

**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: 10.04.2026

## 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, вырабатывать и обосновывать 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 10 з.е. (360 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 216 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:

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

№ п/п	Тематика лекционных занятий / краткое содержание
12	<p>Тема 12. Реактивное программирование и Spring WebFlux</p> <p>Рассматриваемые вопросы:</p> <ul style="list-style-type: none"> <li>- идея неблокирующей обработки запросов и отличие от классического синхронного подхода</li> <li>- основы Project Reactor: Mono, Flux, map, flatMap, filter, обработка ошибок</li> <li>- реактивные REST-контроллеры в Spring WebFlux</li> <li>- практические ограничения реактивного подхода и случаи, когда он оправдан</li> </ul>
13	<p>Тема 13. JDBC и реактивная работа с базой данных: R2DBC</p> <p>Рассматриваемые вопросы:</p> <ul style="list-style-type: none"> <li>- JDBC драйвер</li> <li>- назначение R2DBC как реактивного драйвера доступа к реляционным базам данных</li> <li>- подключение PostgreSQL/H2 через ConnectionFactory и настройки Spring Boot</li> <li>- DatabaseClient, R2dbcEntityTemplate и реактивные репозитории</li> <li>- CRUD-операции, реактивные транзакции и типичные ошибки при работе с Flux/Mono</li> </ul>
14	<p>Тема 14. Итоговая архитектура Java/Spring-приложения</p> <p>Рассматриваемые вопросы:</p> <ul style="list-style-type: none"> <li>- сборка итогового учебного приложения: модели, сервисы, REST/WebFlux API, база данных</li> <li>- архитектурная декомпозиция, принципы MVC и слоистой архитектуры</li> <li>- код-ревью, оформление README, требования к качеству и стилю кода</li> </ul>

## 4.2. Занятия семинарского типа.

### Практические занятия

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

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

#### 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	Анализ больших наборов данных Л. Юре, Р. Ананд, Д. У. Джеффри	<a href="https://e.lanbook.com/book/93571">https://e.lanbook.com/book/93571</a>
2	Программирование на Java Вязовик Н.А.	<a href="https://e.lanbook.com/book/100405">https://e.lanbook.com/book/100405</a>
3	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	<a href="https://e.lanbook.com/book/64399">https://e.lanbook.com/book/64399</a>
4	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	<a href="https://e.lanbook.com/book/93272">https://e.lanbook.com/book/93272</a>
5	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	<a href="https://e.lanbook.com/book/93571">https://e.lanbook.com/book/93571</a>
1	Программирование на языке Java: многопоточные приложения Кожомбердиева Г.И.	<a href="https://e.lanbook.com/book/64399">https://e.lanbook.com/book/64399</a>
2	Обработка естественного языка на Java Риз Р	<a href="https://e.lanbook.com/book/93272">https://e.lanbook.com/book/93272</a>

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

<http://library.mii.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:

заместитель директора

Б.В. Игольников

Согласовано:

Директор

Д.В. Паринов

Руководитель образовательной  
программы

О.Б. Проневич

Председатель учебно-методической  
комиссии

Д.В. Паринов