes, df.isna().sum(), df.duplicated().sum(). Запишите число строк, столбцов, пропусков и дубликатов.
Пояснение и пример: Пример: df = pd.read_csv("orders_raw.csv"); print(df.isna().sum()).
Этап 3. Классификация полей.
Для каждого поля задайте класс: идентификатор, числовой признак, категория, дата/время, целевая переменная. Отдельно отметьте возможные идентифицирующие сведения.
Пояснение и пример: Идентификатор заказа обычно не следует использовать как осмысленный числовой признак.
Этап 4. Исследование выбросов.
Посчитайте describe() для price и quantity; проверьте отрицательные значения и неправдоподобно большие количества. Постройте небольшую таблицу аномалий.
Пояснение и пример: Не удаляйте аномалию автоматически: сначала установите бизнес-правило и возможную причину.
Этап 5. Формирование паспорта.
Подготовьте паспорт датасета: источники, поля, ограничения использования, проблемы качества, ответственные и план исправления.
Пояснение и пример: Разделите подтверждённые факты (набор содержит 4 пропуска) и гипотезы (ошибка выгрузки).

Что должно получиться

Паспорт источников, сводка pandas, классификация полей и перечень аномалий.

Самопроверка и контрольные вопросы

1. Какие действия в работе №19 подтверждают достижение её цели?
2. Какая проверка показала, что получен правильный результат, и что именно она измеряет?
3. Какую типичную ошибку можно допустить в этой работе и как её обнаружить?
4. Какие данные, настройки или права нельзя переносить из учебной среды в рабочую без согласования?

Критерии оценивания

Оценка	Требования к выполнению
5 «отлично»	Выполнены все этапы; результат воспроизводим, проверки (включая негативные, если предусмотрены) подтверждены; примеры адаптированы к заданию; отчёт содержит команды/артефакты и самостоятельные объяснения.
4 «хорошо»	Основные этапы и контрольные результаты выполнены; допущены единичные неточности в объяснениях или оформлении, не влияющие на корректность и безопасность результата.
3 «удовлетворительно»	Получен минимальный рабочий результат, но часть проверок или пояснений отсутствует; есть ошибки, устранённые с подсказкой; отчёт неполон.
2 «неудовлетворительно»	Основной результат отсутствует либо ошибки не устранены; ключевые этапы или доказательства проверки не представлены; допущены критические нарушения безопасности учебного стенда.

Оценивание проводится по фактически продемонстрированному результату, качеству объяснения шагов и полноте отчёта.

Практическое занятие №20. Очистка и преобразование данных

Продолжительность: 6 академических часа(ов).

Проверяемые компетенции:

ПК 2.9 — Осуществлять конвейерную обработку данных для систем ИИ.

ПК 2.10 — Осуществлять мониторинг и оркестрацию конвейеров обработки данных.

ПК 2.11 — Осуществлять контроль качества данных в конвейерах обработки.

Цель занятия: Очистка и стандартизация табличных данных.

Теоретические сведения и объяснение: Очистка должна быть воспроизводима: одинаковый ввод при одинаковых правилах даёт одинаковый результат. Любая потеря строк и замена значений объясняется.

Оборудование и исходные материалы: учебный ПК, Python, pandas, синтетические CSV/JSON, среда Jupyter или редактор кода.

Порядок выполнения и примеры

Этап 1. Постановка правил.

Используйте orders_raw.csv со столбцами order_id, order_date, amount, category. Создайте копии строк с лишними пробелами, пустой категорией, невалидной датой и отрицательной суммой.

Пояснение и пример: Сначала сохраните неизменённый оригинал, затем работайте с копией.

Этап 2. Нормализация строк.

В pandas выполните df["category"] = df["category"].astype("string").str.strip().str.lower(). Пустые строки замените на pd.NA.

Пояснение и пример: Преобразование к строке должно учитывать пропуски; не превращайте NaN бездумно в текст "nan".

Этап 3. Обработка дат и чисел.

Приведите дату: pd.to_datetime(df["order_date"], errors="coerce"); сумму: pd.to_numeric(df["amount"], errors="coerce"). Отдельно посчитайте, сколько значений стало NaT/NaN.

Пояснение и пример: errors="coerce" не исправляет исходную ошибку — он помечает неразобранные

значения.
Этап 4. Решение о пропусках.
Удалите строки без order_id; некорректные суммы поместите в набор исключений. Для пустой категории используйте "unknown" только если это согласовано с задачей.
Пояснение и пример: Не заменяйте пропущенные суммы на ноль без обоснования.
Этап 5. Проверка и экспорт.
Удалите полные дубликаты с подсчётом, убедитесь, что amount >= 0 и order_id уникален; сохраните clean_orders.csv и rejected_orders.csv.
Пояснение и пример: В отчёте укажите числа «было / исправлено / отклонено / осталось».

Что должно получиться

Скрипт очистки, файлы чистых и отклонённых данных, таблица изменений и объяснение правил.

Самопроверка и контрольные вопросы

1. Какие действия в работе №20 подтверждают достижение её цели?
2. Какая проверка показала, что получен правильный результат, и что именно она измеряет?
3. Какую типичную ошибку можно допустить в этой работе и как её обнаружить?
4. Какие данные, настройки или права нельзя переносить из учебной среды в рабочую без согласования?

Критерии оценивания

Оценка	**Требования к выполнению**
5 «отлично»	Выполнены все этапы; результат воспроизводим, проверки (включая негативные, если предусмотрены) подтверждены; примеры адаптированы к заданию; отчёт содержит команды/артефакты и самостоятельные объяснения.
4 «хорошо»	Основные этапы и контрольные результаты выполнены; допущены единичные неточности в объяснениях или оформлении, не влияющие на корректность и безопасность результата.
3 «удовлетворительно»	Получен минимальный рабочий результат, но часть проверок или пояснений отсутствует; есть ошибки, устранённые с подсказкой; отчёт неполон.
2 «неудовлетворительно»	Основной результат отсутствует либо ошибки не устранены; ключевые этапы или доказательства проверки не представлены; допущены критические нарушения безопасности учебного стенда.

Оценивание проводится по фактически продемонстрированному результату, качеству объяснения шагов и полноте отчёта.

Практическое занятие №21. Формирование датасета и контроль качества

Продолжительность: 6 академических часа(ов).

Проверяемые компетенции:

ПК 2.9 — Осуществлять конвейерную обработку данных для систем ИИ.
ПК 2.10 — Осуществлять мониторинг и оркестрацию конвейеров обработки данных.
ПК 2.11 — Осуществлять контроль качества данных в конвейерах обработки.

Цель занятия: Подготовка датасета и проверка качества.

Теоретические сведения и объяснение: Готовый датасет должен сопровождаться схемой, правилами качества и воспроизводимым способом формирования. Разделение train/test выполняется без утечки будущей информации.
Оборудование и исходные материалы: учебный ПК, Python, pandas, синтетические CSV/JSON, среда Jupyter или редактор кода.

Порядок выполнения и примеры

Этап 1. Определение задачи ИИ.
Предположите прогноз суммы следующего заказа по агрегированным признакам клиента. Опишите целевую переменную и признаки, доступные на момент прогноза.
Пояснение и пример: Нельзя включать в признаки сведения, которые стали известны только после прогнозируемого заказа.
Этап 2. Формирование признаков.
На основе clean_orders.csv рассчитайте число прошлых заказов и среднюю прошлую сумму для клиента либо на учебном синтетическом наборе. Запишите типы и назначение признаков.
Пояснение и пример: Пример агрегата pandas:
df.groupby("customer_id")["amount"].agg(["count","mean"]).
Этап 3. Контроль качества.
Для итоговой таблицы проверьте: уникальность ключа наблюдения, отсутствие бесконтрольных NULL в целевой переменной, допустимые диапазоны, совпадение схемы с описанием.
Пояснение и пример: Составьте перечень assert-проверок и укажите ожидаемое поведение при ошибке.
Этап 4. Разделение данных.
При временной природе заказов отсортируйте наблюдения по дате и отделите последние 20% как тестовую часть. Объясните, почему случайное перемешивание может исказить оценку временного прогноза.
Пояснение и пример: Порог 20% — учебное допущение; реальная доля зависит от задачи и объёма.
Этап 5. Версионирование.
Сохраните train.csv, test.csv, schema.json и README с датой формирования, правилами и количеством строк.
Пояснение и пример: Проверьте, что один и тот же наблюдаемый заказ не попал в обе выборки.

Что должно получиться

Пары train/test, схема датасета, перечень тестов качества и описание предотвращения утечки.

Самопроверка и контрольные вопросы

1. Какие действия в работе №21 подтверждают достижение её цели?
2. Какая проверка показала, что получен правильный результат, и что именно она измеряет?
3. Какую типичную ошибку можно допустить в этой работе и как её обнаружить?
4. Какие данные, настройки или права нельзя переносить из учебной среды в рабочую без согласования?

Критерии оценивания

Оценка	**Требования к выполнению**
5 «отлично»	Выполнены все этапы; результат воспроизводим, проверки (включая негативные, если предусмотрены) подтверждены; примеры адаптированы к заданию; отчёт содержит команды/артефакты и самостоятельные объяснения.
4 «хорошо»	Основные этапы и контрольные результаты выполнены; допущены единичные неточности в объяснениях или оформлении,

	не влияющие на корректность и безопасность результата.
3 «удовлетворительно»	Получен минимальный рабочий результат, но часть проверок или пояснений отсутствует; есть ошибки, устранённые с подсказкой; отчёт неполон.
2 «неудовлетворительно»	Основной результат отсутствует либо ошибки не устранены; ключевые этапы или доказательства проверки не представлены; допущены критические нарушения безопасности учебного стенда.

Оценивание проводится по фактически продемонстрированному результату, качеству объяснения шагов и полноте отчёта.

Тема 3.2. Разработка и развертывание конвейера обработки данных

Практическое занятие №22. Разработка DAG и зависимостей задач

Продолжительность: 4 академических часа(ов).

Проверяемые компетенции:

ПК 2.9 — Осуществлять конвейерную обработку данных для систем ИИ.

ПК 2.10 — Осуществлять мониторинг и оркестрацию конвейеров обработки данных.

ПК 2.11 — Осуществлять контроль качества данных в конвейерах обработки.

Цель занятия: Создание учебного DAG в Apache Airflow.

Теоретические сведения и объяснение: DAG описывает граф задач и порядок выполнения, а Airflow управляет запуском и отображает состояние. Зависимости не заменяют собственную обработку исключений.

Оборудование и исходные материалы: учебный ПК, Python и инструменты соответствующего занятия (MongoDB/PySpark/Airflow/Docker/средство визуализации).

Порядок выполнения и примеры

Этап 1. Проектирование графа.

Нарисуйте цепочку extract_orders → validate_orders → transform_orders → load_orders. Укажите вход, выход и условие успешного завершения каждой задачи.

Пояснение и пример: Стрелка означает зависимость: load_orders нельзя запускать до успешной transform_orders.

Этап 2. Создание файла DAG.

В каталоге dags создайте файл orders_pipeline.py, импортируйте DAG, datetime и @task. Объявите with DAG(dag_id="orders_pipeline", schedule="@daily", start_date=datetime(2025,1,1), catchup=False) as dag:.

Пояснение и пример: Дату старта и расписание адаптируйте к локальной учебной среде и версии Airflow.

Этап 3. Определение задач.

Опишите четыре функции через @task; на первом этапе пусть каждая выводит короткое сообщение и возвращает небольшой набор данных/путь к файлу. Свяжите результаты вызовами или используйте оператор >> для задач.

Пояснение и пример: Передавать большие DataFrame через XCom не следует; для них передавайте путь к файлу.

Этап 4. Запуск и наблюдение.

Проверьте, что DAG появился в интерфейсе; выполните ручной запуск, изучите граф и логи. Зафиксируйте состояния success/failed.

Пояснение и пример: Если DAG не обнаружен, сначала проверьте ошибки импорта в его файле.

Этап 5. Негативная проверка.

Добавьте в validate_orders временное исключение ValueError("invalid data") и повторите запуск. Опишите, какая последующая задача не выполнится, затем исправьте причину.

Пояснение и пример: В отчёте покажите обе истории запуска: сбой и успешное исправление.

Что должно получиться

Схема зависимостей, orders_pipeline.py, скриншоты графа и журнал теста ошибки.

Самопроверка и контрольные вопросы

1. Какие действия в работе №22 подтверждают достижение её цели?
2. Какая проверка показала, что получен правильный результат, и что именно она измеряет?
3. Какую типичную ошибку можно допустить в этой работе и как её обнаружить?
4. Какие данные, настройки или права нельзя переносить из учебной среды в рабочую без согласования?

Критерии оценивания

Оценка	**Требования к выполнению**
5 «отлично»	Выполнены все этапы; результат воспроизводим, проверки (включая негативные, если предусмотрены) подтверждены; примеры адаптированы к заданию; отчёт содержит команды/артефакты и самостоятельные объяснения.
4 «хорошо»	Основные этапы и контрольные результаты выполнены; допущены единичные неточности в объяснениях или оформлении, не влияющие на корректность и безопасность результата.
3 «удовлетворительно»	Получен минимальный рабочий результат, но часть проверок или пояснений отсутствует; есть ошибки, устранённые с подсказкой; отчёт неполон.
2 «неудовлетворительно»	Основной результат отсутствует либо ошибки не устранены; ключевые этапы или доказательства проверки не представлены; допущены критические нарушения безопасности учебного стенда.

Оценивание проводится по фактически продемонстрированному результату, качеству объяснения шагов и полноте отчёта.

Практическое занятие №23. Обработка данных в PySpark

Продолжительность: 6 академических часа(ов).

Проверяемые компетенции:

ПК 2.9 — Осуществлять конвейерную обработку данных для систем ИИ.
ПК 2.10 — Осуществлять мониторинг и оркестрацию конвейеров обработки данных.
ПК 2.11 — Осуществлять контроль качества данных в конвейерах обработки.

Цель занятия: Обработка заказов средствами PySpark.

Теоретические сведения и объяснение: PySpark обрабатывает распределённые табличные данные с помощью DataFrame API. Даже локальная лаборатория позволяет изучить схему, фильтрацию и агрегацию.

Оборудование и исходные материалы: учебный ПК, Python и инструменты соответствующего занятия (MongoDB/PySpark/Airflow/Docker/средство визуализации).

Порядок выполнения и примеры

Этап 1. Создание SparkSession.

В Python выполните SparkSession.builder.master("local[*]").appName("orders-lab").getOrCreate(). Убедитесь, что Java и PySpark доступны, а версия совместима со стендом.

Пояснение и пример: Сохраните версию PySpark и режим local[*] в отчёте.
Этап 2. Чтение файла.
Загрузите orders_raw.csv: spark.read.option("header",True).option("inferSchema",True).csv(path).
Вызовите printSchema() и show(5, truncate=False).
Пояснение и пример: Автоопределение типов удобно для лаборатории, но для production лучше явная схема.
Этап 3. Очистка и преобразование.
Импортируйте functions as F; оставьте строки F.col("amount") >= 0, преобразуйте order_date через to_date и очистите category через trim/lower.
Пояснение и пример: Проверьте, какие строки отброшены и что стало с некорректными датами.
Этап 4. Агрегация.
Выполните groupBy("category").agg(F.count("*").alias("orders"), F.sum("amount").alias("revenue")); отсортируйте по revenue по убыванию.
Пояснение и пример: Поясните смысл count, sum и отличие вычисления до action от выполнения show/count.
Этап 5. Запись и проверка.
Сохраните результат в Parquet: result.write.mode("overwrite").parquet("output/orders_summary"). Прочитайте его обратно и сравните количество категорий и суммы.
Пояснение и пример: Осторожно с overwrite: используйте только выделенную учебную папку.

Что должно получиться

Исполняемый PySpark-скрипт, схема DataFrame, агрегация, каталог Parquet и контроль обратного чтения.

Самопроверка и контрольные вопросы

1. Какие действия в работе №23 подтверждают достижение её цели?
2. Какая проверка показала, что получен правильный результат, и что именно она измеряет?
3. Какую типичную ошибку можно допустить в этой работе и как её обнаружить?
4. Какие данные, настройки или права нельзя переносить из учебной среды в рабочую без согласования?

Критерии оценивания

Оценка	**Требования к выполнению**
5 «отлично»	Выполнены все этапы; результат воспроизводим, проверки (включая негативные, если предусмотрены) подтверждены; примеры адаптированы к заданию; отчёт содержит команды/артефакты и самостоятельные объяснения.
4 «хорошо»	Основные этапы и контрольные результаты выполнены; допущены единичные неточности в объяснениях или оформлении, не влияющие на корректность и безопасность результата.
3 «удовлетворительно»	Получен минимальный рабочий результат, но часть проверок или пояснений отсутствует; есть ошибки, устранённые с подсказкой; отчёт неполон.
2 «неудовлетворительно»	Основной результат отсутствует либо ошибки не устранены; ключевые этапы или доказательства проверки не представлены;

	допущены критические нарушения безопасности учебного стенда.

Оценивание проводится по фактически продемонстрированному результату, качеству объяснения шагов и полноте отчёта.

Практическое занятие №24. Контейнеризация и запуск конвейера

Продолжительность: 6 академических часа(ов).

Проверяемые компетенции:

ПК 2.9 — Осуществлять конвейерную обработку данных для систем ИИ.

ПК 2.10 — Осуществлять мониторинг и оркестрацию конвейеров обработки данных.

ПК 2.11 — Осуществлять контроль качества данных в конвейерах обработки.

Цель занятия: Контейнеризация учебного конвейера.

Теоретические сведения и объяснение: Docker Compose описывает состав контейнеров и связи между ними. Цель — воспроизводимый запуск на отдельном стенде без ручной установки каждого пакета на узле.

Оборудование и исходные материалы: учебный ПК, Python и инструменты соответствующего занятия (MongoDB/PySpark/Airflow/Docker/средство визуализации).

Порядок выполнения и примеры

Этап 1. Структура проекта.

Создайте каталог с pipeline.py, requirements.txt, Dockerfile, compose.yaml и подкаталогами data/ и output/. Запишите назначение каждого файла.

Пояснение и пример: Секреты храните во внешнем .env, который не публикуется и исключён из системы версий.

Этап 2. Создание образа.

В Dockerfile используйте совместимый Python-образ, COPY requirements.txt, RUN pip install --no-cache-dir -r requirements.txt, COPY pipeline.py и CMD ["python","pipeline.py"].

Пояснение и пример: Фиксируйте версии библиотек для повторяемости; для PySpark нужен совместимый runtime Java.

Этап 3. Создание Compose-конфигурации.

В compose.yaml определите сервис pipeline: build: ., volumes для ./data:/app/data:ro и ./output:/app/output. Для лаборатории достаточно одного сервиса при отсутствии внешних зависимостей.

Пояснение и пример: Режим :ro защищает входные данные от случайного изменения.

Этап 4. Сборка и запуск.

Выполните docker compose config, docker compose build, docker compose up --abort-on-container-exit. Проверьте вывод приложения и выходные файлы.

Пояснение и пример: Команда config помогает выявить ошибки YAML до фактического запуска.

Этап 5. Диагностика и повтор.

Измените тестовый входной файл, перезапустите конвейер, сравните результат и логи. Опишите обработку ошибки «файл не найден» и корректный код завершения.

Пояснение и пример: Не используйте бесконтрольную очистку volumes и контейнеров на общем сервере.

Что должно получиться

Dockerfile, compose.yaml, схема папок, лог успешного запуска и перечень типовых ошибок.

Самопроверка и контрольные вопросы

1. Какие действия в работе №24 подтверждают достижение её цели?
2. Какая проверка показала, что получен правильный результат, и что именно она измеряет?
3. Какую типичную ошибку можно допустить в этой работе и как её обнаружить?
4. Какие данные, настройки или права нельзя переносить из учебной среды в рабочую без согласования?

Критерии оценивания

Оценка	Требования к выполнению
5 «отлично»	Выполнены все этапы; результат воспроизводим, проверки (включая негативные, если предусмотрены) подтверждены; примеры адаптированы к заданию; отчёт содержит команды/артефакты и самостоятельные объяснения.
4 «хорошо»	Основные этапы и контрольные результаты выполнены; допущены единичные неточности в объяснениях или оформлении, не влияющие на корректность и безопасность результата.
3 «удовлетворительно»	Получен минимальный рабочий результат, но часть проверок или пояснений отсутствует; есть ошибки, устранённые с подсказкой; отчёт неполон.
2 «неудовлетворительно»	Основной результат отсутствует либо ошибки не устранены; ключевые этапы или доказательства проверки не представлены; допущены критические нарушения безопасности учебного стенда.

Оценивание проводится по фактически продемонстрированному результату, качеству объяснения шагов и полноте отчёта.

Тема 3.3. Мониторинг и обеспечение качества конвейера

Практическое занятие №25. Настройка проверок качества и аномалий

Продолжительность: 4 академических часа(ов).

Проверяемые компетенции:

ПК 2.9 — Осуществлять конвейерную обработку данных для систем ИИ.

ПК 2.10 — Осуществлять мониторинг и оркестрацию конвейеров обработки данных.

ПК 2.11 — Осуществлять контроль качества данных в конвейерах обработки.

Цель занятия: Автоматические проверки и поиск аномалий.

Теоретические сведения и объяснение: Контроль качества удобно реализовать как набор тестов, прерывающих конвейер при критическом дефекте. Аномалия — наблюдение, требующее проверки, а не всегда ошибочная запись.

Оборудование и исходные материалы: учебный ПК, Python, pandas, синтетические CSV/JSON, среда Jupyter или редактор кода.

Порядок выполнения и примеры

Этап 1. Каталог проверок.

Для orders_clean определите правила: order_id уникален и непуст; amount >= 0; дата распознаётся; доля пропущенных category <= 5%.

Пояснение и пример: Порог 5% — учебное условие; в реальной системе он утверждается владельцем данных.

Этап 2. Реализация проверок.

Напишите validate(df): assert df["order_id"].notna().all(); assert df["order_id"].is_unique; assert (df["amount"] >= 0).all(). Сформируйте отдельное сообщение о каждой нарушенной проверке.

Пояснение и пример: Лучше собирать список ошибок, чем останавливаться на первой, если это допустимо по критичности.

Этап 3. Обнаружение выбросов.
Для amount вычислите Q1, Q3 и IQR=Q3-Q1; пометьте значения выше Q3+1.5×IQR. Постройте таблицу «order_id / amount / признак аномалии».
Пояснение и пример: Правило IQR — эвристика. Не удаляйте помеченные заказы без разбора.
Этап 4. Тестирование с ошибками.
Создайте четыре контролируемые ошибки: дубликат ID, отрицательная сумма, пустой ID, превышенный процент пропусков. Запустите проверки и сохраните сообщения.
Пояснение и пример: Проверка считается реализованной, когда тест способен обнаружить искусственно внесённый дефект.
Этап 5. Интеграция с конвейером.
Подключите validate к этапу Airflow перед загрузкой результата. Задача должна завершаться ошибкой при нарушении критического правила.
Пояснение и пример: Опишите, где хранится отчёт о проверках и кто получает уведомление.

Что должно получиться

Код проверок, таблица порогов, четыре негативных теста и схема включения в DAG.

Самопроверка и контрольные вопросы

1. Какие действия в работе №25 подтверждают достижение её цели?
2. Какая проверка показала, что получен правильный результат, и что именно она измеряет?
3. Какую типичную ошибку можно допустить в этой работе и как её обнаружить?
4. Какие данные, настройки или права нельзя переносить из учебной среды в рабочую без согласования?

Критерии оценивания

Оценка	**Требования к выполнению**
5 «отлично»	Выполнены все этапы; результат воспроизводим, проверки (включая негативные, если предусмотрены) подтверждены; примеры адаптированы к заданию; отчёт содержит команды/артефакты и самостоятельные объяснения.
4 «хорошо»	Основные этапы и контрольные результаты выполнены; допущены единичные неточности в объяснениях или оформлении, не влияющие на корректность и безопасность результата.
3 «удовлетворительно»	Получен минимальный рабочий результат, но часть проверок или пояснений отсутствует; есть ошибки, устранённые с подсказкой; отчёт неполон.
2 «неудовлетворительно»	Основной результат отсутствует либо ошибки не устранены; ключевые этапы или доказательства проверки не представлены; допущены критические нарушения безопасности учебного стенда.

Оценивание проводится по фактически продемонстрированному результату, качеству объяснения шагов и полноте отчёта.

Практическое занятие №26. Настройка мониторинга и алертинга

Продолжительность: 6 академических часа(ов).

Проверяемые компетенции:

ПК 2.9 — Осуществлять конвейерную обработку данных для систем ИИ.
ПК 2.10 — Осуществлять мониторинг и оркестрацию конвейеров обработки данных.
ПК 2.11 — Осуществлять контроль качества данных в конвейерах обработки.

Цель занятия: Мониторинг работы DAG и уведомления.

Теоретические сведения и объяснение: Мониторинг показывает, завершились ли задачи вовремя и сколько данных обработано. Алертинг уведомляет о нарушении заранее заданного порога.

Оборудование и исходные материалы: учебный ПК, Python и инструменты соответствующего занятия (MongoDB/PySpark/Airflow/Docker/средство визуализации).

Порядок выполнения и примеры

Этап 1. Выбор метрик.

Определите не менее пяти метрик: статус DAG, длительность задачи, число обработанных строк, число отклонённых строк, возраст последней успешной загрузки.

Пояснение и пример: Для каждой метрики задайте единицу измерения, источник и ожидаемое значение.

Этап 2. Сбор событий.

В задачах Airflow добавьте структурированные сообщения логирования о начале, окончании, количестве входных/выходных строк и ошибках.

Пояснение и пример: Лог должен помогать найти запуск, но не содержать пароль или полный чувствительный набор данных.

Этап 3. Разработка порогов.

Для учебного примера задайте тревогу при 2 ошибочных запусках подряд или длительности validate > 10 минут. Создайте таблицу «Метрика / Порог / Уровень / Действие».

Пояснение и пример: Пороги демонстрационные и должны быть уточнены по истории запусков.

Этап 4. Настройка уведомления.

В зависимости от имеющегося стенда настройте on_failure_callback или тестовый webhook/почтовое уведомление через секреты Airflow Connections. Можно использовать локальный имитатор отправки.

Пояснение и пример: Не вставляйте токены в DAG-файл. Не отправляйте реальные письма без учебного согласования.

Этап 5. Проверка срабатывания.

Смоделируйте ошибку validate, убедитесь в статусе failed и наличии события уведомления. После исправления выполните успешный запуск и проверьте отсутствие ложной тревоги.

Пояснение и пример: Оформите два протокола: «сбой» и «восстановление».

Что должно получиться

Таблица метрик и порогов, скриншоты запусков, пример уведомления без секретов и протокол восстановления.

Самопроверка и контрольные вопросы

1. Какие действия в работе №26 подтверждают достижение её цели?
2. Какая проверка показала, что получен правильный результат, и что именно она измеряет?
3. Какую типичную ошибку можно допустить в этой работе и как её обнаружить?
4. Какие данные, настройки или права нельзя переносить из учебной среды в рабочую без согласования?

Критерии оценивания

Оценка	**Требования к выполнению**
5 «отлично»	Выполнены все этапы; результат воспроизводим, проверки (включая негативные, если предусмотрены) подтверждены; примеры адаптированы к заданию; отчёт содержит команды/артефакты и самостоятельные объяснения.

4 «хорошо»	Основные этапы и контрольные результаты выполнены; допущены единичные неточности в объяснениях или оформлении, не влияющие на корректность и безопасность результата.
3 «удовлетворительно»	Получен минимальный рабочий результат, но часть проверок или пояснений отсутствует; есть ошибки, устранённые с подсказкой; отчёт неполон.
2 «неудовлетворительно»	Основной результат отсутствует либо ошибки не устранены; ключевые этапы или доказательства проверки не представлены; допущены критические нарушения безопасности учебного стенда.

Оценивание проводится по фактически продемонстрированному результату, качеству объяснения шагов и полноте отчёта.

Практическое занятие №27. Создание дашборда и настройка контроля доступа

Продолжительность: 4 академических часа(ов).

Проверяемые компетенции:

ПК 2.9 — Осуществлять конвейерную обработку данных для систем ИИ.

ПК 2.10 — Осуществлять мониторинг и оркестрацию конвейеров обработки данных.

ПК 2.11 — Осуществлять контроль качества данных в конвейерах обработки.

Цель занятия: Дашборд качества и разграничение доступа.

Теоретические сведения и объяснение: Дашборд объединяет ключевые показатели качества и состояния конвейера; права доступа предотвращают просмотр и изменение чужих данных.

Оборудование и исходные материалы: учебный ПК, Python и инструменты соответствующего занятия (MongoDB/PySpark/Airflow/Docker/средство визуализации).

Порядок выполнения и примеры

Этап 1. Проектирование показателей.

Определите четыре карточки: последний успешный запуск, число обработанных записей, доля отклонённых, средняя длительность DAG. Добавьте таблицу пяти последних запусков.

Пояснение и пример: Карточка должна показывать дату/единицу измерения; без этого показатель неоднозначен.

Этап 2. Подготовка витрины.

Сформируйте учебную таблицу pipeline_runs(run_id, started_at, status, processed_count, rejected_count, duration_sec). Добавьте не менее 10 синтетических запусков.

Пояснение и пример: Вычислите reject_rate = rejected_count / NULLIF(processed_count + rejected_count,0).

Этап 3. Построение визуализаций.

В доступном инструменте BI либо в Python (pandas + matplotlib/Streamlit) создайте график длительности по времени, круговую/столбчатую диаграмму статусов и таблицу запусков.

Пояснение и пример: Не допускайте, чтобы цвет был единственным способом различения success/failed: нужны текстовые подписи.

Этап 4. Разграничение пользователей.

Опишите роли viewer и operator: viewer только просматривает сводные метрики, operator видит детали ошибок и запускает обновление. Реализуйте права средствами выбранного инструмента или демонстрационной БД.

Пояснение и пример: Проверьте два сценария: разрешённый просмотр и запрещённое изменение для viewer.

Этап 5. Приёмка дашборда.

Проверьте значения двух карточек прямым SQL/pandas-расчётом, зафиксируйте дату обновления и два теста доступа. Подготовьте инструкцию пользователя на 5–7 предложений.

Пояснение и пример: Отдельно отметьте ограничения имитации прав, если выбранный инструмент

не поддерживает управление пользователями.

Что должно получиться

Дашборд с графиком и таблицей, витрина данных, протокол сверки метрик и проверка двух ролей.

Самопроверка и контрольные вопросы

1. Какие действия в работе №27 подтверждают достижение её цели?
2. Какая проверка показала, что получен правильный результат, и что именно она измеряет?
3. Какую типичную ошибку можно допустить в этой работе и как её обнаружить?
4. Какие данные, настройки или права нельзя переносить из учебной среды в рабочую без согласования?

Критерии оценивания

Оценка	**Требования к выполнению**
5 «отлично»	Выполнены все этапы; результат воспроизводим, проверки (включая негативные, если предусмотрены) подтверждены; примеры адаптированы к заданию; отчёт содержит команды/артефакты и самостоятельные объяснения.
4 «хорошо»	Основные этапы и контрольные результаты выполнены; допущены единичные неточности в объяснениях или оформлении, не влияющие на корректность и безопасность результата.
3 «удовлетворительно»	Получен минимальный рабочий результат, но часть проверок или пояснений отсутствует; есть ошибки, устранённые с подсказкой; отчёт неполон.
2 «неудовлетворительно»	Основной результат отсутствует либо ошибки не устранены; ключевые этапы или доказательства проверки не представлены; допущены критические нарушения безопасности учебного стенда.

Оценивание проводится по фактически продемонстрированному результату, качеству объяснения шагов и полноте отчёта.

3.3 Задания для оценки освоения УП.03.01

Раздел 1. Обучение готовых моделей искусственного интеллекта

Тема 1.1. Разработка сценариев обучения готовых моделей.

Практическое занятие №1 Знакомство с готовыми ИИ-моделями

Проверяемые результаты обучения:

- ПК 3.1 Осуществлять выбор готовых моделей искусственного интеллекта
- ПК 3.2 Формировать сценарии обучения готовых моделей искусственного интеллекта
- ПК 3.3 Проводить обучение и последующую калибровку готовых моделей искусственного интеллекта
- ПК 3.4 Контролировать результат обучения
- ПК 3.5 Оформлять результат проведения процедуры обучения
- ПК 3.6 Формировать запросы для работы с искусственным интеллектом с целью визуализации данных
- ПК 3.7 Формировать сценарии обучения готовых моделей искусственного интеллекта
- ПК 3.8 Интегрировать готовые модели искусственного интеллекта в информационные системы с использованием программных интерфейсов
- ПК 3.9 Создавать контент программного продукта средствами нейросетей
- ПК 3.10 Разрабатывать блоки программного кода с использованием нейросетей
- ПК 3.11 Создавать и управлять концепцией ИТ-продукта
- ПК 3.12 Планировать управление требованиями заказчика в проектах любого уровня сложности в области ИТ

1. Цель практической работы.

Познакомиться с возможностями готовых моделей искусственного интеллекта и получить практический опыт их использования для решения задач, связанных с текстом, программным кодом и данными. Научиться формулировать запросы, анализировать ответы ИИ и проверять полученные результаты.

2. Задачи практической работы.

Используя доступную ИИ-модель, выполнить несколько практических задач:

1. Сформировать и сравнить ответы на разные запросы.
2. Использовать ИИ для генерации и анализа текста.
3. Выполнить классификацию информации.
4. Использовать ИИ для поиска ошибки в программном коде.
5. Проанализировать табличные данные.
6. Преобразовать текстовую информацию в структурированный вид.

Самостоятельно выбрать дополнительную задачу для решения с помощью ИИ.

3. Этапы выполнения:

1. Откройте доступную ИИ-модель и создайте новый диалог.
2. Выполните запрос:
   «Объясни простыми словами, что такое искусственный интеллект».

Затем уточните запрос:
«Объясни простыми словами, что такое искусственный интеллект для студента 4 курса, изучающего программирование. Приведи 3 практических примера».

Сравните ответы и сделайте скриншот.

1. Попросите ИИ создать инструкцию по установке программного приложения. Затем попросите изменить её для начинающего пользователя и представить в виде 7 последовательных шагов.

Сравните полученные варианты.

- Передайте ИИ текст:

«Клиент сообщает, что после обновления мобильного приложения перестала работать авторизация. При вводе правильного пароля появляется ошибка. Повторная установка приложения проблему не решила».

Попросите определить категорию проблемы, возможные причины и необходимые дополнительные сведения.

Проверьте, какие выводы являются фактами, а какие предположениями ИИ.

1. Передайте ИИ код:

```
def calculate_average(numbers):
    total = 0
    for i in range(len(numbers) - 1):
        total += numbers[i]
    return total / len(numbers)
```

Попросите найти ошибку и предложить исправленный вариант.
Самостоятельно проверьте код и результат его выполнения.

1. Передайте ИИ данные:

Клавиатура — 5 шт. — 2500 руб.
Мышь — 8 шт. — 1200 руб.
Монитор — 3 шт. — 18500 руб.
Наушники — 6 шт. — 4500 руб.

Попросите рассчитать стоимость каждой позиции и общую сумму. Проверьте расчёты самостоятельно.

1. Передайте ИИ текст:

«Для регистрации необходимо указать имя, email и номер телефона. После регистрации отправляется код подтверждения. Пароль должен содержать не менее восьми символов».

Попросите представить информацию сначала в виде таблицы, затем в формате JSON.

- Самостоятельно выберите ещё одну профессиональную задачу и решите её с помощью ИИ. Например: создание SQL-запроса, генерация тестовых данных, поиск ошибки в коде, составление тест-кейсов или создание технического задания.
- Составьте итоговую таблицу:

Задача | Как использовался ИИ | Результат | Требуется ли проверка

1. Определите одну задачу в собственном проекте, которую в дальнейшем можно решить с помощью ИИ. Укажите: какую проблему решает функция, какие данные получает ИИ и какой результат должен вернуть.

5. Результат: Студент должен представить результаты выполненных заданий, не менее 5 скриншотов взаимодействия с ИИ, проверенный программный код, результаты расчётов, пример JSON и описание предполагаемой AI-функции собственного проекта.

Критерии оценки.

«5» «отлично»

- студент продемонстрировал готовность и способность к самостоятельному выполнению видов работ в стандартных ситуациях;
- все работы выполнил правильно и уверенно;
- проявил активность в освоении новых практических умений;
- содержание отчета по практическим работам полноценно отражает информацию по выполненным видам работ, установленных программой практики;
- оформление отчета по практическим работам полностью соответствует предъявляемым требованиям;

- при выполнении практических работ студент уверенно продемонстрировал обладание необходимыми компонентами общих и профессиональных компетенций;

«4» «хорошо»

- студент продемонстрировал готовность и способность к самостоятельному выполнению видов работ в стандартных ситуациях;
- все работы выполнил правильно и уверенно;
- содержание отчета по практическим работам в целом правильно отражает информацию по выполненным видам работ;
- оформление отчета по практическим работам соответствует предъявляемым требованиям, однако имеются замечания (некоторая неаккуратность, неполное описание проделанной работы и освоенных умений);
- при выполнении практических работ студент продемонстрировал обладание компонентами общих и профессиональных компетенций;
- характеристики студента положительные, имеются отдельные замечания и рекомендации;

«3» «удовлетворительно»

- при выполнении видов работ студент нуждался во внешнем сопровождении и контроле;
- в отчете по практическим работам содержится описание видов выполняемых работ согласно программе дисциплины, однако, записи неполные и неточные;
- отчет по практическим работам оформлен с отклонениями от установленных требований;
- при выполнении практических работ студент продемонстрировал обладание отдельными компонентами общих и профессиональных компетенций;

«**2» «неудовлетворительно»**

– студент не справился с запланированными видами работ;

– при выполнении практических работ студент продемонстрировал слабое обладание отдельными компонентами общих и профессиональных компетенций;

– отчет студентом не подготовлен.

Практическое занятие №2 Эффективное взаимодействие с ИИ: постановка задач и получение качественного ответа

Проверяемые результаты обучения:

- ПК 3.1 Осуществлять выбор готовых моделей искусственного интеллекта
- ПК 3.2 Формировать сценарии обучения готовых моделей искусственного интеллекта
- ПК 3.3 Проводить обучение и последующую калибровку готовых моделей искусственного интеллекта
- ПК 3.4 Контролировать результат обучения
- ПК 3.5 Оформлять результат проведения процедуры обучения

- ПК 3.6 Формировать запросы для работы с искусственным интеллектом с целью визуализации данных
- ПК 3.7 Формировать сценарии обучения готовых моделей искусственного интеллекта
- ПК 3.8 Интегрировать готовые модели искусственного интеллекта в информационные системы с использованием программных интерфейсов
- ПК 3.9 Создавать контент программного продукта средствами нейросетей
- ПК 3.10 Разрабатывать блоки программного кода с использованием нейросетей
- ПК 3.11 Создавать и управлять концепцией ИТ-продукта
- ПК 3.12 Планировать управление требованиями заказчика в проектах любого уровня сложности в области ИТ

1. Цель практической работы

Научиться правильно формулировать задачи для ИИ, составлять понятные и точные запросы, задавать контекст и ограничения, а также оценивать и улучшать полученный результат.

2. Задачи практической работы

Используя доступную ИИ-модель, выполнить серию заданий по составлению и улучшению запросов. Сравнить результаты простых и подробных запросов, определить влияние контекста, ограничений и формата ответа на качество результата.

3. Этапы выполнения

1. Выполните простой запрос:
   «Напиши описание мобильного приложения».

Оцените полученный результат.

1. Улучшите запрос, добавив роль, контекст, задачу и требования:
   «Ты преподаватель программирования. Напиши описание учебного мобильного приложения для студентов 3 курса. Опиши назначение, основные функции и целевую аудиторию. Используй простой технический стиль».

Сравните два результата.

1. Добавьте ограничения к запросу: объём текста, количество пунктов, стиль и формат ответа. Сравните результат с предыдущим вариантом.
2. Сформулируйте запрос на преобразование информации. Например: «Преобразуй описание приложения в таблицу со столбцами: функция, назначение, пользователь, результат».

Проверьте, насколько точно ИИ выполнил требования.

1. Используйте пример в запросе. Покажите ИИ небольшой образец желаемого результата и попросите выполнить аналогичную задачу для другого объекта.

Сравните результат с запросом без примера.

1. Составьте запрос для исправления программного кода. Укажите язык программирования, назначение программы, проблему и требуемый результат. Попросите ИИ не только исправить код, но и объяснить внесённые изменения.
2. Проверьте один из полученных ответов. Найдите неточности, лишнюю информацию или ошибки. Сформулируйте новый запрос с уточнениями и получите улучшенный результат.
3. Самостоятельно составьте качественный запрос для задачи из области программирования или собственного проекта. Используйте структуру: роль → контекст → задача → ограничения → формат результата.
4. Сравните первый и улучшенный запросы и сделайте вывод, какие изменения повлияли на качество ответа.

4. Результат работы: Студент должен представить не менее 3 сравнений запросов и ответов, примеры улучшенных промптов, результаты проверки ответов и один самостоятельно разработанный подробный запрос для собственного проекта.

Критерии оценки.

«5» «отлично»

- студент продемонстрировал готовность и способность к самостоятельному выполнению видов работ в стандартных ситуациях;
- все работы выполнил правильно и уверенно;
- проявил активность в освоении новых практических умений;
- содержание отчета по практическим работам полноценно отражает информацию по выполненным видам работ, установленных программой практики;

- оформление отчета по практическим работам полностью соответствует предъявляемым требованиям;
- при выполнении практических работ студент уверенно продемонстрировал обладание необходимыми компонентами общих и профессиональных компетенций;

«4» «хорошо»

- студент продемонстрировал готовность и способность к самостоятельному выполнению видов работ в стандартных ситуациях;
- все работы выполнил правильно и уверенно;
- содержание отчета по практическим работам в целом правильно отражает информацию по выполненным видам работ;
- оформление отчета по практическим работам соответствует предъявляемым требованиям, однако имеются замечания (некоторая неаккуратность, неполное описание проделанной работы и освоенных умений);
- при выполнении практических работ студент продемонстрировал обладание компонентами общих и профессиональных компетенций;
- характеристики студента положительные, имеются отдельные замечания и рекомендации;

«3» «удовлетворительно»

- при выполнении видов работ студент нуждался во внешнем сопровождении и контроле;
- в отчете по практическим работам содержится описание видов выполняемых работ согласно программе дисциплины, однако, записи неполные и неточные;
- отчет по практическим работам оформлен с отклонениями от установленных требований;
- при выполнении практических работ студент продемонстрировал обладание отдельными компонентами общих и профессиональных компетенций;

«**2» «неудовлетворительно»**

– студент не справился с запланированными видами работ;

– при выполнении практических работ студент продемонстрировал слабое обладание отдельными компонентами общих и профессиональных компетенций;

– отчет студентом не подготовлен.

Практическое занятие №3 Работа ИИ с документами и данными

Проверяемые результаты обучения:

- ПК 3.1 Осуществлять выбор готовых моделей искусственного интеллекта
- ПК 3.2 Формировать сценарии обучения готовых моделей искусственного интеллекта
- ПК 3.3 Проводить обучение и последующую калибровку готовых моделей искусственного интеллекта
- ПК 3.4 Контролировать результат обучения

- ПК 3.5 Оформлять результат проведения процедуры обучения
- ПК 3.6 Формировать запросы для работы с искусственным интеллектом с целью визуализации данных
- ПК 3.7 Формировать сценарии обучения готовых моделей искусственного интеллекта
- ПК 3.8 Интегрировать готовые модели искусственного интеллекта в информационные системы с использованием программных интерфейсов
- ПК 3.9 Создавать контент программного продукта средствами нейросетей
- ПК 3.10 Разрабатывать блоки программного кода с использованием нейросетей
- ПК 3.11 Создавать и управлять концепцией ИТ-продукта
- ПК 3.12 Планировать управление требованиями заказчика в проектах любого уровня сложности в области ИТ

1. Цель практической работы

Научиться использовать ИИ для анализа документов и структурированных данных, извлечения информации, преобразования данных и подготовки кратких выводов.

2. Задачи практической работы

Используя доступную ИИ-модель, выполнить задания по обработке документов и данных: анализ текста, поиск информации, составление краткого содержания, преобразование данных и подготовка выводов.

3. Этапы выполнения

- Загрузите в ИИ небольшой документ или текстовый файл. Попросите определить его тему, основные разделы и ключевые сведения.
- Попросите ИИ составить краткое содержание документа, выделив основные мысли и важные факты.
- Найдите в документе конкретную информацию с помощью ИИ. Например: даты, фамилии, требования, технические характеристики или ключевые показатели.
- Попросите ИИ преобразовать информацию из документа в таблицу. Проверьте, не потерялись ли данные при преобразовании.

- Загрузите таблицу с данными. Попросите ИИ определить основные показатели, найти максимальные и минимальные значения и сформулировать несколько выводов.
- Попросите ИИ найти возможные ошибки или противоречия в предоставленных данных. Самостоятельно проверьте несколько найденных результатов.
- Используйте несколько документов по одной теме. Попросите ИИ сравнить их и определить общие сведения и различия.
- На основе обработанных документов попросите ИИ составить небольшой FAQ из 5–7 вопросов и ответов.
- Самостоятельно выберите документ или набор данных, связанный с программированием или собственным проектом, и выполните его анализ с помощью ИИ.

Результат: Студент должен представить обработанный документ, краткое содержание, таблицу с извлечёнными данными, результаты анализа, найденные ошибки или противоречия и итоговый FAQ

Критерии оценки.

«5» «отлично»

- студент продемонстрировал готовность и способность к самостоятельному выполнению видов работ в стандартных ситуациях;
- все работы выполнил правильно и уверенно;
- проявил активность в освоении новых практических умений;
- содержание отчета по практическим работам полноценно отражает информацию по выполненным видам работ, установленных программой практики;
- оформление отчета по практическим работам полностью соответствует предъявляемым требованиям;
- при выполнении практических работ студент уверенно продемонстрировал обладание необходимыми компонентами общих и профессиональных компетенций;

«4» «хорошо»

- студент продемонстрировал готовность и способность к самостоятельному выполнению видов работ в стандартных ситуациях;
- все работы выполнил правильно и уверенно;
- содержание отчета по практическим работам в целом правильно отражает информацию по выполненным видам работ;
- оформление отчета по практическим работам соответствует предъявляемым требованиям, однако имеются замечания (некоторая неаккуратность, неполное описание проделанной работы и освоенных умений);
- при выполнении практических работ студент продемонстрировал обладание компонентами общих и профессиональных компетенций;

- характеристики студента положительные, имеются отдельные замечания и рекомендации;

«3» «удовлетворительно»

- при выполнении видов работ студент нуждался во внешнем сопровождении и контроле;
- в отчете по практическим работам содержится описание видов выполняемых работ согласно программе дисциплины, однако, записи неполные и неточные;
- отчет по практическим работам оформлен с отклонениями от установленных требований;
- при выполнении практических работ студент продемонстрировал обладание отдельными компонентами общих и профессиональных компетенций;

«2» «неудовлетворительно»

– студент не справился с запланированными видами работ;

– при выполнении практических работ студент продемонстрировал слабое обладание отдельными компонентами общих и профессиональных компетенций;

– отчет студентом не подготовлен.

Практическое занятие №4 Создание собственного ИИ-помощника

Проверяемые результаты обучения:

- ПК 3.1 Осуществлять выбор готовых моделей искусственного интеллекта
- ПК 3.2 Формировать сценарии обучения готовых моделей искусственного интеллекта
- ПК 3.3 Проводить обучение и последующую калибровку готовых моделей искусственного интеллекта
- ПК 3.4 Контролировать результат обучения
- ПК 3.5 Оформлять результат проведения процедуры обучения
- ПК 3.6 Формировать запросы для работы с искусственным интеллектом с целью визуализации данных
- ПК 3.7 Формировать сценарии обучения готовых моделей искусственного интеллекта
- ПК 3.8 Интегрировать готовые модели искусственного интеллекта в информационные системы с использованием программных интерфейсов
- ПК 3.9 Создавать контент программного продукта средствами нейросетей
- ПК 3.10 Разрабатывать блоки программного кода с использованием нейросетей
- ПК 3.11 Создавать и управлять концепцией ИТ-продукта
- ПК 3.12 Планировать управление требованиями заказчика в проектах любого уровня сложности в области ИТ

1. Цель практической работы

Научиться создавать и настраивать собственного ИИ-помощника для решения конкретной задачи, задавать ему роль, правила поведения, формат ответов и проверять качество его работы.

2. Задачи практической работы

Создать собственного ИИ-помощника для учебной или профессиональной задачи. Помощник должен иметь определённую специализацию, инструкции и несколько примеров запросов.

3. Этапы выполнения

1. Выберите назначение помощника.

Например:
«Помощник по программированию Java»;
«Помощник преподавателя»;
«Консультант интернет-магазина»;
«Помощник по технической поддержке».

1. Определите роль и правила помощника.

Пример инструкции:

«Ты — помощник по программированию Java. Помогай студентам разбирать ошибки в коде. Объясняй решения простым языком. Не давай готовый ответ сразу — сначала объясняй причину ошибки. В конце каждого ответа показывай исправленный вариант кода».

1. Добавьте формат ответа.

Например:

«Каждый ответ оформляй в следующем виде:
Проблема:

Причина:
Решение:
Исправленный код:
Рекомендация:»

1. Проверьте помощника на нескольких запросах.

Например:

«Почему этот код не работает?»

```
int age = 17;

if (age >= 18) {
   System.out.println("Доступ разрешен");
} else {
   System.out.println("Доступ запрещен");
}
```

Попросите помощника объяснить работу программы и предложить изменение условия.

1. Добавьте работу с ошибками.

Например, передайте код:

```
public class Main {
   public static void main(String[] args) {
      int a = 10;
      int b = 0;

      System.out.println(a / b);
   }
}
```

Попросите помощника определить проблему, объяснить её причину и предложить безопасный вариант исправления.

1. Добавьте пример ожидаемого поведения помощника.

Например:

Запрос:
«Что такое цикл for?»

Ожидаемый формат:
«Цикл for используется для многократного выполнения блока команд. Сначала задаётся начальное значение, затем условие продолжения и изменение счётчика».

После этого проверьте, будет ли помощник придерживаться заданного стиля.

1. Проверьте помощника на разных типах задач: объяснение кода, поиск ошибки, написание небольшого фрагмента программы, объяснение термина.
2. Проверьте качество ответов. Определите, соблюдает ли помощник заданную роль, форматирует ли ответы правильно и не допускает ли ошибок.
3. Внесите изменения в инструкцию помощника и повторите несколько запросов. Сравните результаты до и после настройки.
4. Определите, какую функцию созданного помощника можно использовать в собственном проекте.

4. Результат: Студент должен представить настроенного ИИ-помощника, его инструкцию, не менее 5 тестовых запросов, примеры полученных ответов и описание возможного использования помощника в собственном проекте..

Критерии оценки.

«5» «отлично»

- студент продемонстрировал готовность и способность к самостоятельному выполнению видов работ в стандартных ситуациях;
- все работы выполнил правильно и уверенно;
- проявил активность в освоении новых практических умений;
- содержание отчета по практическим работам полноценно отражает информацию по выполненным видам работ, установленных программой практики;
- оформление отчета по практическим работам полностью соответствует предъявляемым требованиям;
- при выполнении практических работ студент уверенно продемонстрировал обладание необходимыми компонентами общих и профессиональных компетенций;

«4» «хорошо»

- студент продемонстрировал готовность и способность к самостоятельному выполнению видов работ в стандартных ситуациях;
- все работы выполнил правильно и уверенно;
- содержание отчета по практическим работам в целом правильно отражает информацию по выполненным видам работ;

- оформление отчета по практическим работам соответствует предъявляемым требованиям, однако имеются замечания (некоторая неаккуратность, неполное описание проделанной работы и освоенных умений);
- при выполнении практических работ студент продемонстрировал обладание компонентами общих и профессиональных компетенций;
- характеристики студента положительные, имеются отдельные замечания и рекомендации;

«3» «удовлетворительно»

- при выполнении видов работ студент нуждался во внешнем сопровождении и контроле;
- в отчете по практическим работам содержится описание видов выполняемых работ согласно программе дисциплины, однако, записи неполные и неточные;
- отчет по практическим работам оформлен с отклонениями от установленных требований;
- при выполнении практических работ студент продемонстрировал обладание отдельными компонентами общих и профессиональных компетенций;

«**2» «неудовлетворительно»**

– студент не справился с запланированными видами работ;

– при выполнении практических работ студент продемонстрировал слабое обладание отдельными компонентами общих и профессиональных компетенций;

– отчет студентом не подготовлен.

Практическое занятие №5 Генерация и редактирование изображений

Проверяемые результаты обучения:

- ПК 3.1 Осуществлять выбор готовых моделей искусственного интеллекта
- ПК 3.2 Формировать сценарии обучения готовых моделей искусственного интеллекта
- ПК 3.3 Проводить обучение и последующую калибровку готовых моделей искусственного интеллекта
- ПК 3.4 Контролировать результат обучения
- ПК 3.5 Оформлять результат проведения процедуры обучения
- ПК 3.6 Формировать запросы для работы с искусственным интеллектом с целью визуализации данных
- ПК 3.7 Формировать сценарии обучения готовых моделей искусственного интеллекта
- ПК 3.8 Интегрировать готовые модели искусственного интеллекта в информационные системы с использованием программных интерфейсов
- ПК 3.9 Создавать контент программного продукта средствами нейросетей
- ПК 3.10 Разрабатывать блоки программного кода с использованием нейросетей

- ПК 3.11 Создавать и управлять концепцией ИТ-продукта
- ПК 3.12 Планировать управление требованиями заказчика в проектах любого уровня сложности в области ИТ

1. Цель практической работы

Научиться использовать ИИ для создания изображений по текстовому описанию, изменения готовых изображений и управления результатом с помощью правильно составленного запроса.

2. Задачи практической работы

Создать несколько изображений с помощью ИИ, изменить готовое изображение, попробовать разные стили и сравнить результаты при изменении промпта.

3. Этапы выполнения

1. Создайте простое изображение по описанию.

Пример запроса:

«Создай изображение современного рабочего места программиста: ноутбук, монитор с кодом, клавиатура, чашка кофе, мягкое освещение, минималистичный интерьер».

Пример результата:





Сохраните изображение и промпт.

1. Измените описание изображения.

Добавьте стиль, композицию и освещение:

«Создай минималистичное рабочее место программиста в тёмном стиле, фиолетовая подсветка, два монитора с кодом, вид спереди, реалистичная фотография».

Сравните два результата.

1. Создайте изображение одного объекта в разных стилях.

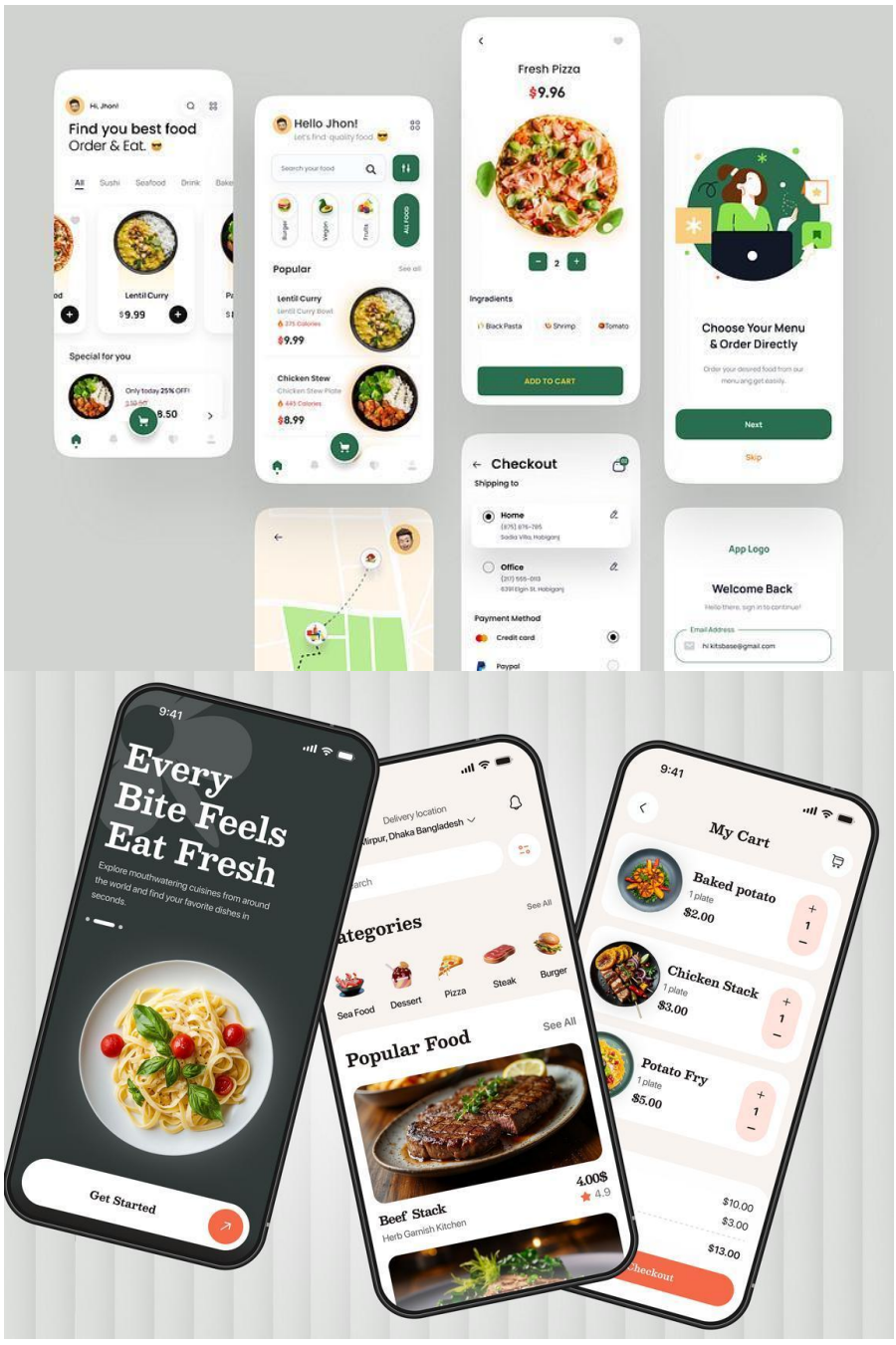
Например, используйте объект «мобильное приложение»:

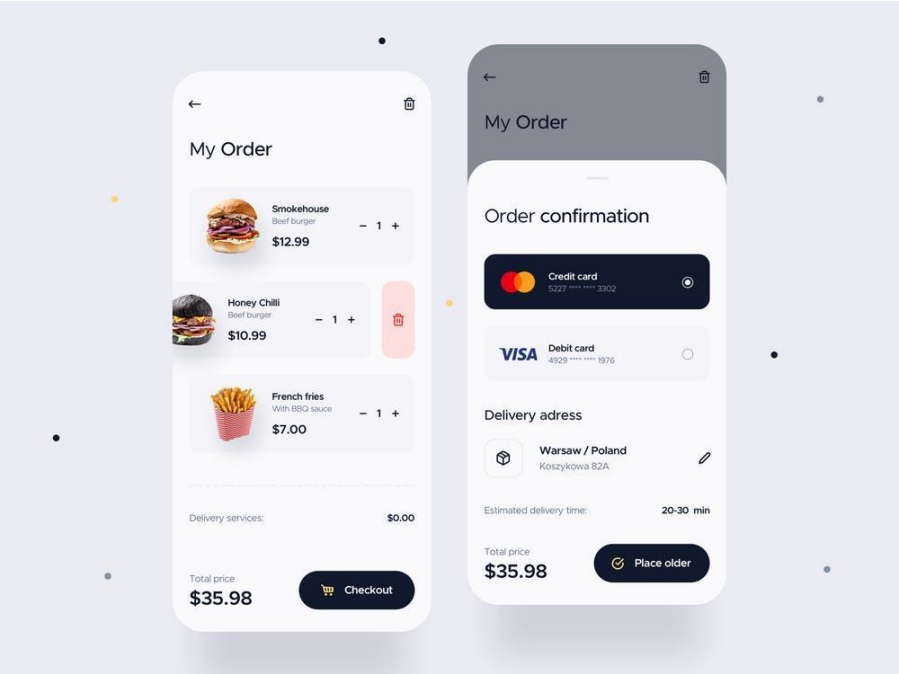
«Интерфейс мобильного приложения для заказа еды, современный UI/UX дизайн».

Затем измените стиль:

«Интерфейс мобильного приложения для заказа еды, современный минималистичный UI/UX, светлая цветовая схема, карточки товаров, нижняя навигация».

Пример визуального направления:





1. Выполните редактирование изображения.

Возьмите любое изображение и попросите ИИ изменить только один элемент.

Например:

«Замени фон на современный офис, сохрани человека, одежду и положение тела без изменений».

Сравните исходное и обработанное изображение.

1. Выполните несколько вариантов одного изображения.

Например:

«Создай три варианта логотипа для приложения по ремонту автомобилей: современный стиль, простая форма, без мелких деталей».

Сравните варианты и выберите наиболее подходящий.

1. Попробуйте управлять композицией.

Создайте изображение по запросу:

«Автомобильный сервис, автомобиль находится справа, слева свободное пространство для текста, горизонтальная композиция, современный корпоративный стиль».

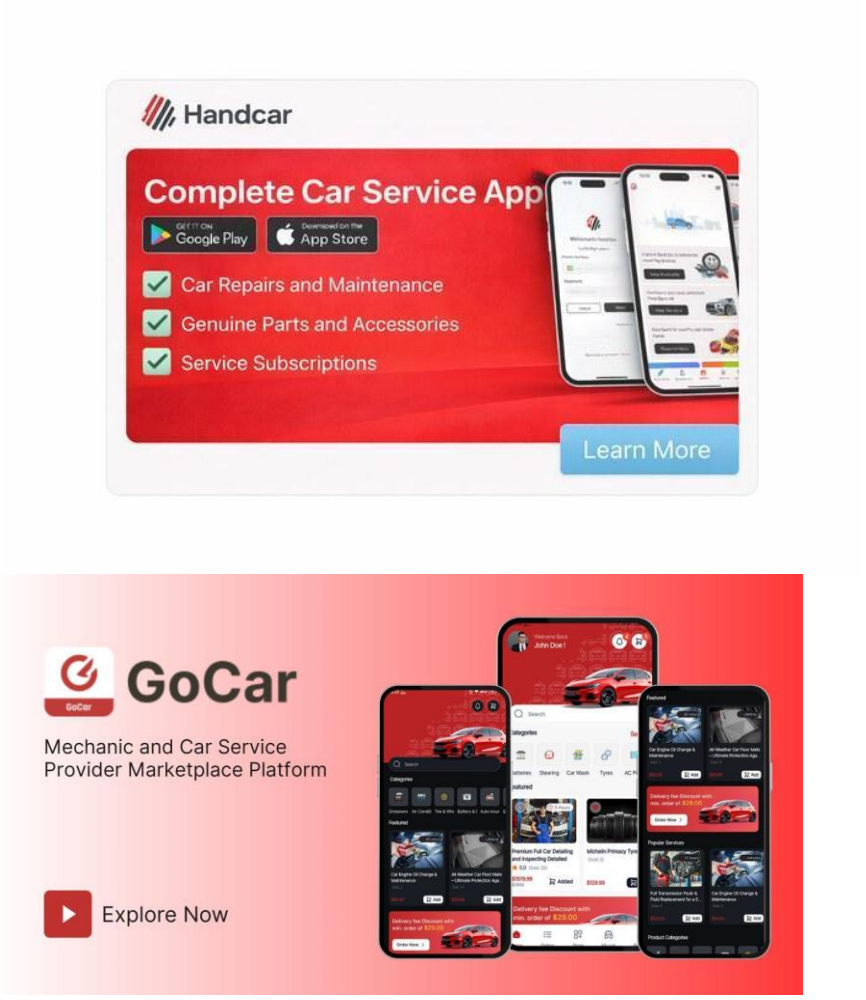
Проверьте, насколько точно ИИ выполнил требования.

1. Создайте изображение для собственного проекта.

Например, для приложения автосервиса:

«Современный рекламный баннер мобильного приложения автосервиса, автомобиль, инструменты, тёмный фон, красные акценты, место для заголовка и кнопки, горизонтальный формат».

Пример направления:



1. Измените созданное изображение с помощью нового запроса. Например:

«Сделай изображение более минималистичным, убери лишние объекты, оставь автомобиль и элементы интерфейса».

Сравните результат с первоначальным вариантом.

1. Проведите небольшой эксперимент. Создайте одно изображение с коротким промптом и второе — с подробным описанием объекта, стиля, композиции, освещения и формата.

Сравните результаты и определите, какие элементы промпта сильнее всего повлияли на изображение.

4. Результат: Студент должен представить не менее 4 сгенерированных изображений, 1 пример редактирования исходного изображения, несколько вариантов одного изображения, использованные промпты и итоговое изображение для собственного проекта.

Критерии оценки.

«5» «отлично»

- студент продемонстрировал готовность и способность к самостоятельному выполнению видов работ в стандартных ситуациях;
- все работы выполнил правильно и уверенно;
- проявил активность в освоении новых практических умений;
- содержание отчета по практическим работам полноценно отражает информацию по выполненным видам работ, установленных программой практики;
- оформление отчета по практическим работам полностью соответствует предъявляемым требованиям;
- при выполнении практических работ студент уверенно продемонстрировал обладание необходимыми компонентами общих и профессиональных компетенций;

«4» «хорошо»

- студент продемонстрировал готовность и способность к самостоятельному выполнению видов работ в стандартных ситуациях;
- все работы выполнил правильно и уверенно;
- содержание отчета по практическим работам в целом правильно отражает информацию по выполненным видам работ;
- оформление отчета по практическим работам соответствует предъявляемым требованиям, однако имеются замечания (некоторая неаккуратность, неполное описание проделанной работы и освоенных умений);
- при выполнении практических работ студент продемонстрировал обладание компонентами общих и профессиональных компетенций;
- характеристики студента положительные, имеются отдельные замечания и рекомендации;

«3» «удовлетворительно»

- при выполнении видов работ студент нуждался во внешнем сопровождении и контроле;
- в отчете по практическим работам содержится описание видов выполняемых работ согласно программе дисциплины, однако, записи неполные и неточные;
- отчет по практическим работам оформлен с отклонениями от установленных требований;
- при выполнении практических работ студент продемонстрировал обладание отдельными компонентами общих и профессиональных компетенций;

«2» «неудовлетворительно»

– студент не справился с запланированными видами работ;

– при выполнении практических работ студент продемонстрировал слабое обладание отдельными компонентами общих и профессиональных компетенций;

– отчет студентом не подготовлен.

Практическое занятие №6 Генерация контента с помощью ИИ

Проверяемые результаты обучения:

- ПК 3.1 Осуществлять выбор готовых моделей искусственного интеллекта
- ПК 3.2 Формировать сценарии обучения готовых моделей искусственного интеллекта
- ПК 3.3 Проводить обучение и последующую калибровку готовых моделей искусственного интеллекта
- ПК 3.4 Контролировать результат обучения
- ПК 3.5 Оформлять результат проведения процедуры обучения
- ПК 3.6 Формировать запросы для работы с искусственным интеллектом с целью визуализации данных
- ПК 3.7 Формировать сценарии обучения готовых моделей искусственного интеллекта
- ПК 3.8 Интегрировать готовые модели искусственного интеллекта в информационные системы с использованием программных интерфейсов
- ПК 3.9 Создавать контент программного продукта средствами нейросетей
- ПК 3.10 Разрабатывать блоки программного кода с использованием нейросетей
- ПК 3.11 Создавать и управлять концепцией ИТ-продукта
- ПК 3.12 Планировать управление требованиями заказчика в проектах любого уровня сложности в области ИТ

1. Цель практической работы

Научиться использовать ИИ для создания различных видов контента: текстов, рекламных материалов, инструкций, презентаций, таблиц и сценариев.

2. Задачи практической работы

С помощью ИИ создать несколько видов контента для разных задач и научиться адаптировать один материал под разные аудитории и форматы.

3. Этапы выполнения

- Создайте информационный текст.

Пример запроса:

«Напиши краткую статью на тему “Искусственный интеллект в мобильных приложениях” для студентов колледжа. Объём 500 слов. Используй понятный язык и приведи 3 практических примера».

1. Преобразуйте созданный текст.

Попросите ИИ представить его в другом формате:

«Преобразуй статью в краткий конспект из 7 основных пунктов».

Затем:

«На основе конспекта составь структуру презентации из 6 слайдов».

1. Создайте рекламный контент.

Пример:

«Создай рекламный текст для мобильного приложения автосервиса. Целевая аудитория — владельцы автомобилей. Текст должен быть коротким, современным и содержать призыв к действию».

Пример результата:

«Автосервис всегда под рукой. Записывайтесь на обслуживание, отслеживайте статус ремонта и получайте уведомления прямо в приложении. Удобно. Быстро. Без лишних звонков».

1. Создайте несколько вариантов одного материала.

Попросите ИИ сделать рекламный текст в трёх вариантах: официальный, дружелюбный и максимально краткий.

Сравните, как изменился стиль и содержание.

1. Создайте инструкцию.

Пример запроса:

«Составь пошаговую инструкцию для пользователя мобильного приложения по записи автомобиля на техническое обслуживание. Используй не более 8 шагов».

Затем попросите представить инструкцию в виде таблицы.

- Создайте сценарий.

Пример:

«Напиши сценарий короткого видеоролика на 30 секунд, рекламирующего мобильное приложение автосервиса. Укажи текст диктора и описание каждого кадра».

1. Создайте структурированные данные.

Попросите ИИ сформировать каталог услуг:

«Создай таблицу из 10 услуг автосервиса. Для каждой укажи название, описание, примерную стоимость и длительность выполнения».

Затем попросите преобразовать таблицу в JSON.

Пример:

```
{
  "name": "Замена масла",
  "price": 2500,
  "duration": "40 минут"
}
```

- Создайте контент для разных пользователей.

Используйте один материал и попросите адаптировать его для:
студента;
клиента;
специалиста;
руководителя.

Сравните полученные варианты.

1. Создайте контент для собственного проекта.

Выберите один формат: описание приложения, инструкция, рекламный текст, FAQ, сценарий, презентация или публикация для социальных сетей.

Сформулируйте подробный запрос и получите итоговый материал.

4. Проверьте результат.

Проверьте фактическую информацию, ошибки, соответствие заданному формату и отсутствие лишнего текста. При необходимости доработайте запрос

5. Результат: Студент должен представить не менее 5 созданных материалов разных типов, несколько вариантов одного контента, пример структурированных данных и один готовый материал для собственного проекта.

Критерии оценки.

«5» «отлично»

- студент продемонстрировал готовность и способность к самостоятельному выполнению видов работ в стандартных ситуациях;
- все работы выполнил правильно и уверенно;
- проявил активность в освоении новых практических умений;
- содержание отчета по практическим работам полноценно отражает информацию по выполненным видам работ, установленных программой практики;
- оформление отчета по практическим работам полностью соответствует предъявляемым требованиям;
- при выполнении практических работ студент уверенно продемонстрировал обладание необходимыми компонентами общих и профессиональных компетенций;

«4» «хорошо»

- студент продемонстрировал готовность и способность к самостоятельному выполнению видов работ в стандартных ситуациях;
- все работы выполнил правильно и уверенно;
- содержание отчета по практическим работам в целом правильно отражает информацию по выполненным видам работ;
- оформление отчета по практическим работам соответствует предъявляемым требованиям, однако имеются замечания (некоторая неаккуратность, неполное описание проделанной работы и освоенных умений);
- при выполнении практических работ студент продемонстрировал обладание компонентами общих и профессиональных компетенций;
- характеристики студента положительные, имеются отдельные замечания и рекомендации;

«3» «удовлетворительно»

- при выполнении видов работ студент нуждался во внешнем сопровождении и контроле;
- в отчете по практическим работам содержится описание видов выполняемых работ согласно программе дисциплины, однако, записи неполные и неточные;
- отчет по практическим работам оформлен с отклонениями от установленных требований;
- при выполнении практических работ студент продемонстрировал обладание отдельными компонентами общих и профессиональных компетенций;

«2» «неудовлетворительно»

– студент не справился с запланированными видами работ;

– при выполнении практических работ студент продемонстрировал слабое обладание отдельными компонентами общих и профессиональных компетенций;

– отчет студентом не подготовлен.

Практическое занятие №7 Выбор ИИ-инструментов для своего проекта

Проверяемые результаты обучения:

- ПК 3.1 Осуществлять выбор готовых моделей искусственного интеллекта
- ПК 3.2 Формировать сценарии обучения готовых моделей искусственного интеллекта
- ПК 3.3 Проводить обучение и последующую калибровку готовых моделей искусственного интеллекта
- ПК 3.4 Контролировать результат обучения
- ПК 3.5 Оформлять результат проведения процедуры обучения
- ПК 3.6 Формировать запросы для работы с искусственным интеллектом с целью визуализации данных
- ПК 3.7 Формировать сценарии обучения готовых моделей искусственного интеллекта
- ПК 3.8 Интегрировать готовые модели искусственного интеллекта в информационные системы с использованием программных интерфейсов
- ПК 3.9 Создавать контент программного продукта средствами нейросетей
- ПК 3.10 Разрабатывать блоки программного кода с использованием нейросетей
- ПК 3.11 Создавать и управлять концепцией ИТ-продукта
- ПК 3.12 Планировать управление требованиями заказчика в проектах любого уровня сложности в области ИТ

1. Цель практической работы

Научиться анализировать задачи собственного проекта и выбирать подходящие ИИ-инструменты в зависимости от требуемого результата.

2. Задачи практической работы

Проанализировать собственный проект, определить задачи, в которых может применяться ИИ, подобрать подходящие инструменты и обосновать их использование.

3. Этапы выполнения

1. Кратко опишите свой проект.

Пример:

«Мобильное приложение автосервиса позволяет клиентам записываться на обслуживание, просматривать историю заказов и получать информацию об услугах».

1. Определите задачи проекта, в которых может использоваться ИИ.

Пример:

«Пользователь описывает проблему автомобиля обычным текстом, а ИИ определяет возможную категорию неисправности и рекомендуемую услугу».

Другие варианты:
генерация описаний товаров;
чат-консультант;
анализ документов;
распознавание изображений;
генерация изображений;
анализ отзывов;
рекомендации пользователю.

1. Определите, какой тип ИИ нужен для каждой задачи.

Пример:

Текстовый запрос клиента → текстовая ИИ-модель.

Фотография повреждения автомобиля → модель анализа изображений.

Генерация рекламного баннера → модель генерации изображений.

Отзывы клиентов → текстовая модель для анализа и классификации.

2. Подберите конкретные инструменты.

Например:

ChatGPT / Claude / Gemini — работа с текстом и документами.

Модели компьютерного зрения — анализ изображений.

Генераторы изображений — создание иллюстраций и рекламных материалов.

API ИИ-модели — интеграция ИИ непосредственно в приложение.

3. Составьте таблицу выбора.

Пример:

Задача | Тип ИИ | Инструмент | Входные данные | Результат

Анализ проблемы автомобиля | Текстовый ИИ | AI API | Описание проблемы | Возможная неисправность

Анализ фотографии | Компьютерное зрение | Vision API | Фотография | Описание повреждения

Создание баннера | Генерация изображений | Image AI | Текстовый запрос | Изображение

Анализ отзывов | Текстовый ИИ | AI API | Текст отзывов | Категория и оценка

4. Сравните несколько инструментов для одной задачи.

Например, для анализа текста сравните два доступных ИИ-сервиса по следующим параметрам:

качество ответа;
скорость;
удобство использования;
возможность работы через API;
формат получаемого результата.

4. Выберите инструменты для собственного проекта.

Для каждой выбранной технологии укажите, какую конкретную функцию она будет выполнять.

Пример:

«Для проекта будет использоваться текстовая ИИ-модель через API. Она будет получать описание проблемы автомобиля и возвращать категорию неисправности, возможные причины и рекомендуемую услугу».

- Определите способ взаимодействия приложения с ИИ.

Пример:

Мобильное приложение → сервер → AI API → результат ИИ → сервер → мобильное приложение.

- Определите возможные ограничения.

Укажите, какие проблемы могут возникнуть: стоимость API, необходимость интернет-соединения, ограничение размера запросов, ошибки модели, некорректные ответы или необходимость проверки результата.

- Подготовьте итоговую схему AI-функций проекта.

Пример:

Пользователь → описание проблемы → сервер → ИИ → анализ → результат → пользователь.

4. Результат: Студент должен представить описание проекта, перечень задач для применения ИИ, таблицу выбранных инструментов, сравнение вариантов, схему взаимодействия приложения с ИИ и обоснование выбора инструментов для собственного проекта.

Критерии оценки.

«5» «отлично»

- студент продемонстрировал готовность и способность к самостоятельному выполнению видов работ в стандартных ситуациях;
- все работы выполнил правильно и уверенно;
- проявил активность в освоении новых практических умений;
- содержание отчета по практическим работам полноценно отражает информацию по выполненным видам работ, установленных программой практики;
- оформление отчета по практическим работам полностью соответствует предъявляемым требованиям;
- при выполнении практических работ студент уверенно продемонстрировал обладание необходимыми компонентами общих и профессиональных компетенций;

«4» «хорошо»

- студент продемонстрировал готовность и способность к самостоятельному выполнению видов работ в стандартных ситуациях;
- все работы выполнил правильно и уверенно;
- содержание отчета по практическим работам в целом правильно отражает информацию по выполненным видам работ;
- оформление отчета по практическим работам соответствует предъявляемым требованиям, однако имеются замечания (некоторая неаккуратность, неполное описание проделанной работы и освоенных умений);
- при выполнении практических работ студент продемонстрировал обладание компонентами общих и профессиональных компетенций;
- характеристики студента положительные, имеются отдельные замечания и рекомендации;

«3» «удовлетворительно»

- при выполнении видов работ студент нуждался во внешнем сопровождении и контроле;
- в отчете по практическим работам содержится описание видов выполняемых работ согласно программе дисциплины, однако, записи неполные и неточные;
- отчет по практическим работам оформлен с отклонениями от установленных требований;
- при выполнении практических работ студент продемонстрировал обладание отдельными компонентами общих и профессиональных компетенций;

«**2» «неудовлетворительно»**

– студент не справился с запланированными видами работ;

– при выполнении практических работ студент продемонстрировал слабое обладание отдельными компонентами общих и профессиональных компетенций;

– отчет студентом не подготовлен.

Практическое занятие №8 Разработка сценария использования ИИ в проекте

Проверяемые результаты обучения:

- ПК 3.1 Осуществлять выбор готовых моделей искусственного интеллекта
- ПК 3.2 Формировать сценарии обучения готовых моделей искусственного интеллекта
- ПК 3.3 Проводить обучение и последующую калибровку готовых моделей искусственного интеллекта
- ПК 3.4 Контролировать результат обучения
- ПК 3.5 Оформлять результат проведения процедуры обучения
- ПК 3.6 Формировать запросы для работы с искусственным интеллектом с целью визуализации данных
- ПК 3.7 Формировать сценарии обучения готовых моделей искусственного интеллекта

- ПК 3.8 Интегрировать готовые модели искусственного интеллекта в информационные системы с использованием программных интерфейсов
- ПК 3.9 Создавать контент программного продукта средствами нейросетей
- ПК 3.10 Разрабатывать блоки программного кода с использованием нейросетей
- ПК 3.11 Создавать и управлять концепцией ИТ-продукта
- ПК 3.12 Планировать управление требованиями заказчика в проектах любого уровня сложности в области ИТ

1. Цель практической работы

Научиться разрабатывать подробный сценарий использования ИИ в собственном проекте: определить действия пользователя, входные данные, обработку информации и результат работы ИИ.

2. Задачи практической работы

Выбрать одну AI-функцию собственного проекта и разработать её полный сценарий работы от действия пользователя до получения результата.

3. Этапы выполнения

1. Выберите функцию, в которой будет использоваться ИИ.

Пример для приложения автосервиса:

«Определение возможной проблемы автомобиля по описанию пользователя».

2. Опишите действие пользователя.

Пример:

«Пользователь открывает раздел “Диагностика”, описывает проблему автомобиля и отправляет запрос».

3. Определите входные данные.

Пример:

Марка: Toyota Corolla
Год: 2018
Проблема: «При запуске двигателя слышен посторонний шум, иногда двигатель работает нестабильно».

4. Определите, что должен сделать ИИ.

Пример:

«ИИ анализирует описание проблемы, определяет возможные категории неисправностей, указывает вероятные причины и рекомендует обратиться на диагностику».

5. Определите формат результата.

Например:

Категория: двигатель

Возможные причины:

- Проблема со свечами зажигания.
- Неисправность системы подачи топлива.
- Необходимость компьютерной диагностики.

Рекомендация: записаться на диагностику двигателя.

6. Составьте полный сценарий:

Пользователь → вводит данные → приложение отправляет запрос → сервер формирует запрос к ИИ → ИИ анализирует данные → возвращает результат → сервер обрабатывает ответ → приложение показывает результат пользователю.

5. Определите данные, которые необходимо передать ИИ.

Пример:

```
{
  "car": "Toyota Corolla",
  "year": 2018,
  "problem": "Двигатель работает нестабильно"
}
```

4. Определите возможные ошибки.

Например:

пользователь не указал описание проблемы;
данных недостаточно;
ИИ не смог определить проблему;
ошибка подключения к API;
ИИ вернул некорректный ответ.

Для каждой ошибки определите действие приложения.

2. Определите ограничения AI-функции.

Например:

«Результат ИИ является предварительной рекомендацией и не заменяет профессиональную диагностику автомобиля».

1. Представьте сценарий в виде схемы.

Пользователь
↓
Ввод данных
↓
Приложение
↓
Сервер
↓
AI API
↓
Обработка результата
↓
Приложение
↓
Результат пользователю

1. Разработайте аналогичный сценарий для собственного проекта.

Можно использовать функции: AI-консультант, анализ документов, рекомендации, генерация текста, анализ изображений, классификация данных или автоматическое создание контента

5. Результат: Студент должен представить описание AI-функции, действия пользователя, входные и выходные данные, последовательность обработки, пример запроса и ответа, обработку ошибок и схему взаимодействия компонентов проекта.

Критерии оценки.

«5» «отлично»

- студент продемонстрировал готовность и способность к самостоятельному выполнению видов работ в стандартных ситуациях;
- все работы выполнил правильно и уверенно;
- проявил активность в освоении новых практических умений;

- содержание отчета по практическим работам полноценно отражает информацию по выполненным видам работ, установленных программой практики;
- оформление отчета по практическим работам полностью соответствует предъявляемым требованиям;
- при выполнении практических работ студент уверенно продемонстрировал обладание необходимыми компонентами общих и профессиональных компетенций;

«4» «хорошо»

- студент продемонстрировал готовность и способность к самостоятельному выполнению видов работ в стандартных ситуациях;
- все работы выполнил правильно и уверенно;
- содержание отчета по практическим работам в целом правильно отражает информацию по выполненным видам работ;
- оформление отчета по практическим работам соответствует предъявляемым требованиям, однако имеются замечания (некоторая неаккуратность, неполное описание проделанной работы и освоенных умений);
- при выполнении практических работ студент продемонстрировал обладание компонентами общих и профессиональных компетенций;
- характеристики студента положительные, имеются отдельные замечания и рекомендации;

«3» «удовлетворительно»

- при выполнении видов работ студент нуждался во внешнем сопровождении и контроле;
- в отчете по практическим работам содержится описание видов выполняемых работ согласно программе дисциплины, однако, записи неполные и неточные;
- отчет по практическим работам оформлен с отклонениями от установленных требований;
- при выполнении практических работ студент продемонстрировал обладание отдельными компонентами общих и профессиональных компетенций;

«2» «неудовлетворительно»

– студент не справился с запланированными видами работ;

– при выполнении практических работ студент продемонстрировал слабое обладание отдельными компонентами общих и профессиональных компетенций;

– отчет студентом не подготовлен.

Тема 1.2. Разработка промптов для искусственного интеллекта

Практическое занятие №9 Разработка и оптимизация промптов для проекта

Проверяемые результаты обучения:

- ПК 3.1 Осуществлять выбор готовых моделей искусственного интеллекта

- ПК 3.2 Формировать сценарии обучения готовых моделей искусственного интеллекта
- ПК 3.3 Проводить обучение и последующую калибровку готовых моделей искусственного интеллекта
- ПК 3.4 Контролировать результат обучения
- ПК 3.5 Оформлять результат проведения процедуры обучения
- ПК 3.6 Формировать запросы для работы с искусственным интеллектом с целью визуализации данных
- ПК 3.7 Формировать сценарии обучения готовых моделей искусственного интеллекта
- ПК 3.8 Интегрировать готовые модели искусственного интеллекта в информационные системы с использованием программных интерфейсов
- ПК 3.9 Создавать контент программного продукта средствами нейросетей
- ПК 3.10 Разрабатывать блоки программного кода с использованием нейросетей
- ПК 3.11 Создавать и управлять концепцией ИТ-продукта
- ПК 3.12 Планировать управление требованиями заказчика в проектах любого уровня сложности в области ИТ

2. Цель практической работы

Научиться создавать, тестировать и улучшать промпты для AI-функции собственного проекта, добиваясь точного, стабильного и структурированного результата.

3. Задачи практической работы

Выбрать AI-функцию проекта и разработать для неё несколько вариантов промпта. Сравнить результаты, выявить недостатки и создать итоговый оптимизированный промпт.

4. Этапы выполнения

1. Выберите AI-функцию проекта.

Пример для приложения автосервиса:

«Анализировать описание проблемы автомобиля и определять возможные причины неисправности».

1. Создайте простой промпт.

Пример:

«Проанализируй проблему автомобиля и скажи, что могло сломаться».

Проверьте полученный результат.

1. Добавьте контекст и роль.

Пример:

«Ты — виртуальный консультант автосервиса. Проанализируй описание проблемы автомобиля и определи возможные причины неисправности. Используй только информацию из запроса пользователя».

Сравните результат с первым вариантом.

1. Добавьте конкретные требования к ответу.

Пример:

«Ответь в следующем формате:
Категория проблемы:
Возможные причины:
Что проверить:
Рекомендация:»

1. Добавьте ограничения.

Например:

«Не утверждай, что неисправность точно установлена. Если информации недостаточно, укажи, какие дополнительные данные необходимо получить».

1. Добавьте пример ожидаемого результата.

Пример:

Запрос пользователя:
«При торможении автомобиль издаёт скрип».

Ожидаемый ответ:

«Категория: тормозная система.
Возможные причины: износ колодок, загрязнение тормозного механизма.
Что проверить: состояние колодок и дисков.
Рекомендация: провести диагностику тормозной системы».

4. Проверьте промпт на нескольких разных запросах.

Например:

«Двигатель плохо запускается утром».

«При движении появился сильный шум».

«Автомобиль тянет вправо».

Проверьте, сохраняется ли заданный формат ответа.

- Найдите недостатки.

Проверьте:
точность ответа;
соблюдение формата;
наличие лишней информации;
стабильность ответов;
обработку недостатка данных.

5. Оптимизируйте промпт.

Итоговый вариант может выглядеть так:

«Ты — виртуальный консультант автосервиса. Проанализируй описание проблемы автомобиля.

Определи наиболее вероятную категорию проблемы и возможные причины. Не утверждай, что причина установлена точно. Если информации недостаточно, укажи, какие данные необходимо получить дополнительно.

Ответ верни строго в формате JSON:

{
"category": "",
"possible_causes": [],
"additional_data": [],
"recommendation": ""
}»

3. Сравните первый и итоговый промпты. Определите, какие изменения улучшили результат.
4. Подготовьте аналогичный оптимизированный промпт для AI-функции собственного проекта

5. Результат: Студент должен представить исходный и улучшенный промпты, результаты их тестирования минимум на 3 запросах, анализ полученных результатов и итоговый оптимизированный промпт для собственного проекта.

Критерии оценки.

«5» «отлично»

- студент продемонстрировал готовность и способность к самостоятельному выполнению видов работ в стандартных ситуациях;
- все работы выполнил правильно и уверенно;
- проявил активность в освоении новых практических умений;
- содержание отчета по практическим работам полноценно отражает информацию по выполненным видам работ, установленных программой практики;
- оформление отчета по практическим работам полностью соответствует предъявляемым требованиям;
- при выполнении практических работ студент уверенно продемонстрировал обладание необходимыми компонентами общих и профессиональных компетенций;

«4» «хорошо»

- студент продемонстрировал готовность и способность к самостоятельному выполнению видов работ в стандартных ситуациях;
- все работы выполнил правильно и уверенно;
- содержание отчета по практическим работам в целом правильно отражает информацию по выполненным видам работ;
- оформление отчета по практическим работам соответствует предъявляемым требованиям, однако имеются замечания (некоторая неаккуратность, неполное описание проделанной работы и освоенных умений);
- при выполнении практических работ студент продемонстрировал обладание компонентами общих и профессиональных компетенций;
- характеристики студента положительные, имеются отдельные замечания и рекомендации;

«3» «удовлетворительно»

- при выполнении видов работ студент нуждался во внешнем сопровождении и контроле;
- в отчете по практическим работам содержится описание видов выполняемых работ согласно программе дисциплины, однако, записи неполные и неточные;
- отчет по практическим работам оформлен с отклонениями от установленных требований;
- при выполнении практических работ студент продемонстрировал обладание отдельными компонентами общих и профессиональных компетенций;

«**2» «неудовлетворительно»**

– студент не справился с запланированными видами работ;

– при выполнении практических работ студент продемонстрировал слабое обладание отдельными компонентами общих и профессиональных компетенций;
– отчет студентом не подготовлен.

Практическое занятие №10 Структурированный ответ ИИ и JSON

Проверяемые результаты обучения:

- ПК 3.1 Осуществлять выбор готовых моделей искусственного интеллекта
- ПК 3.2 Формировать сценарии обучения готовых моделей искусственного интеллекта
- ПК 3.3 Проводить обучение и последующую калибровку готовых моделей искусственного интеллекта
- ПК 3.4 Контролировать результат обучения
- ПК 3.5 Оформлять результат проведения процедуры обучения
- ПК 3.6 Формировать запросы для работы с искусственным интеллектом с целью визуализации данных
- ПК 3.7 Формировать сценарии обучения готовых моделей искусственного интеллекта
- ПК 3.8 Интегрировать готовые модели искусственного интеллекта в информационные системы с использованием программных интерфейсов
- ПК 3.9 Создавать контент программного продукта средствами нейросетей
- ПК 3.10 Разрабатывать блоки программного кода с использованием нейросетей
- ПК 3.11 Создавать и управлять концепцией ИТ-продукта
- ПК 3.12 Планировать управление требованиями заказчика в проектах любого уровня сложности в области ИТ

2. Цель практической работы

Научиться получать от ИИ предсказуемые структурированные ответы и использовать формат JSON для дальнейшей обработки данных в программном приложении.

3. Задачи практической работы

Разработать несколько запросов, в которых ИИ должен возвращать информацию в заданной структуре. Научиться задавать поля JSON, проверять корректность полученного результата и подготавливать его для использования в проекте.

4. Этапы выполнения

1. Получите обычный ответ ИИ.

Пример запроса:

«Проанализируй проблему автомобиля: двигатель плохо запускается и работает нестабильно».

Обратите внимание, что формат ответа может отличаться при каждом запросе.

- Задайте структуру ответа.

Пример:

«Ответь в формате:
Проблема:
Возможные причины:
Рекомендация:»

Сравните результат с предыдущим ответом.

1. Попросите ИИ вернуть результат в формате JSON.

Пример запроса:

«Проанализируй проблему автомобиля и верни результат только в формате JSON».

Пример ответа:

```
{
  "problem": "Двигатель плохо запускается",
  "possible_causes": [
    "Проблемы со свечами зажигания",
    "Проблемы с аккумулятором"
  ],
  "recommendation": "Провести диагностику двигателя"
}
```

- Определите обязательные поля.

Например:

```
{
  "category": "",
  "problem": "",
  "possible_causes": [],
```

```
  "recommendation": ""
}
```

Попросите ИИ всегда возвращать именно эти поля.

2. Добавьте дополнительные данные.

Например:

```
{
  "car": {
    "brand": "Toyota",
    "model": "Corolla",
    "year": 2018
  },
  "problem": "Двигатель работает нестабильно",
  "category": "Двигатель",
  "possible_causes": [
    "Свечи зажигания",
    "Топливная система"
  ],
  "recommendation": "Провести диагностику"
}
```

1. Проверьте JSON.

Скопируйте полученный ответ и проверьте:
правильность скобок;
кавычки;
запятые;
названия полей;
соответствие типов данных.

- Используйте JSON в программе.

Например, JavaScript:

```
const response = `{
  "category": "Двигатель",
  "recommendation": "Провести диагностику"
}`;

const data = JSON.parse(response);

console.log(data.category);
```

```
console.log(data.recommendation);
```

Результат:

```
Двигатель
Провести диагностику
```

1. Проверьте обработку нескольких объектов.

Пример:

```
{
  "problems": [
    {
      "category": "Двигатель",
      "priority": "Высокая"
    },
    {
      "category": "Тормозная система",
      "priority": "Средняя"
    }
  ]
}
```

Попросите ИИ сформировать такой ответ для нескольких обращений клиентов.

1. Проверьте некорректные и неполные данные.

Например:

«Проанализируй проблему: автомобиль плохо работает».

Попросите ИИ вернуть JSON и указать, какой информации недостаточно.

1. Подготовьте JSON для AI-функции собственного проекта.

Определите, какие данные приложение будет отправлять ИИ и какие данные получать обратно.

Пример:

Приложение → описание проблемы → ИИ → JSON → приложение → отображение результата

5. Результат: Студент должен представить несколько примеров структурированных ответов, корректный JSON, пример его обработки в коде, обработку неполных данных и структуру JSON для AI-функции собственного проекта

Критерии оценки.

«5» «отлично»

- студент продемонстрировал готовность и способность к самостоятельному выполнению видов работ в стандартных ситуациях;
- все работы выполнил правильно и уверенно;
- проявил активность в освоении новых практических умений;
- содержание отчета по практическим работам полноценно отражает информацию по выполненным видам работ, установленных программой практики;
- оформление отчета по практическим работам полностью соответствует предъявляемым требованиям;
- при выполнении практических работ студент уверенно продемонстрировал обладание необходимыми компонентами общих и профессиональных компетенций;

«4» «хорошо»

- студент продемонстрировал готовность и способность к самостоятельному выполнению видов работ в стандартных ситуациях;
- все работы выполнил правильно и уверенно;
- содержание отчета по практическим работам в целом правильно отражает информацию по выполненным видам работ;
- оформление отчета по практическим работам соответствует предъявляемым требованиям, однако имеются замечания (некоторая неаккуратность, неполное описание проделанной работы и освоенных умений);
- при выполнении практических работ студент продемонстрировал обладание компонентами общих и профессиональных компетенций;
- характеристики студента положительные, имеются отдельные замечания и рекомендации;

«3» «удовлетворительно»

- при выполнении видов работ студент нуждался во внешнем сопровождении и контроле;
- в отчете по практическим работам содержится описание видов выполняемых работ согласно программе дисциплины, однако, записи неполные и неточные;
- отчет по практическим работам оформлен с отклонениями от установленных требований;
- при выполнении практических работ студент продемонстрировал обладание отдельными компонентами общих и профессиональных компетенций;

«**2» «неудовлетворительно»**

– студент не справился с запланированными видами работ;

– при выполнении практических работ студент продемонстрировал слабое обладание отдельными компонентами общих и профессиональных компетенций;
– отчет студентом не подготовлен.

Практическое занятие №11 Цепочки обработки с использованием ИИ

Проверяемые результаты обучения:

- ПК 3.1 Осуществлять выбор готовых моделей искусственного интеллекта
- ПК 3.2 Формировать сценарии обучения готовых моделей искусственного интеллекта
- ПК 3.3 Проводить обучение и последующую калибровку готовых моделей искусственного интеллекта
- ПК 3.4 Контролировать результат обучения
- ПК 3.5 Оформлять результат проведения процедуры обучения
- ПК 3.6 Формировать запросы для работы с искусственным интеллектом с целью визуализации данных
- ПК 3.7 Формировать сценарии обучения готовых моделей искусственного интеллекта
- ПК 3.8 Интегрировать готовые модели искусственного интеллекта в информационные системы с использованием программных интерфейсов
- ПК 3.9 Создавать контент программного продукта средствами нейросетей
- ПК 3.10 Разрабатывать блоки программного кода с использованием нейросетей
- ПК 3.11 Создавать и управлять концепцией ИТ-продукта
- ПК 3.12 Планировать управление требованиями заказчика в проектах любого уровня сложности в области ИТ

2. Цель практической работы

Научиться создавать цепочки последовательных операций, в которых результат одного этапа используется на следующем этапе обработки с помощью ИИ.

3. Задачи практической работы

Разработать несколько цепочек обработки данных с использованием ИИ и создать одну цепочку для собственного проекта.

4. Этапы выполнения

1. Познакомьтесь с принципом цепочки.

Пример:

Текст пользователя → классификация → анализ → генерация ответа → проверка → итоговый результат.

1. Создайте простую цепочку обработки текста.

Исходный текст:

«Клиент пишет, что после последнего обновления приложения не может войти в аккаунт».

Первый запрос:

«Определи категорию обращения».

Результат:

«Проблема с авторизацией».

Второй запрос:

«Проанализируй обращение и определи возможные причины проблемы».

Третий запрос:

«Составь ответ клиенту на основе проведённого анализа».

- Объедините результаты этапов.

Схема:

Текст обращения
↓
Определение категории
↓
Поиск возможных причин
↓
Формирование ответа
↓
Проверка ответа
↓
Ответ клиенту

1. Создайте цепочку для обработки программного кода.

Например:

Код → поиск ошибки → объяснение ошибки → исправление → создание теста.

Исходный код:

```
numbers = [10, 20, 30, 40]

average = sum(numbers) / len(numbers)

print("Среднее значение:", average)
```

Попросите ИИ сначала проверить код, затем объяснить его работу и создать несколько тестовых данных.

1. Создайте цепочку обработки документа.

Пример:

Документ → извлечение информации → классификация → структурирование → итоговый отчёт.

Например, из технического задания сначала извлечь требования, затем разделить их на функциональные и нефункциональные, а после этого представить результат в таблице.

1. Добавьте проверку результата.

Пример:

Данные → ИИ анализирует → второй запрос проверяет результат → исправление ошибок → итоговый ответ.

Попросите ИИ:

«Проверь предыдущий результат. Найди фактические ошибки, пропущенные данные и несоответствия заданным требованиям».

1. Создайте цепочку с использованием JSON.

Пример:

```
{
  "text": "Клиент не может войти в приложение",
```

```
  "category": "Авторизация",
  "priority": "Высокая"
}
```

Следующий этап получает этот JSON и формирует ответ пользователю.

1. Разработайте цепочку для собственного проекта.

Например, для приложения автосервиса:

Описание проблемы
↓
Определение категории
↓
Определение возможных причин
↓
Формирование рекомендации
↓
Проверка результата
↓
Ответ пользователю

1. Опишите каждый этап цепочки.

Для каждого этапа укажите:
входные данные;
запрос к ИИ;
результат;
данные, передаваемые на следующий этап.

1. Проверьте работу цепочки на нескольких примерах. Определите, на каком этапе могут возникнуть ошибки и что произойдёт при некорректном результате

5. Результат: Студент должен представить минимум две цепочки обработки, описание каждого этапа, примеры запросов и ответов, одну цепочку для собственного проекта и схему её работы

Критерии оценки.

«5» «отлично»

- студент продемонстрировал готовность и способность к самостоятельному выполнению видов работ в стандартных ситуациях;
- все работы выполнил правильно и уверенно;
- проявил активность в освоении новых практических умений;

- содержание отчета по практическим работам полноценно отражает информацию по выполненным видам работ, установленных программой практики;
- оформление отчета по практическим работам полностью соответствует предъявляемым требованиям;
- при выполнении практических работ студент уверенно продемонстрировал обладание необходимыми компонентами общих и профессиональных компетенций;

«4» «хорошо»

- студент продемонстрировал готовность и способность к самостоятельному выполнению видов работ в стандартных ситуациях;
- все работы выполнил правильно и уверенно;
- содержание отчета по практическим работам в целом правильно отражает информацию по выполненным видам работ;
- оформление отчета по практическим работам соответствует предъявляемым требованиям, однако имеются замечания (некоторая неаккуратность, неполное описание проделанной работы и освоенных умений);
- при выполнении практических работ студент продемонстрировал обладание компонентами общих и профессиональных компетенций;
- характеристики студента положительные, имеются отдельные замечания и рекомендации;

«3» «удовлетворительно»

- при выполнении видов работ студент нуждался во внешнем сопровождении и контроле;
- в отчете по практическим работам содержится описание видов выполняемых работ согласно программе дисциплины, однако, записи неполные и неточные;
- отчет по практическим работам оформлен с отклонениями от установленных требований;
- при выполнении практических работ студент продемонстрировал обладание отдельными компонентами общих и профессиональных компетенций;

«2» «неудовлетворительно»

– студент не справился с запланированными видами работ;

– при выполнении практических работ студент продемонстрировал слабое обладание отдельными компонентами общих и профессиональных компетенций;

– отчет студентом не подготовлен.

Тема 1.3. Интеграция искусственного интеллекта в информационные системы

Практическое занятие №12 Подключение ИИ через API

Проверяемые результаты обучения:

- ПК 3.1 Осуществлять выбор готовых моделей искусственного интеллекта
- ПК 3.2 Формировать сценарии обучения готовых моделей искусственного интеллекта
- ПК 3.3 Проводить обучение и последующую калибровку готовых моделей искусственного интеллекта
- ПК 3.4 Контролировать результат обучения
- ПК 3.5 Оформлять результат проведения процедуры обучения
- ПК 3.6 Формировать запросы для работы с искусственным интеллектом с целью визуализации данных
- ПК 3.7 Формировать сценарии обучения готовых моделей искусственного интеллекта
- ПК 3.8 Интегрировать готовые модели искусственного интеллекта в информационные системы с использованием программных интерфейсов
- ПК 3.9 Создавать контент программного продукта средствами нейросетей
- ПК 3.10 Разрабатывать блоки программного кода с использованием нейросетей
- ПК 3.11 Создавать и управлять концепцией ИТ-продукта
- ПК 3.12 Планировать управление требованиями заказчика в проектах любого уровня сложности в области ИТ

2. Цель практической работы

Научиться подключать готовую ИИ-модель через API, отправлять запросы из программы, получать ответ и обрабатывать его в приложении.

3. Задачи практической работы

Создать небольшое приложение, которое отправляет запрос к ИИ через API, получает ответ и выводит его пользователю.

4. Этапы выполнения

- Познакомьтесь со схемой взаимодействия.

Приложение → HTTP-запрос → AI API → ответ JSON → приложение → результат пользователю.

2. Получите API-ключ выбранного ИИ-сервиса.

API-ключ нельзя размещать непосредственно в исходном коде или передавать другим пользователям.

3. Создайте простой Python-проект.

Установите библиотеку:

```
pip install requests
```

Создайте переменную окружения для ключа API:

```
import os

API_KEY = os.getenv("AI_API_KEY")
```

4. Подготовьте HTTP-запрос.

Общий пример:

```
import os
import requests

API_KEY = os.getenv("AI_API_KEY")

url = "https://api.example.com/v1/chat"

headers = {
    "Authorization": f"Bearer {API_KEY}",
    "Content-Type": "application/json"
}

data = {
    "model": "your-model",
    "messages": [
        {
            "role": "user",
            "content": "Объясни, что такое API простыми словами"
        }
    ]
}

response = requests.post(
    url,
    headers=headers,
    json=data
)

print(response.json())
```

Примечание: адрес API, название модели и структура запроса зависят от выбранного сервиса. Используйте актуальную документацию выбранного API.

5. Получите текст ответа.

Если API возвращает JSON, сначала сохраните результат:

```
result = response.json()

print(result)
```

После изучения структуры ответа получите нужное поле:

```
answer = result["answer"]

print(answer)
```

Название поля необходимо заменить на фактическое поле, которое возвращает используемый API.

6. Сделайте программу интерактивной.

Например:

```
question = input("Введите вопрос: ")

data = {
    "model": "your-model",
    "messages": [
        {
            "role": "user",
            "content": question
        }
    ]
}

response = requests.post(
    url,
    headers=headers,
    json=data
)

print(response.json())
```

Теперь пользователь может самостоятельно вводить запросы.

7. Добавьте обработку ошибок.

```
try:
    response = requests.post(
        url,
        headers=headers,
        json=data,
        timeout=30
    )

    response.raise_for_status()

    result = response.json()
    print(result)

except requests.exceptions.Timeout:
    print("Время ожидания ответа истекло")

except requests.exceptions.RequestException as error:
    print("Ошибка API:", error)
```

8. Передайте ИИ данные из приложения.

Например:

```
problem = input("Опишите проблему: ")

prompt = f"""
Проанализируй проблему пользователя:
{problem}

Определи категорию проблемы и предложи возможное решение.
"""

data = {
    "model": "your-model",
    "messages": [
        {
            "role": "user",
            "content": prompt
        }
    ]
}
```

9. Реализуйте небольшую AI-функцию для собственного проекта.

Например, для приложения автосервиса:

Пользователь → описание неисправности → сервер → AI API → анализ → результат пользователю.

Пример результата:

```
{
  "category": "Двигатель",
  "recommendation": "Провести диагностику системы зажигания"
}
```

10. Проверьте работу программы на нескольких запросах и убедитесь, что приложение корректно получает и отображает ответы

5. Результат: Студент должен представить работающую программу с подключением к AI API, пример HTTP-запроса, обработку JSON-ответа, обработку ошибок и одну AI-функцию, связанную с собственным проектом

Критерии оценки.

«5» «отлично»

- студент продемонстрировал готовность и способность к самостоятельному выполнению видов работ в стандартных ситуациях;
- все работы выполнил правильно и уверенно;
- проявил активность в освоении новых практических умений;
- содержание отчета по практическим работам полноценно отражает информацию по выполненным видам работ, установленных программой практики;
- оформление отчета по практическим работам полностью соответствует предъявляемым требованиям;
- при выполнении практических работ студент уверенно продемонстрировал обладание необходимыми компонентами общих и профессиональных компетенций;

«4» «хорошо»

- студент продемонстрировал готовность и способность к самостоятельному выполнению видов работ в стандартных ситуациях;
- все работы выполнил правильно и уверенно;
- содержание отчета по практическим работам в целом правильно отражает информацию по выполненным видам работ;
- оформление отчета по практическим работам соответствует предъявляемым требованиям, однако имеются замечания (некоторая неаккуратность, неполное описание проделанной работы и освоенных умений);

- при выполнении практических работ студент продемонстрировал обладание компонентами общих и профессиональных компетенций;
- характеристики студента положительные, имеются отдельные замечания и рекомендации;

«3» «удовлетворительно»

- при выполнении видов работ студент нуждался во внешнем сопровождении и контроле;
- в отчете по практическим работам содержится описание видов выполняемых работ согласно программе дисциплины, однако, записи неполные и неточные;
- отчет по практическим работам оформлен с отклонениями от установленных требований;
- при выполнении практических работ студент продемонстрировал обладание отдельными компонентами общих и профессиональных компетенций;

«**2» «неудовлетворительно»**

– студент не справился с запланированными видами работ;
– при выполнении практических работ студент продемонстрировал слабое обладание отдельными компонентами общих и профессиональных компетенций;
– отчет студентом не подготовлен.

Практическое занятие №13 Разработка AI-модуля приложения

Проверяемые результаты обучения:

- ПК 3.1 Осуществлять выбор готовых моделей искусственного интеллекта
- ПК 3.2 Формировать сценарии обучения готовых моделей искусственного интеллекта
- ПК 3.3 Проводить обучение и последующую калибровку готовых моделей искусственного интеллекта
- ПК 3.4 Контролировать результат обучения
- ПК 3.5 Оформлять результат проведения процедуры обучения
- ПК 3.6 Формировать запросы для работы с искусственным интеллектом с целью визуализации данных
- ПК 3.7 Формировать сценарии обучения готовых моделей искусственного интеллекта
- ПК 3.8 Интегрировать готовые модели искусственного интеллекта в информационные системы с использованием программных интерфейсов
- ПК 3.9 Создавать контент программного продукта средствами нейросетей
- ПК 3.10 Разрабатывать блоки программного кода с использованием нейросетей
- ПК 3.11 Создавать и управлять концепцией ИТ-продукта
- ПК 3.12 Планировать управление требованиями заказчика в проектах любого уровня сложности в области ИТ

2. Цель практической работы

Научиться выделять AI-функцию в отдельный программный модуль, организовывать его взаимодействие с API и подключать его к приложению.

3. Задачи практической работы

Разработать отдельный AI-модуль, который получает данные от приложения, отправляет запрос к ИИ и возвращает обработанный результат.

4. Этапы выполнения

1. Определите назначение AI-модуля.

Пример для приложения автосервиса:

«Модуль получает описание проблемы автомобиля и возвращает категорию проблемы, возможные причины и рекомендацию».

Схема работы:

Приложение → AI-модуль → AI API → AI-модуль → приложение.

2. Определите входные и выходные данные.

Вход:

```
{
  "car": "Toyota Corolla",
  "year": 2018,
  "problem": "Двигатель работает нестабильно"
}
```

Выход:

```
{
  "category": "Двигатель",
  "possible_causes": [
    "Свечи зажигания",
    "Топливная система"
  ],
  "recommendation": "Провести диагностику"
```

```
}
```

3. Создайте отдельный файл для AI-модуля.

Например:

```
project/
├── main.py
├── ai_service.py
└── config.py
```

4. Создайте AI-сервис.

Пример:

```
# ai_service.py

import os
import requests

API_KEY = os.getenv("AI_API_KEY")

def ask_ai(prompt):
    url = "https://api.example.com/v1/chat"

    headers = {
        "Authorization": f"Bearer {API_KEY}",
        "Content-Type": "application/json"
    }

    data = {
        "model": "your-model",
        "messages": [
            {
                "role": "user",
                "content": prompt
            }
        ]
    }

    response = requests.post(
        url,
        headers=headers,
        json=data,
        timeout=30
```

```
    )

    response.raise_for_status()

    return response.json()
```

Адрес API и структура запроса зависят от выбранного AI-сервиса.

5. Создайте функцию для конкретной задачи.

```
def analyze_car_problem(car, year, problem):

    prompt = f"""
    Ты — консультант автосервиса.

    Автомобиль: {car}
    Год выпуска: {year}
    Проблема: {problem}

    Определи категорию проблемы,
    возможные причины и рекомендации.
    """

    return ask_ai(prompt)
```

6. Подключите AI-модуль к основной программе.

```
from ai_service import analyze_car_problem

result = analyze_car_problem(
    "Toyota Corolla",
    2018,
    "Двигатель работает нестабильно"
)

print(result)
```

Теперь основная программа не содержит непосредственно код работы с API.

7. Добавьте обработку ошибок.

```
def analyze_car_problem(car, year, problem):

    try:
        prompt = f"""
```

```
        Автомобиль: {car}
        Год: {year}
        Проблема: {problem}

        Проанализируй проблему.
        """

        return ask_ai(prompt)

    except Exception as error:
        return {
            "error": "Не удалось получить ответ ИИ",
            "details": str(error)
        }
```

8. Добавьте структурированный результат.

Попросите модель возвращать JSON:

```
prompt = f"""
Проанализируй проблему автомобиля.

Верни результат строго в JSON:
{{
   "category": "",
   "possible_causes": [],
   "recommendation": ""
}}

Автомобиль: {car}
Год: {year}
Проблема: {problem}
"""
```

9. Подключите модуль к пользовательскому интерфейсу.

Схема:

```
Пользователь
    ↓
Форма приложения
    ↓
AI-модуль
    ↓
AI API
```

↓
Обработка JSON
↓
Интерфейс приложения

Например:

```
problem = input("Опишите проблему автомобиля: ")

result = analyze_car_problem(
   "Toyota Corolla",
   2018,
   problem
)

print(result)
```

10.Протестируйте AI-модуль.

Проверьте:
обычный запрос;
пустой запрос;
слишком короткое описание;
ошибку API;
некорректный ответ;
несколько разных пользовательских запросов.

11.Адаптируйте AI-модуль под собственный проект.

Можно реализовать AI-консультанта, анализатор документов, генератор описаний, систему рекомендаций, анализ отзывов или другую функцию

5. Результат: Студент должен представить отдельный AI-модуль, его подключение к приложению, код взаимодействия с API, структуру входных и выходных данных, обработку ошибок и работающую AI-функцию собственного проекта

Критерии оценки.

«5» «отлично»

- студент продемонстрировал готовность и способность к самостоятельному выполнению видов работ в стандартных ситуациях;
- все работы выполнил правильно и уверенно;
- проявил активность в освоении новых практических умений;

- содержание отчета по практическим работам полноценно отражает информацию по выполненным видам работ, установленных программой практики;
- оформление отчета по практическим работам полностью соответствует предъявляемым требованиям;
- при выполнении практических работ студент уверенно продемонстрировал обладание необходимыми компонентами общих и профессиональных компетенций;

«4» «хорошо»

- студент продемонстрировал готовность и способность к самостоятельному выполнению видов работ в стандартных ситуациях;
- все работы выполнил правильно и уверенно;
- содержание отчета по практическим работам в целом правильно отражает информацию по выполненным видам работ;
- оформление отчета по практическим работам соответствует предъявляемым требованиям, однако имеются замечания (некоторая неаккуратность, неполное описание проделанной работы и освоенных умений);
- при выполнении практических работ студент продемонстрировал обладание компонентами общих и профессиональных компетенций;
- характеристики студента положительные, имеются отдельные замечания и рекомендации;

«3» «удовлетворительно»

- при выполнении видов работ студент нуждался во внешнем сопровождении и контроле;
- в отчете по практическим работам содержится описание видов выполняемых работ согласно программе дисциплины, однако, записи неполные и неточные;
- отчет по практическим работам оформлен с отклонениями от установленных требований;
- при выполнении практических работ студент продемонстрировал обладание отдельными компонентами общих и профессиональных компетенций;

«**2» «неудовлетворительно»**

– студент не справился с запланированными видами работ;

– при выполнении практических работ студент продемонстрировал слабое обладание отдельными компонентами общих и профессиональных компетенций;

– отчет студентом не подготовлен.

Практическое занятие №14 Интеграция ИИ в информационную систему

Проверяемые результаты обучения:

- ПК 3.1 Осуществлять выбор готовых моделей искусственного интеллекта
- ПК 3.2 Формировать сценарии обучения готовых моделей искусственного интеллекта

- ПК 3.3 Проводить обучение и последующую калибровку готовых моделей искусственного интеллекта
- ПК 3.4 Контролировать результат обучения
- ПК 3.5 Оформлять результат проведения процедуры обучения
- ПК 3.6 Формировать запросы для работы с искусственным интеллектом с целью визуализации данных
- ПК 3.7 Формировать сценарии обучения готовых моделей искусственного интеллекта
- ПК 3.8 Интегрировать готовые модели искусственного интеллекта в информационные системы с использованием программных интерфейсов
- ПК 3.9 Создавать контент программного продукта средствами нейросетей
- ПК 3.10 Разрабатывать блоки программного кода с использованием нейросетей
- ПК 3.11 Создавать и управлять концепцией ИТ-продукта
- ПК 3.12 Планировать управление требованиями заказчика в проектах любого уровня сложности в области ИТ

2. Цель практической работы

Научиться интегрировать AI-модуль в существующую информационную систему и организовать обмен данными между интерфейсом, сервером, AI-сервисом и пользователем.

3. Задачи практической работы

Добавить AI-функцию в существующее приложение. Организовать передачу данных от пользователя на сервер, отправку запроса в ИИ и возврат результата обратно в интерфейс.

4. Этапы выполнения

1. Определите AI-функцию информационной системы.

Пример для системы автосервиса:

«Пользователь вводит описание проблемы автомобиля, а система с помощью ИИ формирует предварительную рекомендацию».

Схема:

Пользователь → интерфейс → сервер → AI API → сервер → интерфейс.

2. Определите API-метод для AI-функции.

Например:

POST /api/ai/diagnose

Запрос:

```
{
  "car": "Toyota Corolla",
  "year": 2018,
  "problem": "Двигатель работает нестабильно"
}
```

3. Создайте серверный маршрут.

Пример на Node.js + Express:

```
app.post("/api/ai/diagnose", async (req, res) => {
    const { car, year, problem } = req.body;

    try {
        const result = await analyzeCarProblem(
            car,
            year,
            problem
        );

        res.json(result);
    } catch (error) {
        res.status(500).json({
            error: "Ошибка AI-сервиса"
        });
    }
});
```

4. Реализуйте функцию обращения к ИИ.

```
async function analyzeCarProblem(car, year, problem) {
    const prompt = `
        Ты консультант автосервиса.

        Автомобиль: ${car}
        Год выпуска: ${year}
        Проблема: ${problem}

        Определи возможные причины
```

```
        и предложи рекомендацию.
    `;

    // Здесь выполняется запрос к выбранному AI API
    const response = await callAI(prompt);

    return response;
}
```

5. Подключите AI-функцию к интерфейсу.

Пример на JavaScript:

```
async function diagnoseCar() {
    const problem = document.querySelector("#problem").value;

    const response = await fetch("/api/ai/diagnose", {
        method: "POST",
        headers: {
            "Content-Type": "application/json"
        },
        body: JSON.stringify({
            car: "Toyota Corolla",
            year: 2018,
            problem: problem
        })
    });

    const result = await response.json();

    document.querySelector("#result").textContent =
        result.recommendation;
}
```

6. Добавьте кнопку в интерфейс.

```
<textarea id="problem"
          placeholder="Опишите проблему автомобиля">
</textarea>

<button onclick="diagnoseCar()">
    Получить рекомендацию
</button>
```

```
<div id="result"></div>
```

7. Организуйте обработку результата.

Например, AI возвращает:

```
{
  "category": "Двигатель",
  "possible_causes": [
    "Свечи зажигания",
    "Топливная система"
  ],
  "recommendation": "Провести диагностику двигателя"
}
```

Интерфейс может вывести:

Категория: Двигатель

Возможные причины:
• Свечи зажигания
• Топливная система

Рекомендация:
Провести диагностику двигателя.

8. Добавьте обработку ошибок.

```
try {
    const response = await fetch("/api/ai/diagnose", options);

    if (!response.ok) {
        throw new Error("Ошибка сервера");
    }

    const result = await response.json();

} catch (error) {
    document.querySelector("#result").textContent =
        "Не удалось получить рекомендацию";
}
```

9. Продумайте безопасность.

API-ключ ИИ не должен находиться во frontend-коде. Запрос к AI API должен выполняться на сервере.

Неправильно:

```
const API_KEY = "secret-key";
```

Правильно:

```
Frontend
  ↓
Backend
  ↓
AI API
```

10.Протестируйте интеграцию.

Проверьте обычный запрос, пустые данные, некорректные данные, ошибку API и несколько различных пользовательских запросов.

11.Реализуйте аналогичную AI-функцию в собственном проекте.

Например:
AI-консультант;
анализ отзывов;
генерация описаний;
анализ документов;
рекомендации пользователю;
поиск информации;
анализ изображений.

5. Результат: Студент должен представить работающую AI-функцию внутри информационной системы, серверный API-метод, код взаимодействия frontend и backend, обработку результата и ошибок, а также схему взаимодействия компонентов системы

Критерии оценки.

«5» «отлично»

- студент продемонстрировал готовность и способность к самостоятельному выполнению видов работ в стандартных ситуациях;
- все работы выполнил правильно и уверенно;
- проявил активность в освоении новых практических умений;

- содержание отчета по практическим работам полноценно отражает информацию по выполненным видам работ, установленных программой практики;
- оформление отчета по практическим работам полностью соответствует предъявляемым требованиям;
- при выполнении практических работ студент уверенно продемонстрировал обладание необходимыми компонентами общих и профессиональных компетенций;

«4» «хорошо»

- студент продемонстрировал готовность и способность к самостоятельному выполнению видов работ в стандартных ситуациях;
- все работы выполнил правильно и уверенно;
- содержание отчета по практическим работам в целом правильно отражает информацию по выполненным видам работ;
- оформление отчета по практическим работам соответствует предъявляемым требованиям, однако имеются замечания (некоторая неаккуратность, неполное описание проделанной работы и освоенных умений);
- при выполнении практических работ студент продемонстрировал обладание компонентами общих и профессиональных компетенций;
- характеристики студента положительные, имеются отдельные замечания и рекомендации;

«3» «удовлетворительно»

- при выполнении видов работ студент нуждался во внешнем сопровождении и контроле;
- в отчете по практическим работам содержится описание видов выполняемых работ согласно программе дисциплины, однако, записи неполные и неточные;
- отчет по практическим работам оформлен с отклонениями от установленных требований;
- при выполнении практических работ студент продемонстрировал обладание отдельными компонентами общих и профессиональных компетенций;

«2» «неудовлетворительно»

– студент не справился с запланированными видами работ;

– при выполнении практических работ студент продемонстрировал слабое обладание отдельными компонентами общих и профессиональных компетенций;

– отчет студентом не подготовлен.

Практическое занятие №15 Обработка данных информационной системы с помощью ИИ

Проверяемые результаты обучения:

- ПК 3.1 Осуществлять выбор готовых моделей искусственного интеллекта

- ПК 3.2 Формировать сценарии обучения готовых моделей искусственного интеллекта
- ПК 3.3 Проводить обучение и последующую калибровку готовых моделей искусственного интеллекта
- ПК 3.4 Контролировать результат обучения
- ПК 3.5 Оформлять результат проведения процедуры обучения
- ПК 3.6 Формировать запросы для работы с искусственным интеллектом с целью визуализации данных
- ПК 3.7 Формировать сценарии обучения готовых моделей искусственного интеллекта
- ПК 3.8 Интегрировать готовые модели искусственного интеллекта в информационные системы с использованием программных интерфейсов
- ПК 3.9 Создавать контент программного продукта средствами нейросетей
- ПК 3.10 Разрабатывать блоки программного кода с использованием нейросетей
- ПК 3.11 Создавать и управлять концепцией ИТ-продукта
- ПК 3.12 Планировать управление требованиями заказчика в проектах любого уровня сложности в области ИТ

2. Цель практической работы

Научиться передавать данные информационной системы в ИИ, выполнять их анализ и получать результат, который можно использовать в приложении.

3. Задачи практической работы

Реализовать AI-функцию, которая получает данные информационной системы, обрабатывает их с помощью ИИ и возвращает структурированный результат.

4. Этапы выполнения

1. Определите данные, которые будут обрабатываться.

Пример для автосервиса:

```
{
  "client": "Иван",
  "car": "Toyota Corolla",
  "orders": [
    {
```

```
      "service": "Замена масла",
      "price": 2500
    },
    {
      "service": "Замена колодок",
      "price": 8500
    }
  ]
}
```

2. Определите задачу для ИИ.

Например:

«Проанализировать историю заказов клиента и сформировать краткое описание, основные виды обслуживания и рекомендации».

3. Получите данные из информационной системы.

Пример Node.js:

```
const [rows] = await db.query(
   "SELECT service, price FROM orders WHERE client_id = ?",
   [clientId]
);

console.log(rows);
```

4. Подготовьте данные для передачи ИИ.

```
const prompt = `
Проанализируй историю обслуживания клиента.

Данные:
${JSON.stringify(rows)}

Определи:
1. Основные выполненные работы.
2. Общую сумму расходов.
3. Наиболее частую услугу.
4. Краткую рекомендацию клиенту.
`;
```

5. Передайте данные в AI-модуль.

```
const result = await analyzeWithAI(prompt);

console.log(result);
```

6. Попросите ИИ вернуть структурированный результат.

Например:

```
{
  "total": 11000,
  "main_services": [
    "Замена масла",
    "Замена тормозных колодок"
  ],
  "most_common_service": "Замена масла",
  "recommendation": "Рекомендуется регулярно проводить техническое
обслуживание."
}
```

7. Обработайте результат в программе.

```
console.log("Общая сумма:", result.total);

console.log(
    "Основные услуги:",
    result.main_services.join(", ")
);

console.log(
    "Рекомендация:",
    result.recommendation
);
```

8. Реализуйте анализ отзывов пользователей.

Например, из базы данных получаем:

```
const [reviews] = await db.query(
    "SELECT text FROM reviews"
);
```

Передаём отзывы ИИ:

```
const prompt = `
Проанализируй отзывы клиентов:
```

```
${JSON.stringify(reviews)}

Для каждого отзыва определи:
- категорию;
- тональность;
- основную проблему.
`;
```

Пример результата:

```
{
  "reviews": [
    {
      "category": "Обслуживание",
      "sentiment": "Положительная",
      "problem": null
    },
    {
      "category": "Сроки",
      "sentiment": "Отрицательная",
      "problem": "Долгое ожидание ремонта"
    }
  ]
}
```

9. Добавьте проверку данных перед отправкой.

```
if (!rows || rows.length === 0) {
    return {
        error: "Данные для анализа отсутствуют"
    };
}
```

Также не следует передавать ИИ лишние персональные или конфиденциальные данные.

10. Отобразите результат в информационной системе.

Например:

Анализ клиента

Количество заказов: 2
Общая сумма: 11 000 руб.

Основные услуги: замена масла, замена колодок

Рекомендация:
Регулярно проходить техническое обслуживание.

11.Реализуйте аналогичную функцию для собственного проекта.

Можно использовать:
анализ заказов;
анализ отзывов;
рекомендации пользователю;
классификацию заявок;
анализ документов;
формирование отчётов;
поиск закономерностей в данных.

12.Протестируйте функцию.

Проверьте работу с обычными данными, пустой выборкой, большим количеством записей и некорректными данными.

5. Результат: Студент должен представить AI-функцию, которая получает данные из информационной системы, передаёт их в ИИ, получает структурированный результат и отображает его в приложении. Также необходимо представить код получения данных, формирования запроса, обработки результата и обработки ошибок

Критерии оценки.

«5» «отлично»

- студент продемонстрировал готовность и способность к самостоятельному выполнению видов работ в стандартных ситуациях;
- все работы выполнил правильно и уверенно;
- проявил активность в освоении новых практических умений;
- содержание отчета по практическим работам полноценно отражает информацию по выполненным видам работ, установленных программой практики;
- оформление отчета по практическим работам полностью соответствует предъявляемым требованиям;
- при выполнении практических работ студент уверенно продемонстрировал обладание необходимыми компонентами общих и профессиональных компетенций;

«4» «хорошо»

- студент продемонстрировал готовность и способность к самостоятельному выполнению видов работ в стандартных ситуациях;
- все работы выполнил правильно и уверенно;
- содержание отчета по практическим работам в целом правильно отражает информацию по выполненным видам работ;
- оформление отчета по практическим работам соответствует предъявляемым требованиям, однако имеются замечания (некоторая неаккуратность, неполное описание проделанной работы и освоенных умений);
- при выполнении практических работ студент продемонстрировал обладание компонентами общих и профессиональных компетенций;
- характеристики студента положительные, имеются отдельные замечания и рекомендации;

«3» «удовлетворительно»

- при выполнении видов работ студент нуждался во внешнем сопровождении и контроле;
- в отчете по практическим работам содержится описание видов выполняемых работ согласно программе дисциплины, однако, записи неполные и неточные;
- отчет по практическим работам оформлен с отклонениями от установленных требований;
- при выполнении практических работ студент продемонстрировал обладание отдельными компонентами общих и профессиональных компетенций;

«**2» «неудовлетворительно»**

– студент не справился с запланированными видами работ;
– при выполнении практических работ студент продемонстрировал слабое обладание отдельными компонентами общих и профессиональных компетенций;
– отчет студентом не подготовлен.

Тема 1.4. Управление жизненным циклом моделей и AI-решений

Практическое занятие №16 Тестирование и ошибки AI-решения

Проверяемые результаты обучения:

- ПК 3.1 Осуществлять выбор готовых моделей искусственного интеллекта
- ПК 3.2 Формировать сценарии обучения готовых моделей искусственного интеллекта
- ПК 3.3 Проводить обучение и последующую калибровку готовых моделей искусственного интеллекта
- ПК 3.4 Контролировать результат обучения
- ПК 3.5 Оформлять результат проведения процедуры обучения
- ПК 3.6 Формировать запросы для работы с искусственным интеллектом с целью визуализации данных

- ПК 3.7 Формировать сценарии обучения готовых моделей искусственного интеллекта
- ПК 3.8 Интегрировать готовые модели искусственного интеллекта в информационные системы с использованием программных интерфейсов
- ПК 3.9 Создавать контент программного продукта средствами нейросетей
- ПК 3.10 Разрабатывать блоки программного кода с использованием нейросетей
- ПК 3.11 Создавать и управлять концепцией ИТ-продукта
- ПК 3.12 Планировать управление требованиями заказчика в проектах любого уровня сложности в области ИТ

2. Цель практической работы

Научиться тестировать AI-функцию, находить ошибки в работе ИИ, проверять качество результатов и вести версии промптов и AI-модуля

3. Задачи практической работы

Протестировать AI-функцию собственного проекта на разных входных данных, проверить обработку ошибок и сравнить несколько версий AI-решения.

4. Этапы выполнения

1. Выберите AI-функцию проекта.

Пример:

«ИИ анализирует описание неисправности автомобиля и формирует рекомендацию».

2. Подготовьте тестовые данные.

Нормальный запрос:

```
{
  "problem": "При торможении появляется скрип"
}
```

Пустой запрос:

```
{
```

```
  "problem": ""
}
```

Недостаточный запрос:

```
{
  "problem": "Машина плохо работает"
}
```

Некорректные данные:

```
{
  "problem": 12345
}
```

3. Проверьте AI-функцию на каждом тестовом случае.

Зафиксируйте:
входные данные;
ответ ИИ;
наличие ошибки;
корректность результата;
необходимость дополнительной проверки.

4. Проверьте обработку ошибок API.

Например:

```
try {
    const response = await fetch("/api/ai/analyze", {
        method: "POST",
        headers: {
            "Content-Type": "application/json"
        },
        body: JSON.stringify(data)
    });

    if (!response.ok) {
        throw new Error("AI API недоступен");
    }

    const result = await response.json();

} catch (error) {
    console.error(error);
```

```
    return {
        error: "Не удалось получить ответ ИИ"
    };
}
```

5. Проверьте некорректный ответ ИИ.

Например, приложение ожидает:

```
{
  "category": "Двигатель",
  "recommendation": "Провести диагностику"
}
```

Но ИИ вернул:

Возможно, проблема связана с двигателем. Рекомендуется обратиться в сервис.

Определите, сможет ли приложение обработать такой ответ. Если нет — измените промпт или добавьте проверку результата.

6. Добавьте проверку обязательных полей.

```
function validateAIResult(result) {
    if (!result.category) {
        return false;
    }

    if (!result.recommendation) {
        return false;
    }

    return true;
}
```

7. Создайте первую версию промпта.

Проанализируй проблему автомобиля и предложи возможное решение.

Сохраните её как версию 1.0.

8. Улучшите промпт.

Версия 2.0:

Ты — консультант автосервиса.

Проанализируй описание проблемы автомобиля.
Определи категорию проблемы и возможные причины.
Не утверждай, что причина установлена точно.

Верни результат в формате JSON:
```
{
  "category": "",
  "possible_causes": [],
  "recommendation": ""
}
```

Сравните результаты двух версий.

9. Проведите повторное тестирование версии 2.0 на тех же данных.

Сравните:
точность;
структуру ответа;
количество ошибок;
стабильность результата;
соответствие требованиям.

10. Ведите журнал версий.

Пример:

Версия 1.0
Промпт: базовый запрос
Проблема: ответы имеют разный формат

Версия 2.0
Промпт: добавлена роль и JSON
Результат: структура ответа стала предсказуемой

Версия 2.1
Промпт: добавлена обработка недостатка данных
Результат: ИИ запрашивает дополнительную информацию

11. Проведите итоговое тестирование AI-функции на нескольких различных запросах.

Определите, какие ошибки удалось устранить и какие проблемы ещё требуют доработки.

12.Подготовьте итоговую версию AI-решения для собственного проекта

5. Результат: Студент должен представить тестовые данные, результаты тестирования, не менее двух версий промпта, журнал изменений, обработку ошибок, проверку ответа ИИ и итоговую протестированную версию AI-функции

Критерии оценки.

«5» «отлично»

- студент продемонстрировал готовность и способность к самостоятельному выполнению видов работ в стандартных ситуациях;
- все работы выполнил правильно и уверенно;
- проявил активность в освоении новых практических умений;
- содержание отчета по практическим работам полноценно отражает информацию по выполненным видам работ, установленных программой практики;
- оформление отчета по практическим работам полностью соответствует предъявляемым требованиям;
- при выполнении практических работ студент уверенно продемонстрировал обладание необходимыми компонентами общих и профессиональных компетенций;

«4» «хорошо»

- студент продемонстрировал готовность и способность к самостоятельному выполнению видов работ в стандартных ситуациях;
- все работы выполнил правильно и уверенно;
- содержание отчета по практическим работам в целом правильно отражает информацию по выполненным видам работ;
- оформление отчета по практическим работам соответствует предъявляемым требованиям, однако имеются замечания (некоторая неаккуратность, неполное описание проделанной работы и освоенных умений);
- при выполнении практических работ студент продемонстрировал обладание компонентами общих и профессиональных компетенций;
- характеристики студента положительные, имеются отдельные замечания и рекомендации;

«3» «удовлетворительно»

- при выполнении видов работ студент нуждался во внешнем сопровождении и контроле;
- в отчете по практическим работам содержится описание видов выполняемых работ согласно программе дисциплины, однако, записи неполные и неточные;
- отчет по практическим работам оформлен с отклонениями от установленных требований;
- при выполнении практических работ студент продемонстрировал обладание отдельными компонентами общих и профессиональных компетенций;

«2» «неудовлетворительно»
– студент не справился с запланированными видами работ;
– при выполнении практических работ студент продемонстрировал слабое обладание отдельными компонентами общих и профессиональных компетенций;
– отчет студентом не подготовлен.

Практическое занятие №17 Версионирование AI-решения

Проверяемые результаты обучения:

- ПК 3.1 Осуществлять выбор готовых моделей искусственного интеллекта
- ПК 3.2 Формировать сценарии обучения готовых моделей искусственного интеллекта
- ПК 3.3 Проводить обучение и последующую калибровку готовых моделей искусственного интеллекта
- ПК 3.4 Контролировать результат обучения
- ПК 3.5 Оформлять результат проведения процедуры обучения
- ПК 3.6 Формировать запросы для работы с искусственным интеллектом с целью визуализации данных
- ПК 3.7 Формировать сценарии обучения готовых моделей искусственного интеллекта
- ПК 3.8 Интегрировать готовые модели искусственного интеллекта в информационные системы с использованием программных интерфейсов
- ПК 3.9 Создавать контент программного продукта средствами нейросетей
- ПК 3.10 Разрабатывать блоки программного кода с использованием нейросетей
- ПК 3.11 Создавать и управлять концепцией ИТ-продукта
- ПК 3.12 Планировать управление требованиями заказчика в проектах любого уровня сложности в области ИТ

2. Цель практической работы

Научиться оценивать качество разработанного AI-решения, проверять соответствие поставленной задаче и подготовить его к демонстрации.

3. Задачи практической работы

Провести итоговую оценку AI-функции собственного проекта, проверить её работу на тестовых данных, оформить результаты и подготовить короткую демонстрацию.

4. Этапы выполнения

1. Опишите разработанную AI-функцию.

Пример:

«AI-модуль анализирует описание неисправности автомобиля, определяет категорию проблемы, возможные причины и формирует рекомендацию».

2. Определите критерии оценки.

Например:

точность результата;
соответствие формату;
скорость получения ответа;
устойчивость к разным запросам;
обработка ошибок;
удобство использования.

3. Подготовьте тестовые данные.

Пример:

Тест 1:
"При торможении появился скрип"

Ожидаемый результат:
Категория — тормозная система.

Тест 2:
"Двигатель плохо запускается утром"

Ожидаемый результат:
Категория — двигатель.

Тест 3:
"Машина плохо работает"

Ожидаемый результат:
ИИ должен указать, что информации недостаточно.

4. Проведите тестирование и занесите результаты в таблицу.

Пример:

Тест | Ожидаемый результат | Полученный результат | Результат проверки

1 | Тормозная система | Тормозная система | Соответствует
2 | Двигатель | Двигатель | Соответствует
3 | Запрос уточняющей информации | Запрос уточнения | Соответствует

5. Оцените качество AI-ответов.

Проверьте:

«ИИ правильно определяет категорию?»

«Ответ содержит только необходимую информацию?»

«Соблюдается заданный формат?»

«Обрабатываются ли ошибочные данные?»

6. Сравните работу до и после оптимизации.

Пример:

Версия 1.0:
ИИ возвращает произвольный текст.

Версия 2.0:
ИИ возвращает структурированный JSON.

Версия 2.1:
ИИ возвращает JSON и сообщает о недостатке данных.

Сделайте вывод, какие изменения улучшили работу системы.

7. Подготовьте демонстрацию.

Продемонстрируйте:

1. интерфейс приложения;
2. ввод данных пользователем;
3. отправку запроса;
4. обработку данных ИИ;
5. получение результата;
6. обработку ошибки.

8. Подготовьте небольшой отчёт.

В отчёте укажите:

название проекта;
назначение AI-функции;
используемый AI-инструмент;
сценарий работы;
пример запроса;
пример ответа;
результаты тестирования;
обнаруженные ошибки;
выполненные улучшения.

9. Подготовьте демонстрационный пример.

Например:

Пользователь:
"При торможении появился металлический скрип".

ИИ:
Категория: тормозная система.
Возможные причины:
– износ тормозных колодок;
– загрязнение тормозного механизма.

Рекомендация:
Провести диагностику тормозной системы.

10. Проведите итоговую демонстрацию преподавателю и объясните, какую задачу решает AI-функция, как она работает и какие ограничения имеет

5. Результат: Студент должен представить работающую AI-функцию, результаты её тестирования, таблицу проверки, сравнение версий, краткий отчёт и демонстрацию работы AI-решения

Критерии оценки.

«5» «отлично»

- студент продемонстрировал готовность и способность к самостоятельному выполнению видов работ в стандартных ситуациях;
- все работы выполнил правильно и уверенно;
- проявил активность в освоении новых практических умений;
- содержание отчета по практическим работам полноценно отражает информацию по выполненным видам работ, установленных программой практики;

- оформление отчета по практическим работам полностью соответствует предъявляемым требованиям;
- при выполнении практических работ студент уверенно продемонстрировал обладание необходимыми компонентами общих и профессиональных компетенций;

«4» «хорошо»

- студент продемонстрировал готовность и способность к самостоятельному выполнению видов работ в стандартных ситуациях;
- все работы выполнил правильно и уверенно;
- содержание отчета по практическим работам в целом правильно отражает информацию по выполненным видам работ;
- оформление отчета по практическим работам соответствует предъявляемым требованиям, однако имеются замечания (некоторая неаккуратность, неполное описание проделанной работы и освоенных умений);
- при выполнении практических работ студент продемонстрировал обладание компонентами общих и профессиональных компетенций;
- характеристики студента положительные, имеются отдельные замечания и рекомендации;

«3» «удовлетворительно»

- при выполнении видов работ студент нуждался во внешнем сопровождении и контроле;
- в отчете по практическим работам содержится описание видов выполняемых работ согласно программе дисциплины, однако, записи неполные и неточные;
- отчет по практическим работам оформлен с отклонениями от установленных требований;
- при выполнении практических работ студент продемонстрировал обладание отдельными компонентами общих и профессиональных компетенций;

«**2» «неудовлетворительно»**

– студент не справился с запланированными видами работ;

– при выполнении практических работ студент продемонстрировал слабое обладание отдельными компонентами общих и профессиональных компетенций;

– отчет студентом не подготовлен.

Практическое занятие №18 Оценка и представление результатов разработки с использованием AI

Проверяемые результаты обучения:

- ПК 3.1 Осуществлять выбор готовых моделей искусственного интеллекта
- ПК 3.2 Формировать сценарии обучения готовых моделей искусственного интеллекта

- ПК 3.3 Проводить обучение и последующую калибровку готовых моделей искусственного интеллекта
- ПК 3.4 Контролировать результат обучения
- ПК 3.5 Оформлять результат проведения процедуры обучения
- ПК 3.6 Формировать запросы для работы с искусственным интеллектом с целью визуализации данных
- ПК 3.7 Формировать сценарии обучения готовых моделей искусственного интеллекта
- ПК 3.8 Интегрировать готовые модели искусственного интеллекта в информационные системы с использованием программных интерфейсов
- ПК 3.9 Создавать контент программного продукта средствами нейросетей
- ПК 3.10 Разрабатывать блоки программного кода с использованием нейросетей
- ПК 3.11 Создавать и управлять концепцией ИТ-продукта
- ПК 3.12 Планировать управление требованиями заказчика в проектах любого уровня сложности в области ИТ

2. Цель практической работы

Освоить навыки обработки и структурирования ответов от LLM для использования в приложениях.

3. Задачи практической работы

– Разработать скрипты для обработки и структурирования ответов;
– Реализовать извлечение необходимой информации из ответов LLM;
– Преобразовать ответы в требуемый формат;
– Реализовать обработку ошибок и некорректных ответов.

4. Этапы выполнения

1. Проанализируйте типы ответов, которые возвращает LLM на различные запросы:
   - Определите структуру ответов для разных сценариев;
   - Выявите возможные проблемы с ответами (неполные, некорректные).
2. Разработайте функции для обработки ответов, которые должны:
   - Очищать ответ от лишних пробелов и символов;
   - Извлекать JSON-данные из текстового ответа (если возможно);
   - Структурировать текстовые ответы (разбивать на части, выделять ключевые моменты).
3. Реализуйте обработку ошибок в ответах:
   - Проверка на пустой ответ;
   - Проверка на обрыв ответа (если достигнут max_tokens);

- Проверка на некорректный JSON (с повторной попыткой);
- Повторные запросы при некорректных ответах.

4. Интегрируйте обработку ответов с вашим скриптом взаимодействия с LLM:
   - Модифицируйте функцию получения ответа для автоматической обработки;
   - Добавьте логирование ошибок.
5. Протестируйте обработку ответов на различных типах запросов:
   - Запрос с ожиданием JSON-ответа;
   - Запрос с ожиданием структурированного текста;
   - Запрос с возможным обрывом ответа.

5. Теоретический материал

Обработка ответов LLM – это процесс преобразования сырого текстового ответа модели в структурированный формат, пригодный для использования в приложении.

Методы обработки ответов:

Метод	Описание	Применение
Парсинг	Извлечение данных с помощью регулярных выражений	Структурированные ответы
JSON-декодирование	Преобразование JSON-строки в объект	API-интеграция
Семантический анализ	Извлечение смысловых компонентов	Анализ текста
Пост-обработка	Очистка, форматирование	Улучшение качества

6. Результат: В ходе работы должны быть разработаны функции обработки и структурирования ответов от LLM, реализована обработка ошибок и некорректных ответов, проведено тестирование на различных типах запросов.

Критерии оценки.

«5» «отлично»

- студент продемонстрировал готовность и способность к самостоятельному выполнению видов работ в стандартных ситуациях;
- все работы выполнил правильно и уверенно;
- проявил активность в освоении новых практических умений;
- содержание отчета по практическим работам полноценно отражает информацию по выполненным видам работ, установленных программой практики;
- оформление отчета по практическим работам полностью соответствует предъявляемым требованиям;
- при выполнении практических работ студент уверенно продемонстрировал обладание необходимыми компонентами общих и профессиональных компетенций;

«4» «хорошо»

- студент продемонстрировал готовность и способность к самостоятельному выполнению видов работ в стандартных ситуациях;
- все работы выполнил правильно и уверенно;
- содержание отчета по практическим работам в целом правильно отражает информацию по выполненным видам работ;
- оформление отчета по практическим работам соответствует предъявляемым требованиям, однако имеются замечания (некоторая неаккуратность, неполное описание проделанной работы и освоенных умений);
- при выполнении практических работ студент продемонстрировал обладание компонентами общих и профессиональных компетенций;
- характеристики студента положительные, имеются отдельные замечания и рекомендации;

«3» «удовлетворительно»

- при выполнении видов работ студент нуждался во внешнем сопровождении и контроле;
- в отчете по практическим работам содержится описание видов выполняемых работ согласно программе дисциплины, однако, записи неполные и неточные;
- отчет по практическим работам оформлен с отклонениями от установленных требований;
- при выполнении практических работ студент продемонстрировал обладание отдельными компонентами общих и профессиональных компетенций;

«**2» «неудовлетворительно»**

– студент не справился с запланированными видами работ;

– при выполнении практических работ студент продемонстрировал слабое обладание отдельными компонентами общих и профессиональных компетенций;

– отчет студентом не подготовлен.

4. Контроль приобретения практического опыта. Оценка по учебной практике (по профилю специальности)

4.1. Общие положения

Целью оценки учебной практики является оценка:

1) профессиональных и общих компетенций;

2) практического опыта и умений.

Зачет по учебной практике выставляется на основании данных аттестационного листа (характеристики профессиональной деятельности обучающегося/студента на практике) с указанием видов работ, выполненных обучающимся во время практики, их объема, качества выполнения в соответствии с технологией и (или) требованиями организации, в которой проходила практика.

4.2. Виды работ практики и проверяемые результаты обучения по профессиональным модулям

Таблица 7

Практический опыт	Виды и объем работ на учебной практике, требования к их выполнению и/ или условия выполнения	Документ, подтверждающий качество выполнения работ
1	2	3
- Разрабатывает, оптимизирует и производит оценку сложности алгоритмов для ИИ-программ - Использует библиотеки и инструменты для работы с алгоритмами и данными (например: Pandas, NumPy, Scikit-learn) - Применяет структуры данных (деревья, графы, списки) для реализации алгоритмов - Разрабатывает модульные ИИ-системы, соответствующие требованиям производительности и безопасности - Внедряет разработанные ИИ-модули в комплексные программные системы	1.Формировать алгоритмы разработки программных модулей в соответствии с техническим заданием 2.Использовать систему контроля версий программного кода с учетом обеспечения возможности организации групповой разработки 3.Документировать логику и поведение интерфейса: описывать взаимосвязи, взаимодействия и возможные состояния элементов графического пользовательского интерфейса 4.Использовать специализированные инструменты для прототипирования и	аттестационный лист и отчет, подтверждающие практический опыт, полученный на практике

- Оптимизирует код и работает с интерфейсами для взаимодействия между модулями - Оформляет, документирует и структурирует код для последующей поддержки - Использует инструменты статического анализа кода для выявления ошибок и улучшения качества - Работает с системами документирования кода (например, Doxygen, Sphinx) - Управляет проектами с использованием систем контроля версий для организации командной работы - Разрешает конфликты при слиянии веток и использует pull request для рецензирования кода - Настраивает процессы CI/CD для автоматического тестирования и развертывания кода - Отлаживает программные модули с использованием пошаговой проверки - Применяет методы логирования и профилирования производительности - Использует специальные средства для отладки многопоточных программ - Выполняет статическое тестирование программного кода на предмет выявления ошибок/дефектов алгоритмов, в том числе – на наличие обработки исключений - Выполняет тестирование программных модулей в соответствии с тест-планом - Генерирует тестовые данные - Выполняет интеграционное тестирование в соответствии с заданием - Выполняет регрессионное тестирование в соответствии с заданием - Работает с CI/CD пайплайнами для автоматизации тестирования	проектирования графических пользовательских интерфейсов 5 Разрабатывать программные модули с использованием алгоритмов и методов искусственного интеллекта 6 Осуществлять интеграцию программных модулей и компонентов программного обеспечения. 7 Разрабатывать программные интерфейсы и процедуры сборки и развертывания программного обеспечения 8 Осуществлять миграцию, преобразование и конвертацию данных программных систем 9 Разрабатывать и интегрировать серверные программные модули для ИИ-решений 10 Разрабатывать и настраивать программные интерфейсы взаимодействия с ИИ-компонентами и внешними системами 11 Выполнять развертывание, тестирование и обеспечение работоспособности серверных компонентов ИИ-решений 12.Разрабатывать программные модули в соответствии с техническим заданием 13 Оформлять программный код в соответствии с техническим заданием 14 Выполнять отладку программных модулей с использованием специализированных программных средств 15 Выполнять тестирование программного кода 16 Составлять тестовые сценарии 17 Разрабатывать программный код мобильных приложений в соответствии с техническим заданием и утвержденными спецификациями 18 Осуществлять оптимизацию программного кода мобильных приложений в соответствии с техническим заданием, оценивая трудоемкость и сроки выполнения работ 19 Выполнять разработку программных модулей	

- Разрабатывает тестовые сценарии в соответствии с тестовым планом (тестирование производительности, надежности, UI-тестирование), в том числе с применением средств автоматизации проектирования - Разрабатывает тестовые пакеты и задания на выполнение тестирования - Производит оценку тестовых данных на предмет покрытия строк и покрытия ветвей, выполняет валидацию данных - Автоматизирует создание и выполнение тестовых сценариев - Составляет формализованные описания решений поставленных задач в соответствии с требованиями технического задания или внутренних документов организации - Разрабатывает алгоритмы решения поставленных задач в соответствии с требованиями технического задания или внутренних документов организации - Проверяет корректность алгоритмов решения поставленных задач - Создает программный код в соответствии с техническим заданием (готовыми спецификациями) - Оптимизирует программный код с использованием специализированных программных средств - Производит оценку сроков выполнения поставленных задач и согласовывает их - Создает программный код в строгом соответствии с техническим заданием и готовыми спецификациями - Оптимизирует программный код с использованием профильных инструментов и средств	20 Выполнять тестирование программных модулей и ИИ-решений 21 Осуществлять защиту данных в мобильных приложениях 22 Интегрировать информационную систему с существующими информационными системами заказчика 23 Выполнять непрерывную интеграцию и непрерывное развертывание программного обеспечения в процессе разработки 24 Управлять конфигурациями и инфраструктурой 25 Управлять версиями и кодом	

- Производит оценку реалистичных сроков выполнения поставленных задач и согласовывает их - Разрабатывает и проверяет интерфейсные тексты с учетом глоссария терминов - Детально описывает логику работы, взаимосвязи, взаимодействие и возможные состояния элементов интерфейса - Разрабатывает и оформляет проектную документацию на графический интерфейс - Проектирует графический пользовательский интерфейс согласно утвержденной концепции или готовому образцу - Создает интерактивные прототипы графического интерфейса - Создает эскизы (наброски) графических пользовательских интерфейсов - Работает с программами и инструментами прототипирования интерфейсов - Выявляет типы тестовых данных для выполнения тестирования ПО - Выделяет тестовые данные из имеющихся машинных файлов - Выделяет тестовые данные из внемашинных документов - Генерирует тестовые данные (при необходимости) - Формирует и представляет отчетность о подготовке тестовых данных в соответствии с установленными регламентами - Проверяет корректность начального состояния компонентов инструментария и тестируемого ПО перед началом тестирования - Выполняет тестовые процедуры на тестовых данных - Сравнивает фактический и ожидаемый результаты выполнения тестовых процедур		

- Формирует и представляет отчетность о выполнении процесса тестирования ПО в соответствии с установленными регламентами - Анализирует и выявляет проблемы сопряжения неоднородных модулей и компонентов компьютерного программного обеспечения - Разрабатывает и документирует программные интерфейсы - Разрабатывает процедуры сборки модулей и компонентов компьютерного программного обеспечения - Разрабатывает процедуры развертывания и обновления компьютерного программного обеспечения - Разрабатывает процедуры миграции и преобразования (конвертации) данных - Собирает программные модули и компоненты в программный продукт - Вносит изменения в процедуры сборки модулей и компонентов компьютерного программного обеспечения, развертывания компьютерного программного обеспечения, миграции и преобразования данных - Подключает программный продукт к компонентам внешней среды - Проверяет работоспособность выпусков программного продукта - Разрабатывает безопасные методы аутентификации и авторизации пользователей - Обрабатывает и хранит конфиденциальные данные - Отслеживает и обрабатывает уязвимости безопасности - Использует шифрование для защиты данных в покое и в движении		

- Использует шифрование данных для защиты конфиденциальной информации, такой как пароли, персональные данные пользователей и другие чувствительные данные - Реализует механизмы аутентификации и авторизации для обеспечения доступа только авторизованным пользователям - Применяет механизмы хеширования для защиты паролей пользователей от несанкционированного доступа - Обеспечивает безопасность передачи данных между клиентскими устройствами и серверами с использованием протоколов шифрования, таких как SSL/TLS - Разрабатывает механизмы контроля доступа к данным, чтобы предотвратить несанкционированное чтение, изменение или удаление данных - Проектирует и реализует системы резервного копирования и восстановления данных для обеспечения их сохранности в случае сбоев или потери устройства - Тестирует приложения на уязвимости безопасности, такие как SQL-инъекции, межсайтовые сценарии и другие уязвимости, и принимает меры по их устранению - Соблюдает законодательство и нормативные требования в области защиты данных - Интегрирует информационную систему с существующими системами заказчика - Разрабатывает API для интеграции информационной системы - Тестирует и отлаживает интеграцию информационной системы		

- Проектирует интерфейсы обмена данными в соответствии с трудовым заданием - Разрабатывает интерфейсы обмена данными в соответствии с трудовым заданием - Создает и настраивает сборочные и развертывающие пайплайны (pipelines) - Автоматизирует тестирование, сборку и доставку приложений - Использует инструменты для автоматизации CI/CD - Использует контейнеры для изоляции этапов выполнения pipelines - Разрабатывает и настраивает CI/CD пайплайны для автоматической сборки, тестирования и развертывания веб-приложений - Использует инструменты для автоматического тестирования кода - Разрабатывает и настраивает системы контроля версий, такие как Git, для управления исходным кодом и версионирования приложений - Настраивает инфраструктуру для развертывания веб-приложений, включая сервера приложений, базы данных и другие необходимые компоненты - Настраивает конфигурацию инфраструктуры и управляет ею через код - Использует инструменты для автоматизации конфигурации - Создает и поддерживает сценарии управления конфигурациями - Разрабатывает и настраивает конфигурационные файлы для различных компонентов веб-приложений, таких как серверы приложений, базы данных и другие сервисы - Автоматизирует процесс развертывания и конфигурации		

инфраструктуры с использованием инструментов - Управляет версиями конфигурационных файлов и инфраструктуры с использованием систем контроля версий - Использует системы контроля версий - Работает с репозиториями кода и ветками разработки - Разрешает конфликты и объединяет код - Использует системы контроля версий для управления и отслеживания изменений в коде приложений - Разрабатывает и поддерживает процессы работы с Git, включая создание веток, слияние изменений и управление конфликтами - Настраивает инфраструктуру для хранения и управления кодом приложений с использованием репозиториев Git		
- Идентифицирует проблемы, связанные с нормальным функционированием базы данных - Восстанавливает систему - Администрирует сервер баз данных - Участвует в администрировании отдельных компонентов серверов - Документирует результаты аудита безопасности информации - Использует процедуру резервного копирования баз данных - Использует процедуру восстановления баз данных - Подготавливает документацию по формированию требований к хранилищам банка данных - Проектирует, разрабатывает и эксплуатирует базы данных	1. Установка СУБД и настройка параметров серверной службы. 2. Настройка средств администрирования и анализ выполнения запросов для выявления и устранения проблем. 3. Создание учетных записей, ролей и назначение прав доступа в СУБД. 4. Настройка аутентификации, аудита и контроль соблюдения политик безопасности. 5. Планирование и выполнение резервного копирования, мониторинг и хранение резервных копий. 6. Восстановление базы данных из резервной копии, проверка целостности и документирование результата. 7. Анализ предметной области, разработка ER-модели и документации требований к базе данных. 8. Разработка логической и	аттестационный лист и отчет, подтверждающие практический опыт, полученный на практике

- Планирует процедуры резервного копирования данных - Запускает процедуру резервного копирования данных - Осуществляет мониторинг выполнения процедур резервного копирования данных - Контролирует завершение процедуры резервного копирования данных - Проводит повторную процедуру резервного копирования данных в случае ее нештатного завершения - Хранит резервные копии БД - Назначает приоритет запросов к БД - Оптимизирует выполнение статистически значимых запросов к БД - Создает программный код в соответствии с техническим заданием (готовыми спецификациями) - Оптимизирует программный код с использованием специализированных программных средств - Определяет источники больших данных для анализа, идентифицирует внешние и внутренние источники данных для проведения аналитических работ - Получает и фильтрует большие объемы данных из гетерогенных источников - Извлекает, проверяет и очищает большие объемы данных из гетерогенных источников - Агрегирует большие объемы данных из гетерогенных источников и разрабатывает их представление - Оценивает соответствие набора данных предметной области и задачам аналитических работ - Знает возможности	физической моделей, нормализация данных, определение ключей и связей. 9. Создание таблиц и ограничений целостности средствами SQL, заполнение и импорт данных. 10. Разработка SQL-запросов, процедур и триггеров, анализ плана выполнения и настройка индексов. 11. Реализация ролевой модели безопасности, настройка аудита и защиты данных. 12. Выполнение CRUD-операций и агрегации в NoSQL-базе, настройка индексов и программного кода доступа к данным. 13. Профилирование и классификация источников данных, определение требований к качеству и формату набора данных. 14. Очистка, преобразование, разметка и формирование датасета, контроль качества результата. 15. Разработка DAG с задачами, зависимостями, расписанием и подключениями в Apache Airflow. 16. Реализация обработки данных в PySpark и контейнеризация конвейера в Docker Compose. 17. Настройка автоматических проверок качества и обнаружения аномалий данных. 18. Настройка мониторинга, дашборда, алертинга и контроля доступа к данным конвейера.	

имеющейся у исполнителя методологической и технологической инфраструктуры анализа больших данных - Знает предметную область анализа - Знает теоретические и прикладные основы анализа больших данных - Знает современные методы и инструментальные средства анализа больших данных - Знает современный опыт использования анализа больших данных - Осуществляет стратегическое и оперативное руководство кадровыми и материальными ресурсами в процессах получения, размещения, хранения, распределения, миграции, архивирования и удаления данных - Осуществляет стратегическое и оперативное руководство материально-техническими ресурсами и элементами технологической инфраструктуры больших данных - Разрабатывает модели данных, адаптированные к технологиям больших данных - Интегрирует большие данные с системами хранения данных организации - Создает облачные репозитории и интегрирует их с системами хранения данных организации - Создает параллельные системы хранения и обработки информации - Разрабатывает методы и регламенты реализации различных режимов обработки больших данных, в том числе режима реального времени - Осуществляет мониторинг и оценивает производительность обработки данных в организации, разрабатывает предложения по		

повышению производительности обработки больших данных - Разрабатывает методы и регламенты преобразования данных - Разрабатывает и согласовывает стратегию формирования информационных активов организации - Разрабатывает методы и регламенты оценки информационных активов организации - Формирует состав персонала, отвечающего за приобретение и получение больших данных, и координирует его деятельность - Оценивает эффективность системы хранения и обработки данных организации - Разрабатывает предложения по развитию и совершенствованию системы получения, хранения, передачи, обработки больших данных - Формирует цели, приоритеты и ограничения управления качеством данных, адаптирует цели к изменениям внутренних и внешних условий, потребностей и возможностей, в том числе технологических - Разрабатывает стратегию управления качеством больших данных - Выделяет ресурсы для управления качеством больших данных - Реализует стратегию обеспечения качества больших данных - Анализирует реализацию стратегии обеспечения качества больших данных - Проводит оценку качества больших данных - Разрешает проблемы и инциденты с качеством больших данных - Оценивает риски		

использования больших данных		
- Подбирает и настраивает готовые модели ИИ с учетом поставленных задач, анализирует результаты их применения - Создает сценарии обучения, подготавливает данные для обучения, настраивает гиперпараметры для достижения оптимального результата - Обучает модели на подготовленных данных, применяет методы калибровки для улучшения точности моделей - Оценивает эффективность обученных моделей, корректирует обучение при необходимости, анализирует ошибки и улучшает модель - Создает отчеты по обучению моделей, использует инструменты для визуализации (Matplotlib, Seaborn) для наглядного представления данных - Формирует запросы для получения и анализа данных, строит графики и диаграммы для визуализации результатов работы ИИ - Определяет источники больших данных для анализа, идентифицирует внешние и внутренние источники данных для проведения аналитических работ - Получает и фильтрует большие объемы данных из гетерогенных источников - Извлекает, проверяет и очищает большие объемы данных из гетерогенных источников - Агрегирует большие объемы данных из гетерогенных источников и разрабатывает их представление - Оценивает соответствие набора данных предметной области и задачам аналитических работ - Выбирает методы и инструментальные средства	1.Осуществлять выбор готовых моделей искусственного интеллекта 2.Формировать сценарии обучения готовых моделей искусственного интеллекта 3.Проводить обучение и последующую калибровку готовых моделей искусственного интеллекта 4.Контролировать результат обучения 5.Оформлять результат проведения процедуры обучения 6.Формировать запросы для работы с искусственным интеллектом с целью визуализации данных 7.Формировать сценарии обучения готовых моделей искусственного интеллекта 8.Интегрировать готовые модели искусственного интеллекта в информационные системы с использованием программных интерфейсов 9.Создавать контент программного продукта средствами нейросетей 10.Разрабатывать блоки программного кода с использованием нейросетей 11.Создавать и управлять концепцией ИТ-продукта 12.Планировать управление требованиями заказчика в проектах любого уровня сложности в области ИТ	аттестационный лист и отчет, подтверждающие практический опыт, полученный на практике

анализа больших данных для проведения аналитических работ - Разрабатывает, проверяет и оценивает используемые модели больших данных - Адаптирует и развертывает модели больших данных в предметной среде - Выбирает средства представления результатов аналитики больших данных - Подготавливает отчет по результатам аналитических работ - Разрабатывает код ИС и базы данных ИС в рамках выполнения работ по созданию (модификации) и сопровождению ИС - Верифицирует код ИС и базы данных ИС относительно дизайна ИС и структуры баз данных ИС в рамках выполнения работ по созданию (модификации) и сопровождению ИС - Устраняет обнаруженные несоответствия в коде ИС и базах данных ИС относительно дизайна ИС и структуры баз данных ИС в рамках выполнения работ по созданию (модификации) и сопровождению ИС - Разрабатывает интерфейсы обмена данными между ИС заказчика в рамках выполнения работ по созданию (модификации) и сопровождению ИС - Разрабатывает форматы обмена данными между ИС заказчика в рамках выполнения работ по созданию (модификации) и сопровождению ИС - Разрабатывает технологии обмена данными между ИС заказчика в соответствии с трудовым заданием в рамках выполнения работ по созданию (модификации) и сопровождению ИС - Перерабатывает тексты различной тематики (рерайт) - Выполняет поисковую		

оптимизацию и адаптирует текстовые материалы - Создает программный код в соответствии с техническим заданием (готовыми спецификациями) - Оптимизирует программный код с использованием специализированных программных средств - Собирает идеи о требованиях к ИТ-продукту из различных источников - Разрабатывает функциональные требования к ИТ-продукту - Создает макеты интерфейса ИТ-продукта - Разрабатывает нефункциональные требования к ИТ-продукту - Организует разработку и разрабатывает план управления требованиями заказчика в проектах в области ИТ любого уровня сложности - Согласовывает план управления требованиями в проектах в области ИТ любого уровня сложности с заинтересованными лицами - Утверждает план управления требованиями заказчика в проектах в области ИТ любого уровня сложности		

5. Контрольно-оценочные материалы для учебной практики

Зачет по учебной практике выставляется на основании:

- данных аттестационного листа (характеристики профессиональной деятельности обучающегося/студента на практике) с указанием видов работ, выполненных обучающимся во время практики, их объема, качества выполнения в соответствии с технологией и (или) требованиями организации, в которой проходила практика;
- выполненных практических работ;
- отчета по практическим работам, оформленным в соответствии с требованиями.