

Документ подписан простой электронной подписью

Информация о владельце:

ФИО: Порохня Андрей Алексеевич

Должность: И.о. директора института перспективной инженерии

Дата подписания: 19.06.2026 16:38:42

Уникальный программный ключ:

990bea3aeec48c23bd8699e48288f8d1960c600

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение высшего
образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

УТВЕРЖДАЮ

И.о. директора института

А.А. Порохня

Ф.И.О.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Междисциплинарный проектный практикум-3

Направление подготовки
Направленность (профиль)

09.04.04 Программная инженерия
Инженерия сложных программных
систем

Год начала обучения

2026

Форма обучения

очная

Реализуется в семестрах

5,6

Разработано

Заведующая межинститутской базовой
кафедрой департамента цифровых,
робототехнических систем и электроники
(должность разработчика)

Новикова Елена Николаевна

Ф.И.О.

Ставрополь 2026 г.

1. Цель и задачи освоения дисциплины (модуля)

Цель: Формирование у студентов комплексных практических компетенций в области сквозной разработки программных продуктов с использованием современных технологических стеков (Java/Spring Boot + React), методологий управления проектами, архитектурных паттернов, инструментов тестирования, контейнеризации и DevOps-практик, а также развитие навыков командной работы и самостоятельного проектирования

Задачи освоения дисциплины:

1. **Проектные:** Научить планированию проектов, декомпозиции, управлению сроками и ресурсами с использованием Git и Jira/Trello.
2. **Архитектурные:** Сформировать навыки проектирования архитектуры с применением PCMEF, Clean Architecture, DDD, событийно-ориентированного подхода.
3. **Бэкенд-разработка:** Обеспечить практические навыки разработки на Java/Spring Boot: REST API, ORM, аутентификация, асинхронность.
4. **Фронтенд-разработка (углубление React):** Углубить знания React: продвинутые хуки, роутинг, Redux Toolkit, React Query, тестирование, оптимизация.
5. **Тестирование и качество** Научить применению JUnit, Mockito, TestContainers, Jest, RTL, Cypress, JMeter.
6. **Контейнеризация и DevOps:** Сформировать навыки работы с Docker, CI/CD, мониторингом.
7. **Работа с данными:** Проектирование БД, сложные SQL-запросы, кэширование.
8. **ML-интеграция:** Интеграция готовых ML-моделей.
9. **Документирование:** Оформление документации, презентация, безопасность.

Место дисциплины (модуля) в структуре образовательной программы

Дисциплина «Междисциплинарный проектный практикум-3» относится к части, формируемой участниками образовательных отношений.

3. Перечень планируемых результатов обучения по дисциплине (модулю), соотнесённых с планируемыми результатами освоения образовательной программы

Код, формулировка компетенции	Код, формулировка индикатора	Планируемые результаты обучения по дисциплине (модулю), характеризующие этапы формирования компетенций, индикаторов
ПК-1. Способен осуществлять управление программными проектами: планировать, контролировать сроки и ресурсы,	ИД-1пк-1 Применяет методы декомпозиции работ (WBS), оценки трудоёмкости (PERT, Planning Poker) и календарного планирования (диаграмма Ганта, критический путь) для формирования плана проекта	Способен разрабатывать иерархическую структуру работ (WBS), оценивать трудоёмкость задач методами PERT и Planning Poker, строить диаграмму Ганта и определять критический путь для сквозного проекта «Умный дом»

Код, формулировка компетенции	Код, формулировка индикатора	Планируемые результаты обучения по дисциплине (модулю), характеризующие этапы формирования компетенций, индикаторов
применять системы контроля версий и инструменты управления задачами	ИД-2пк-1 Использует системы контроля версий (Git, Git Flow) и платформы совместной разработки (GitLab, GitHub) для организации командной работы	Способен организовать командную работу с использованием Git (Git Flow, ветвление, Pull Requests, разрешение конфликтов) и платформ совместной разработки (GitHub/GitLab) в рамках сквозного проекта
	ИД-3пк-1 Применяет инструменты управления задачами (Jira, Trello, YouTrack) для отслеживания прогресса (burndown chart), ведения бэклога и контроля исполнения	Способен вести бэклог проекта, отслеживать прогресс с помощью burndown chart и контролировать исполнение задач с использованием инструментов управления задачами (Jira/Trello/Yougile)
	ИД-4пк-1 Анализирует план-фактные отклонения по срокам, ресурсам и бюджету, управляет изменениями и рисками, разрабатывает корректирующие действия	Способен анализировать план-фактные отклонения по срокам и ресурсам, идентифицировать и анализировать риски, разрабатывать корректирующие действия для учебного проекта
ПК-2. Способен выполнять проектирование и разработку архитектуры программных систем среднего и крупного масштаба с применением современных	ИД-1пк-2 Выделяет и анализирует функциональные и нефункциональные требования (FURPS+), выбирает архитектурный стиль (монолит, микросервисы, EDA, Clean Architecture) на основе сценариев качества	Способен выделять и анализировать функциональные и нефункциональные требования, выбирать архитектурный стиль (монолит, микросервисы, событийно-ориентированная архитектура) на основе сценариев качества для системы «Умный дом»

Код, формулировка компетенции	Код, формулировка индикатора	Планируемые результаты обучения по дисциплине (модулю), характеризующие этапы формирования компетенций, индикаторов
паттернов и методов проектирования	ИД-2пк-2 Документирует архитектуру с использованием нотаций C4 и UML, фиксирует архитектурные решения в формате ADR (Architecture Decision Record)	Способен документировать архитектуру программной системы с использованием нотаций C4 (уровни Контекста, Контейнеров, Компонентов) в PlantUML и фиксировать архитектурные решения в формате ADR
	ИД-3пк-2 Применяет порождающие, структурные и поведенческие паттерны GoF, а также архитектурные паттерны (CQRS, Event Sourcing, Saga, Circuit Breaker)	Способен применять порождающие, структурные и поведенческие паттерны GoF, а также архитектурные паттерны (CQRS, Event Sourcing, Saga) при проектировании распределённой системы «Умный дом»
	ИД-4пк-2 Реализует многослойную архитектуру (Clean Architecture, Hexagonal, PCMEF), обеспечивая независимость от фреймворков через Dependency Inversion	Способен реализовывать многослойную архитектуру (Clean Architecture, PCMEF), обеспечивая независимость от фреймворков через принцип инверсии зависимостей (Dependency Inversion)
ПК-3. Владеет навыками разработки программного обеспечения: способен реализовывать алгоритмы, писать код на современных языках	ИД-1пк-3 Разрабатывает алгоритмы и реализует их на современных языках программирования (C++, C#, Python, Java, JavaScript/TypeScript, Go) с использованием эффективных структур данных и стандартных	Способен разрабатывать алгоритмы и реализовывать их на Java/Spring Boot для бэкенда и на JavaScript/TypeScript/React для фронтенда с использованием эффективных структур данных и стандартных библиотек

Код, формулировка компетенции	Код, формулировка индикатора	Планируемые результаты обучения по дисциплине (модулю), характеризующие этапы формирования компетенций, индикаторов
программирования, создавать веб- и desktop-приложения, работать с базами данных	библиотек	
	ИД-2пк-3 Создает клиент-серверные, веб- (frontend + backend) и desktop-приложения с графическим интерфейсом, применяя современные фреймворки (React, Spring, Django, WPF) и протоколы (REST, gRPC, WebSocket)	Способен создавать клиент-серверные веб-приложения с бэкендом на Java/Spring Boot (REST API, JPA/Hibernate, JWT, RabbitMQ) и фронтендом на React (продвинутые хуки, Redux Toolkit, React Router, React Query, WebSocket)
	ИД-3пк-3 Проектирует, мигрирует и оптимизирует запросы к реляционным и нереляционным БД с использованием SQL и ORM (Hibernate, Entity Framework, Django ORM)	Способен проектировать реляционные схемы баз данных (PostgreSQL, TimescaleDB), выполнять миграции (Liquibase/Flyway) и оптимизировать запросы с использованием SQL и Hibernate (JPA)
	ИД-4пк-3 Пишет модульный, документированный и тестируемый код, соблюдая стандарты оформления и принципы SOLID, использует системы сборки (Make, Gradle, npm)	Способен писать модульный, документированный и тестируемый код на Java и JavaScript/TypeScript, соблюдая стандарты оформления и принципы SOLID, используя системы сборки Maven/Gradle и npm/Vite
ПК-4. Способен проводить тестирование, отладку и оценивать	ИД-1пк-4 Применяет методы тест-дизайна (классы эквивалентности, граничные значения, pairwise) и пишет	Способен применять методы тест-дизайна (классы эквивалентности, граничные значения) и писать unit-тесты (JUnit, Mockito, Jest, RTL),

Код, формулировка компетенции	Код, формулировка индикатора	Планируемые результаты обучения по дисциплине (модулю), характеризующие этапы формирования компетенций, индикаторов
качество программного обеспечения, применяя инструменты автоматизированного тестирования и методы обеспечения надежности и безопасности	unit-, интеграционные и E2E-тесты с использованием JUnit, pytest, Jest, Cypress, Selenium	интеграционные тесты (TestContainers) и E2E-тесты (Cypress) для полного стека приложения
	ИД-2пк-4 Использует инструменты отладки (GDB, IDE debugger), профилирования (perf, JProfiler) и нагрузочного тестирования (JMeter, k6) для выявления узких мест	Способен использовать инструменты отладки (IDE debugger), профилирования (JProfiler, async profiler) и нагрузочного тестирования (JMeter, k6) для выявления узких мест производительности
	ИД-3пк-4 Проводит нагрузочное, стресс-тестирование и оценку надёжности (MTBF, MTTR) с применением chaos-инжиниринга, статического (SAST) и динамического (DAST) анализа безопасности	Способен проводить нагрузочное тестирование REST API, применять статический анализ кода (SonarQube) и базовый динамический анализ безопасности (OWASP ZAP) для оценки качества и надёжности ПО
ПК-5. Способен осуществлять сопровождение, настройку, развертывание и эксплуатацию программного обеспечения, в том числе в средах	ИД-1пк-5 Контейнеризует приложения с помощью Docker (Dockerfile, multi-stage builds) и оркестрирует многоконтейнерные стеки через Docker Compose	Способен контейнеризовать сервисы (бэкенд, фронтенд, БД, брокер сообщений) с помощью Docker (Dockerfile, многостадийная сборка, healthcheck) и оркестрировать многоконтейнерный стек через Docker Compose
	ИД-2пк-5 Разворачивает и настраивает инфраструктуру	Способен разворачивать и настраивать приложение на VPS (Yandex Cloud /

Код, формулировка компетенции	Код, формулировка индикатора	Планируемые результаты обучения по дисциплине (модулю), характеризующие этапы формирования компетенций, индикаторов
контейнеризации и облачных платформах	на облачных платформах (VM, Managed DB, Object Storage, K8s) с использованием IaC (Terraform, Ansible)	AWS) с использованием Nginx, SSL (Let's Encrypt) и базовыми подходами Infrastructure as Code
	ИД-3пк-5 Настраивает CI/CD пайплайны (GitLab CI, GitHub Actions) для автоматической сборки, тестирования и развёртывания приложений	Способен настраивать CI/CD пайплайны (GitHub Actions) для автоматической сборки, тестирования (unit, integration, E2E), статического анализа и развёртывания приложения
ПК-6. Владеет методами работы с данными и информационными системами: способен проектировать базы данных, выполнять анализ и обработку данных, интегрировать ПО с внешними системами	ИД-1пк-6 Проектирует реляционные схемы (нормализация до 3НФ) и нереляционные структуры данных, управляет миграциями БД (Liquibase, Flyway)	Способен проектировать реляционные схемы баз данных (нормализация до 3НФ), создавать ER-диаграммы и управлять миграциями БД с использованием Liquibase или Flyway
	ИД-2пк-6 Выполняет интеграцию с внешними API, брокерами сообщений (Kafka, RabbitMQ) и legacy-системами, обеспечивая идемпотентность и отказоустойчивость	Способен выполнять интеграцию с внешними REST API, настраивать асинхронное взаимодействие через брокер сообщений (RabbitMQ) и обеспечивать идемпотентность при интеграции
ПК-7. Способен применять современные информационные	ИД-1пк-7 Использует современные IDE, линтеры, форматтеры и инструменты статического анализа	Способен использовать современные IDE (IntelliJ IDEA, VS Code), линтеры (Checkstyle, ESLint), форматтеры и инструменты статического анализа

Код, формулировка компетенции	Код, формулировка индикатора	Планируемые результаты обучения по дисциплине (модулю), характеризующие этапы формирования компетенций, индикаторов
технологии, инструменты и платформы (включая отечественные), использовать DevOps-практики, облачные сервисы и средства автоматизации	(SonarQube) для обеспечения качества кода	(SonarQube) для обеспечения качества кода
	ИД-2пк-7 Настраивает мониторинг, логирование и алертинг (Prometheus, Grafana, ELK/Loki, Sentry) для отслеживания состояния ПО	Способен настраивать мониторинг (Prometheus + Grafana), централизованное логирование (Loki + Promtail) и алертинг для отслеживания состояния приложения в production-среде
ПК-8. Способен адаптировать и применять методы искусственного интеллекта и машинного обучения для решения профессиональных задач, включая интеграцию готовых моделей в программные продукты	ИД-1пк-8 Выполняет предобработку и анализ данных (EDA, визуализация, кодирование, масштабирование) для подготовки датасета к обучению	Способен выполнять предобработку и первичный анализ данных (EDA, визуализация, кодирование категориальных признаков, масштабирование) для подготовки данных к интеграции с ML-моделью
	ИД-3пк-8 Интегрирует предобученные модели и API ИИ (Hugging Face, OpenAI, YOLO) в программные продукты, разворачивает их как микросервисы (FastAPI, Flask, Streamlit, Docker) и оптимизирует инференс (кэширование, similarity cache, батчинг, асинхронная обработка через очереди)	Способен интегрировать готовые ML-модели (Hugging Face, scikit-learn) в веб-приложение через REST API, реализовывать асинхронный инференс через очередь задач (RabbitMQ) и кэширование результатов (Redis)
ПК-9. Умеет оформлять техническую документацию	ИД-1пк-9 Оформляет техническую документацию (ТЗ, пояснительная записка,	Способен оформлять техническую документацию проекта (README, Wiki, API-спецификацию

Код, формулировка компетенции	Код, формулировка индикатора	Планируемые результаты обучения по дисциплине (модулю), характеризующие этапы формирования компетенций, индикаторов
документацию, готовить презентации и отчеты, вести деловую коммуникацию на русском и иностранном языках в профессиональной среде	руководство пользователя, API-спецификация) в форматах Markdown, OpenAPI, DocBook в соответствии с ГОСТ (ЕСПД, ISO/IEC 12207) или современными стандартами	OpenAPI/Swagger) в формате Markdown, включая архитектурные диаграммы в нотации C4 (PlantUML)
	ИД-2пк-9 Готовит презентационные материалы и публично защищает результаты проектной деятельности, аргументированно отвечая на вопросы	Способен готовить презентационные материалы и публично защищать результаты проектной деятельности (сквозной и самостоятельный проекты), аргументированно отвечая на вопросы по архитектуре, реализации и тестированию
ПК-10. Готов соблюдать правовые нормы, стандарты и требования информационной безопасности, участвовать в разработке нормативной и технической документации, учитывать законодательство в области ИТ	ИД-1пк-10 Применяет основные правовые нормы (лицензирование ПО, авторское право, ФЗ-152, ФЗ-149) и стандарты ИБ (ГОСТ Р ИСО/МЭК 27001, ГОСТ Р 56545-2015) при разработке и эксплуатации ПО	Способен применять основные правовые нормы (лицензирование ПО, авторское право, требования ФЗ-152 о персональных данных) при разработке и документировании программного обеспечения
	ИД-2пк-10 Реализует меры защиты информации:	Способен реализовывать меры защиты информации: аутентификацию с

Код, формулировка компетенции	Код, формулировка индикатора	Планируемые результаты обучения по дисциплине (модулю), характеризующие этапы формирования компетенций, индикаторов
	аутентификацию (JWT, OAuth2), авторизацию (RBAC), шифрование, защиту от типовых уязвимостей OWASP Top 10 (SQL-инъекции, XSS, CSRF, IDOR) с учётом требований регуляторов (ФСТЭК)	использованием JWT (access/refresh токены) или OAuth2, ролевую авторизацию (RBAC), хеширование паролей (BCrypt), защиту от типовых уязвимостей OWASP Top 10 (SQL-инъекции, XSS, CSRF)

4. Объем учебной дисциплины и формы контроля *

Объем занятий: всего: 4 з.е. 144 ак.ч.	ОФО, в астр. часах
Контактная работа:	54
Лекции/из них практическая подготовка	18
Лабораторных работ/из них практическая подготовка	36
Практических занятий/из них практическая подготовка	
Самостоятельная работа	162
Формы контроля	
Экзамен	
Зачет	
Зачет с оценкой	±
Расчетно-графические работы	
Курсовые работы	
Контрольные работы	

* Дисциплина (модуль) предусматривает применение электронного обучения, дистанционных образовательных технологий

5. Содержание дисциплины (модуля), структурированное по темам (разделам) с указанием количества часов и видов занятий

№	Раздел (тема) дисциплины и краткое содержание	Формируемые компетенции, индикаторы	очная форма			
			Контактная работа обучающихся с преподавателем /из них в форме практической подготовки, часов			Самостоятельная работа, часов
			Лекции	Практические занятия	Лабораторные работы	
1	Анализ предметной области и архитектурное видение (C4: Контекст и Контейнеры) Анализ предметной области «Умный дом». Выделение стейкхолдеров (владелец дома, инженер поддержки, разработчик устройства). Определение функциональных и нефункциональных требований. Формулировка границ системы (что входит/не входит). Построение диаграммы контекста (C4 Level 1) и диаграммы контейнеров (C4 Level 2) в PlantUML. Оценка рисков и план-фактный анализ начального этапа проекта.	ИД-1пк-2 ИД-2пк-2 ИД-4пк-1			2	10
2	Детальное проектирование сервисов и API-контрактов (API First, OpenAPI) Выделение бизнес-доменов (Управление устройствами, Телеметрия, Пользователи, Сценарии). Определение REST эндпоинтов для каждого сервиса. Создание OpenAPI спецификаций (YAML) для Device Service и User Service. Описание моделей данных (сущностей). Проектирование схемы JWT-аутентификации. Построение обновлённой диаграммы контейнеров с указанием протоколов взаимодействия.	ИД-1пк-2 ИД-2пк-2 ИД-4пк-2			2	10

3	Асинхронное взаимодействие и событийно-ориентированная архитектура (EDA, RabbitMQ) Понятие событийно-ориентированной архитектуры (EDA). Установка и настройка RabbitMQ через Docker. Модификация Device Service для публикации событий (DeviceStateChangedEvent). Создание Notification Service для подписки на события и логирования. Демонстрация слабой связанности сервисов. Построение диаграммы последовательности включения устройства.	ИД-3пк-2 ИД-2пк-6 ИД-1пк-4			2	10
4	Контейнеризация и оркестрация (Docker & Docker Compose) Принципы контейнеризации. Структура Dockerfile, многостадийная сборка, слои. Создание Dockerfile для сервисов (бэкенд, фронтенд). Настройка healthcheck и непривилегированного пользователя. Создание docker-compose.yml для всего стека (бэкенд, БД, RabbitMQ, Redis). Настройка сетей (bridge) и именованных томов. Запуск и остановка многоконтейнерного приложения. Принципы 12-factor app: конфигурация через переменные окружения (.env).	ИД-1пк-5 ИД-4пк-2			2	10
5	Реализация бэкенда: REST API, ORM, бизнес-логика (Spring Boot + JPA) Создание JPA-сущностей (Device, User, Telemetry). Настройка связей между сущностями (@OneToMany, @ManyToOne). Создание Spring Data JPA репозиторий. Реализация сервисного слоя с бизнес-логикой (CRUD операции). Реализация REST-контроллеров. Настройка подключения к PostgreSQL (HikariCP). Выполнение миграций базы данных (Liquibase/Flyway).	ИД-1пк-3 ИД-3пк-3 ИД-4пк-3			2	10

6	Аутентификация и авторизация (JWT, Spring Security) Настройка Spring Security. Реализация UserDetailsService. Генерация и валидация JWT (access и refresh токены). Создание JWT-фильтра для аутентификации запросов. Реализация эндпоинтов регистрации, логина и обновления токена. Хеширование паролей (BCrypt). Настройка защиты эндпоинтов по ролям (USER, ADMIN, SUPPORT).	ИД-2пк-10 ИД-4пк-3 ИД-6пк-3			2	10
7	Модульное и интеграционное тестирование бэкенда (JUnit, Mockito, TestContainers) Написание unit-тестов для сервисного слоя с Mockito. Написание тестов для репозитория с @DataJpaTest. Написание интеграционных тестов для REST API с @SpringBootTest. Настройка TestContainers для PostgreSQL. Измерение покрытия кода (JaCoCo). Применение TDD (Test-Driven Development) для ключевых модулей.	ИД-1пк-4 ИД-2пк-4			2	10
8	Документирование API (Swagger/OpenAPI) и написание интеграционных тестов Настройка springdoc-openapi для автоматической генерации OpenAPI документации. Настройка Swagger UI. Добавление аннотаций для улучшения документации (@Operation, @ApiResponse, @Schema). Написание сквозных интеграционных тестов (E2E) для ключевых пользовательских сценариев. Валидация соответствия API спецификации. Генерация клиентского SDK из OpenAPI.	ИД-1пк-9 ИД-1пк-4 ИД-2пк-2			2	10

9	Итоговая интеграция бэкенда, рефакторинг и защита проекта (1 часть) Интеграция всех сервисов бэкенда (синхронное и асинхронное взаимодействие). Рефакторинг кода (устранение дублирования, длинных методов, «запахов» кода по Фаулеру). Анализ технического долга (Technical Debt) и составление плана его устранения. Подготовка презентации проекта (1 часть). Проведение ретроспективы команды. Защита промежуточных результатов	ИД-1пк-1 ИД-1пк-2 ИД-1пк-3 ИД-1пк-4, ИД-2пк-9			2	10
	ИТОГО за 5 семестр				18	90
10	Углубленный React: компоненты, хуки, роутинг (React Router v6, Vite) Создание React-проекта через Vite. Маршрутизация в SPA: BrowserRouter, Routes, Route, Link, useParams, useNavigate. Продвинутое хуки: useMemo, useCallback, React.memo для оптимизации ререндеров. useRef для работы с DOM. Создание кастомных хуков. Ленивая загрузка компонентов (React.lazy, Suspense) для code splitting.	ИД-2пк-3, ИД-4пк-3 ИД-2пк-3			2	4
11	Продвинутое управление состоянием (Redux Toolkit / RTK Query) Проблемы Context API (лишние ререндеры, сложность с асинхронностью). Redux Toolkit: createSlice, configureStore, useSelector, useDispatch. RTK Query для работы с API: встроенное кэширование, автоматическая инвалидация, генерация хуков. Настройка хранилища для аутентификации и устройств.	ИД-2пк-3			2	4
12	Интеграция с API: React Query (TanStack Query) и CORS Проблемы нативного fetch/axios. TanStack Query (React Query): useQuery (получение данных, кэширование, фоновая синхронизация), useMutation (изменение данных). QueryClient, инвалидация кэша. Оптимистичные обновления. Настройка CORS на бэкенде (django-cors-headers / Spring @CrossOrigin).	ИД-2пк-3 ИД-2пк-6			2	4

13	Формы и валидация (React Hook Form + Zod / Formik + Yup) React Hook Form - производительная библиотека для форм (минимальное количество ререндеров). Zod - схема валидации (TypeScript-first). Интеграция React Hook Form + Zod (resolver). Обработка ошибок и состояние загрузки. Альтернативно: Formik + Yup. Реализация форм добавления/редактирования устройства и профиля пользователя.	ИД-2пк-3 ИД-2пк-10			2	4
14	Тестирование React (Jest + React Testing Library) Jest как тестовый раннер. React Testing Library (RTL): render, screen, fireEvent, userEvent. Query методы (getBy, findBy, queryBy). Асинхронное тестирование (waitFor, findBy). Мокирование API-запросов с MSW (Mock Service Worker). Достижение покрытия не менее 60% для ключевых компонентов.	ИД-1пк-4 ИД-4пк-3			2	4
15	Кэширование на бэкенде с Redis (Cache-Aside, инвалидация) Запуск Redis через Docker. Настройка кэширования в Spring Boot (@Cacheable, @CacheEvict). Реализация стратегии Cache-Aside. Инвалидация кэша при обновлении данных. Кэширование результатов тяжелых запросов (список устройств, статистика). Оценка эффективности кэширования (hit/miss ratio).	ИД-3пк-3 ИД-1пк-7			2	4
16	WebSockets и real-time уведомления (Spring WebSocket + STOMP) Настройка WebSocket на бэкенде (Spring WebSocket + STOMP). Реализация отправки уведомлений при изменении состояния устройства. Подключение STOMP-клиента на фронтенде (STOMP.js, SockJS). Отображение real-time уведомлений. Интеграция с RabbitMQ как брокером сообщений для WebSocket.	ИД-2пк-3 ИД-2пк-6			2	4

17	Нагрузочное тестирование и профилирование (JMeter, JProfiler) Настройка JMeter для нагрузочного тестирования REST API (100 concurrent users). Измерение пропускной способности (RPS), latency (p95, p99), процента ошибок. Профилирование производительности с JProfiler / async profiler. Оптимизация «горячих точек» (N+1 запросы, неоптимальные алгоритмы).	ИД-2пк-4 ИД-3пк-4			2	4
18	CI/CD пайплайн (GitHub Actions: сборка, тесты, деплой) Создание GitHub Actions workflow для бэкенда (сборка, тесты). Создание workflow для фронтенда (сборка, линтинг, тесты). Настройка условного деплоя на staging при пуше в develop. Добавление статического анализа кода (Checkstyle/ESLint, SonarQube). Использование секретов GitHub для деплой-токенов.	ИД-3пк-5 ИД-1пк-7			2	4
19	Docker-композиция полного стека, healthcheck, оптимизация образов Интеграция всего стека в единую Docker Compose конфигурацию (бэкенд, фронтенд, БД, Redis, RabbitMQ). Настройка healthcheck для всех сервисов. Оптимизация Dockerfile (многостадийная сборка, .dockerignore, выбор slim/alpine образов). Настройка сетей (bridge) и именованных томов для персистентности данных.	ИД-1пк-5 ИД-4пк-2			2	4
20	Развертывание на VPS (Yandex Cloud / AWS) + Nginx + SSL Аренда VPS (Yandex Cloud / AWS / Digital Ocean). Настройка SSH-доступа и базового фаервола (UFW). Установка Docker и Docker Compose. Настройка Nginx как reverse proxy. Получение SSL-сертификата (Let's Encrypt / Certbot). Развертывание приложения в облаке. Настройка HTTPS.	ИД-2пк-5 ИД-3пк-5 ИД-2пк-10			2	4
21	Интеграция готовой ML-модели (классификация текстов / рекомендации) Выбор ML-задачи (классификация текстов отзывов / рекомендации устройств). Загрузка предобученной модели (Hugging Face transformers / scikit-learn). Создание REST API для инференса модели на Spring Boot / FastAPI. Интеграция вызова API на фронтенде (например, анализ тональности отзыва). Оценка качества инференса (метрики).	ИД-1пк-8 ИД-3пк-8			2	4

22	Мониторинг и логирование (Prometheus + Grafana + Loki) Настройка сбора метрик с Micrometer → Prometheus. Настройка визуализации метрик в Grafana (дашборд: RPS, latency, ошибки). Настройка централизованного сбора логов (Loki + Promtail). Настройка алертов (например, при высоком проценте ошибок или превышении latency).	ИД-2пк-7 ИД-1пк-7			2	4
23	Рефакторинг архитектуры: от PCMEF к Clean Architecture Трансформация существующей PCMEF-архитектуры в Clean Architecture. Выделение слоев: Entities, Use Cases, Interface Adapters, Frameworks. Применение Dependency Inversion Principle (интерфейсы репозитория). Выделение доменных моделей (Value Objects, Aggregates). Обеспечение независимости от фреймворков.	ИД-1пк-2 ИД-4пк-2			2	4
24	Асинхронный инференс ML-модели через очередь (RabbitMQ + Worker) Настройка очереди RabbitMQ для задач инференса ML-модели. Создание Worker на Java/Spring Boot, забирающего задачи из очереди и выполняющего инференс. Сохранение результатов в Redis/БД. Реализация polling-механизма на фронте для получения результатов. Кэширование результатов (exact match + similarity caching).	ИД-3пк-8 ИД-2пк-6			2	4
25	E2E-тестирование с Cypress Установка и настройка Cypress. Написание E2E-теста на авторизацию (логин → редирект на дашборд). Написание теста на создание и удаление устройства. Написание теста на срабатывание уведомления (при изменении состояния устройства). Использование cy.intercept для мокирования API. Интеграция E2E-тестов в CI/CD пайплайн.	ИД-1пк-4 ИД-2пк-10			2	4

26	Самостоятельный проект: презентация ТЗ и архитектуры (свободный выбор стека) Выбор темы (отличной от сквозного проекта). Составление технического задания. Выделение стейкхолдеров и требований. Проектирование архитектуры (С4, выбор технологического стека). Создание OpenAPI спецификаций. Оценка трудоемкости (WBS, PERT). Составление плана работ. Презентация ТЗ и архитектуры (10-15 минут).	ИД-1пк-1 ИД-1пк-2 ИД-1пк-3 ИД-1пк-4 ИД-2пк-2 ИД-1пк-9 ИД-2пк-9			2	4
27	Самостоятельный проект: реализация и защита (свободный выбор стека) Реализация бэкенда и фронтенда (или desktop-приложения) согласно ТЗ на выбранном стеке (Java/Spring Boot + React, Python/Django + React, Node.js + Angular/Vue, ASP.NET Core, Go + Svelte и т.д.). Написание тестов (unit, integration, E2E). Контейнеризация (Docker) и развертывание (VPS или локально). Подготовка документации (README, API, отчет). Подготовка презентации и финальная защита проекта.	ИД-1пк-3 ИД-2пк-3 ИД-3пк-3 ИД-1пк-4 ИД-2пк-4 ИД-1пк-5 ИД-2пк-5 ИД-3пк-5 ИД-2пк-6 ИД-3пк-8 ИД-1пк-9 ИД-2пк-9 ИД-2пк-10			2	4
	ИТОГО за 6 семестр				36	72
	ИТОГО				54	162

6. Фонд оценочных средств по дисциплине (модулю)

Фонд оценочных средств (ФОС) по дисциплине (модулю) базируется на перечне осваиваемых компетенций с указанием индикаторов. ФОС обеспечивает объективный контроль достижения запланированных результатов обучения. ФОС включает в себя:

- описание показателей и критериев оценивания компетенций на различных этапах их формирования, описание шкал оценивания;
- методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций (включаются в методические указания по тем видам работ, которые предусмотрены учебным планом и предусматривают оценку сформированности компетенций);
- типовые оценочные средства, необходимые для оценки знаний, умений и уровня сформированности компетенций.

ФОС является приложением к данной программе дисциплины.

7. Методические указания для обучающихся по освоению дисциплины

Приступая к работе, каждый студент должен принимать во внимание следующие положения.

Дисциплина построена по тематическому принципу, каждая тема представляет собой логически завершённый раздел.

Практические занятия проводятся с целью закрепления усвоенной информации, приобретения навыков ее применения при решении практических задач в соответствующей предметной области.

Самостоятельная работа студентов направлена на самостоятельное изучение дополнительного материала, подготовку к лабораторным занятиям, а также выполнения всех видов самостоятельной работы.

Для успешного освоения дисциплины, необходимо выполнить все виды самостоятельной работы, используя рекомендуемые источники информации.

8. Учебно-методическое и информационное обеспечение дисциплины

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

8.1.1. Перечень основной литературы:

1. Сэм Ньюмен. Создание микросервисов / Пер. с англ. - М.: ДМК Пресс, 2024. - 590 с. - ISBN 978-5-93700-214-7.
Ключевые разделы: Принципы микросервисной архитектуры, декомпозиция приложений, межсервисное взаимодействие, API Gateway.
2. Крейг Уоллс. Spring Boot в действии / Пер. с англ. - М.: Манн, Иванов и Фербер, 2022. - 464 с. - ISBN 978-5-00195-146-3.
Ключевые разделы: Создание REST API, Spring Data JPA, тестирование Spring-приложений, Spring Security.
3. Эрик Эванс. Предметно-ориентированное проектирование (DDD): структуризация сложных программных систем / Пер. с англ. - М.: Вильямс, 2021. - 448 с. - ISBN 978-5-907144-38-7.
Ключевые разделы: Ограниченные контексты, агрегаты, репозитории, доменные события.
4. Мартин Фаулер. Рефакторинг: улучшение существующего кода / Пер. с англ. - СПб.: Питер, 2021. - 448 с. - ISBN 978-5-4461-1359-8.
Ключевые разделы: Каталог рефакторингов, запахи кода, техники улучшения архитектуры.
5. Стив Макконнелл. Совершенный код / Пер. с англ. - М.: Русская редакция, 2020. - 896 с. - ISBN 978-5-7502-0064-4.
Ключевые разделы: Проектирование классов, работа с подпрограммами, отладка, качество программного обеспечения.
6. Алексей Васильев. Java 17: основы программирования. - СПб.: Наука и Техника, 2022. - 640 с. - ISBN 978-5-94387-932-6.
Ключевые разделы: Синтаксис Java, ООП, коллекции, работа с потоками.

8.1.2. Перечень дополнительной литературы:

1. Карл Айгнс. Современный анализ требований / Пер. с англ. - М.: ДМК Пресс, 2020. - 394 с. - ISBN 978-5-97060-835-5.
Ключевые разделы: Выявление стейкхолдеров, сбор требований, управление изменениями.
2. Крис Ричардсон. Микросервисы. Паттерны разработки и рефакторинга / Пер. с англ. - СПб.: Питер, 2022. - 544 с. - ISBN 978-5-4461-1718-3.
Ключевые разделы: Saga, CQRS, Event Sourcing, Circuit Breaker.
3. Роберт Мартин. Чистая архитектура. Искусство разработки программного обеспечения / Пер. с англ. - СПб.: Питер, 2021. - 352 с. - ISBN 978-5-4461-1272-0.

Ключевые разделы: Dependency Inversion, независимость от фреймворков, границы в архитектуре.

4. Даниэль Брайант, Мартин Карпентер. RabbitMQ. Профессиональная разработка распределённых приложений. - М.: ДМК Пресс, 2021. - 320 с. - ISBN 978-5-97060-887-4.

Ключевые разделы: exchanges, queues, binding, кластеризация.

5. Томас Виттиг, Вольфганг Гемлих. Docker на практике / Пер. с англ. - М.: ДМК Пресс, 2022. - 524 с. - ISBN 978-5-97060-935-2.
Ключевые разделы: Dockerfile, слои, Docker Compose, оптимизация образов.
6. Юрий Гурин. PostgreSQL изнутри. - М.: ДМК Пресс, 2021. - 488 с. - ISBN 978-5-97060-917-8.

8.2. Перечень учебно-методического обеспечения самостоятельной работы обучающихся по дисциплине

8.3. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины

Отечественные ресурсы (доступны и рекомендованы)

1. Habr - раздел «Программирование»
<https://habr.com/ru/hub/programming/>
Назначение: Публикации по Java, Spring Boot, Docker, архитектуре ПО. Примеры статей: анализ C4-модели, руководства по RabbitMQ, Spring Security.
2. PGConf.Russia - материалы конференций
<https://pgconf.ru/>
Назначение: Доклады по PostgreSQL, включая документацию на русском языке .
3. МойКруг - профессиональное сообщество
<https://moikrug.ru/>
Назначение: Изучение требований работодателей к компетенциям Java-разработчиков.
4. GitHub - русскоязычные README
<https://github.com/>
Назначение: Поиск open-source проектов с русскоязычной документацией, в том числе C4-PlantUML .
Иностранные ресурсы (свободно распространяемые)
5. Официальная документация Spring Boot
<https://docs.spring.io/spring-boot/>
Назначение: Полное руководство по Spring Boot, включая Web, Data, Security, Actuator .
6. Официальная документация React
<https://ru.react.dev/>
Назначение: Документация React на русском языке: компоненты, хуки, роутинг, управление состоянием .
7. Docker Documentation
<https://docs.docker.com/>
Назначение: Официальная документация Docker: установка, Dockerfile, Docker Compose .
8. PostgreSQL Documentation (русский перевод)
<https://postgrespro.ru/docs/postgresql>
Назначение: Полная документация PostgreSQL на русском языке .
9. RabbitMQ Tutorials

<https://www.rabbitmq.com/getstarted.html>

Назначение: Официальные руководства RabbitMQ с примерами на Java .

10. PlantUML Documentation

<https://plantuml.com/ru/>

Назначение: Документация PlantUML на русском языке: диаграммы классов, последовательности, C4-диаграммы .

11. GitHub Actions Documentation

<https://docs.github.com/ru/actions>

Назначение: Официальная документация GitHub Actions на русском языке .

12. Prometheus Documentation

<https://prometheus.io/docs/>

Назначение: Документация Prometheus: сбор метрик, настройка экспортеров.

13. Grafana Documentation

<https://grafana.com/docs/>

Назначение: Документация Grafana: создание дашбордов, визуализация метрик .

14. OpenAPI Specification (Swagger)

<https://swagger.io/specification/>

Назначение: Спецификация OpenAPI для документирования REST API .

9. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине (модулю), включая перечень программного обеспечения и информационных справочных систем

При чтении лекций используется компьютерная техника, демонстрации презентационных мультимедийных материалов. На семинарских и практических занятиях студенты представляют презентации, подготовленные ими в часы самостоятельной работы.

Информационные справочные системы:

Информационно-справочные и информационно-правовые системы, используемые при изучении дисциплины:

1. Правовые и нормативно-технические системы

- КонсультантПлюс – справочно-правовая система (доступ через компьютерные классы вуза)
- Гарант – справочно-правовая система (доступ через компьютерные классы вуза)
- Росстандарт (<https://www.rst.gov.ru>) – официальный сайт Федерального агентства по техническому регулированию и метрологии
- ФСТЭК России (<https://fstec.ru>) – официальный сайт Федеральной службы по техническому и экспортному контролю
- Электронный фонд правовых и нормативно-технических документов (<https://docs.cntd.ru>) – сборник ГОСТ, СНиП, СанПиН

2. Техническая документация

- GitHub Docs (<https://docs.github.com/ru>) – документация по GitHub и GitHub Actions на русском языке
- Docker Docs (<https://docs.docker.com>) – официальная документация Docker
- Spring Boot Documentation (<https://docs.spring.io/spring-boot>) – официальная документация Spring Boot

- React Documentation (<https://ru.react.dev>) – официальная документация React на русском языке
- PostgreSQL Documentation (<https://postgrespro.ru/docs>) – документация PostgreSQL на русском языке
- RabbitMQ Documentation (<https://www.rabbitmq.com/docs>) – официальная документация RabbitMQ
- Prometheus Documentation (<https://prometheus.io/docs>) – официальная документация Prometheus
- Grafana Documentation (<https://grafana.com/docs>) – официальная документация Grafana
- PlantUML Documentation (<https://plantuml.com/ru>) – документация PlantUML на русском языке
- OpenAPI Specification (<https://swagger.io/specification>) – спецификация OpenAPI
- Cypress Documentation (<https://docs.cypress.io>) – официальная документация Cypress

3. Профессиональные сообщества и блоги

- Habr (<https://habr.com/ru/hub/programming>) – публикации по программированию, архитектуре, DevOps
- Habr Q&A (<https://qna.habr.com>) – 问答 сообщество разработчиков
- Stack Overflow на русском (<https://ru.stackoverflow.com>) – 问答 сообщество
- Medium – публикации по React, Spring Boot, микросервисам (свободный доступ)

4. Агрегаторы документации

- DevDocs (<https://devdocs.io>) – агрегированная документация по API (Spring, React, Docker, PostgreSQL)
- GeeksforGeeks (<https://www.geeksforgeeks.org>) – учебные материалы по Java, Spring, React
- Baeldung (<https://www.baeldung.com>) – учебные материалы по Spring Boot, Java, тестированию

Программное обеспечение:

1. Среды разработки (IDE)

- IntelliJ IDEA Community Edition – разработка Java/Spring Boot приложений
- VS Code – разработка React-приложений, написание документации и скриптов
- Postman – тестирование REST API

2. Языки программирования и среды выполнения

- Java Development Kit (JDK) 17+ – компиляция и выполнение Java-кода
- Node.js – среда выполнения JavaScript для разработки React-приложений
- Python 3.10+ (опционально) – для написания ML-скриптов и тестов

3. Системы сборки и управления зависимостями

- Maven – сборка Java-проектов, управление зависимостями
- Gradle – альтернативная система сборки Java-проектов
- npm – менеджер пакетов для JavaScript/TypeScript
- Yarn – альтернативный менеджер пакетов для JavaScript/TypeScript
- Vite – сборщик frontend-проектов (React)

4. Системы контроля версий и платформы совместной разработки
 - Git – распределённая система контроля версий
 - GitHub Desktop – графический интерфейс для работы с Git
 - GitLab CE/EE – платформа для хранения репозитория и CI/CD
 - GitHub – платформа для хранения репозитория и CI/CD
5. Базы данных и инструменты для работы с данными
 - PostgreSQL 15+ – реляционная база данных
 - pgAdmin 4 – графический клиент для управления PostgreSQL
 - DBeaver – универсальный клиент для работы с базами данных
 - TimescaleDB – расширение PostgreSQL для хранения временных рядов
 - Redis – in-memory хранилище для кэширования и сессий
 - Redis Insight – графический клиент для управления Redis
6. Контейнеризация и оркестрация
 - Docker Desktop (Windows/macOS) – платформа для контейнеризации приложений
 - Docker Engine (Linux) – платформа для контейнеризации приложений
 - Docker Compose – оркестрация многоконтейнерных приложений
 - Docker Hub – реестр Docker-образов
7. Брокеры сообщений и очереди
 - RabbitMQ – брокер сообщений с плагином управления (Management UI)
8. Инструменты тестирования
 - JUnit 5 – фреймворк для модульного тестирования Java
 - Mockito – фреймворк для создания мок-объектов в Java
 - TestContainers – библиотека для тестирования с реальными контейнерами
 - AssertJ – библиотека для fluent assertions в Java
 - Jest – фреймворк для тестирования JavaScript/React
 - React Testing Library – библиотека для тестирования React-компонентов
 - Cypress – фреймворк для E2E-тестирования веб-приложений
 - Apache JMeter – инструмент для нагрузочного тестирования
 - k6 – инструмент для нагрузочного тестирования (опционально)
9. Инструменты профилирования и отладки
 - JProfiler – профилировщик Java-приложений (ознакомительная версия)
 - Async Profiler – профилировщик с низким оверхедом
 - Java Flight Recorder (JFR) – встроенный инструмент профилирования JDK
 - Browser DevTools – инструменты отладки фронтенда (встроены в браузеры)
10. Мониторинг и логирование
 - Prometheus – сбор метрик
 - Grafana – визуализация метрик и создание дашбордов
 - Loki – централизованное логирование
 - Promtail – агент для сбора логов в Loki
11. Инструменты документирования и моделирования
 - PlantUML – создание диаграмм (C4, UML) из текстового описания
 - Swagger UI / OpenAPI – интерактивная документация REST API
 - MkDocs – генератор статической документации из Markdown
 - Docusaurus – фреймворк для создания сайтов документации

12. Утилиты и вспомогательные инструменты

- curl – командная строка для HTTP-запросов
- jq – обработка JSON в командной строке
- OpenSSL – генерация SSL-сертификатов
- Certbot (Let's Encrypt) – получение бесплатных SSL-сертификатов
- Nginx – веб-сервер и reverse proxy

13. Операционные системы

- Windows 10/11 – для локальной разработки
- Ubuntu 22.04 LTS / 24.04 LTS – для серверного развертывания и локальной разработки
- macOS 12+ – для локальной разработки (при наличии техники)

14. Облачные платформы (опционально)

- Yandex Cloud – российская облачная платформа
- VK Cloud Solutions – российская облачная платформа
- SberCloud – российская облачная платформа
- AWS Free Tier – зарубежная облачная платформа (при наличии доступа)

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

Лекционные занятия	Учебная аудитория для проведения учебных занятий, оснащенная мультимедийным оборудованием и техническими средствами обучения.
Лабораторные работы	Учебная аудитория для проведения учебных занятий, оснащенная мультимедийным оборудованием и техническими средствами обучения.
Самостоятельная работа	Помещение для самостоятельной работы обучающихся оснащенное компьютерной техникой с возможностью подключения к сети "Интернет" и возможностью доступа к электронной информационно-образовательной среде университета

11. Особенности освоения дисциплины (модуля) лицами с ограниченными возможностями здоровья

Обучающимся с ограниченными возможностями здоровья предоставляются специальные учебники, учебные пособия и дидактические материалы, специальные технические средства обучения коллективного и индивидуального пользования, услуги ассистента (помощника), оказывающего обучающимся необходимую техническую помощь, а также услуги сурдопереводчиков и тифлосурдопереводчиков.

Освоение дисциплины (модуля) обучающимися с ограниченными возможностями здоровья может быть организовано совместно с другими обучающимися, а также в отдельных группах.

Освоение дисциплины (модуля) обучающимися с ограниченными возможностями здоровья осуществляется с учетом особенностей психофизического развития, индивидуальных возможностей и состояния здоровья.

В целях доступности получения высшего образования по образовательной программе лицами с ограниченными возможностями здоровья при освоении дисциплины (модуля) обеспечивается:

- 1) для лиц с ограниченными возможностями здоровья по зрению:

- присутствие ассистента, оказывающий студенту необходимую техническую помощь с учетом индивидуальных особенностей (помогает занять рабочее место, передвигаться, прочитать и оформить задание, в том числе, записывая под диктовку),
 - письменные задания, а также инструкции о порядке их выполнения оформляются увеличенным шрифтом,
 - специальные учебники, учебные пособия и дидактические материалы (имеющие крупный шрифт или аудиофайлы),
 - индивидуальное равномерное освещение не менее 300 люкс,
 - при необходимости студенту для выполнения задания предоставляется увеличивающее устройство;
- 2) для лиц с ограниченными возможностями здоровья по слуху:
- присутствие ассистента, оказывающий студенту необходимую техническую помощь с учетом индивидуальных особенностей (помогает занять рабочее место, передвигаться, прочитать и оформить задание, в том числе, записывая под диктовку),
 - обеспечивается наличие звукоусиливающей аппаратуры коллективного пользования, при необходимости обучающемуся предоставляется звукоусиливающая аппаратура индивидуального пользования;
 - обеспечивается надлежащими звуковыми средствами воспроизведения информации;
- 3) для лиц с ограниченными возможностями здоровья, имеющих нарушения опорно-двигательного аппарата (в том числе с тяжелыми нарушениями двигательных функций верхних конечностей или отсутствием верхних конечностей):
- письменные задания выполняются на компьютере со специализированным программным обеспечением или надиктовываются ассистенту;
 - по желанию студента задания могут выполняться в устной форме.

12. Особенности реализации дисциплины с применением дистанционных образовательных технологий и электронного обучения

Согласно части 1 статьи 16 Федерального закона от 29 декабря 2012 г. № 273-ФЗ «Об образовании в Российской Федерации» под *электронным обучением* понимается организация образовательной деятельности с применением содержащейся в базах данных и используемой при реализации образовательных программ информации и обеспечивающих ее обработку информационных технологий, технических средств, а также информационно-телекоммуникационных сетей, обеспечивающих передачу по линиям связи указанной информации, взаимодействие обучающихся и педагогических работников. Под *дистанционными образовательными технологиями* понимаются образовательные технологии, реализуемые в основном с применением информационно-телекоммуникационных сетей при опосредованном (на расстоянии) взаимодействии обучающихся и педагогических работников.

Реализация дисциплины может быть осуществлена с применением дистанционных образовательных технологий и электронного обучения полностью или частично. Компоненты УМК дисциплины (рабочая программа дисциплины, оценочные и методические материалы, формы аттестации), реализуемой с применением дистанционных образовательных технологий и электронного обучения, содержат указание на их использование.

При организации образовательной деятельности с применением дистанционных образовательных технологий и электронного обучения могут предусматриваться асинхронный и синхронный способы осуществления взаимодействия участников образовательных отношений посредством информационно-телекоммуникационной сети «Интернет».

При применении дистанционных образовательных технологий и электронного обучения в расписании по дисциплине указываются: способы осуществления

взаимодействия участников образовательных отношений посредством информационно-телекоммуникационной сети «Интернет» (ВКС-видеоконференцсвязь, ЭТ – электронное тестирование); ссылки на электронную информационно-образовательную среду СКФУ, на образовательные платформы и ресурсы иных организаций, к которым предоставляется открытый доступ через информационно-телекоммуникационную сеть «Интернет»; для синхронного обучения - время проведения онлайн-занятий и преподаватели; для асинхронного обучения - авторы онлайн-курсов.

При организации промежуточной аттестации с применением дистанционных образовательных технологий и электронного обучения используются Методические рекомендации по применению технических средств, обеспечивающих объективность результатов при проведении промежуточной и государственной итоговой аттестации по образовательным программам высшего образования - программам бакалавриата, программам специалитета и программам магистратуры с применением дистанционных образовательных технологий (Письмо Минобрнауки России от 07.12.2020 г. № МН-19/1573-АН "О направлении методических рекомендаций").

Реализация дисциплины с применением электронного обучения и дистанционных образовательных технологий осуществляется с использованием электронной информационно-образовательной среды СКФУ, к которой обеспечен доступ обучающихся через информационно-телекоммуникационную сеть «Интернет», или с использованием ресурсов иных организаций, в том числе платформ, предоставляющих сервисы для проведения видеоконференций, онлайн-встреч и дистанционного обучения (МТС-Линк), а также с использованием возможностей социальных сетей для осуществления коммуникации обучающихся и преподавателей.

Учебно-методическое обеспечение дисциплины, реализуемой с применением электронного обучения и дистанционных образовательных технологий, включает представленные в электронном виде рабочую программу, учебно-методические пособия или курс лекций, методические указания к выполнению различных видов учебной деятельности обучающихся, предусмотренных дисциплиной, и прочие учебно-методические материалы, размещенные в информационно-образовательной среде СКФУ.