

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 - программы 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): 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 РУТ (МИИТ)
ID подписи: 937226
Подписал: руководитель образовательной программы
Проневич Ольга Борисовна
Дата: 04.05.2026

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 семействе Java-технологий;
- изучение языка программирования Java в составе технологии JavaSE;
- изучение основ использования JDK SE при проектировании Java приложений;
- использование интегрированной среды разработки программных проектов (IDE) для проектирования и отладки различных видов Java приложений;
- обеспечение качества в проектах в области ИТ в соответствии с установленными регламентами;
- распространение информации в проектах в области ИТ 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:

- Представление о функциональных возможностях языка.
- основные функции системы автоматической сборки и управления зависимостями Maven;
- основные принципы и шаблоны GRASP (Информационный эксперт, Создатель, Контроллер, Слабое зацепление, Высокая сплоченность), базовые шаблоны GOF (Прототип, фабрика, строитель, одиночка, наблюдатель,

приспособленец, стратегия, команда и др.), знать шаблон Модель-Представление-Контроллер (MVC);

- нотацию языка моделирования UML в части диаграмм классов, диаграмм последовательности

- основные средства и принципы организации стандартной библиотеки Java

Уметь:

- Эффективно использовать инструментарий высокоуровневых языков программирования для анализа больших данных;

- разрабатывать и реализовывать автоматизированные тесты с целью верификации корректности реализованных программ с использованием библиотеки junit;

- разрабатывать и визуализировать модель классов системы на языке UML;

- осуществлять объектно-ориентированную декомпозицию программ на языке Java с разделением на модули (пакеты) с последующей реализацией;

- проводить объектную декомпозицию информационной системы, выработать и обосновать архитектурное решение.

Владеть:

- Основными средства поддержки процедурного стиля программирования (типы данных, переменные, структура программы, функции, структурные типы) языка программирования Java;

- средствами языка Java поддерживающими основные принципы объектно-ориентированного подхода (поддержка абстракции, инкапсуляции, иерархии, модульности, типизации, параллелизма и сохраняемости в Java);

- основными средствами обобщенного программирования на Java (родовые компоненты);

- знаниями об основных современных средах разработки для Java;

- навыками разработки программ на языках высокого уровня.

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 №5
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	Тема 1. Введение в Java и средства разработки Рассматриваемые вопросы: - место Java в современных Java-технологиях; обзор областей применения языка - JVM, JDK, JRE; установка JDK и настройка среды разработки - структура Java-программы, класс, метод main, компиляция и запуск - знакомство с IDE, созданием проекта и базовой отладкой
2	Тема 2. Основы синтаксиса Java Рассматриваемые вопросы: - переменные, константы, примитивные типы данных и ссылочные типы - операторы, выражения, преобразование типов - класс на базовом уровне - методы, параметры, возвращаемые значения, область видимости
3	Тема 3. Управляющие конструкции и массивы Рассматриваемые вопросы: - условные операторы, циклы, массивы - простые алгоритмы обработки данных: поиск, подсчет, сортировка

№ п/п	Тематика лекционных занятий / краткое содержание
4	<p>Тема 4. Классы и объекты. Инкапсуляция</p> <p>Рассматриваемые вопросы:</p> <ul style="list-style-type: none"> - объектно-ориентированный подход: класс, объект, состояние и поведение - поля, методы, конструкторы, this, static - модификаторы доступа, инкапсуляция, геттеры и сеттеры - пакеты, import, базовая организация кода проекта
5	<p>Тема 5. Наследование и полиморфизм</p> <p>Рассматриваемые вопросы:</p> <ul style="list-style-type: none"> - наследование классов: extends, super, вызов конструкторов родителя - переопределение методов, аннотация @Override - полиморфизм, приведение типов, использование базового типа - класс Object: toString(), equals(), hashCode()
6	<p>Тема 6. Абстракция и интерфейсы</p> <p>Рассматриваемые вопросы:</p> <ul style="list-style-type: none"> - абстрактные классы и абстрактные методы - интерфейсы, implements, default-методы - композиция как альтернатива наследованию - enum, внутренние и вложенные классы на обзорном уровне
7	<p>Тема 7. Строки, исключения и основы тестирования</p> <p>Рассматриваемые вопросы:</p> <ul style="list-style-type: none"> - обработка строк: String, StringBuilder, основные методы строк - иерархия исключений, checked и unchecked exceptions - try/catch/finally, throw, throws, собственные исключения - принципы отладки и базовые автоматизированные тесты на JUnit
8	<p>Тема 8. Коллекции и обобщённое программирование</p> <p>Рассматриваемые вопросы:</p> <ul style="list-style-type: none"> - обобщения: generic-классы и generic-методы, параметр типа T - основные коллекции: List, Set, Map; классы ArrayList, HashSet, HashMap - итерация по коллекциям, класс Collections - сравнение объектов: Comparable и Comparator
9	<p>Тема 9. Файлы, сборка проекта и доступ к данным</p> <p>Рассматриваемые вопросы:</p> <ul style="list-style-type: none"> - потоки ввода-вывода, работа с файлами, Path и Files (NIO.2) на базовом уровне - Maven/Gradle: структура проекта, зависимости, жизненный цикл сборки - синхронное подключение к базе данных через JDBC: Connection, PreparedStatement, ResultSet - понятие транзакции и подготовка к переходу к Spring-приложениям
10	<p>Тема 10. Spring Core: IoC и Dependency Injection</p> <p>Рассматриваемые вопросы:</p> <ul style="list-style-type: none"> - назначение Spring Framework и роль IoC-контейнера - понятие Bean, жизненный цикл и область видимости бинов - конфигурация через аннотации: @Component, @Service, @Repository, @Bean - внедрение зависимостей через конструктор; @Autowired и принципы слабой связанности
11	<p>Тема 11. Spring Boot и разработка REST API</p> <p>Рассматриваемые вопросы:</p> <ul style="list-style-type: none"> - создание проекта Spring Boot, автоконфигурация, @SpringBootApplication - структура приложения: controller, service, repository, model/dto - REST-контроллеры: @RestController, @RequestMapping, @GetMapping, @PostMapping - обработка HTTP-запросов и ответов, DTO, базовая валидация входных данных
12	<p>Тема 12. Реактивное программирование и Spring WebFlux</p> <p>Рассматриваемые вопросы:</p>

№ п/п	Тематика лекционных занятий / краткое содержание
	<ul style="list-style-type: none"> - идея неблокирующей обработки запросов и отличие от классического синхронного подхода - основы Project Reactor: Mono, Flux, map, flatMap, filter, обработка ошибок - реактивные REST-контроллеры в Spring WebFlux - практические ограничения реактивного подхода и случаи, когда он оправдан
13	Тема 13. JDBC и реактивная работа с базой данных: R2DBC Рассматриваемые вопросы: <ul style="list-style-type: none"> - JDBC драйвер - назначение R2DBC как реактивного драйвера доступа к реляционным базам данных - подключение PostgreSQL/H2 через ConnectionFactory и настройки Spring Boot - DatabaseClient, R2dbcEntityTemplate и реактивные репозитории - CRUD-операции, реактивные транзакции и типичные ошибки при работе с Flux/Mono
14	Тема 14. Итоговая архитектура Java/Spring-приложения Рассматриваемые вопросы: <ul style="list-style-type: none"> - сборка итогового учебного приложения: модели, сервисы, REST/WebFlux API, база данных - архитектурная декомпозиция, принципы MVC и слоистой архитектуры - код-ревью, оформление README, требования к качеству и стилю кода

4.2. Занятия семинарского типа.

Практические занятия

№ п/п	Тематика практических занятий/краткое содержание
1	Алгоритмический практикум: разбор задач и оформление решений Рассматриваемые вопросы: <ul style="list-style-type: none"> - разбор условия задачи, выделение входных и выходных данных; - решение задач на строки, массивы, циклы и условия; - линейный поиск, поиск максимума/минимума, частотный анализ; - простые сортировки и сравнение подходов к решению; - оформление решения в виде методов и проверка на граничных случаях.
2	Объектная модель без привязки к фреймворкам Рассматриваемые вопросы: <ul style="list-style-type: none"> - выделение сущностей, связей и ответственности классов; - построение простой UML-диаграммы классов; - инкапсуляция, валидация состояния объектов, работа с конструкторами; - использование интерфейсов для отделения контракта от реализации; - проверка модели через небольшой консольный сценарий.
3	Использование фреймворков в разработке, базовая архитектура приложения Рассматриваемые вопросы: <ul style="list-style-type: none"> - создание CRUD-логики без базы данных; - использование коллекций для хранения заявок и пользователей; - поиск, фильтрация и сортировка элементов; - обработка некорректного ввода и пользовательских ошибок; - разделение кода на пакеты: model, service, repository, ui.
4	Импорт, очистка и сохранение данных из файлов Рассматриваемые вопросы: <ul style="list-style-type: none"> - чтение данных из CSV/текстового файла; - очистка строк, проверка формата и обработка пропусков; - преобразование строк в объекты предметной области;

№ п/п	Тематика практических занятий/краткое содержание
	<ul style="list-style-type: none"> – сохранение результата в JSON или текстовый отчет; – использование try-with-resources и корректная обработка I/O-ошибок.
5	Параллельная обработка набора данных Рассматриваемые вопросы: <ul style="list-style-type: none"> – разбиение задачи на независимые части; – использование ExecutorService или CompletableFuture; – сбор результатов из нескольких задач; – поиск и исправление типовых ошибок синхронизации; – сравнение последовательного и параллельного варианта на учебном наборе данных.
6	SQL-слой без ORM: реализация доступа к данным через JDBC Рассматриваемые вопросы: <ul style="list-style-type: none"> – подключение к учебной базе данных через JDBC-драйвер; – создание таблиц и выполнение параметризованных SQL-запросов; – реализация DAO-слоя без JPA и Hibernate; – операции create, read, update, delete; – управление транзакциями и обработка SQLException.
7	Неблокирующий доступ к данным: R2DBC в сервисном слое Рассматриваемые вопросы: <ul style="list-style-type: none"> – подключение реактивного драйвера R2DBC; – создание неблокирующих операций чтения и записи; – использование Mono и Flux в сервисном слое; – обработка ошибок в реактивной цепочке; – ограничения реактивного подхода при работе с транзакциями и внешними ресурсами.
8	Сравнительный эксперимент: JDBC против реактивного стека Рассматриваемые вопросы: <ul style="list-style-type: none"> – реализация одной и той же задачи двумя способами: JDBC и R2DBC/WebFlux; – сравнение блокирующего и неблокирующего выполнения; – анализ структуры кода, сложности сопровождения и читаемости; – проверка поведения при нескольких одновременных запросах; – вывод: когда уместен JDBC, а когда оправдан реактивный стек.
9	Учебный REST-сервис без JPA: API поверх собственного слоя данных Рассматриваемые вопросы: <ul style="list-style-type: none"> – проектирование DTO и сервисного слоя; – создание REST-эндпоинтов для учебной предметной области; – подключение JDBC или R2DBC на выбор; – валидация входных данных и единый формат ответа об ошибке; – проверка API через Postman, curl или аналогичный инструмент.
10	Итоговый практический мини-проект и защита технического решения Рассматриваемые вопросы: <ul style="list-style-type: none"> – выбор предметной области мини-проекта; – описание архитектуры и структуры пакетов; – реализация минимального рабочего приложения; – подготовка README с инструкцией запуска; – краткая защита выбора: JDBC или реактивный стек, сильные и слабые стороны решения.

4.3. Самостоятельная работа обучающихся.

№ п/п	Вид самостоятельной работы
1	Работа с учебной литературой

№ п/п	Вид самостоятельной работы
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	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:

Экзамен в 5 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