

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 РУТ (МИИТ)
Тимониным В.С.

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: 15.08.2025

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 к организации процессов разработки программного обеспечения .

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 4 з.е. (144 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 №6
Контактная работа при проведении учебных занятий (всего):	64	64
В том числе:		
Занятия лекционного типа	32	32
Занятия семинарского типа	32	32

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 80 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	<p>Тема 1. Введение в Java и инструменты разработки.</p> <p>Рассматриваемые вопросы:</p> <ul style="list-style-type: none"> – 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	<p>Тема 2. Работа памяти в Java.</p> <p>Рассматриваемые вопросы:</p> <ul style="list-style-type: none"> – 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	<p>Тема 3. Объектно-ориентированное программирование, классы и коллекции.</p> <p>Рассматриваемые вопросы:</p> <ul style="list-style-type: none"> – классы, объекты, поля, методы, конструкторы и модификаторы доступа; – инкапсуляция, наследование, полиморфизм и абстракция; – строки и имутабельность: String, StringBuilder, StringBuffer; – коллекции List, Set, Map, контракт equals/hashCode, Comparable и Comparator.
4	<p>Тема 4. Паттерны проектирования на Java и применение принципов ООП.</p> <p>Рассматриваемые вопросы:</p> <ul style="list-style-type: none"> – 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	<p>Тема 5. Работа с файлами, потоками, Stream API и исключениями.</p> <p>Рассматриваемые вопросы:</p> <ul style="list-style-type: none"> – 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	<p>Тема 6. Сериализация объектов и форматы обмена данными: XML, JSON, Protobuf.</p> <p>Рассматриваемые вопросы:</p> <ul style="list-style-type: none"> – 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"> – 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	<p>Тема 8. Рефлексия и работа с базами данных: SQL, JDBC, R2DBC.</p> <p>Рассматриваемые вопросы:</p> <ul style="list-style-type: none"> – 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	<p>Тема 9. Процессоры аннотаций, Lombok и генерация кода.</p> <p>Рассматриваемые вопросы:</p> <ul style="list-style-type: none"> – 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	<p>Тема 10. Spring Framework, Spring Boot и Dependency Injection.</p> <p>Рассматриваемые вопросы:</p> <ul style="list-style-type: none"> – 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	<p>Тема 11. Контейнеризация Spring Boot-приложений, Docker, Docker Compose, GraalVM.</p> <p>Рассматриваемые вопросы:</p> <ul style="list-style-type: none"> – 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	<p>Тема 12. Тестирование Java-приложений: JUnit, Mockito, Spring Boot Test, Testcontainers.</p> <p>Рассматриваемые вопросы:</p> <ul style="list-style-type: none"> – 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.
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"> – клиент-серверное взаимодействие, HTTP и принципы REST API; – HTTP-методы, endpoint, request/response, status codes; – контроллеры, сервисы, DTO и валидация входных данных; – базовые подходы к защите backend-приложений.
14	<p>Тема 14. Архитектурные паттерны backend-приложений: Hexagonal Architecture, Clean Architecture, DDD.</p> <p>Рассматриваемые вопросы:</p> <ul style="list-style-type: none"> – разделение бизнес-логики и инфраструктуры;

№ п/п	Тематика лекционных занятий / краткое содержание
	<ul style="list-style-type: none"> – Domain Layer, Application Layer, Infrastructure Layer; – гексагональная архитектура, порты и адаптеры; – принципы чистой архитектуры и предметно-ориентированного проектирования.
15	<p>Тема 15. Микросервисы и Spring Cloud: Gateway, Config, Discovery и отказоустойчивость.</p> <p>Рассматриваемые вопросы:</p> <ul style="list-style-type: none"> – 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.
16	<p>Тема 16. Основы ORM, JPA и сущностей.</p> <p>Рассматриваемые вопросы:</p> <ul style="list-style-type: none"> – объектно-реляционное отображение и назначение ORM; – JPA: сущности, идентификаторы, наследование и коллекции; – отношения между сущностями и ассоциации; – преимущества и ограничения ORM-подхода.

4.2. Занятия семинарского типа.

Практические занятия

№ п/п	Тематика практических занятий/краткое содержание
1	<p>Тема 1. Анализ входных данных по варианту.</p> <p>Рассматриваемые вопросы:</p> <ul style="list-style-type: none"> – 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	<p>Тема 2. Рефакторинг анализатора данных через ООП и паттерны.</p> <p>Рассматриваемые вопросы:</p> <ul style="list-style-type: none"> – 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	<p>Тема 3. Сохранение результата анализа в JSON и Java Serialization.</p> <p>Рассматриваемые вопросы:</p> <ul style="list-style-type: none"> – 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	<p>Тема 4. Загрузка большого JSON в PostgreSQL через JDBC, реактивный драйвер и парсинг JSON в БД.</p> <p>Рассматриваемые вопросы:</p> <ul style="list-style-type: none"> – 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	<p>Тема 5. Аннотированная плагиновая система анализаторов.</p> <p>Рассматриваемые вопросы:</p>

№ п/п	Тематика практических занятий/краткое содержание
	<ul style="list-style-type: none"> – 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	<p>Тема 6. Unit-тестирование аннотированной плагинной системы.</p> <p>Рассматриваемые вопросы:</p> <ul style="list-style-type: none"> – 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	<p>Тема 7. Базовое Spring Boot-приложение и тестирование.</p> <p>Рассматриваемые вопросы:</p> <ul style="list-style-type: none"> – 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	<p>Тема 8. REST API, DTO, валидация и обработка ошибок.</p> <p>Рассматриваемые вопросы:</p> <ul style="list-style-type: none"> – 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	<p>Тема 9. Взаимосвязи сущностей</p> <p>Рассматриваемые вопросы:</p> <ul style="list-style-type: none"> – определение сущности – идентификатор определения сущности – определение отношений между сущностями
10	<p>Тема 10. Иерархия классов сущностей</p> <p>Рассматриваемые вопросы:</p> <ul style="list-style-type: none"> – тип наследования SINGLE_TABLE – тип наследования TABLE_PER_CLASS – тип наследования JOINED
11	<p>Тема 11. Работа с EntityManager</p> <p>Рассматриваемые вопросы:</p> <ul style="list-style-type: none"> – реализация методов getCompany(int) – реализация методов saveCompany(Company) – реализация методов init()
12	<p>Тема 12. Работа с JPQL и Criteria API</p> <p>Рассматриваемые вопросы:</p> <ul style="list-style-type: none"> – возможности JPA для поиска сущностей: JPQL и Criteria API. – использование JPQL и Criteria API – интеграция с фреймворком Spring
13	<p>Тема 13. Разработка приложения для блогов</p> <p>Рассматриваемые вопросы:</p> <ul style="list-style-type: none"> – описание сущностей: класс DomainObject, класс BlogPost – реализация сервисных объектов: класс BlogServiceImpl, класс BlogPostServiceImpl – конфигурации контейнера Spring и Persistence Unit

4.3. Самостоятельная работа обучающихся.

№ п/п	Вид самостоятельной работы
1	Работа с учебной литературой
2	Участие в онлайн-конференциях и мастер-классах
3	Поиск алгоритмов обработки данных в открытых источниках
4	Подготовка к промежуточной аттестации.
5	Подготовка к текущему контролю.

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

№ п/п	Библиографическое описание	Место доступа
1	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
2	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
3	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

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:

Экзамен в 6 семестре.

10. Оценочные материалы.

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

Авторы:

доцент, к.н. Академии "Высшая
инженерная школа"

Б.В. Игольников

Согласовано:

Директор

Д.В. Паринов

Руководитель образовательной
программы

О.Б. Проневич

Председатель учебно-методической
комиссии

Д.В. Паринов