

**МИНИСТЕРСТВО ТРАНСПОРТА РОССИЙСКОЙ ФЕДЕРАЦИИ**  
**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ**  
**УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**  
**«РОССИЙСКИЙ УНИВЕРСИТЕТ ТРАНСПОРТА»**  
**(РУТ (МИИТ))**



Рабочая программа дисциплины (модуля),  
как компонент образовательной программы  
базового высшего образования  
по направлению подготовки  
09.03.01 Информатика и вычислительная техника,  
утвержденной директором РУТ (МИИТ)  
Париновым Д.В.

**РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ (МОДУЛЯ)**

**Программирование на Java**

Направление подготовки: 09.03.01 Информатика и вычислительная техника

Направленность (профиль): IT-сервисы и технологии обработки данных на транспорте

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 РУТ (МИИТ)  
ID подписи: 937226  
Подписал: руководитель образовательной программы  
Проневич Ольга Борисовна  
Дата: 10.06.2026

## 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):

**ОПК-3** - Способен использовать современные информационные технологии и программно-аппаратные средства, в том числе отечественного производства, при решении задач профессиональной деятельности;

**ПК-1** - Способен анализировать большие данные с использованием существующей в организации методологической и технологической инфраструктуры.

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 плагиновые 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):	192	96	96
В том числе:			
Занятия лекционного типа	64	32	32
Занятия семинарского типа	128	64	64

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 96 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	<p>Тема 7. Многопоточность, конкурентность и реактивное программирование.</p> <p>Рассматриваемые вопросы:</p> <ul style="list-style-type: none"> <li>- потоки выполнения, Runnable, Thread, ExecutorService;</li> <li>- конкурентность и параллелизм, синхронизация и проблемы потокобезопасности;</li> <li>- потокобезопасные коллекции и средства java.util.concurrent;</li> <li>- основы реактивного подхода и неблокирующей обработки задач.</li> </ul>
8	<p>Тема 8. Рефлексия и работа с базами данных: SQL, JDBC, R2DBC.</p> <p>Рассматриваемые вопросы:</p> <ul style="list-style-type: none"> <li>- рефлексия: Class, поля, методы, конструкторы и создание объектов во время выполнения;</li> <li>- основы SQL и взаимодействие Java-приложения с реляционной базой данных;</li> <li>- JDBC: подключение, запросы, ResultSet, транзакции;</li> <li>- R2DBC и реактивный доступ к данным.</li> </ul>
9	<p>Тема 9. Процессоры аннотаций, Lombok и генерация кода.</p> <p>Рассматриваемые вопросы:</p> <ul style="list-style-type: none"> <li>- аннотации Java: runtime- и compile-time-аннотации;</li> <li>- Annotation Processing API и генерация исходного кода на этапе компиляции;</li> <li>- использование Lombok и анализ преимуществ/ограничений генерации кода;</li> <li>- размещение generated-sources и подключение сгенерированных классов к проекту.</li> </ul>
10	<p>Тема 10. Spring Framework, Spring Boot и Dependency Injection.</p> <p>Рассматриваемые вопросы:</p> <ul style="list-style-type: none"> <li>- IoC и Dependency Injection как основа Spring-приложений;</li> <li>- Spring Bean, ApplicationContext и жизненный цикл компонентов;</li> <li>- конфигурация Spring Boot-приложения, автоконфигурация и starter-зависимости;</li> <li>- организация слоёв приложения и внедрение зависимостей в сервисы.</li> </ul>
11	<p>Тема 11. Контейнеризация Spring Boot-приложений, Docker, Docker Compose, GraalVM.</p> <p>Рассматриваемые вопросы:</p> <ul style="list-style-type: none"> <li>- контейнеризация и отличие контейнера от виртуальной машины;</li> <li>- Dockerfile, Docker Image, Docker Container и основные команды Docker;</li> <li>- запуск Spring Boot-приложения и инфраструктуры через Docker Compose;</li> <li>- Spring Boot Compose Support, GraalVM и нативная сборка Java-приложений.</li> </ul>
12	<p>Тема 12. Тестирование Java-приложений: JUnit, Mockito, Spring Boot Test, Testcontainers.</p> <p>Рассматриваемые вопросы:</p> <ul style="list-style-type: none"> <li>- виды тестирования: unit, integration, end-to-end, пирамида тестирования;</li> <li>- JUnit и структура тестового класса;</li> <li>- Mockito и изоляция зависимостей в unit-тестах;</li> <li>- Spring Boot Test и Testcontainers для интеграционных тестов.</li> </ul>
13	<p>Тема 13. Backend-разработка на Spring Boot: REST API, архитектура и безопасность.</p> <p>Рассматриваемые вопросы:</p> <ul style="list-style-type: none"> <li>- клиент-серверное взаимодействие, HTTP и принципы REST API;</li> <li>- HTTP-методы, endpoint, request/response, status codes;</li> <li>- контроллеры, сервисы, DTO и валидация входных данных;</li> <li>- базовые подходы к защите backend-приложений.</li> </ul>
14	<p>Тема 14. Архитектурные паттерны backend-приложений: Hexagonal Architecture, Clean Architecture, DDD.</p> <p>Рассматриваемые вопросы:</p> <ul style="list-style-type: none"> <li>- разделение бизнес-логики и инфраструктуры;</li> </ul>

№ п/п	Тематика лекционных занятий / краткое содержание
	<ul style="list-style-type: none"> <li>- Domain Layer, Application Layer, Infrastructure Layer;</li> <li>- гексагональная архитектура, порты и адаптеры;</li> <li>- принципы чистой архитектуры и предметно-ориентированного проектирования.</li> </ul>
15	<p>Тема 15. Микросервисы и Spring Cloud: Gateway, Config, Discovery и отказоустойчивость.</p> <p>Рассматриваемые вопросы:</p> <ul style="list-style-type: none"> <li>- монолитная и микросервисная архитектура, границы микросервисов;</li> <li>- Spring Cloud Gateway, Config Server и Service Discovery;</li> <li>- межсервисное взаимодействие и проблемы распределённых систем;</li> <li>- отказоустойчивость, retry, timeout, circuit breaker.</li> </ul>
16	<p>Тема 16. Основы ORM, JPA и сущностей.</p> <p>Рассматриваемые вопросы:</p> <ul style="list-style-type: none"> <li>- объектно-реляционное отображение и назначение ORM;</li> <li>- JPA: сущности, идентификаторы, наследование и коллекции;</li> <li>- отношения между сущностями и ассоциации;</li> <li>- преимущества и ограничения ORM-подхода.</li> </ul>
17	<p>Тема 17. Структура JPA-приложения и управление данными.</p> <p>Рассматриваемые вопросы:</p> <ul style="list-style-type: none"> <li>- Persistence Unit и конфигурация JPA-приложения;</li> <li>- EntityManager API и жизненный цикл сущностей;</li> <li>- транзакции в JPA;</li> <li>- обработка параллельного чтения и обновления данных.</li> </ul>
18	<p>Тема 18. Запросы, оптимизация Hibernate и интеграция со Spring.</p> <p>Рассматриваемые вопросы:</p> <ul style="list-style-type: none"> <li>- HQL, JPQL, Query API и Criteria API;</li> <li>- аспекты производительности Hibernate и оптимизация выборки;</li> <li>- Hibernate Reactive и особенности неблокирующего доступа к данным;</li> <li>- интеграция JPA/Hibernate со Spring, DAO и управление транзакциями.</li> </ul>
19	<p>Тема 19. Наблюдаемость backend-приложений: логирование, метрики, трассировка.</p> <p>Рассматриваемые вопросы:</p> <ul style="list-style-type: none"> <li>- Observability и Monitoring, основные сигналы: logs, metrics, traces;</li> <li>- уровни логирования и правила безопасного логирования;</li> <li>- метрики приложения и технические метрики инфраструктуры;</li> <li>- OpenTelemetry, Prometheus и Grafana для наблюдаемости backend-систем.</li> </ul>
20	<p>Тема 20. gRPC, GraphQL и WebSocket как альтернативы REST API.</p> <p>Рассматриваемые вопросы:</p> <ul style="list-style-type: none"> <li>- ограничения REST API и выбор способа коммуникации;</li> <li>- gRPC и Protocol Buffers;</li> <li>- GraphQL: схема, запросы и преимущества для клиентских приложений;</li> <li>- WebSocket и realtime-коммуникация.</li> </ul>
21	<p>Тема 21. Асинхронная коммуникация: Kafka, RabbitMQ, события и очереди сообщений.</p> <p>Рассматриваемые вопросы:</p> <ul style="list-style-type: none"> <li>- синхронное и асинхронное взаимодействие приложений;</li> <li>- message broker, очередь сообщений, события и команды;</li> <li>- основы Kafka и RabbitMQ;</li> <li>- event-driven architecture и обработка фоновых задач.</li> </ul>
22	<p>Тема 22. Сборка и управление зависимостями Java-проектов.</p> <p>Рассматриваемые вопросы:</p> <ul style="list-style-type: none"> <li>- назначение build tool и задачи сборки Java-проекта;</li> </ul>

№ п/п	Тематика лекционных занятий / краткое содержание
	- 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 имутабельных 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: - потоковая обработка JSON без полной загрузки в память;

№ п/п	Тематика практических занятий/краткое содержание
	<ul style="list-style-type: none"> <li>- загрузка данных через JDBC и реактивный драйвер;</li> <li>- загрузка JSON в PostgreSQL и парсинг данных на стороне БД;</li> <li>- измерение производительности, хранение статуса набора данных и формирование отчёта.</li> </ul>
5	<p>Тема 5. Аннотированная плагиновая система анализаторов.</p> <p>Рассматриваемые вопросы:</p> <ul style="list-style-type: none"> <li>- создание интерфейса плагина анализатора и собственной аннотации;</li> <li>- использование annotation processor для поиска и регистрации анализаторов;</li> <li>- автоматическое подключение расчётов без ручного изменения основного кода;</li> <li>- расширение отчёта результатами найденных плагинов.</li> </ul>
6	<p>Тема 6. Unit-тестирование аннотированной плагиновой системы.</p> <p>Рассматриваемые вопросы:</p> <ul style="list-style-type: none"> <li>- написание unit-тестов для parser, analyzer, plugin registry и report builder;</li> <li>- использование JUnit и Mockito для проверки бизнес-логики;</li> <li>- проверка работы аннотаций и подключаемых анализаторов;</li> <li>- контроль корректной обработки ошибок и некорректных строк.</li> </ul>
7	<p>Тема 7. Базовое Spring Boot-приложение и тестирование.</p> <p>Рассматриваемые вопросы:</p> <ul style="list-style-type: none"> <li>- перенос консольного анализатора в Spring Boot-приложение;</li> <li>- создание controller, service и configuration-слоёв;</li> <li>- приём данных через HTTP и возврат результата анализа;</li> <li>- покрытие web- и service-слоёв тестами.</li> </ul>
8	<p>Тема 8. REST API, DTO, валидация и обработка ошибок.</p> <p>Рассматриваемые вопросы:</p> <ul style="list-style-type: none"> <li>- разделение внутренних моделей и request/response DTO;</li> <li>- валидация входных данных и формирование понятных сообщений об ошибках;</li> <li>- единая обработка исключений через RestControllerAdvice;</li> <li>- тестирование controller- и service-слоёв.</li> </ul>
9	<p>Тема 9. Взаимосвязи сущностей в JPA.</p> <p>Рассматриваемые вопросы:</p> <ul style="list-style-type: none"> <li>- определение сущности и идентификатора;</li> <li>- определение отношений между сущностями (@OneToMany, @ManyToOne, @ManyToMany);</li> <li>- каскадные операции и lazy/eager загрузка.</li> </ul>
10	<p>Тема 10. Иерархия классов сущностей.</p> <p>Рассматриваемые вопросы:</p> <ul style="list-style-type: none"> <li>- стратегии наследования: SINGLE_TABLE, TABLE_PER_CLASS, JOINED;</li> <li>- отображение абстрактных классов и полиморфные запросы;</li> <li>- выбор подходящей стратегии для предметной области.</li> </ul>
11	<p>Тема 11. Работа с EntityManager.</p> <p>Рассматриваемые вопросы:</p> <ul style="list-style-type: none"> <li>- реализация методов getCompany(int), saveCompany(Company);</li> <li>- реализация методов init(), merge, remove;</li> <li>- жизненный цикл сущностей (new, managed, detached, removed).</li> </ul>
12	<p>Тема 12. Работа с JPQL и Criteria API.</p> <p>Рассматриваемые вопросы:</p> <ul style="list-style-type: none"> <li>- возможности JPA для поиска сущностей: JPQL и Criteria API;</li> <li>- параметризованные запросы, агрегатные функции, Join;</li> <li>- интеграция с фреймворком Spring (@PersistenceContext).</li> </ul>
13	<p>Тема 13. Разработка приложения для блогов.</p> <p>Рассматриваемые вопросы:</p> <ul style="list-style-type: none"> <li>- описание сущностей: DomainObject, BlogPost, Comment;</li> </ul>

№ п/п	Тематика практических занятий/краткое содержание
	<ul style="list-style-type: none"> <li>- реализация сервисных объектов: BlogServiceImpl, BlogPostServiceImpl;</li> <li>- конфигурация контейнера Spring и Persistence Unit.</li> </ul>
14	<p>Тема 14. PostgreSQL и repository-слой.</p> <p>Рассматриваемые вопросы:</p> <ul style="list-style-type: none"> <li>- подключение Spring Boot-приложения к PostgreSQL;</li> <li>- создание таблиц и сохранение запусков анализа, результатов и проблемных строк;</li> <li>- реализация repository-слоя и методов получения отчётов по id и списком;</li> <li>- проверка repository/service-слоёв тестами.</li> </ul>
15	<p>Тема 15. Асинхронные задачи анализа.</p> <p>Рассматриваемые вопросы:</p> <ul style="list-style-type: none"> <li>- перевод запуска анализа в асинхронный режим;</li> <li>- создание задачи анализа с jobId и статусами NEW, PROCESSING, COMPLETED, FAILED;</li> <li>- получение статуса задачи и результата анализа по API;</li> <li>- обработка ошибок и корректное обновление статусов.</li> </ul>
16	<p>Тема 16. Docker Compose для приложения и базы данных.</p> <p>Рассматриваемые вопросы:</p> <ul style="list-style-type: none"> <li>- создание Dockerfile для Spring Boot-приложения;</li> <li>- описание compose.yml для приложения и PostgreSQL;</li> <li>- настройка подключения через environment variables и Docker-профиль;</li> <li>- проверка работы API при запуске через Docker Compose.</li> </ul>
17	<p>Тема 17. Кэширование и оптимизация чтения отчётов.</p> <p>Рассматриваемые вопросы:</p> <ul style="list-style-type: none"> <li>- кэширование повторных запросов отчёта по id;</li> <li>- использование локального кэша или Redis/Spring Cache;</li> <li>- инвалидация или обновление кэша при создании новых отчётов;</li> <li>- сравнение времени ответа до и после кэширования.</li> </ul>
18	<p>Тема 18. Телеметрия, сбор логов и наблюдаемость приложения.</p> <p>Рассматриваемые вопросы:</p> <ul style="list-style-type: none"> <li>- структурированное логирование и correlationId/requestId;</li> <li>- логирование старта, завершения и ошибок анализа;</li> <li>- измерение времени выполнения и сбор базовых метрик;</li> <li>- health check и endpoint для просмотра простой статистики приложения.</li> </ul>
19	<p>Тема 19. Интеграционное тестирование с Testcontainers.</p> <p>Рассматриваемые вопросы:</p> <ul style="list-style-type: none"> <li>- запуск PostgreSQL в контейнере для тестов;</li> <li>- тестирование repository и service слоёв с реальной БД;</li> <li>- проверка миграций и корректности работы с транзакциями.</li> </ul>
20	<p>Тема 20. Пагинация и сортировка в REST API.</p> <p>Рассматриваемые вопросы:</p> <ul style="list-style-type: none"> <li>- использование Pageable и Page из Spring Data;</li> <li>- кастомные запросы с сортировкой;</li> <li>- добавление параметров page, size, sort в контроллеры.</li> </ul>
21	<p>Тема 21. Обработка больших файлов с использованием Stream API и NIO.2.</p> <p>Рассматриваемые вопросы:</p> <ul style="list-style-type: none"> <li>- чтение многогигабайтных файлов;</li> <li>- параллельная обработка строк;</li> <li>- использование Files.lines() и параллельных стримов.</li> </ul>
22	<p>Тема 22. Реализация паттерна Observer / Event Listener в Spring.</p> <p>Рассматриваемые вопросы:</p> <ul style="list-style-type: none"> <li>- создание событий и слушателей;</li> </ul>

№ п/п	Тематика практических занятий/краткое содержание
	<ul style="list-style-type: none"> <li>- асинхронная обработка событий (@Async);</li> <li>- слабое связывание компонентов.</li> </ul>
23	<p>Тема 23. Создание собственного Spring Boot стартера.</p> <p>Рассматриваемые вопросы:</p> <ul style="list-style-type: none"> <li>- автоконфигурация и conditional-аннотации;</li> <li>- создание мета-информации;</li> <li>- тестирование стартера.</li> </ul>
24	<p>Тема 24. Реактивные запросы с WebClient.</p> <p>Рассматриваемые вопросы:</p> <ul style="list-style-type: none"> <li>- неблокирующий вызов внешних API;</li> <li>- обработка ошибок, retry, timeout;</li> <li>- объединение нескольких реактивных вызовов.</li> </ul>
25	<p>Тема 25. Миграции схемы базы данных с Flyway / Liquibase.</p> <p>Рассматриваемые вопросы:</p> <ul style="list-style-type: none"> <li>- создание миграций;</li> <li>- откат (undo) и проверка состояния;</li> <li>- интеграция с Spring Boot.</li> </ul>
26	<p>Тема 26. Мониторинг с Micrometer, Prometheus и Grafana.</p> <p>Рассматриваемые вопросы:</p> <ul style="list-style-type: none"> <li>- экспорт метрик приложения;</li> <li>- создание кастомных счетчиков и гистограмм;</li> <li>- визуализация в Grafana.</li> </ul>
27	<p>Тема 27. Создание REST-клиента на Feign / RestClient.</p> <p>Рассматриваемые вопросы:</p> <ul style="list-style-type: none"> <li>- декларативный вызов внешних сервисов;</li> <li>- настройка логов, ретраев, error decoder;</li> <li>- интеграция с Spring Cloud.</li> </ul>
28	<p>Тема 28. Защита REST API с помощью Spring Security и JWT.</p> <p>Рассматриваемые вопросы:</p> <ul style="list-style-type: none"> <li>- аутентификация и авторизация;</li> <li>- выдача и проверка JWT-токенов;</li> <li>- защита эндпоинтов.</li> </ul>
29	<p>Тема 29. Graceful shutdown и health indicators.</p> <p>Рассматриваемые вопросы:</p> <ul style="list-style-type: none"> <li>- настройка корректного завершения приложения;</li> <li>- кастомные индикаторы здоровья;</li> <li>- подготовка к оркестрации.</li> </ul>
30	<p>Тема 30. Профилирование приложения.</p> <p>Рассматриваемые вопросы:</p> <ul style="list-style-type: none"> <li>- использование JProfiler / Async Profiler;</li> <li>- поиск узких мест по памяти и CPU;</li> <li>- оптимизация на основе профилирования.</li> </ul>
31	<p>Тема 31. Отправка email-уведомлений.</p> <p>Рассматриваемые вопросы:</p> <ul style="list-style-type: none"> <li>- настройка JavaMailSender;</li> <li>- создание шаблонов писем;</li> <li>- асинхронная отправка.</li> </ul>
32	<p>Тема 32. Итоговый проект: многомодульный Spring Boot приложение.</p> <p>Рассматриваемые вопросы:</p> <ul style="list-style-type: none"> <li>- разделение на модули (api, core, infrastructure);</li> </ul>

№ п/п	Тематика практических занятий/краткое содержание
	- 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	Выполнение курсовой работы.
8	Подготовка к промежуточной аттестации.
9	Подготовка к текущему контролю.

#### 4.4. Примерный перечень тем курсовых работ

1. Разработка Spring Boot сервиса для анализа логов веб-сервера
2. Разработка Spring Boot приложения для асинхронного анализа больших JSON-файлов
3. Разработка Spring Boot сервиса хранения и просмотра отчётов анализа данных
4. Разработка защищённого Spring Boot API для анализа пользовательских данных
5. Разработка Spring Boot приложения с plugin-системой анализаторов данных
6. Разработка Spring Boot сервиса телеметрии и мониторинга анализа данных
7. Разработка Spring Boot приложения для кэширования и оптимизации доступа к отчётам
8. Система мониторинга состояния подвижного состава с использованием Java и MQTT
9. Разработка Android-приложения для отслеживания грузов

## 10. Анализ и обработка больших данных в транспортной логистике

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

№ п/п	Библиографическое описание	Место доступа
1	Программирование на java : учебно-методическое пособие / составитель Ю. А. Крыжановская. — Воронеж : ВГУ, 2012. — 52 с.	<a href="https://e.lanbook.com/book/357971">https://e.lanbook.com/book/357971</a>
2	Пономарчук, Ю. В. Программирование на языке Java : учебное пособие / Ю. В. Пономарчук, И. В. Кузнецов. — Хабаровск : ДВГУПС, 2021. — 103 с.	<a href="https://e.lanbook.com/book/259451">https://e.lanbook.com/book/259451</a>

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 семестре.

Курсовая работа в 5 семестре.

Экзамен в 6 семестре.

#### 10. Оценочные материалы.

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

Авторы:

заместитель директора

Б.В. Игольников

Согласовано:

Руководитель образовательной  
программы

О.Б. Проневич

Председатель учебно-методической  
комиссии

Д.В. Паринов