

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Тольяттинский государственный университет»

Б1.В.ДВ.04.01

(индекс дисциплины)

**РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ**

**Разработка веб-сервисов с интеграцией искусственного интеллекта**

(наименование дисциплины)

по направлению подготовки  
09.03.04 Программная инженерия

направленность (профиль)  
Программная инженерия с применением ИИ-технологий

Форма обучения: очная

Год набора: 2026

Общая трудоемкость: 5 ЗЕ

**Распределение часов дисциплины по семестрам**

Семестр	7	Итого
	экзамен	
Вид занятий	Форма контроля	
Лекции	32	32
Лабораторные		
Практические	32	32
Руководство: курсовые работы (проекты) / РГР		
Промежуточная аттестация	0,35	0,35
Контактная работа	64,35	64,35
Самостоятельная работа	80	80
Контроль	35,65	35,65
<b>Итого</b>	<b>180</b>	<b>180</b>

Рабочую программу составил(и):

Доцент института цифровых технологий, канд.пед.наук. Ерофеева Е.А.

*(должность, ученое звание, степень, Фамилия И.О.)*

Рецензирование рабочей программы дисциплины:



Отсутствует



Рецензент

*(должность, ученое звание, степень, Фамилия И.О.)*

Рабочая программа дисциплины составлена на основании ФГОС ВО и учебного плана направления подготовки

09.03.04 Программная инженерия

**Срок действия рабочей программы дисциплины до «31» августа 2030 г.**

УТВЕРЖДЕНО

На заседании института цифровых технологий

(протокол заседания № 1 от «05» сентября 2025 г.).

## 1. Цель освоения дисциплины

Цель освоения дисциплины – формирование у обучающихся профессиональных компетенций в области проектирования, разработки и интеграции веб-сервисов, использующих методы искусственного интеллекта; освоение технологий создания серверных и клиентских компонентов, взаимодействия с моделями машинного обучения, обеспечения корректности, надёжности и производительности веб-решений с ИИ-функциональностью.

## 2. Место дисциплины в структуре ОПОП ВО

Дисциплины и практики, на освоении которых базируется данная дисциплина: Тестирование и верификация систем с ИИ-компонентами, Системы искусственного интеллекта, Архитектура информационных систем и методы интеграции.

Дисциплины и практики, для которых освоение данной дисциплины необходимо как предшествующее: Производственная практика (преддипломная практика), Выполнение и защита выпускной квалификационной работы.

## 3. Планируемые результаты обучения

Формируемые и контролируемые компетенции (код и наименование)	Индикаторы достижения компетенций (код и наименование)	Планируемые результаты обучения
ПК-7 Способен разрабатывать сервисы на основе аналитики больших данных	ПК-7.1. Понимает принципы разработки сервисов, приемы аналитики больших данных	Знать: принципы разработки сервисов; методы аналитики (дескриптивная, диагностическая, предиктивная, предписывающая аналитика). Уметь: проектировать сервисно-ориентированную архитектуру. Владеть: навыками проектирования API для аналитических сервисов.
	ПК-7.2. Умеет разрабатывать сервисы на основе аналитики больших данных	Знать: как работать с потоковыми данными. Уметь: разрабатывать микросервисы, которые предоставляют аналитические функции (например, рекомендации, прогнозирование). Владеть: практическими навыками создания веб-сервисов с интеграцией аналитических библиотек.

#### 4. Структура и содержание дисциплины

Модуль (раздел)	Вид учебной работы	Наименование тем занятий (учебной работы)	Семестр	Объем, ч.	Баллы	Интерактив, ч.	Формы текущего контроля (наименование оценочного средства)
Модуль 1. Архитектура веб-сервисов и основы интеграции ИИ	Лек 1	Тема 1. Архитектура современного веб-приложения: клиент, сервер, API, микросервисы	7	2	–	–	
	Лек 2	Тема 2. Принципы проектирования API. REST, gRPC, веб-хуки	7	2	–	–	
	Лек 3	Тема 3. Инфраструктура для веб-сервисов: контейнеризация, Docker, оркестраторы	7	2	–	–	
	Лек 4	Тема 4. Способы интеграции ИИ-моделей в веб-сервисы: прямой вызов, сервис-модель, ML-node	7	2	–	–	
	Пр 1	ПР1. Проектирование структуры веб-сервиса: модули, конфигурация, маршруты	7	2	–	–	Отчет по практической работе 1
	Пр 2	ПР2. Создание минимального REST-сервиса (FastAPI / Flask / Spring)	7	2	–	–	Отчет по практической работе 2
	Пр 3	ПР3. Работа с маршрутизацией и обработкой запросов	7	2	–	–	Отчет по практической работе 3
	Пр 4	ПР4. Создание контейнера Docker для веб-сервиса	7	2	–	–	Отчет по практической работе 4
Модуль 2. Интеграция	Лек 5	Тема 5. Модели ИИ как сервисы: REST-модели, RPC-модели, модельные контейнеры	7	2	–	–	

Модуль (раздел)	Вид учебной работы	Наименование тем занятий (учебной работы)	Семестр	Объем, ч.	Баллы	Интерактив, ч.	Формы текущего контроля (наименование оценочного средства)
моделей ИИ в веб-приложения	Лек 6	Тема 6. Предобработка данных и форматирование входов/выходов моделей	7	2	–	–	
	Лек 7	Тема 7. Схемы взаимодействия backend ↔ модель (Batch, realtime, streaming)	7	2	–	–	
	Лек 8	Тема 8. Паттерны интеграции ИИ- модуля в микросервисную архитектуру	7	2	–	–	
	Пр 5	ПР5. Подключение готовой ML-модели к веб-API	7	2	–	–	Отчет по практической работе 5
	Пр 6	ПР6. Реализация эндпоинта /predict с использованием модели	7	2	–	–	Отчет по практической работе 6
	Пр 7	ПР7. Предобработка данных: валидация, нормализация, сериализация	7	2	–	–	Отчет по практической работе 7
	Пр 8	ПР8. Интеграция модели в отдельный контейнер ML-node	7	2	–	–	Отчет по практической работе 8
Модуль 3. Хранение данных, очереди, асинхронность	Лек 9	Тема 9. Базы данных для веб-сервисов: реляционные и NoSQL	7	2	–	–	
	Лек 10	Тема 10. Асинхронные веб-сервисы, event-loop, background-tasks	7	2	–	–	
	Лек 11	Тема 11. Системы очередей и брокеры сообщений (RabbitMQ, Kafka)	7	2	–	–	
	Лек 12	Тема 12. Логирование, мониторинг, метрики и трассировка	7	2	–	–	

Модуль (раздел)	Вид учебной работы	Наименование тем занятий (учебной работы)	Семестр	Объем, ч.	Баллы	Интерактив, ч.	Формы текущего контроля (наименование оценочного средства)
	Пр 9	ПР9. Подключение базы данных, создание схемы и CRUD-операций	7	2	–	–	Отчет по практической работе 9
	Пр 10	ПР10. Асинхронный API: реализация фоновых задач	7	2	–	–	Отчет по практической работе 10
	Пр 11	ПР11. Связка сервис → очередь → модель	7	2	–	–	Отчет по практической работе 11
	Пр 12	ПР12. Логирование и мониторинг веб-сервиса	7	2	–	–	Отчет по практической работе 12
Модуль 4. Тестирование, безопасность и деплой веб-сервисов с ИИ	Лек 13	Тема 13. Тестирование API и ИИ-эндпоинтов: unit, integration, e2e	7	2	–	–	
	Лек 14	Тема 14. Проверка корректности моделей в составе сервиса	7	2	–	–	
	Лек 15	Тема 15. Безопасность веб-сервисов с ИИ, защита API, rate limiting	7	2	–	–	
	Лек 16	Тема 16. Деплой, CI/CD, автоматизация обновлений модели	7	2	–	–	
	Пр 13	ПР13. Тестирование API (Postman/pytest)	7	2	–	–	Отчет по практической работе 13
	Пр 14	ПР14. Проверка корректности модели в продакшен-сценариях	7	2	–	–	Отчет по практической работе 14
	Пр 15	ПР15. Реализация базовых механизмов безопасности (JWT, токены)	7	2	–	–	Отчет по практической работе 15
	Пр 16	ПР16. Деплой веб-сервиса и модели (Docker Compose / Kubernetes)	7	2	–	–	Отчет по практической работе 16

Модуль (раздел)	Вид учебной работы	Наименование тем занятий (учебной работы)	Семестр	Объем, ч.	Баллы	Интерактив, ч.	Формы текущего контроля (наименование оценочного средства)
	СР	Самостоятельное изучение методических рекомендаций при подготовке к практическим работам.	7	80	—	—	
	ПА	Промежуточная аттестация	7	0,35	—	—	
	Контроль	Экзамен	7	36,65	—	—	Вопросы к экзамену
Итого:				180			

## 5. Образовательные технологии

При изучении дисциплины используются следующие образовательные технологии:

- **технологии традиционного обучения** в форме лекций, практических работ и самостоятельной работы обучающихся;
- **технология проектного обучения**, предусматривающая разработку фрагментов веб-сервисов, интеграцию ИИ-модулей, защиту результатов практических работ и демонстрацию работоспособности созданных решений;
- **практико-ориентированное обучение**, основанное на выполнении реальных задач по разработке API, интеграции моделей машинного обучения, работе с контейнерами, очередями и базами данных.

Технологии традиционного обучения включают объяснительно-иллюстративный формат, применяемый во всех модулях курса при изучении архитектуры веб-сервисов, принципов интеграции ИИ и моделей взаимодействия компонентов.

Организация учебного процесса предполагает активную позицию обучающихся при выполнении практических заданий: разработку серверных модулей, реализацию ИИ-функциональности в составе веб-сервисов, настройку инфраструктуры, анализ корректности интеграции и представление полученных результатов.

На практических занятиях обучающиеся:

- демонстрируют работоспособность разработанных веб-сервисов и ИИ-модулей;
- обосновывают принятые архитектурные решения;
- обсуждают возникающие технические проблемы интеграции и пути их решения;
- проводят мини-защиту выполненной работы.

Итоговая аттестация проводится в форме **экзамена**, включающего оценку теоретических знаний и практических навыков разработки веб-сервисов с интеграцией искусственного интеллекта.

## 6. Методические указания по освоению дисциплины

### 6.1. Рекомендации по подготовке к практическим занятиям

Обучающимся следует:

- при подготовке к практическим занятиям использовать не только материалы лекций и учебную литературу, но и официальную документацию по используемым технологиям (фреймворки для веб-разработки, библиотеки ИИ, инструменты контейнеризации, системы управления БД);
- заранее познакомиться с примерами исходного кода, структурой API и форматами данных, применяемыми в задании;
- в начале занятия уточнить у преподавателя моменты, вызвавшие затруднения, чтобы избежать ошибок при реализации серверной логики или интеграции моделей ИИ;
- на занятии доводить каждое задание до работоспособного состояния, демонстрировать корректную работу разработанного сервиса, уметь пояснить принятые архитектурные решения.

Для того чтобы практические занятия приносили максимальную пользу, необходимо помнить, что выполнение заданий тесно связано с лекционным материалом: архитектурой веб-сервисов, методами интеграции моделей искусственного интеллекта, принципами обработки данных, методами деплоя и обеспечением работоспособности сервиса. Только после усвоения этих основ будет возможна качественная реализация серверных модулей, маршрутов API и ИИ-функциональности в практических работах.

При самостоятельной работе над заданиями необходимо:

- обосновывать каждый этап разработки (выбор архитектуры, схемы API, способ интеграции модели, структуру хранения данных);



- анализировать несколько вариантов решения (например, прямой вызов модели или вынесение её в отдельный сервис) и выбирать наиболее рациональный с точки зрения производительности, устойчивости и удобства поддержки;
- предварительно составлять план разработки: структуру каталогов проекта, точки входа, взаимодействующие компоненты, последовательность выполнения задач;
- сопровождать решения комментариями, диаграммами, схемами вызовов или архитектурными эскизами, если требуется представить логику работы сервиса.

Каждая практическая работа должна завершаться получением функционирующего результата — корректно работающего веб-сервиса или его модуля, готового к интеграции, тестированию или деплою. Полученный результат обучающийся должен проверить самостоятельно: протестировать основные маршруты API, корректность взаимодействия модуля ИИ, поведение сервиса при ошибочных данных.

Развитие устойчивых навыков разработки достигается регулярной самостоятельной практикой, анализом примеров, экспериментированием с архитектурой и изучением профессиональной документации.

## **6.2. Рекомендации по подготовке к экзамену**

Подготовка к экзамену направлена на систематизацию, закрепление и обобщение теоретических знаний и практических навыков, полученных в ходе изучения дисциплины. Экзамен проверяет сформированность элементов компетенций, связанных с разработкой веб-сервисов, проектированием архитектуры, интеграцией моделей искусственного интеллекта и обеспечением работоспособности сервисов.

Для успешной подготовки обучающимся рекомендуется:  
последовательно повторить ключевые темы лекционного курса:

- архитектура веб-приложений и принципы проектирования API;
- методы интеграции моделей искусственного интеллекта в веб-сервисы;
- основы контейнеризации и деплоя;
- работа с базами данных, очередями сообщений и асинхронными компонентами;
- обеспечение безопасности, логирование и мониторинг веб-сервисов;

проанализировать выполненные практические работы, уделив внимание:

- корректности структуры серверного приложения;
- способам передачи данных и форматам запросов/ответов;
- работе API-эндпоинтов и взаимодействию с ИИ-модулем;
- работе с контейнерами, очередями и базой данных;
- настройке окружения и использованию документации;

пересмотреть примеры интеграции ИИ-моделей, изученные в курсе:

- особенности предобработки данных;
- схемы взаимодействия сервиса и модели;
- возможные ошибки интеграции и способы их устранения;

систематизировать знания по архитектурным решениям, применяемым при построении веб-сервисов:

- проектирование микросервисов;
- разделение функциональности;
- типовые точки отказа и методы повышения устойчивости.

Подготовка к экзамену должна проводиться последовательно на протяжении всего семестра. Рекомендуется активно участвовать в обсуждениях на занятиях, выполнять все практические задания, анализировать примеры реализации и регулярно обращаться к документации используемых технологий. Такая работа формирует глубокое понимание материала и обеспечивает успешное прохождение экзамена.

## 7. Оценочные средства

### 7.1. Паспорт оценочных средств

Семестр	Код контролируемой компетенции (или ее части)	Наименование оценочного средства
7	ПК-7	Вопросы к экзамену 1-78 Отчеты по практическим работам 1-16

### 7.2. Типовые задания или иные материалы, необходимые для текущего контроля

#### 7.2.1. Пример практической работы

##### **Практическая работа 1. Проектирование структуры веб-сервиса: модули, конфигурация, маршруты**

Цель работы: Получить навыки проектирования архитектуры веб-сервиса на ранних этапах разработки; научиться выделять основные модули приложения, определять структуру каталогов, конфигурационные файлы и маршруты API; сформировать понимание принципов построения сервисно-ориентированных приложений и интеграции ИИ-модулей.

##### **1. Ознакомьтесь с ключевыми понятиями**

Прочитайте краткие определения:

- **веб-сервис** — серверное приложение, предоставляющее функциональность по протоколу HTTP/HTTPS;
- **API (Application Programming Interface)** — набор маршрутов и методов, через которые клиент взаимодействует с сервисом;
- **маршрут (endpoint)** — адрес и метод (GET, POST и др.), обрабатывающий запрос;
- **модуль веб-сервиса** — логически обособленная часть функциональности (auth, users, ml, data, admin и др.);
- **конфигурация** — параметры запуска сервиса: переменные окружения, настройки БД, пути к модели и др.;
- **структура проекта** — иерархия каталогов, файлов кода, шаблонов, моделей, конфигураций;
- **точка входа** — основной файл, запускающий приложение.

##### **2. Изучите предложенный шаблон структуры проекта**

Преподаватель выдаёт 2–3 примера структур веб-сервисов (скриншоты, PDF, Git-репозитории).

Ознакомьтесь с ними и определите:

- какие каталоги выделены (app, routers, services, models, config);
- какие модули присутствуют;
- какие маршруты реализованы;
- какие конфигурации используются;
- где располагаются файлы модели ИИ (если есть).

##### **3. Спроектируйте структуру собственного веб-сервиса**

Создайте проект (в любой среде: PyCharm, VS Code, IntelliJ, WebStorm или просто как схема).

Заполните таблицу 1. Модули веб-сервиса

№	Название модуля	Назначение модуля	Основные функции	Связанные маршруты
1				

2				
3				
4				

#### 4. Определите конфигурацию веб-сервиса

Укажите параметры, необходимые для работы сервиса (переменные окружения, базы данных, путь к модели).

Таблица 2. Конфигурационные параметры

№	Параметр	Значение (пример)	Назначение
1			
2			
3			
4			

#### 5. Разработайте маршруты (эндпоинты) сервиса

Определите планируемые API-методы: их назначение, методы HTTP, входные и выходные данные.

Таблица 3. Маршруты сервиса

№	Endpoint	Метод	Назначение	Входные данные	Выходные данные
1					
2					
3					
4					

#### 6. Постройте схему архитектуры веб-сервиса

Нарисуйте схему (в любой форме: блок-схема, граф, архитектурная диаграмма):

Должны быть отражены:

- модули сервиса;
- маршруты и связи между ними;
- конфигурационные зависимости;
- при наличии — модуль ИИ (ml-service, model endpoint).

Можно выполнить в любой программе: draw.io, Figma, diagrams.net, Miro или вручную.

#### 7. Подготовьте отчёт, включающий следующую структуру

- Название работы
- Цель работы
- Краткое описание понятий: веб-сервис, API, маршрут, конфигурация, модуль
- Анализ примеров архитектур (по предложенным шаблонам)
- Спроектированная структура веб-сервиса (таблица модулей)
- Конфигурация веб-сервиса (таблица параметров)
- Маршруты веб-сервиса (таблица endpoint'ов)
- Архитектурная схема
- Выводы (не менее 5 пунктов)

#### Требования к оформлению

Отчет должен содержать подробное описание (включая иллюстрации). Отчёт по практическому занятию выполняется на страницах формата А4 в электронном виде.

При оформлении отчёта используется сквозная нумерация страниц, считая титульный лист первой страницей. Номер страницы на титульном листе не ставится. Номера страницы ставятся по центру сверху.

При оформлении отчёта соблюдать следующие требования:

- Для заголовков: полужирный шрифт, 14 пт, центрированный.

- Для основного текста: нежирный шрифт, 14 пт, выравнивание по ширине.
- Во всех случаях тип шрифта – Times New Roman, отступ абзаца 1.25 см, полуторный междустрочный интервал.
- Поля: левое – 2 см, правое, верхнее и нижнее – 1 см.

### Процедура оценивания

Оценка выполненной практической работы проводится по следующим критериям:

1. Наличие всей существенной информации по работе
2. Точность и полнота предоставляемых сведений
3. Непротиворечивость приводимой информации
4. Правильность интерпретаций и выводов, которые сделаны по результатам работы
5. Степень достижения обучающимся поставленной цели
6. Обоснованность применяемого решения
7. Грамотность (содержательная) используемых формулировок

**Критерии оценки** за отчеты по практическим работам:

Формы текущего контроля	Критерии и нормы оценки	
Отчеты по практическим работам 1-16	Отлично	<ul style="list-style-type: none"> <li>• Работа выполнена в полном объеме, все этапы и задания реализованы без пропусков.</li> <li>• Обучающийся демонстрирует глубокое понимание темы, способность к критическому анализу, выдвижению собственных обоснованных идей и решений.</li> <li>• Отчет безупречно структурирован, соответствует всем стандартам оформления. Материал изложен логично, последовательно, ясно и грамотно. Все выводы аргументированы и подкреплены результатами выполненной работы.</li> <li>• Продemonстрировано уверенное применение теоретических знаний для решения поставленной практической задачи.</li> <li>• Графики, схемы, таблицы используются уместно, информативно и наглядно, существенно облегчая понимание результатов.</li> </ul>
	Хорошо	<ul style="list-style-type: none"> <li>• Работа выполнена практически в полном объеме, могут иметься незначительные недочеты или пропуски второстепенных деталей.</li> <li>• Работа в основном выполнена самостоятельно. Обучающийся показывает твердое понимание материала, но анализ может быть не столь глубоким, как на "отлично". Собственные выводы присутствуют, но могут быть недостаточно развернутыми.</li> <li>• Отчет хорошо структурирован и оформлен, однако могут присутствовать небольшие погрешности в стиле или оформлении. Логика изложения в целом сохранена.</li> <li>• Визуальные материалы присутствуют и отражают суть работы, но их качество или информативность могут быть несколько ниже.</li> </ul>

Формы текущего контроля	Критерии и нормы оценки	
	Удовлетворительно	<ul style="list-style-type: none"> <li>• Работа выполнена формально, в минимально допустимом объеме. Значительная часть заданий выполнена поверхностно или с ошибками. Могут быть пропущены ключевые элементы работы.</li> <li>• * Работа выполнена с существенной помощью преподавателя или следованием шаблону без глубокого осмысления. Анализ слабый, выводы носят общий характер или отсутствуют.</li> <li>• Отчет имеет нарушения в структуре и оформлении. Логика изложения нарушена, язык изложения может быть неточным, присутствуют грамматические ошибки.</li> <li>• Применение знаний на практике: Теоретические</li> <li>• Визуальные материалы отсутствуют, либо их использование неоправданно и не помогает раскрыть содержание.</li> </ul>
	Неудовлетворительно	<ul style="list-style-type: none"> <li>• Работа не выполнена или выполнена в ничтожном объеме, не позволяющем проверить усвоение материала. Ключевые задания не выполнены.</li> <li>• Обучающийся не понимает сути выполненной работы и не может ответить на вопросы по базовым понятиям темы.</li> <li>• Отчет не структурирован, не оформлен или представлен в виде, не поддающемся проверке.</li> <li>• Работа содержит грубые концептуальные или фактические ошибки.</li> </ul>

### 7.3. Оценочные средства для промежуточной аттестации по итогам освоения дисциплины

#### 7.3.1. Вопросы к промежуточной аттестации

Семестр 7

№	Вопросы к экзамену
1.	Дайте определение Jakarta EE и перечислите ключевые спецификации, используемые при разработке веб-сервисов.
2.	Объясните назначение контейнера сервлетов и его роль в архитектуре Jakarta EE.
3.	Перечислите основные компоненты веб-приложения Jakarta EE и охарактеризуйте их.
4.	Опишите жизненный цикл сервлета и его влияние на обработку HTTP-запросов.
5.	Объясните назначение слоистой архитектуры (Controller → Service → Repository) в Jakarta EE.
6.	Раскройте понятие "enterprise-уровня" приложений и их особенности по сравнению с обычными веб-сервисами.
7.	Охарактеризуйте роль Jakarta RESTful Web Services (JAX-RS) при создании API.
8.	Перечислите способы маршрутизации HTTP-запросов в JAX-RS и принципы их работы.

№	Вопросы к экзамену
9.	Объясните работу аннотаций JAX-RS (@GET, @POST, @Path, @Produces, @Consumes).
10.	Охарактеризуйте роль Jakarta Servlet API в работе веб-приложения.
11.	Объясните разницу между Jakarta JSON-B и Jakarta JSON-P и сферы их применения.
12.	Опишите процесс сериализации и десериализации объектов в Jakarta JSON-B.
13.	Объясните назначение DTO и их роль в веб-приложениях.
14.	Раскройте важность валидации входных данных и перечислите механизмы Jakarta Bean Validation.
15.	Объясните, что такое схема данных API (API contract) и для чего она применяется.
16.	Интеграция моделей ИИ в Jakarta-приложения
17.	Перечислите подходы к интеграции внешних ML/AI-моделей через HTTP в Jakarta EE.
18.	Объясните особенности реализации клиентских вызовов к ML-модулю с использованием JAX-RS Client API.
19.	Опишите этапы подготовки входных данных в Jakarta-сервисе перед передачей модели.
20.	Перечислите типичные ошибки взаимодействия Jakarta-бэкенда с внешним ML-сервисом.
21.	Объясните назначение выделенного ML-сервиса и преимущества его использования.
22.	Охарактеризуйте работу Jakarta Concurrency API при фоновой обработке AI-задач.
23.	Опишите архитектурные схемы вызова модели ИИ из Jakarta-приложения.
24.	Объясните необходимость обработки ошибок модели на стороне веб-сервиса.
25.	Раскройте требования к форматам данных при интеграции Jakarta-сервиса с ML-моделью.
26.	Дайте определение Jakarta Persistence (JPA) и перечислите основные компоненты ORM.
27.	Объясните назначение EntityManager и его роль в работе с базой данных.
28.	Перечислите типичные шаблоны работы репозитория в Jakarta EE.
29.	Объясните принципы транзакционности и роль Jakarta Transaction API.
30.	Охарактеризуйте использование JPQL и его отличия от SQL.
31.	Опишите подходы к оптимизации JPA-запросов в веб-сервисах.
32.	Объясните назначение Jakarta Messaging (JMS) и его использование в распределённых системах.
33.	Перечислите основные модели доставки сообщений JMS.
34.	Опишите архитектуру веб-приложения, использующего JMS для обработки AI-задач.
35.	Раскройте роль Message-driven beans (MDB) при работе с очередями.
36.	Объясните преимущества использования брокера сообщений для интеграции ИИ.
37.	Перечислите основные механизмы Jakarta Security.
38.	Объясните принципы работы фильтров безопасности в веб-приложениях.
39.	Охарактеризуйте роли и принципы ролевой модели безопасности в Jakarta EE.

№	Вопросы к экзамену
40.	Объясните назначение HTTP-аутентификации и методов её реализации.
41.	Раскройте понятие CORS и опишите его конфигурацию в приложении Jakarta EE.
42.	Объясните, как защищать REST-эндпоинты от несанкционированного доступа.
43.	Охарактеризуйте методы защиты от атак SQL-injection и XSS в Java-приложениях.
44.	Дайте определение Docker-контейнеру и опишите его преимущества в разработке Jakarta-приложений.
45.	Опишите структуру Dockerfile для Java/Jakarta-сервиса.
46.	Перечислите типичные ошибки при контейнеризации Jakarta-приложений.
47.	Объясните назначение Docker Compose при работе с многосервисной системой.
48.	Раскройте особенности деплоя Jakarta-сервисов в Kubernetes.
49.	Объясните подходы к контейнеризации ML-модулей и их интеграции с Jakarta-бэкендом.
50.	Объясните назначение журналирования с использованием Jakarta Logging.
51.	Перечислите подходы к централизованному сбору логов Jakarta-сервисов.
52.	Охарактеризуйте принципы работы систем APM (Application Performance Monitoring).
53.	Опишите виды метрик, используемых для мониторинга работы веб-сервиса.
54.	Объясните назначение трассировки запросов в распределённых сервисах.
55.	Перечислите подходы к оптимизации производительности Jakarta-приложений.
56.	Объясните разницу между горизонтальным и вертикальным масштабированием.
57.	Охарактеризуйте факторы, влияющие на время ответа JAX-RS-эндпоинтов.
58.	Опишите способы оптимизации сетевых вызовов к внешним ML-сервисам.
59.	Объясните назначение кэширования результатов модели и его варианты.
60.	Оцените влияние асинхронности на производительность веб-сервиса.
61.	Дайте определение CI/CD в контексте Jakarta EE.
62.	Объясните этапы автоматической сборки Jakarta-приложения.
63.	Перечислите инструменты для CI/CD в Java/Jakarta-экосистеме.
64.	Опишите разницу между dev, staging и production окружениями.
65.	Приведите этапы деплоя Jakarta-приложения в контейнеризированной среде.
66.	Объясните назначение health-checks в оркестрации сервисов.
67.	Объясните роль спецификации OpenAPI при разработке Jakarta-REST-сервисов.
68.	Перечислите требования к качественной документации API.
69.	Опишите этапы тестирования REST-эндпоинтов в Jakarta EE.
70.	Охарактеризуйте назначение интеграционных тестов при работе Jakarta-сервиса с ML-модулем.
71.	Оцените важность контрактного тестирования при взаимодействии микросервисов.

№	Вопросы к экзамену
72.	Раскройте принципы чистой архитектуры (Clean Architecture) применительно к Jakarta-сервисам.
73.	Перечислите правила разделения ответственности между слоями приложения.
74.	Объясните понятие “точка отказа” в архитектуре и способы её устранения.
75.	Опишите подходы к проектированию API для взаимодействия с ИИ-модулем.
76.	Раскройте преимущества использования CDI (Contexts and Dependency Injection).
77.	Объясните, как внедрение зависимостей упрощает архитектуру веб-приложения.
78.	Опишите жизненный цикл HTTP-запроса в приложении Jakarta EE.

### 7.3.2. Критерии и нормы оценки

#### Процедура оценивания по билетам

Ответы на вопросы билета обеспечивают возможность адекватной оценки знаний и профессиональной подготовки бакалавров. Важным фактором при этом является умение экзаменуемого оперировать в своем ответе ссылками на соответствующие положения учебной и научной литературы. По результатам выполнения практического задания определяется уровень сформированности профессиональных компетенций обучающимся по использованию современных технологий решения прикладных задач предметной области.

Процедура проведения экзамена:

- группа не более чем из 10 обучающихся входит в аудиторию, где будет проходить экзамен, остальные остаются ожидать очередь в коридоре;
- каждый обучающийся вытягивает 1 билет из банка билетов и идет готовиться к ответу на вопросы билета;
- по готовности, обучающиеся подходят и отвечают по вопросам билета;
- после того, как обучающийся завершил ответ на вопросы и получил оценку, он покидает аудиторию, а его билет ложится к оставшимся в банке билетам;
- на место вышедшего обучающегося входит очередной студент, которые вытягивает билет и идет готовиться для ответа;
- и так до последнего отвечающего обучающегося.

Требования к ответу:

- ответ должен быть научным, логически стройным, опираться на соответствующие теоретические положения и концепции;
- ответ следует строить в единстве теории и практики с подтверждением теоретических положений реальными практическими примерами;
- практические задания должны быть выполнены на компьютере с использованием соответствующих программных средств.

Оценивание:

- порядок ответов на вопросы билета определяется самим обучающимся;
- при необходимости дополнительные вопросы задаются обучающемуся после ответа на все три вопроса билета;



- оценка результатов производится по четырехбалльной системе («отлично», «хорошо», «удовлетворительно», «неудовлетворительно»);
- оценка объявляется после завершения ответа обучающегося на дополнительные вопросы.

**Критерии оценки:**

- оценка «отлично» выставляется обучающемуся, если он исчерпывающе и грамотно дал ответы на вопросы экзаменационного билета или при ответе допустил небольшую неточность на 1 вопрос, но при этом смог грамотно ответить на дополнительные вопросы;
- оценка «хорошо» выставляется обучающемуся, если он исчерпывающе и грамотно дал ответ на 1 вопрос экзаменационного билета, а на другой только тезисные высказывания или допустил небольшие неточности при ответе на вопросы экзаменационного билета и дал краткие ответы на дополнительные вопросы;
- оценка «удовлетворительно» выставляется обучающемуся, если он не смог дать ответ на один из вопросов экзаменационного билета или ответил на все вопросы, но при этом ответы содержали только тезисные высказывания;
- оценка «неудовлетворительно» выставляется обучающемуся, если он не дал ответ на вопросы экзаменационного билета или в ответе содержались фундаментальные ошибки.

## 8. Учебно-методическое и информационное обеспечение дисциплины

### 8.1. Обязательная литература

№ п/п	Авторы, составители	Заглавие (заголовок)	Тип (учебник, учебное пособие, учебно-методическое пособие, практикум, др.)	Год издания	Количество в научной библиотеке / Наименование ЭБС
1.	М. Л. Аншина	Аншина М. Л. Архитектура приложений и данных [Электронный ресурс] : учебное пособие / М. Л. Аншина. — М. : МИРЭА — Российский технологический университет, 2024.	учебное пособие	2025	ЭБС «Лань»
2.	Е. Л. Турнецкая	Тестирование и контроль качества программного обеспечения : учебное пособие / Е. Л. Турнецкая, А. В. Аграновский, А. А. Сенцов. — Санкт-Петербург : ГУАП, 2023. — 118 с. — ISBN 978-5-8088-1891-0. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <a href="https://e.lanbook.com/book/461498">https://e.lanbook.com/book/461498</a> (дата обращения: 28.11.2025)	учебное пособие	2023	ЭБС «Лань»
3.	Ю. А. Антохина, М. Л. Кричевский, Ю. А. Мартынова, А. А. Оводенко	Искусственный интеллект. Инноватика : учебное пособие / Ю. А. Антохина, М. Л. Кричевский, Ю. А. Мартынова, А. А. Оводенко. — Санкт-Петербург : ГУАП, 2023. — 320 с. — ISBN 978-5-8088-1830-9. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <a href="https://e.lanbook.com/book/341003">https://e.lanbook.com/book/341003</a> (дата обращения: 28.11.2025).	учебное пособие	2023	ЭБС «Лань»
4.	А. Н. Баланов	Машинное обучение и искусственный интеллект : учебное пособие для вузов / А. Н. Баланов. — 2-е изд., стер. — Санкт-Петербург : Лань, 2025. — 172 с. — ISBN 978-5-507-52891-2. — Текст : электронный // Лань : электронно-библиотечная	Учебное пособие	2025	ЭБС «Лань»

<b>№ п/п</b>	<b>Авторы, составители</b>	<b>Заглавие (заголовок)</b>	<b>Тип (учебник, учебное пособие, учебно- методическое пособие, практикум, др.)</b>	<b>Год издания</b>	<b>Количество в научной библиотеке / Наименование ЭБС</b>
		система. — URL: <a href="https://e.lanbook.com/book/462248">https://e.lanbook.com/book/462248</a> (дата обращения: 28.11.2025).			
5.	М. М. Маран	Маран, М. М. Программная инженерия : Учебное пособие для вузов / М. М. Маран. — 3-е изд., стер. — Санкт-Петербург : Лань, 2022. — 196 с. — ISBN 978-5-8114-9323-4. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <a href="https://e.lanbook.com/book/189470">https://e.lanbook.com/book/189470</a> (дата обращения: 28.11.2025).	Учебное пособие	2022	ЭБС «Лань»

## 8.2. Дополнительная литература

<b>№ п/п</b>	<b>Авторы, составители</b>	<b>Заглавие (заголовок)</b>	<b>Тип (учебник, учебное пособие, учебно- методическое пособие, практикум, др.)</b>	<b>Год издания</b>	<b>Количество в научной библиотеке / Наименование ЭБС</b>
1.	А. А. Бубнов	Бубнов, А. А. Тестирование программного обеспечения : учебное пособие / А. А. Бубнов, С. А. Бубнов, В. В. Тишкина. — Рязань : РГРТУ, 2024. — 164 с. — ISBN 978-5-7722-0421-4. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <a href="https://e.lanbook.com/book/494540">https://e.lanbook.com/book/494540</a> (дата обращения: 28.11.2025).	учебное пособие	2024	ЭБС «IPRbooks»
2.	Е. Л. Турнецкая	Турнецкая, Е. Л. Программная инженерия. Тестирование и контроль	учебное пособие	2025	ЭБС «IPRbooks»

<b>№ п/п</b>	<b>Авторы, составители</b>	<b>Заглавие (заголовок)</b>	<b>Тип (учебник, учебное пособие, учебно- методическое пособие, практикум, др.)</b>	<b>Год издания</b>	<b>Количество в научной библиотеке / Наименование ЭБС</b>
		качества программного обеспечения : учебное пособие для вузов / Е. Л. Турнецкая, А. В. Аграновский. — Санкт- Петербург : Лань, 2025. — 172 с. — ISBN 978-5-507-51677-3. — Текст : электронный // Лань : электронно- библиотечная система. — URL: <a href="https://e.lanbook.com/book/455672">https://e.lanbook.com/book/455672</a> (дата обращения: 28.11.2025).			
3.	Е. Л. Романов	Романов, Е. Л. Программная инженерия : учебное пособие / Е. Л. Романов. — Новосибирск : НГТУ, 2017. — 395 с. — ISBN 978-5-7782-3455-0. — Текст : электронный // Лань : электронно- библиотечная система. — URL: <a href="https://e.lanbook.com/book/118221">https://e.lanbook.com/book/118221</a> (дата обращения: 28.11.2025).	учебное пособие	2017	ЭБС «IPRbooks»

### 8.3. Перечень профессиональных баз данных и информационных справочных систем

№ пп	Наименование	Ссылка
1	Springer Nature (Полнотекстовая коллекция журналов)	<a href="https://www.springernature.com/gp/products">https://www.springernature.com/gp/products</a>
2	Springer eBooks (Полнотекстовая коллекция электронных книг издательства Springer Nature)	<a href="https://link.springer.com/">https://link.springer.com/</a>
3	«Кодекс»	<a href="https://kodeks.ru/">https://kodeks.ru/</a>
4	ELIBRARY.RU (электронная библиотека научных публикаций)	<a href="http://elibrary.ru">http://elibrary.ru</a>
5	"Гарант"	<a href="https://www.garant.ru/">https://www.garant.ru/</a>
6	"КонсультантПлюс"	<a href="https://www.consultant.ru/">https://www.consultant.ru/</a>
7	Техэксперт	<a href="https://cntd.ru/">https://cntd.ru/</a>

### 8.4. Перечень программного обеспечения

№ п/п	Наименование ПО	Реквизиты договора (дата, номер, срок действия)
1	OpenJDK 17 / 21 (реализация Java)	Лицензия: GNU GPL v2 with Classpath Exception (свободное ПО)
2	Jakarta EE (GlassFish / Payara / WildFly) — сервер приложений	GlassFish: CDDL + GPL (свободное ПО); Payara Community Edition: Apache License 2.0; WildFly: LGPL v2.1
3	Maven (система сборки)	Лицензия: Apache License 2.0 (свободное ПО)
4	Postman (инструмент для тестирования REST-API)	Бесплатная версия для образовательного и личного использования
5	Docker Desktop / Docker Engine (контейнеризация)	Лицензия: Apache License 2.0, бесплатная версия для обучения
6	Docker Compose	Лицензия: Apache License 2.0 (свободное ПО)
7	Git (система контроля версий)	Лицензия: GNU GPL v2 (свободное ПО)
8	GitHub (web-интерфейс)	Бесплатный доступ для образовательного использования (Free Tier)
9	H2 / PostgreSQL (БД для разработки и тестирования)	H2 — MPL 2.0 (free), PostgreSQL — PostgreSQL License (свободное ПО)
10	OpenAPI / Swagger UI (генерация и документация API)	Лицензия: Apache License 2.0 (свободное ПО)
11	Curl / httpie (утилиты для работы с HTTP-запросами)	Curl — MIT License; httpie — BSD License
12	VS Code / IntelliJ IDEA Community Edition (IDE)	VS Code — MIT License; IntelliJ CE — Apache License 2.0

**8.5. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине**

№ п/п	Наименование оборудованных учебных кабинетов, лабораторий, мастерских и др. объектов для проведения практических и лабораторных занятий, помещений для самостоятельной работы обучающихся (номер аудитории)	Перечень основного оборудования
1	Компьютерный класс. Учебная аудитория для проведения занятий лекционного типа. Учебная аудитория для проведения занятий семинарского типа. Учебная аудитория для проведения лабораторных работ. Учебная аудитория для курсового проектирования (выполнения курсовых работ). Учебная аудитория для проведения групповых и индивидуальных консультаций. Учебная аудитория для проведения занятий текущего контроля и промежуточной аттестации. (УЛК-401).	Компьютер (монитор 19", системный блок Pentium (R) Dual-Core E5500 2,8 GHz / 4 Gb / 500 Gb), столы ученические, столы компьютерные, стол преподавательский, стулья, доска аудиторная (меловая).
2	Помещение для самостоятельной работы обучающихся (УЛК-105).	Стол, стулья, стеллажи (в т.ч. выставочные) с книгами, компьютеры, мобильные рабочие места.
3	Помещение для самостоятельной работы обучающихся (УЛК-406).	Стол компьютерный, стулья, микрокомпьютеры raspberry pi 32 bit.