МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ "РЯЗАНСКИЙ ГОСУДАРСТВЕННЫЙ РАДИОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ИМЕНИ В.Ф. УТКИНА"

СОГЛАСОВАНО
Зав. выпускающей кафедры

УТВЕРЖДАЮ

Проектирование интеллектуальных программных систем

рабочая программа дисциплины (модуля)

Закреплена за кафедрой	**Вычислительной и прикладной математики**
Учебный план	09.03.04_25_00_ ИИ_В2.plx 09.03.04 Программная инженерия
Квалификация	**бакалавр**
Форма обучения	**очная**
Общая трудоемкость	**5 ЗЕТ**

Распределение часов дисциплины по семестрам

Семестр (<Курс>.<Семестр на курсе>)	**7 (4.1)**		Итого	
Недель	16			
Вид занятий	УП	РП	УП	РП
Лекции	32	32	32	32
Лабораторные	16	16	16	16
Практические	32	32	32	32
Иная контактная работа	0,65	0,65	0,65	0,65
Консультирование перед экзаменом и практикой	2	2	2	2
Итого ауд.	82,65	82,65	82,65	82,65
Контактная работа	82,65	82,65	82,65	82,65
Сам. работа	28,3	28,3	28,3	28,3
Часы на контроль	53,35	53,35	53,35	53,35
Письменная работа на курсе	15,7	15,7	15,7	15,7
Итого	180	180	180	180

г. Рязань

Программу составил(и):
д.т.н., проф., Белов В.В.

Рабочая программа дисциплины
Проектирование интеллектуальных программных систем

разработана в соответствии с ФГОС ВО:
ФГОС ВО - бакалавриат по направлению подготовки 09.03.04 Программная инженерия (приказ Минобрнауки России от 19.09.2017 г. № 920)

составлена на основании учебного плана:
09.03.04 Программная инженерия
утвержденного учёным советом вуза от 29.08.2025 протокол № 1.

Рабочая программа одобрена на заседании кафедры
Вычислительной и прикладной математики

Протокол от 15.06.2026 г. № 7
Срок действия программы: 20262030 уч.г.
Зав. кафедрой Овечкин Геннадий Владимирович

Визирование РПД для исполнения в очередном учебном году

Рабочая программа пересмотрена, обсуждена и одобрена для исполнения в 2026-2027 учебном году на заседании кафедры
Вычислительной и прикладной математики

Протокол от __ _____ 2026 г. № __

Зав. кафедрой _____

Визирование РПД для исполнения в очередном учебном году

Рабочая программа пересмотрена, обсуждена и одобрена для исполнения в 2027-2028 учебном году на заседании кафедры
Вычислительной и прикладной математики

Протокол от __ _____ 2027 г. № __

Зав. кафедрой _____

Визирование РПД для исполнения в очередном учебном году

Рабочая программа пересмотрена, обсуждена и одобрена для исполнения в 2028-2029 учебном году на заседании кафедры
Вычислительной и прикладной математики

Протокол от __ _____ 2028 г. № __

Зав. кафедрой _____

Визирование РПД для исполнения в очередном учебном году

Рабочая программа пересмотрена, обсуждена и одобрена для исполнения в 2029-2030 учебном году на заседании кафедры

Вычислительной и прикладной математики

Протокол от __ _____ 2029 г. № __

Зав. кафедрой _____

1. ЦЕЛИ ОСВОЕНИЯ ДИСЦИПЛИНЫ (МОДУЛЯ)	
1.1	Цель дисциплины: Формирование компетенций в проектировании масштабируемых, поддерживаемых и этичных архитектур интеллектуальных систем через интеграцию ИИ-компонентов, проектирование интерфейсов и качество документирования. Развитие стремления к научному поиску путей совершенствования своей работы посредством глубокого понимания процессов и явлений из области будущей профессиональной деятельности.
1.2	Задачами дисциплины является формирование умений и навыков самостоятельной разработки интеллектуальных программных систем на основе изучения фундаментальных аспектов проектирования: Декомпозиция систем (обработка данных, модель, инференс, обратная связь). Паттерны интеграции (Pipeline, Adapter) для соединения ML-моделей и традиционного ПО. Представление результатов (UI/UX для визуализации предсказаний и объяснений).
1.3	Документирование (фиксация архитектурных решений и спецификаций интерфейсов в соответствии со стандартами)

2. МЕСТО ДИСЦИПЛИНЫ (МОДУЛЯ) В СТРУКТУРЕ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ	
Цикл (раздел) ОП:	Б1.В
2.1	**Требования к предварительной подготовке обучающегося:**
2.1.1	Введение в машинное обучение
2.1.2	Основы систем искусственного интеллекта
2.1.3	Математическая статистика для систем ИИ
2.1.4	Управление проектами в сфере ИИ
2.1.5	Большие языковые модели и генеративный ИИ
2.1.6	Автоматизированное машинное обучение
2.2	**Дисциплины (модули) и практики, для которых освоение данной дисциплины (модуля) необходимо как предшествующее:**
2.2.1	Производственная практика
2.2.2	Преддипломная практика
2.2.3	Выполнение и защита выпускной квалификационной работы
2.2.4	Этика и правовые основы искусственного интеллекта
2.2.5	Системы компьютерного зрения
2.2.6	Рекомендательные системы и системы поддержки принятия решений

3. КОМПЕТЕНЦИИ ОБУЧАЮЩЕГОСЯ, ФОРМИРУЕМЫЕ В РЕЗУЛЬТАТЕ ОСВОЕНИЯ ДИСЦИПЛИНЫ (МОДУЛЯ)	
В результате освоения дисциплины (модуля) обучающийся должен	
3.1	**Знать:**
3.1.1	• основные подходы к проектированию ИПС;
3.1.2	• модели представления знаний (семантические сети, фреймы, продукционные системы);
3.1.3	• алгоритмы машинного обучения и их применение;
3.1.4	• технологии разработки интеллектуальных агентов;
3.1.5	• стандарты и инструменты для создания онтологий.
3.1.6	• факт, что ИИ-системы могут порождать этические проблемы (например, дискриминация, непрозрачность, манипуляция);
3.1.7	• основные этические принципы (отсутствие дискриминации, справедливость, человекоориентированность, ответственность, безопасность, прозрачность, автономия человека и т.д.);
3.1.8	• подходы к этической оценке ИИ (риск-ориентированный подход, бенчмаркинг, анкетирование на предмет оценки уровня этичности ИИ-системы со стороны разработчика или эксплуатанта системы);
3.1.9	• общую цель команды и механизмы участия в обсуждении задач, касающихся обработки данных, построения моделей или архитектурных решений;
3.1.10	• формы активного участия в обсуждении постановки задач с учётом требований заказчика и особенностей данных, изложения предложений по задачам и распределению ролей;
3.1.11	• подходы к определению и формализации проблемы предметной области, решение которой требует применения искусственного интеллекта
3.1.12	• принципы разработки архитектуры ИИ, подходы к проектированию и выбору подходящего стека технологий ИИ для RnD и промышленной разработки
3.1.13	• принципы, позволяющие осуществлять ведение (запуск и управление) проектов в области ИИ, в том числе подбор команды, планирование и контроль задач, оценка ресурсов

3.2	**Уметь:**
3.2.1	• выявлять ценностные конфликты в конкретных рабочих ситуациях (например, при разработке рекомендательной системы, системы распознавания изображений и т.п.);
3.2.2	• формулировать предложения, ориентируясь на техническую сторону задачи. Способен формулировать собственное понимание задач и уточнять его у других.
3.2.3	• принимать участие в обсуждении постановки задач с учётом требований заказчика и особенностей данных, формулирует предложения по задачам и распределению ролей, умеет вносить предложения по отбору релевантных метрик по проекту,
3.2.4	• согласовывать цели проекта с возможностями проектируемой ИИ-системы и ограничениями имеющейся инфраструктуры;
3.2.5	• анализировать требования к ИПС и формулировать техническое задание;
3.2.6	• проектировать архитектуру ИПС с учётом выбранной предметной области;
3.2.7	• разрабатывать базы знаний и онтологии;
3.2.8	• реализовывать алгоритмы машинного обучения на практике;
3.2.9	• оценивать эффективность и надёжность разработанных систем.
3.3	**Владеть:**
3.3.1	• навыками оценки потенциальных последствий внедрения ИИ-систем для различных групп пользователей;
3.3.2	• навыками совместной постановки целей проекта и формулирования задач в диалоге с командой, участия в приоритезации задач и распределении ответственности;
3.3.3	• навыками анализа требований (разрешения противоречий, приоритезации) в плане выбора технологий;
3.3.4	• навыками развития архитектуры системы ИИ на этапах жизненного цикла и ответа на вызовы масштабируемости и производительности;
3.3.5	• навыками управления продуктом с ИИ, выполнения продуктового дизайна цифровых сервисов с ИИ (в том числе концепция MVP, итеративное развитие продукта, Customer Development и другими методами проверки продуктовых гипотез), мониторинга ML-моделей, использования инструментов управления ЖЦ моделей машинного обучения (MLFlow, ClearML)
3.3.6	• навыками работы с инструментами для проектирования ИПС (Protégé, Weka, TensorFlow);
3.3.7	• методами тестирования и отладки интеллектуальных систем;
3.3.8	• приёмами документирования и сопровождения ИПС.

4. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

Код занятия	Наименование разделов и тем /вид занятия/	Семестр / Курс	Часов	Компетен-ции	Литература	Форма контроля
	Раздел 1. Архитектура интеллектуальных программных систем					
1.1	Введение в проектирование интеллектуальных систем /Тема/	7	0			
1.2	Жизненный цикл интеллектуальных систем: сбор и анализ требований, проектирование архитектуры, подготовка данных, интеграция моделей, тестирование, развёртывание и непрерывный мониторинг. Кейс-стади провала стартапа, использовавшего Jupyter Notebook в качестве основы для production-решения: обсуждаются проблемы с воспроизводимостью, отсутствием контроля версий и невозможностью горизонтального масштабирования. Связь с смежными дисциплинами иллюстрируется на примере перехода от курса «Математическая статистика для систем ИИ» к практическому проектированию: как статистические методы трансформируются в компоненты архитектуры, а не остаются изолированными алгоритмами /Лек/	7	2		Л1.3 Л1.2 Л1.5 Л1.1 Л1.4Л2.2 Л2.1Л3.1 Э1 Э2 Э3	Экзамен

1.3	Сбор требований проводится по шаблону: – Определение бизнес-целей (например, «снижение времени обработки заявок клиентов на 30%»). – Формулировка функциональных требований (об-работка текстовых отзывов, генерация отчетов) и нефункциональных (время ответа API ≤ 500 мс). Создание архитектурного эскиза в инструменте draw.io: – Выделение компонентов системы (модуль предобработки данных, модель машинного обучения, сервис обратной связи). – Фиксация границ ответственности: например, «модель не обрабатывает ошибки валидации данных, это задача отдельного компонента». – Пример итогового результата: схема с подписями, поясняющими взаимодействие компонентов через REST API. /Пр/	7	4		Л1.3 Л1.2 Л1.5 Л1.1 Л1.4Л2.2 Л2.1Л3.1 Э1 Э2 Э3	Экзамен
1.4	Архитектурные стили для интеллектуальных систем /Тема/	7	0			
1.5	Монолитная архитектура анализируется на примере MVP (минимально жизнеспособного продукта) с использованием языковой модели: преимущества – простота развертывания и отладки; риски – сложность масштабирования отдельных модулей. Микросервисы демонстрируются на кейсе системы рекомендаций для маркетплейса: сервис для обработки изображений товаров, отдельный модуль для коллаборативной фильтрации, сервис генерации персональных предложений. Обсуждаются накладные расходы на управление сетевыми взаимодействиями. Критерии выбора включают: – Операционные требования (например, необходимость горячего обновления модели без остановки системы). – Компетенции команды (опыт работы с контейнеризацией, распределёнными системами). /Лек/	7	2		Л1.3 Л1.2 Л1.5 Л1.1 Л1.4Л2.2 Л2.1Л3.1 Э1 Э2 Э3	Экзамен
1.6	Реализация заглушки ML-модели через FastAPI: – Создание эндпоинта POST /predict с валидацией входных данных с помощью Pydantic-моделей (например, проверка диапазона значений для числовых признаков). – Обработка ошибок: возврат кода 400 при некорректных данных с пояснением в теле ответа. – Тестирование через Swagger UI: отправка тестовых запросов с валидными и аномальными данными. /Лаб/	7	2		Л1.3 Л1.2 Л1.5 Л1.1 Л1.4Л2.2 Л2.1Л3.1 Э1 Э2 Э3	Экзамен
1.7	Паттерны проектирования для интеграции компонентов /Тема/	7	0			

1.8	Паттерн Pipeline детализируется на примере системы анализа медицинских изображений: – Этап 1: нормализация изображений (фиксация параметров в момент обучения). – Этап 2: детекция аномалий с помощью сверточной сети. – Этап 3: генерация объяснений для врача через методы интерпретируемости. Паттерн Adapter иллюстрируется кейсом интеграции модели, обученной в TensorFlow, в систему, построенную на PyTorch: адаптер преобразует форматы тензоров и управляет сессиями вычислений /Лек/	7	2		Л1.3 Л1.2 Л1.5 Л1.1 Л1.4Л2.2 Л2.1Л3.1 Э1 Э2 Э3	Экзамен
1.9	Проектирование OpenAPI-спецификации: – Студенты описывают контракты для всех эндпоинтов проекта (например, GET /health для проверки статуса сервиса, POST /feedback для сбора корректировок от пользователей). – Используют инструменты Swagger Editor для визуализации спецификации. Фиксация архитектурного решения (ADR): – По шаблону ГОСТ Р 7.0.97-2016 описывают проблему («необходимость поддержки нескольких форматов входных данных»), варианты решений (единый парсер vs. отдельные адаптеры), обоснование выбора /Пр/	7	4		Л1.3 Л1.2 Л1.5 Л1.1 Л1.4Л2.2 Л2.1Л3.1 Э1 Э2 Э3	Экзамен
	Раздел 2. Документирование проекта					
2.1	Документирование в соответствии с ГОСТ /Тема/	7	0			
2.2	Требования ГОСТ 19.101-77 к структуре: – Техническое задание (ТЗ) должно содержать разделы: «Основание для разработки», «Требования к надежности», «Условия эксплуатации». – Руководство программиста включает схемы алгоритмов, описание протоколов взаимодействия. Пример ADR по ГОСТ Р 7.0.97-2016: – Заголовок: «Выбор фреймворка для веб-интерфейса». – Раздел «Обоснование»: сравнение Streamlit (быстрая разработка) и Flask (гибкость). – Раздел «Риски»: необходимость обучения команды при выборе Flask. /Лек/	7	2		Л1.3 Л1.2 Л1.5 Л1.1 Л1.4Л2.2 Л2.1Л3.1 Э1 Э2 Э3	Экзамен
2.3	Проектирование OpenAPI-спецификации и ADR: – Описываются контракты для эндпоинтов (GET /health, POST /feedback). – Фиксируются архитектурное решение (например, выбор паттерна Pipeline) по шаблону ГОСТ Р 7.0.97-2016 с обоснованием /Пр/	7	4		Л1.3 Л1.2 Л1.5 Л1.1 Л1.4Л2.2 Л2.1Л3.1 Э1 Э2 Э3	Экзамен
	Раздел 3. Проектирование надёжных механизмов для взаимодействия с ML-моделями					
3.1	Стратегии инференса и обработки ошибок /Тема/	7	0			

3.2	Синхронный и асинхронный инференс сравниваются на примерах: – Чат-бот с синхронным ответом (максимальное время ожидания – 2 секунды). – Система обработки сканов документов с асинхронной очередью (пользователь получает уведомление по email после завершения). Паттерн Circuit Breaker объясняется на кейсе падения внешнего API для геокодирования: при превышении порога ошибок система временно переключается на кэшированные данные /Лек/	7	2		Л1.3 Л1.2 Л1.5 Л1.1 Л1.4Л2.2 Л2.1Л3.1 Э1 Э2 Э3	Экзамен
3.3	ETL-конвейер с валидацией данных: – Загрузка CSV-файла с клиентскими данными, проверка на пропуски и выбросы (например, возраст > 120 лет). – Эмуляция сбоя: искусственное добавление строк с некорректными типами данных. – Логирование ошибок с указанием номера строки и типа нарушения (используется библиотека logging с уровнем WARNING) /Лаб/	7	2		Л1.3 Л1.2 Л1.5 Л1.1 Л1.4Л2.2 Л2.1Л3.1 Э1 Э2 Э3	Экзамен
3.4	Проектирование надёжных data- конвейеров /Тема/	7	0			
3.5	Версионирование данных демонстрируется через инструмент DVC: – Фиксация хешей датасетов при обучении модели. – Откат к предыдущей версии при обнаружении проблем. Этические аспекты обсуждение на примере кредитного скоринга: – Анализ признаков, коррелирующих с гендером или этнической принадлежностью. – Техники смягчения смещений: перебалансировка классов, постобработка предсказаний /Лек/	7	2		Л1.3 Л1.2 Л1.5 Л1.1 Л1.4Л2.2 Л2.1Л3.1 Э1 Э2 Э3	Экзамен
3.6	Реализация модуля предобработки: – Создание pipeline с использованием scikit-learn: иммутабельность стадий (например, StandardScaler не изменяет исходные данные). – Интеграция валидации через Pydantic: проверка соответствия формата JSON-запроса ожидаемой схеме. – Обработка edge cases: данные с пустыми полями, экстремальные значения (например, отрицательный доход). /Пр/	7	4		Л1.3 Л1.2 Л1.5 Л1.1 Л1.4Л2.2 Л2.1Л3.1 Э1 Э2 Э3	Экзамен
	Раздел 4. Тестирование интеллектуальной ПС					
4.1	Тестирование интеграции компонентов /Тема/	7	0			
4.2	Контрактное тестирование иллюстрируется примером: – Сервис предсказания должен возвращать поле «confidence» в диапазоне [0, 1] при успешном запросе. – При отсутствии этого поля тест падает, даже если точность модели высока. Стратегии логирования: – Разделение логов по уровням: DEBUG (отладка этапов обработки), ERROR (критические сбои с трейсбэком), AUDIT (запись входных данных для аудита) /Лек/	7	2		Л1.3 Л1.2 Л1.5 Л1.1 Л1.4Л2.2 Л2.1Л3.1 Э1 Э2 Э3	Экзамен

4.3	Тестирование API через Postman: – Создание коллекции запросов: валидные данные, пустой JSON, массив вместо числа. – Проверка HTTP-статусов: 200 для успешных запросов, 400 для некорректных данных, 503 при недоступности модели. – Автоматизация через Newman: запуск тестов в конвейере CI/CD. /Лаб/	7	2		Л1.3 Л1.2 Л1.5 Л1.1 Л1.4Л2.2 Л2.1Л3.1 Э1 Э2 Э3	Экзамен
	Раздел 5. Развёртывание интеллектуальных ПС					
5.1	Контейнеризация и основы развёртывания /Тема/	7	0			
5.2	Docker для ML-проектов: – Структура Dockerfile: кэширование зависимостей через слои, оптимизация размера образа. – Пример: запуск Streamlit-приложения внутри контейнера с пробросом порта 8501. Выбор облачной платформы: – Serverless (AWS Lambda) для эпизодических задач с низкой нагрузкой. – Виртуальные машины (Yandex Compute Cloud) для моделей, требующих GPU /Лек/	7	2		Л1.3 Л1.2 Л1.5 Л1.1 Л1.4Л2.2 Л2.1Л3.1 Э1 Э2 Э3	Экзамен
5.3	Реализация паттерна Pipeline в контейнере: – Этапы: загрузка данных из CSV → предобработка (удаление выбросов) → инференс через заглушку. – Интеграция обработки ошибок: при падении одного этапа система возвращает структурированное сообщение с указанием проблемного шага. – Логирование в stdout с пометкой этапа выполнения. /Лаб/	7	2		Л1.3 Л1.2 Л1.5 Л1.1 Л1.4Л2.2 Л2.1Л3.1 Э1 Э2 Э3	Экзамен
5.4	Кросс-функциональные требования /Тема/	7	0			
5.5	Производительность: – Компромисс latency vs. throughput: для мобильного приложения важна минимальная задержка, для batch-обработки — пропускная способность. Дрейф данных: – Алгоритм детекции: еженедельное сравнение распределения входных данных с помощью теста Колмогорова-Смирнова. – Автоматическая отправка уведомления администратору при p-value < 0.05. Объяснимость как требование: – Сценарий для банковской системы: модель с точностью 85% и интерпретируемыми правилами предпочтительнее «черного ящика» с точностью 87%. /Лек/	7	2		Л1.3 Л1.2 Л1.5 Л1.1 Л1.4Л2.2 Л2.1Л3.1 Э1 Э2 Э3	Экзамен
5.6	Настройка мониторинга: – Логирование статистик данных (среднее, дисперсия) на этапе обучения и инференса. – Простой скрипт для сравнения распределений: генерация алерта при обнаружении значимых различий. /Лаб/	7	2		Л1.3 Л1.2 Л1.5 Л1.1 Л1.4Л2.2 Л2.1Л3.1 Э1 Э2 Э3	Экзамен
	Раздел 6. Этические нормы и архитектурные решения					
6.1	Этические принципы и Human-AI collaboration /Тема/	7	0			

6.2	Human-in-the-loop проектируется как обязательный компонент: – Схема сбора корректировок: например, врач подтверждает или отклоняет диагноз, предложенный моделью. – Механизм переобучения: автоматическая отправка одобренных данных в датасет для дообучения. Соответствие GDPR: – Анонимизация на этапе проектирования: хеширование идентификаторов пользователей перед сохранением в логах. – Право на удаление данных: специальный эндпоинт DELETE /user_data.. /Лек/	7	2		Л1.3 Л1.2 Л1.5 Л1.1 Л1.4Л2.2 Л2.1Л3.1 Э1 Э2 Э3	Экзамен
6.3	Разработка интерфейса с акцентом на объяснимость: – интеграция LIME для генерации текстовых пояснений («Предсказание основано на высоком уровне холестерина и возрасте»). – Визуализация зон внимания модели на изображениях (Grad-CAM для медицинских снимков). /Пр/	7	4		Л1.3 Л1.2 Л1.5 Л1.1 Л1.4Л2.2 Л2.1Л3.1 Э1 Э2 Э3	Экзамен
	Раздел 7. Безопасность и защита данных					
7.1	Проактивное выявление уязвимостей на этапе проектирования /Тема/	7	0			
7.2	Защита от adversarial-атак: – Валидация входных данных: ограничение длины текстовых полей, фильтрация специальных символов. – Пример атаки: подстановка изображения с шумом, приводящего к ошибке классификации. Анонимизация для GDPR: – Маскирование персональных данных: замена ФИО на хеш-идентификаторы. –Логирование только агрегированных метрик (количество запросов в минуту), а не сырых данных. /Лек/	7	2		Л1.3 Л1.2 Л1.5 Л1.1 Л1.4Л2.2 Л2.1Л3.1 Э1 Э2 Э3	Экзамен
7.3	Валидация и анонимизация: – Реализация middleware для FastAPI: проверка размера JSON-тела запроса (ограничение 1 МБ). – Генерация ошибки 413 (Payload Too Large) при превышении лимита. – Интеграция функции анонимизации email-адресов в логах (замена domain.com на ***) /Лаб/	7	2		Л1.3 Л1.2 Л1.5 Л1.1 Л1.4Л2.2 Л2.1Л3.1 Э1 Э2 Э3	Экзамен
	Раздел 8. Визуальное представление результатов					
8.1	Проектирование интерфейсов для пользователей и администраторов /Тема/	7	0			
8.2	UI/UX для аналитиков включает: – Интерактивные графики в Plotly: возможность фильтрации данных по временным периодам. – Визуализацию SHAP-значений для объяснения решений модели (например, влияние «возраста» на вероятность отказа в кредите). Системы алертов для администраторов: – Уведомления о превышении latency в инференсе. – Панель мониторинга с индикаторами загрузки CPU/GPU. /Лек/	7	2		Л1.3 Л1.2 Л1.5 Л1.1 Л1.4Л2.2 Л2.1Л3.1 Э1 Э2 Э3	Экзамен

8.3	Интеграция ML-модели в веб-приложение: – Использование Streamlit для создания дашборда: загрузка файла с данными, отображение предсказаний, интерактивные виджеты (слайдеры для изменения порога классификации). – Добавление элементов Human-in-the-loop: кнопка «Сообщить об ошибке» с отправкой комментария в базу данных /Пр/	7	4		Л1.3 Л1.2 Л1.5 Л1.1 Л1.4Л2.2 Л2.1Л3.1 Э1 Э2 Э3	Экзамен
	Раздел 9. Гибридные архитектуры					
9.1	Преимущества комбинирования методов машинного обучения, экспертных правил и традиционных алгоритмов /Тема/	7	0			
9.2	Чат-бот с гибридной логикой: – Языковая модель генерирует черновик ответа. – Экспертные правила проверяют фактологическую корректность (например, соответствие дозировок лекарств стандартам). Гибридные рекомендации: – Коллаборативная фильтрация для персонализации. – Правила бизнес-логики: исключение товаров с отрицательными отзывами из выдачи /Лек/	7	2		Л1.3 Л1.2 Л1.5 Л1.1 Л1.4Л2.2 Л2.1Л3.1 Э1 Э2 Э3	Экзамен
9.3	Создание интерактивного дашборда: – Визуализация распределения предсказаний с помощью гистограммы. – Интеграция SHAP для детального анализа: при наведении на график отображается вклад каждого признака. – Тестирование на реальных данных: загрузка набора отзывов о ресторанах и анализ тональности /Лаб/	7	2		Л1.3 Л1.2 Л1.5 Л1.1 Л1.4Л2.2 Л2.1Л3.1 Э1 Э2 Э3	Экзамен
	Раздел 10. Декомпозиция промышленных решений					
10.1	Case study – анализ реальных архитектур /Тема/	7	0			
10.2	ChatGPT-подобная система: – Компоненты: LLM-ядро, модуль инструментов (калькулятор, поиск в интернете), система модерации. – Ограничения: стоимость инференса (кэширование частых запросов), latency (асинхронная генерация длинных ответов). Отказоустойчивость: – Использование очередей сообщений (RabbitMQ) для буферизации запросов при пиковой нагрузке. – Fallback на упрощённую модель при превышении времени ответа. /Лек/	7	2		Л1.3 Л1.2 Л1.5 Л1.1 Л1.4Л2.2 Л2.1Л3.1 Э1 Э2 Э3	Экзамен
10.3	Подготовка документации: – Создание архитектурных диаграмм в draw.io с подписями, соответствующими ГОСТ 19.101-77. – Написание инструкции по запуску: требования к Python-версии, установка зависимостей через requirements.txt. – Финальное тестирование демо-версии: проверка работоспособности на чистом окружении (Docker-контейнер). /Пр/	7	4		Л1.3 Л1.2 Л1.5 Л1.1 Л1.4Л2.2 Л2.1Л3.1 Э1 Э2 Э3	Экзамен
	Раздел 11. Представление знаний в интеллектуальных ПС					

11.1	Protégé – инструмент представления онтологий /Тема/	7	0			
11.2	Знакомство с интерфейсом Protégé и создание пер-вой онтологии – Запуск Protégé, создание нового проекта (OWL ontology). – Настройка IRI (имени онтологии). – Знакомство с основными вкладками: Entities, Classes, Object Proper-ties, Data Properties, Individuals. – Создание простой онтологии (например, «Домашние животные»: классы Животное, Кошка, Собака; свойства имеетКличку, имеетПороду, имеетВозраст). Построение иерархии классов – Создание классов и подклассов (наследование через rdfs:subClassOf). – Использование Thing как корневого класса. – Пример: онтология «Университет» (классы Человек → Студент, Преподаватель; Курс → Обязательный, Факультативный). Определение свойств: Object Properties и Data Properties – Задание доменов (domain) и диапазонов (range) для свойств. – Различие между объектными (связи между экземплярами) и данными (литералы) свойствами. – Пример: для класса Студент создать свойства изучаетКурс (Object Property) и имеетСреднийБалл (Data Property, тип xsd:decimal). /Лек/	7	2		Л1.3 Л1.2 Л1.5 Л1.1 Л1.4Л2.2 Л2.1Л3.1 Э1 Э2 Э3	Экзамен
11.3	Начальные манипуляции с онтологиями /Тема/	7	0			
11.4	Создание и заполнение экземпляров (individuals) – Добавление экземпляров классов. – Задание значений свойств для экземпляров. – Пример: создать экземпляры Иван_Иванов (класс Студент), Математика_101 (класс Курс), связать их через свойство изучаетКурс. Работа с ограничениями (cardinality, functional, inverse) – Установка ограничений на свойства: functional (единственное значение), inverse (обратное свойство), minCardinality, maxCardinality. – Пример: свойство имеетПаспорта у класса Человек – functional (один паспорт); имеетДетей – minCardinality 0. /Лек/	7	2		Л1.3 Л1.2 Л1.5 Л1.1 Л1.4Л2.2 Л2.1Л3.1 Э1 Э2 Э3	Экзамен

11.5	Создание defined classes (классов с условиями) – Использование Description Logic (DL) для определения классов через логические выражения (например, Студент & (изучаетКурс только Математика)). – Проверка вывода с помощью Reasoner (HermiT, Pellet). Выполнение SPARQL-запросов – Написание запросов для выборки данных из онтологии. – Примеры: «Найти всех студентов со средним баллом > 4.0». «Найти курсы, которые изучают хотя бы 5 студентов». – Использование вкладки SPARQL Query в Protégé. Работа с правилами (SWRL, SQWRL) – Изучение синтаксиса SWRL и SQWRL – Создание правил для автоматического вывода (например, «Если студент имеет средний балл > 4.5, то он получает стипендию»). – Создание правил для проверки информации в онтологии (например, выявление противоречий в данных о товарах, выявление сомнительных данных) – Тестирование правил и анализ результатов. /Пр/	7	4		Л1.3 Л1.2 Л1.5 Л1.1 Л1.4Л2.2 Л2.1Л3.1 Э1 Э2 Э3	Экзамен
11.6	Проверка онтологии с помощью Reasoner – Активация решателя (HermiT/Pellet). – Анализ согласованности онтологии (Check Consistency). – Выявление и исправление логических противоречий. – Пример: обнаружить конфликт, если экземпляр одновременно при-надлежит взаимоисключающим классам. /Лаб/	7	2		Л1.3 Л1.2 Л1.5 Л1.1 Л1.4Л2.2 Л2.1Л3.1 Э1 Э2 Э3	Экзамен
	Раздел 12. Промежуточная аттестация					
12.1	Экзамен и КП /Тема/	7	0			
12.2	Приём экзамена /ИКР/	7	0,35		Л1.3 Л1.2 Л1.5 Л1.1 Л1.4Л2.2 Л2.1Л3.1 Э1 Э2 Э3	
12.3	Защита КП /ИКР/	7	0,3		Л1.3 Л1.2 Л1.5 Л1.1 Л1.4Л2.2 Л2.1Л3.1 Э1 Э2 Э3	
12.4	Подготовка к экзамену /Экзамен/	7	53,35		Л1.3 Л1.2 Л1.5 Л1.1 Л1.4Л2.2 Л2.1Л3.1 Э1 Э2 Э3	
12.5	Подготовка КП /КПКР/	7	15,7		Л1.3 Л1.2 Л1.5 Л1.1 Л1.4Л2.2 Л2.1Л3.1 Э1 Э2 Э3	
12.6	Консультация /Кнс/	7	2		Л1.3 Л1.2 Л1.5 Л1.1 Л1.4Л2.2 Л2.1Л3.1 Э1 Э2 Э3	

	Раздел 13. Самостоятельная работа					
13.1	Самостоятельная работа /Тема/	7	0			
13.2	Изучение конспекта лекции и рекомендованной литературы, решение типовых задач по теме. Подготовка к экзамену /Ср/	7	28,3		Л1.3 Л1.2 Л1.5 Л1.1 Л1.4Л2.2 Л2.1Л3.1 Э1 Э2 Э3	Экзамен

5. ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ПО ДИСЦИПЛИНЕ (МОДУЛЮ)

Фонд оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине представлен в Приложении к рабочей программе дисциплины (см. документ «Оценочные материалы о дисциплине «Проектирование интеллектуальных программных систем»»).

6. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

6.1. Рекомендуемая литература

6.1.1. Основная литература

№	Авторы, составители	Заглавие	Издательство, год	Количество/ название ЭБС
Л1.1	Сорокин А. Б., Железняк Л. М.	Проектирование интеллектуальных систем	Москва: РТУ МИРЭА, 2023, 104 с.	978-5-7339-1974-4, https://e.lanbook.com/book/386195
Л1.2	Белов В.В., Чистякова В.И.	Проектирование информационных систем : учеб.	М.: КУРС, 2018, 395с.	978-5-906923-53-0, 1
Л1.3	Гвоздева Т. В., Баллод Б. А.	Проектирование информационных систем. Стандартизация	Санкт-Петербург: Лань, 2021, 252 с.	978-5-8114-7963-4, https://e.lanbook.com/book/169810
Л1.4	Огарок А. Л.	Проектирование интеллектуальных информационных систем : учебное пособие	Москва: РТУ МИРЭА, 2024, 104 с.	978-5-7339-2320-8, https://e.lanbook.com/book/448943
Л1.5	Гринченко Н.Н., Громов А.Ю., Хизриева Н.И.	Проектирование информационных систем: учебник : Учебник	Рязань: КУРС, 2023,	, https://elib.rsreu.ru/ebs/download/3596

6.1.2. Дополнительная литература

№	Авторы, составители	Заглавие	Издательство, год	Количество/ название ЭБС
Л2.1	Гвоздева Т. В., Баллод Б. А.	Проектирование информационных систем. Методы и средства структурно-функционального проектирования. Практикум : учебное пособие для спо	Санкт-Петербург: Лань, 2024, 148 с.	978-5-507-47555-1, https://e.lanbook.com/book/388976
Л2.2	Корячко В.П., Бакулева М.А., Орешков В.И.	Интеллектуальные системы и нечеткая логика: учебник : Учебник	Рязань: КУРС, 2023,	, https://elib.rsreu.ru/ebs/download/3615

6.1.3. Методические разработки

№	Авторы, составители	Заглавие	Издательство, год	Количество/ название ЭБС
Л3.1	Белов В.В., Чистякова В.И.	Проектирование программно-информационных систем: метод. указ. по оформлению пояснительной записки к курсовому проекту по дисциплинам «Проектирование информационных систем», «Проектирование программных систем» : Методические указания	Рязань: , 2020,	, https://elib.rsreu.ru/ebs/download/2810

6.2. Перечень ресурсов информационно-телекоммуникационной сети "Интернет"

Э1	Библиотека ресурсов РГРТУ https://elib.rsreu.ru/ebs
Э2	Электронно-библиотечная система IPRbooks https://www.iprbookshop.ru/
Э3	Электронно-библиотечная система издательства «Лань» http://e.lanbook.com

6.3 Перечень программного обеспечения и информационных справочных систем

6.3.1 Перечень лицензионного и свободно распространяемого программного обеспечения, в том числе отечественного производства

Наименование	Описание
Операционная система Windows	Коммерческая лицензия
Kaspersky Endpoint Security	Коммерческая лицензия
Adobe Acrobat Reader	Свободное ПО
LibreOffice	Свободное ПО
Интерпретатор Python	Свободное ПО
Python	Свободно распространяемое программное обеспечение под лицензиями
Anaconda3	Свободно распространяемое программное обеспечение под лицензиями
Chrome	Свободное ПО
PyCharm	Свободное ПО

6.3.2 Перечень информационных справочных систем

6.3.2.1	Справочная правовая система «КонсультантПлюс» (договор об информационной поддержке №1342/455-100 от 28.10.2011 г.)
6.3.2.2	Система КонсультантПлюс http://www.consultant.ru
6.3.2.3	Информационно-правовой портал ГАРАНТ.РУ http://www.garant.ru

7. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

1	206-1 учебно-административный корпус. Учебная аудитория для проведения занятий лекционного и семинарского типа, групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации 42 мест, 1 ПК: ЦП: Intel i7 6700 ОЗУ: 8 Гб ПЗУ: 120Gb ssd + 500 Гб (1 шт. преподавательский) Телевизор: PHILIPS U7PEL4606H/60 документ-камера: AVER Media POB3 (AverVision 330) документ-камера: AverVisionF33 POE7D;
2	206-2 учебно-административный корпус. Аудитория для самостоятельной работы 18 мест, Телевизор PHILIPS 46PFL3208T/60; 18 ПК с возможностью подключения к сети «Интернет» и обеспечением доступа в электронную информационно-образовательную среду: ЦП: AMD a10 7800 ОЗУ: 8 Gb ПЗУ: 500-1000 Gb (18 шт.)
3	206-2 учебно-административный корпус. Учебная аудитория для проведения занятий лекционного и семинарского типа, групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации 18 мест, Телевизор PHILIPS 46PFL3208T/60; 18 ПК с возможностью подключения к сети «Интернет» и обеспечением доступа в электронную информационно-образовательную среду: ЦП: AMD a10 7800 ОЗУ: 8 Gb ПЗУ: 500-1000 Gb (18 шт.)

4	206-3 учебно-административный корпус. Учебная аудитория для самостоятельной работы Проектор: InFocus LP640 31 ПК с возможностью подключения к сети «Интернет» и обеспечением доступа в электронную информационно-образовательную среду: ЦП: Intel i7 6700 ОЗУ: 8 Гб ПЗУ: 120Gb ssd + 500 Гб (31 шт.)
5	206-3 учебно-административный корпус. Учебная аудитория для проведения практический занятий, лабораторных работ Проектор: InFocus LP640 31 ПК с возможностью подключения к сети «Интернет» и обеспечением доступа в электронную информационно-образовательную среду: ЦП: Intel i7 6700 ОЗУ: 8 Гб ПЗУ: 120Gb ssd + 500 Гб (31 шт.))
6	206-4 учебно-административный корпус. Учебная аудитория для проведения практический занятий, лабораторных работ 30 ПК с возможностью подключения к сети «Интернет» и обеспечением доступа в электронную информационно-образовательную среду: ЦП: Intel i5-12400 ОЗУ: 16 Гб ПЗУ: ssd 512 Гб (18 шт. моноблок) ЦП: Intel i7 6700 ОЗУ: 8 Гб ПЗУ: 120Gb ssd + 500 Гб (12 шт.)
7	206-4 учебно-административный корпус. Учебная аудитория для самостоятельной работы 30 ПК с возможностью подключения к сети «Интернет» и обеспечением доступа в электронную информационно-образовательную среду: ЦП: Intel i5-12400 ОЗУ: 16 Гб ПЗУ: ssd 512 Гб (18 шт. моноблок) ЦП: Intel i7 6700 ОЗУ: 8 Гб ПЗУ: 120Gb ssd + 500 Гб (12 шт.)
8	110 учебно-административный корпус. Учебная аудитория для проведения занятий лекционного и семинарского типа, групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации 20 мест (+20 мест за компьютерными столами) Проектор: HITACHI CP-X400 3LCD 21 ПК с возможностью подключения к сети «Интернет» и обеспечением доступа в электронную информационно-образовательную среду: ЦП: Intel i5-12400 ОЗУ: 16 Гб ПЗУ: ssd 512 Гб (20 шт. моноблок + 1шт. преподавательский)
9	110 учебно-административный корпус. Аудитория для самостоятельной работы 20 мест (+20 мест за компьютерными столами) Проектор: HITACHI CP-X400 3LCD 21 ПК с возможностью подключения к сети «Интернет» и обеспечением доступа в электронную информационно-образовательную среду: ЦП: Intel i5-12400 ОЗУ: 16 Гб ПЗУ: ssd 512 Гб (20 шт. моноблок + 1шт. преподавательский)

8. МЕТОДИЧЕСКИЕ МАТЕРИАЛЫ ПО ДИСЦИПЛИНЕ (МОДУЛЮ)

Методическое обеспечение дисциплины приведено в приложении к рабочей программе дисциплины (см. документ Методические материалы по дисциплине «Проектирование интеллектуальных программных систем»).

Оператор ЭДО ООО "Компания "Тензор"

ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ

ПОДПИСАНО ЗАВЕДУЮЩИМ КАФЕДРЫ — **ФГБОУ ВО "РГРТУ", РГРТУ,** Овечкин Геннадий Владимирович, Заведующий кафедрой ВПМ — **03.11.26** 16:20 (MSK) — Простая подпись

ПОДПИСАНО ЗАВЕДУЮЩИМ ВЫПУСКАЮЩЕЙ КАФЕДРЫ — **ФГБОУ ВО "РГРТУ", РГРТУ,** Овечкин Геннадий Владимирович, Заведующий кафедрой ВПМ — **03.11.26** 16:20 (MSK) — Простая подпись