МИНИСТЕРСТВО ТРАНСПОРТА РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«РОССИЙСКИЙ УНИВЕРСИТЕТ ТРАНСПОРТА»
(РУТ (МИИТ)

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

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ (МОДУЛЯ)

Программирование на Java

Направление подготовки: 09.03.01 Информатика и вычислительная техника

Направленность (профиль): Цифровая инженерия транспортных процессов

Форма обучения: Очная

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

Простая электронная подпись, выданная ФГАОУ ВО РУТ (МИИТ)
ID подписи: 937226
Подписал: руководитель образовательной программы Проневич Ольга Борисовна
Дата: 22.10.2026

1. Общие сведения о дисциплине (модуле).

Целями освоения учебной дисциплины являются формирование у студентов базы знаний и навыков в области программирования на языке высокого уровня Java, а также формирование и закрепление у студентов компетенций в области прикладной информатики для решения следующих профессиональных задач проектного вида деятельности:

- ознакомление студентов с современным представлением о Java-технологиях, применяемых при разработке прикладных и серверных программных систем;
- изучение языка программирования Java в составе платформы Java SE, включая основные средства объектно-ориентированного программирования, работу с коллекциями, исключениями, файлами и потоками выполнения;
- изучение основ использования JDK и инструментов сборки при проектировании, разработке и сопровождении Java-приложений;
- использование интегрированной среды разработки программных проектов (IDE) для проектирования, отладки, тестирования и документирования Java-приложений;
- обеспечение качества в проектах в области ИТ за счёт применения модульного тестирования, обработки ошибок, структурирования кода и соблюдения установленных требований к разработке;
- распространение и представление результатов разработки в проектах в области ИТ, включая подготовку программного кода, документации и демонстрацию работы приложения в соответствии с трудовым заданием.

2. Планируемые результаты обучения по дисциплине (модулю).

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

ПК-2 - Способен осуществлять концептуальное, функциональное и логическое проектирование систем среднего и крупного масштаба и сложности ;

ПК-7 - Способен организовывать логистическую деятельность по перевозке грузов в цепи поставок.

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

Знать:

- архитектуру платформы Java, назначение JDK, JVM, среды разработки, средств сборки и управления зависимостями Java-проектов;

- синтаксис языка Java, систему типов, особенности работы памяти, принципы обработки исключений и организации модульной структуры приложения;

- принципы объектно-ориентированного проектирования, SOLID, GRASP, базовые и архитектурные шаблоны проектирования, включая MVC, слоистую, чистую и гексагональную архитектуру;

- средства стандартной библиотеки Java для работы со строками, коллекциями, обобщениями, файлами, потоками ввода-вывода, сериализацией, XML, JSON и Protobuf;

- основы многопоточности, конкурентного программирования, потокобезопасных коллекций, асинхронной и реактивной обработки данных;

- принципы работы с реляционными базами данных в Java: SQL, JDBC, R2DBC, ORM, JPA, Hibernate, транзакции, миграции схемы и оптимизация запросов;

- основы Spring Framework и Spring Boot: IoC/DI, Bean, ApplicationContext, REST API, DTO, валидация, обработка ошибок, безопасность и структура backend-приложения;

- подходы к тестированию Java- и Spring Boot-приложений с использованием JUnit, Mockito, Spring Boot Test и Testcontainers;

- принципы контейнеризации и развёртывания Java-приложений с использованием Docker, Docker Compose, профилей конфигурации, переменных окружения и нативной сборки;

- современные подходы к построению backend-систем: микросервисы, Spring Cloud, синхронная и асинхронная коммуникация, gRPC, GraphQL, WebSocket, Kafka, RabbitMQ, кэширование, наблюдаемость и базовая интеграция интеллектуальных сервисов.

Уметь:

- настраивать рабочее окружение Java-разработчика, создавать, собирать, запускать и сопровождать Java-проекты с использованием современных IDE и систем сборки;

- проектировать классы, интерфейсы, пакеты и модули приложения, выполнять объектно-ориентированную декомпозицию предметной области и обосновывать архитектурные решения;

- разрабатывать консольные и backend-приложения на Java, реализовывать обработку данных, работу с файлами, сериализацию, чтение и запись структурированных форматов данных;

- применять коллекции, обобщённые типы, регулярные выражения, исключения, потоки ввода-вывода и механизмы многопоточности для решения прикладных задач;

- реализовывать аннотированные и плагинные механизмы расширения приложения с использованием reflection, annotation processor и общих интерфейсов;

- проектировать и реализовывать REST API на Spring Boot с разделением на controller, service и repository слои, использованием DTO, валидации, централизованной обработки ошибок и базовой защиты приложения;

- организовывать хранение и обработку данных в PostgreSQL, использовать SQL, JDBC, R2DBC, JPA/Hibernate и repository-слой для доступа к данным;

- разрабатывать асинхронные задачи обработки данных, применять кэширование и оптимизировать чтение отчётов и результатов анализа;

- писать unit-, интеграционные и web-тесты для Java- и Spring Boot-приложений, использовать мокирование зависимостей и тестовые контейнеры внешних сервисов;

- контейнеризировать приложение и базу данных, запускать их через Docker Compose, настраивать профили окружения, health check, логирование, метрики и базовую наблюдаемость;

- моделировать и реализовывать прикладные сценарии обработки данных, в том числе данные транспортно-логистических и производственных процессов, с формированием отчётов и аналитических выводов.

Владеть:

- навыками разработки программ на Java с применением процедурного, объектно-ориентированного, обобщённого и модульного подходов;

- инструментами разработки, сборки, управления зависимостями и сопровождения Java-проектов в командной и индивидуальной работе;

- приёмами проектирования устойчивой структуры приложения, выделения доменной логики, инфраструктурного слоя и интерфейсов взаимодействия;

- навыками работы со стандартной библиотекой Java, коллекциями, потоками ввода-вывода, сериализацией, форматами XML/JSON/Protobuf и обработкой больших наборов данных;

- средствами многопоточной, асинхронной и реактивной обработки задач, а также методами обеспечения потокобезопасности и управляемости выполнения;

- практическими навыками создания Spring Boot-приложений, REST API, сервисного и repository-слоёв, механизмов валидации, обработки ошибок и тестирования;

- инструментами доступа к данным и сопровождения базы данных: SQL, PostgreSQL, JDBC, R2DBC, JPA/Hibernate, Flyway или Liquibase;

- методами автоматизированного тестирования, включая JUnit, Mockito, Spring Boot Test и Testcontainers, а также навыками анализа результатов тестирования;

- инструментами контейнеризации, конфигурирования и локального развёртывания Java-приложений с использованием Docker и Docker Compose;

- средствами повышения эксплуатационной готовности backend-приложений: кэшированием, структурированным логированием, метриками, трассировкой, health check и мониторингом;

- навыками подготовки, документирования и защиты программного решения, включая описание архитектуры, API, сценариев запуска и результатов анализа данных.

3. Объем дисциплины (модуля).

3.1. Общая трудоемкость дисциплины (модуля).

Общая трудоемкость дисциплины (модуля) составляет 8 з.е. (288 академических часа(ов).

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

Тип учебных занятий	Количество часов		
	Всего	Семестр	
		№5	№6
Контактная работа при проведении учебных занятий (всего):	144	64	80
В том числе:			
Занятия лекционного типа	64	32	32
Занятия семинарского типа	80	32	48

3.3. Объем дисциплины (модуля) в форме самостоятельной работы обучающихся, а также в форме контактной работы обучающихся с педагогическими работниками и (или) лицами, привлекаемыми к реализации образовательной программы на иных условиях, при проведении промежуточной аттестации составляет 144 академических часа (ов).

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

4. Содержание дисциплины (модуля).

4.1. Занятия лекционного типа.

№ п/п	Тематика лекционных занятий / краткое содержание
1	Тема 1. Введение в Java и инструменты разработки. Рассматриваемые вопросы: - области применения Java, роль JVM и этапы выполнения Java-программы; - настройка JDK, IDE и системы сборки; подключение зависимостей; - структура первой программы: классы, методы, переменные, запуск приложения.
2	Тема 2. Работа памяти в Java. Рассматриваемые вопросы: - области памяти Stack, Heap и Metaspace; - примитивные и ссылочные типы, создание объектов и передача ссылок; - механизм Garbage Collector, типовые ошибки NullPointerException и утечки памяти
3	Тема 3. Объектно-ориентированное программирование, классы и коллекции. Рассматриваемые вопросы: - классы, объекты, поля, методы, конструкторы и модификаторы доступа; - инкапсуляция, наследование, полиморфизм и абстракция; - строки и иммутабельность: String, StringBuilder, StringBuffer; - коллекции List, Set, Map, контракт equals/hashCode, Comparable и Comparator.
4	Тема 4. Паттерны проектирования на Java и применение принципов ООП. Рассматриваемые вопросы: - назначение паттернов проектирования и связь паттернов с принципами ООП; - порождающие паттерны: Singleton, Factory Method, Abstract Factory, Builder; - структурные и поведенческие паттерны: Adapter, Decorator, Facade, Strategy и др.; - выбор паттерна и оценка влияния паттернов на сопровождаемость кода.
5	Тема 5. Работа с файлами, потоками, Stream API и исключениями. Рассматриваемые вопросы: - чтение и запись файлов, абсолютные и относительные пути; - байтовые и символьные потоки, Files, Path и NIO.2; - проверяемые и непроверяемые исключения, try-catch-finally, try-with-resources; - Stream API для обработки коллекций и файловых данных.
6	Тема 6. Сериализация объектов и форматы обмена данными: XML, JSON, Protobuf. Рассматриваемые вопросы: - сериализация и десериализация объектов, отличие объекта в памяти от представления в файле; - текстовые и бинарные форматы обмена данными; - работа с XML, JSON и Protobuf в Java-приложениях; - практические сценарии хранения и передачи данных между приложениями.

№ п/п	Тематика лекционных занятий / краткое содержание
7	Тема 7. Многопоточность, конкурентность и реактивное программирование. Рассматриваемые вопросы: - потоки выполнения, Runnable, Thread, ExecutorService; - конкурентность и параллелизм, синхронизация и проблемы потокобезопасности; - потокобезопасные коллекции и средства java.util.concurrent; - основы реактивного подхода и неблокирующей обработки задач.
8	Тема 8. Рефлексия и работа с базами данных: SQL, JDBC, R2DBC. Рассматриваемые вопросы: - рефлексия: Class, поля, методы, конструкторы и создание объектов во время выполнения; - основы SQL и взаимодействие Java-приложения с реляционной базой данных; - JDBC: подключение, запросы, ResultSet, транзакции; - R2DBC и реактивный доступ к данным.
9	Тема 9. Процессоры аннотаций, Lombok и генерация кода. Рассматриваемые вопросы: - аннотации Java: runtime- и compile-time-аннотации; - Annotation Processing API и генерация исходного кода на этапе компиляции; - использование Lombok и анализ преимуществ/ограничений генерации кода; - размещение generated-sources и подключение сгенерированных классов к проекту.
10	Тема 10. Spring Framework, Spring Boot и Dependency Injection. Рассматриваемые вопросы: - IoC и Dependency Injection как основа Spring-приложений; - Spring Bean, ApplicationContext и жизненный цикл компонентов; - конфигурация Spring Boot-приложения, автоконфигурация и starter-зависимости; - организация слоёв приложения и внедрение зависимостей в сервисы.
11	Тема 11. Контейнеризация Spring Boot-приложений, Docker, Docker Compose, GraalVM. Рассматриваемые вопросы: - контейнеризация и отличие контейнера от виртуальной машины; - Dockerfile, Docker Image, Docker Container и основные команды Docker; - запуск Spring Boot-приложения и инфраструктуры через Docker Compose; - Spring Boot Compose Support, GraalVM и нативная сборка Java-приложений.
12	Тема 11. Контейнеризация Spring Boot-приложений, Docker, Docker Compose, GraalVM. Рассматриваемые вопросы: - контейнеризация и отличие контейнера от виртуальной машины; - Dockerfile, Docker Image, Docker Container и основные команды Docker; - запуск Spring Boot-приложения и инфраструктуры через Docker Compose; - Spring Boot Compose Support, GraalVM и нативная сборка Java-приложений.
13	Тема 12. Тестирование Java-приложений: JUnit, Mockito, Spring Boot Test, Testcontainers. Рассматриваемые вопросы: - виды тестирования: unit, integration, end-to-end, пирамида тестирования; - JUnit и структура тестового класса; - Mockito и изоляция зависимостей в unit-тестах; - Spring Boot Test и Testcontainers для интеграционных тестов.
14	Тема 13. Backend-разработка на Spring Boot: REST API, архитектура и безопасность. Рассматриваемые вопросы: - клиент-серверное взаимодействие, HTTP и принципы REST API;

№ п/п	Тематика лекционных занятий / краткое содержание
	- HTTP-методы, endpoint, request/response, status codes; - контроллеры, сервисы, DTO и валидация входных данных; - базовые подходы к защите backend-приложений.
15	Тема 14. Архитектурные паттерны backend-приложений: Hexagonal Architecture, Clean Architecture, DDD. Рассматриваемые вопросы: - разделение бизнес-логики и инфраструктуры; - Domain Layer, Application Layer, Infrastructure Layer; - гексагональная архитектура, порты и адаптеры; - принципы чистой архитектуры и предметно-ориентированного проектирования.
16	Тема 15. Микросервисы и Spring Cloud: Gateway, Config, Discovery и отказоустойчивость. Рассматриваемые вопросы: - монолитная и микросервисная архитектура, границы микросервисов; - Spring Cloud Gateway, Config Server и Service Discovery; - межсервисное взаимодействие и проблемы распределённых систем; - отказоустойчивость, retry, timeout, circuit breaker
17	Тема 16. Основы ORM, JPA и сущностей. Рассматриваемые вопросы: - объектно-реляционное отображение и назначение ORM; - JPA: сущности, идентификаторы, наследование и коллекции; - отношения между сущностями и ассоциации; - преимущества и ограничения ORM-подхода.
18	Тема 18. Запросы, оптимизация Hibernate и интеграция со Spring. Рассматриваемые вопросы: - HQL, JPQL, Query API и Criteria API; - аспекты производительности Hibernate и оптимизация выборок; - Hibernate Reactive и особенности неблокирующего доступа к данным; - интеграция JPA/Hibernate со Spring, DAO и управление транзакциями.
19	Тема 19. Наблюдаемость backend-приложений: логирование, метрики, трассировка. Рассматриваемые вопросы: - Observability и Monitoring, основные сигналы: logs, metrics, traces; - уровни логирования и правила безопасного логирования; - метрики приложения и технические метрики инфраструктуры; - OpenTelemetry, Prometheus и Grafana для наблюдаемости backend-систем.
20	Тема 20. gRPC, GraphQL и WebSocket как альтернативы REST API. Рассматриваемые вопросы: - ограничения REST API и выбор способа коммуникации; - gRPC и Protocol Buffers; - GraphQL: схема, запросы и преимущества для клиентских приложений; - WebSocket и realtime-коммуникация.
21	Тема 21. Асинхронная коммуникация: Kafka, RabbitMQ, события и очереди сообщений. Рассматриваемые вопросы: - синхронное и асинхронное взаимодействие приложений; - message broker, очередь сообщений, события и команды; - основы Kafka и RabbitMQ; - event-driven architecture и обработка фоновых задач.
22	Тема 22. Сборка и управление зависимостями Java-проектов. Рассматриваемые вопросы:

№ п/п	Тематика лекционных занятий / краткое содержание
	- назначение build tool и задачи сборки Java-проекта; - зависимости, transitive dependencies и репозитории зависимостей; - Maven и Gradle: структура проекта, lifecycle/tasks, плагины; - управление версиями, профили сборки и подготовка артефактов.
23	Тема 23. Кэширование и производительность backend-систем. Рассматриваемые вопросы: - показатели производительности: latency, throughput, bottleneck; - профилирование и поиск узких мест в приложении; - кэширование, Spring Cache и Redis; - стратегии инвалидации кэша и оценка эффекта оптимизации.
24	Тема 24. Миграции базы данных и управление схемой. Рассматриваемые вопросы: - версионирование схемы базы данных и жизненный цикл миграций; - Flyway и Liquibase, changelog и schema version; - проблемы ручного изменения схемы в production; - zero-downtime изменения и безопасное обновление БД.
25	Тема 25. Spring AI: интеграция LLM, RAG, embeddings, vector store в Spring Boot. Рассматриваемые вопросы: - назначение Spring AI и сценарии применения AI в Spring Boot-приложениях; - LLM, AI model provider, ChatModel и ChatClient; - embeddings, vector store и поиск по смысловому представлению данных; - RAG-подход и интеграция внешних источников знаний в приложение.

4.2. Занятия семинарского типа.

Практические занятия

№ п/п	Тематика практических занятий/краткое содержание
1	Тема 1. Анализ входных данных по варианту. Рассматриваемые вопросы: - создание консольного Java-приложения и проекта через Gradle; - чтение строк из System.in и разбор данных по варианту; - преобразование корректных строк в Java-объекты, обработка некорректных строк; - расчёт статистики и вывод итогового отчёта в консоль.
2	Тема 2. Рефакторинг анализатора данных через ООП и паттерны. Рассматриваемые вопросы: - выделение модели данных, parser, analyzer и report builder; - использование enum, коллекций и иммутабельных моделей; - применение паттернов Facade и Strategy; - обновление структуры проекта и документации README.
3	Тема 3. Сохранение результата анализа в JSON и Java Serialization. Рассматриваемые вопросы: - сохранение распарсенных данных в JSON-файл; - сериализация итогового отчёта стандартными средствами Java; - чтение ранее сохранённого отчёта и восстановление объекта; - реализация режимов запуска через параметры командной строки.
4	Тема 4. Загрузка большого JSON в PostgreSQL через JDBC, реактивный драйвер и парсинг JSON в БД. Рассматриваемые вопросы:

№ п/п	Тематика практических занятий/краткое содержание
	- потоковая обработка JSON без полной загрузки в память; - загрузка данных через JDBC и реактивный драйвер; - загрузка JSON в PostgreSQL и парсинг данных на стороне БД; - измерение производительности, хранение статуса набора данных и формирование отчёта.
5	Тема 5. Аннотированная плагинная система анализаторов. Рассматриваемые вопросы: - создание интерфейса плагина анализатора и собственной аннотации; - использование annotation processor для поиска и регистрации анализаторов; - автоматическое подключение расчётов без ручного изменения основного кода; - расширение отчёта результатами найденных плагинов.
6	Тема 6. Unit-тестирование аннотированной плагинной системы. Рассматриваемые вопросы: - написание unit-тестов для parser, analyzer, plugin registry и report builder; - использование JUnit и Mockito для проверки бизнес-логики; - проверка работы аннотаций и подключаемых анализаторов; - контроль корректной обработки ошибок и некорректных строк.
7	Тема 7. Базовое Spring Boot-приложение и тестирование. Рассматриваемые вопросы: - перенос консольного анализатора в Spring Boot-приложение; - создание controller, service и configuration-слоёв; - приём данных через HTTP и возврат результата анализа; - покрытие web- и service-слоёв тестами.
8	Тема 8. REST API, DTO, валидация и обработка ошибок. Рассматриваемые вопросы: - разделение внутренних моделей и request/response DTO; - валидация входных данных и формирование понятных сообщений об ошибках; - единая обработка исключений через RestControllerAdvice; - тестирование controller- и service-слоёв.
9	Тема 9. Взаимосвязи сущностей в JPA. Рассматриваемые вопросы: - определение сущности и идентификатора; - определение отношений между сущностями (@OneToMany, @ManyToOne, @ManyToMany); - каскадные операции и lazy/eager загрузка.
10	Тема 10. Иерархия классов сущностей. Рассматриваемые вопросы: - стратегии наследования: SINGLE_TABLE, TABLE_PER_CLASS, JOINED; - отображение абстрактных классов и полиморфные запросы; - выбор подходящей стратегии для предметной области.
11	Тема 11. Работа с EntityManager. Рассматриваемые вопросы: - реализация методов getCompany(int), saveCompany(Company); - реализация методов init(), merge, remove; - жизненный цикл сущностей (new, managed, detached, removed).
12	Тема 12. Работа с JPQL и Criteria API. Рассматриваемые вопросы: - возможности JPA для поиска сущностей: JPQL и Criteria API; - параметризованные запросы, агрегатные функции, Join; - интеграция с фреймворком Spring (@PersistenceContext).
13	Тема 13. Разработка приложения для блогов. Рассматриваемые вопросы:

№ п/п	Тематика практических занятий/краткое содержание
	- описание сущностей: DomainObject, BlogPost, Comment; - реализация сервисных объектов: BlogServiceImpl, BlogPostServiceImpl; - конфигурация контейнера Spring и Persistence Unit.
14	Тема 14. PostgreSQL и repository-слой. Рассматриваемые вопросы: - подключение Spring Boot-приложения к PostgreSQL; - создание таблиц и сохранение запусков анализа, результатов и проблемных строк; - реализация repository-слоя и методов получения отчётов по id и списком; - проверка repository/service-слоёв тестами.
15	Тема 15. Асинхронные задачи анализа. Рассматриваемые вопросы: - перевод запуска анализа в асинхронный режим; - создание задачи анализа с jobId и статусами NEW, PROCESSING, COMPLETED, FAILED; - получение статуса задачи и результата анализа по API; - обработка ошибок и корректное обновление статусов.
16	Тема 16. Docker Compose для приложения и базы данных. Рассматриваемые вопросы: - создание Dockerfile для Spring Boot-приложения; - описание compose.yml для приложения и PostgreSQL; - настройка подключения через environment variables и Docker-профиль; - проверка работы API при запуске через Docker Compose.
17	Тема 17. Кэширование и оптимизация чтения отчётов. Рассматриваемые вопросы: - кэширование повторных запросов отчёта по id; - использование локального кэша или Redis/Spring Cache; - инвалидация или обновление кэша при создании новых отчётов; - сравнение времени ответа до и после кэширования.
18	Тема 18. Телеметрия, сбор логов и наблюдаемость приложения. Рассматриваемые вопросы: - структурированное логирование и correlationId/requestId; - логирование старта, завершения и ошибок анализа; - измерение времени выполнения и сбор базовых метрик; - health check и endpoint для просмотра простой статистики приложения.
19	Тема 19. Интеграционное тестирование с Testcontainers. Рассматриваемые вопросы: - запуск PostgreSQL в контейнере для тестов; - тестирование repository и service слоёв с реальной БД; - проверка миграций и корректности работы с транзакциями.
20	Тема 20. Пагинация и сортировка в REST API. Рассматриваемые вопросы: - использование Pageable и Page из Spring Data; - кастомные запросы с сортировкой; - добавление параметров page, size, sort в контроллеры.
21	Тема 21. Обработка больших файлов с использованием Stream API и NIO.2. Рассматриваемые вопросы: - чтение многогигабайтных файлов; - параллельная обработка строк; - использование Files.lines() и параллельных стримов.
22	Тема 22. Реализация паттерна Observer / Event Listener в Spring. Рассматриваемые вопросы:

№ п/п	Тематика практических занятий/краткое содержание
	- создание событий и слушателей; - асинхронная обработка событий (@Async); - слабое связывание компонентов.
23	Тема 23. Создание собственного Spring Boot стартера. Рассматриваемые вопросы: - автоконфигурация и conditional-аннотации; - создание мета-информации; - тестирование стартера.
24	Тема 24. Реактивные запросы с WebClient. Рассматриваемые вопросы: - неблокирующий вызов внешних API; - обработка ошибок, retry, timeout; - объединение нескольких реактивных вызовов.
25	Тема 25. Миграции схемы базы данных с Flyway / Liquibase. Рассматриваемые вопросы: - создание миграций; - откат (undo) и проверка состояния; - интеграция с Spring Boot.
26	Тема 26. Мониторинг с Micrometer, Prometheus и Grafana. Рассматриваемые вопросы: - экспорт метрик приложения; - создание кастомных счетчиков и гистограмм; - визуализация в Grafana.
27	Тема 27. Создание REST-клиента на Feign / RestClient. Рассматриваемые вопросы: - декларативный вызов внешних сервисов; - настройка логов, ретраев, error decoder; - интеграция с Spring Cloud.
28	Тема 28. Защита REST API с помощью Spring Security и JWT. Рассматриваемые вопросы: - аутентификация и авторизация; - выдача и проверка JWT-токенов; - защита эндпоинтов.
29	Тема 29. Graceful shutdown и health indicators. Рассматриваемые вопросы: - настройка корректного завершения приложения; - кастомные индикаторы здоровья; - подготовка к оркестрации.
30	Тема 30. Профилирование приложения. Рассматриваемые вопросы: - использование JProfiler / Async Profiler; - поиск узких мест по памяти и CPU; - оптимизация на основе профилирования.
31	Тема 31. Отправка email-уведомлений. Рассматриваемые вопросы: - настройка JavaMailSender; - создание шаблонов писем; - асинхронная отправка.
32	Тема 32. Итоговый проект: многомодульный Spring Boot приложение. Рассматриваемые вопросы:

№ п/п	Тематика практических занятий/краткое содержание
	- разделение на модули (api, core, infrastructure); - настройка сборки Gradle/Maven; - сборка Docker-образа и запуск всего стека.

4.3. Самостоятельная работа обучающихся.

№ п/п	Вид самостоятельной работы
1	Работа с учебной литературой и официальной документацией Java, Spring Boot, JPA/Hibernate, Docker и PostgreSQL.
2	Самостоятельное изучение материалов лекций и разбор примеров кода по темам дисциплины.
3	Выполнение сквозного проектного задания по разработке Java/Spring Boot-приложения.
4	Подготовка отчётов по практическим заданиям, оформление README и инструкции по запуску проекта.
5	Разработка и запуск unit-, integration- и repository-тестов.
6	Проведение самостоятельных замеров производительности, анализа логов, метрик и результатов кэширования.
7	Подготовка к промежуточной аттестации.
8	Подготовка к текущему контролю.

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

№ п/п	Библиографическое описание	Место доступа
1	Кожомбердиева, Г. И. Программирование на языке Java: многопоточные приложения : учебное пособие / Г. И. Кожомбердиева. — Санкт-Петербург : ПГУПС, 2012. — 44 с. — ISBN 978-7641-0401-0	https://e.lanbook.com/book/64399
2	Риз, Р. Обработка естественного языка на Java : учебное пособие / Р. Риз ; перевод с английского А. В. Снастина. — Москва : ДМК Пресс, 2016. — 264 с. — ISBN 978-5-97060-331-4	https://e.lanbook.com/book/93272
3	Юре, Л. Анализ больших наборов данных / Л. Юре, Р. Ананд, Д. У. Джеффри ; перевод с английского А. А. Слинкин. — Москва : ДМК Пресс, 2016. — 498 с. — ISBN 978-5-97060-190-7	https://e.lanbook.com/book/93571

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

http://library.miit.ru/ - электронно-библиотечная система Научно-технической библиотеки МИИТ

https://e.lanbook.com/ - электронно-библиотечная система Лань

http://opencsv.sourceforge.net/ - библиотека основных команд OpenCV

https://spring.io/quickstart - руководство фреймворк Spring

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

Java EE 7 SD

Microsoft Office 2007

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

Компьютер преподавателя

Компьютеры студентов

экран для проектора, маркерная доска,

Проектор

9. Форма промежуточной аттестации:

Зачет в 5, 6 семестрах.

10. Оценочные материалы.

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

Авторы:

директор	Б.В. Игольников

Согласовано:

Директор	Д.В. Паринов
Руководитель образовательной программы	О.Б. Проневич
Председатель учебно-методической комиссии	Д.В. Паринов