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 ФГАОУ ВО РУТ
(МИИТ) Париновым Д.В.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ (МОДУЛЯ)

Программирование

Направление подготовки: 09.03.01 Информатика и вычислительная техника

Направленность (профиль): Технологии разработки программного обеспечения

Форма обучения: Очная

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

Простая электронная подпись, выданная ФГАОУ ВО РУТ (МИИТ)
ID подписи: 937226
Подписал: директор центра Проневич Ольга Борисовна
Дата: 12.11.2026

1. Общие сведения о дисциплине (модуле).

Дисциплина «Программирование» формирует базовый профессиональный контур инженера-программиста, востребованный работодателями транспортной, логистической и финансовой инфраструктуры, где корпоративные системы разрабатываются на языке Java и ощущается дефицит младших разработчиков, способных читать чужой код, формализовать задачу и поставлять проверенные изменения. Содержание соответствует профессиональному стандарту «Программист» и реализуется в отечественном контуре – Axiom JDK, Astra Linux, GitFlic и GitVerse, PostgreSQL, Р7-Офис и МойОфис. Обучение открывается историей языков программирования и обзором того, какие языки для каких задач применяются, и последовательно ведёт от первой программы к объектной модели, коллекциям, файлам, лямбда-выражениям и автотестам, а сопутствующие темы – контроль версий, сборка и базы данных – даются кратко в прикладном объёме. Итогом освоения является репозиторий тестируемого приложения.

Целью освоения дисциплины является формирование базовой способности разрабатывать программное обеспечение на языке Java – формализовать задачу, проектировать алгоритм, писать и оформлять программный код, создавать проверочные данные, выполнять автоматизированную проверку и отладку, управлять версиями и представлять изменение для командного рецензирования.

Для достижения поставленной цели в рамках дисциплины решается комплекс задач, направленных на формирование у обучающихся способности: формализовать ограниченную программную задачу, определить входные данные, алгоритм, граничные условия и ожидаемый результат. Разрабатывать структурированный программный код на языке Java с применением стандартной библиотеки и объектно-ориентированного подхода. Оформлять программный код и сопроводительные материалы в соответствии с установленными правилами. Разрабатывать контрольные примеры и тестовые наборы данных для нормальных, граничных и ошибочных сценариев. Проверять работоспособность программы, интерпретировать диагностические сведения и устранять ошибки на уровне модулей и их взаимодействий. Вести историю изменений, сравнивать и объединять версии программного кода. Оценивать небольшую задачу, участвовать в планировании, подготавливать изменение к рецензированию и обрабатывать замечания.

2. Планируемые результаты обучения по дисциплине (модулю).

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

ОПК-6 - Способен разрабатывать алгоритмы и компьютерные программы, пригодные для практического применения;

ППК-Р1 - Способен разрабатывать и отлаживать программный код;

ППК-Р6 - Способен участвовать в промышленной разработке программного обеспечения.

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

Знать:

- методы создания и документирования контрольных примеров и тестовых наборов данных;
- требования к структуре и форматам хранения тестовых наборов данных;
- правила, алгоритмы и технологии создания тестовых наборов данных;
- методы и средства проверки работоспособности компьютерного программного обеспечения;
- государственные стандарты испытания автоматизированных систем;
- руководящие документы по стандартизации требований к документам автоматизированных систем;
- методы создания и документирования контрольных примеров и тестовых наборов данных;
- требования к структуре и форматам хранения тестовых наборов данных;
- правила, алгоритмы и технологии создания тестовых наборов данных;
- методы и средства проверки работоспособности компьютерного программного обеспечения;
- государственные стандарты испытания автоматизированных систем;
- руководящие документы по стандартизации требований к документам автоматизированных систем;
- алгоритмы решения типичных задач, области и способы их применения;
- нотации и программное обеспечение для графического отображения алгоритмов;
- методы и приемы алгоритмизации поставленных задач;

- синтаксис выбранного языка программирования, особенности программирования на этом языке, стандартные библиотеки языка программирования;

- методологии разработки компьютерного программного обеспечения;

- технологии программирования;

- нормативно-технические документы (стандарты и регламенты), определяющие требования к оформлению программного кода;

- основные стандарты оформления технической документации на компьютерное программное обеспечение;

- возможности используемой системы управления версиями и вспомогательных инструментальных программных средств;

- установленный регламент использования системы управления версиями;

- методы и приемы отладки программного кода;

- типы и форматы сообщений об ошибках, предупреждений;

- способы использования технологических журналов, форматы и типы записей журналов;

- принципы Agile и их применение в промышленных проектах;

- процессы code review, принципы коллективного владения кодом (collective code ownership).

Уметь:

- разрабатывать и оформлять контрольные примеры для проверки работоспособности компьютерного программного обеспечения;

- готовить тестовые наборы данных в соответствии с выбранной методикой тестирования компьютерного программного обеспечения;

- применять методы и средства проверки работоспособности компьютерного программного обеспечения;

- интерпретировать диагностические данные проверки работоспособности компьютерного программного обеспечения;

- анализировать значения полученных характеристик компьютерного программного обеспечения;

- разрабатывать и оформлять контрольные примеры для проверки работоспособности компьютерного программного обеспечения;

- готовить тестовые наборы данных в соответствии с выбранной методикой тестирования компьютерного программного обеспечения;

- применять методы и средства проверки работоспособности компьютерного программного обеспечения;

- интерпретировать диагностические данные проверки работоспособности компьютерного программного обеспечения;

- анализировать значения полученных характеристик компьютерного программного обеспечения;

- использовать методы и приемы формализации и алгоритмизации поставленных задач;

- применять алгоритмы решения типовых задач в соответствующих областях;

- применять выбранные языки программирования для написания программного кода;

- использовать выбранную среду программирования;

- использовать возможности имеющейся технической и/или программной архитектуры для написания программного кода;

- применять заданные стандарты и шаблоны для составления и оформления технической документации;

- применять нормативно-технические документы (стандарты и регламенты), определяющие требования к оформлению программного кода;

- применять инструментарий для создания и актуализации исходных текстов программ;

- регистрировать изменения исходного текста программного кода в системе управления версиями;

- сохранять изменения программного кода в соответствии с регламентом управления версиями;

- выполнять слияние, разделение и сравнение исходных текстов программного кода;

- выявлять ошибки в программном коде;

- отлаживать программный код на уровне программных модулей;

- отлаживать программный код на уровне межмодульных взаимодействий и взаимодействий с окружением;

- оценивать объем задачи и срок ее выполнения, участвовать в планировании спринтов;

- работать в команде с использованием инструментов управления проектами.

Владеть:

- навыками реализации алгоритмов в программном коде на языке Java с использованием среды разработки;

- навыками отладки программного кода инструментальными средствами среды разработки;

- навыками применения системы управления версиями при сопровождении исходного кода;

- навыками автоматизированной проверки работоспособности программного обеспечения средствами библиотек модульного тестирования;

- навыками подготовки изменений программного кода к командному рецензированию.

3. Объем дисциплины (модуля).

3.1. Общая трудоемкость дисциплины (модуля).

Общая трудоемкость дисциплины (модуля) составляет 9 з.е. (324 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	
		№1	№2
Контактная работа при проведении учебных занятий (всего):	176	80	96
В том числе:			
Занятия лекционного типа	64	32	32
Занятия семинарского типа	112	48	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 148 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	Введение в программирование и история языков Рассматриваемые вопросы: - понятие программы и программирования как инженерной деятельности; - эволюция от машинных кодов и ассемблера к языкам высокого уровня; - основные парадигмы программирования.
2	Современные языки и экосистема Java Рассматриваемые вопросы: - какие языки применяются для системных, прикладных, веб- и мобильных задач; - рейтинги популярности языков и их динамика; - платформа Java: JDK, JRE, JVM и отечественные дистрибутивы.
3	Постановка задачи и алгоритмизация Рассматриваемые вопросы: - формализация входных данных, результатов и ограничений; - блок-схемы, псевдокод и таблицы решений; - проверка алгоритма ручной трассировкой.
4	Первая программа на Java Рассматриваемые вопросы: - установка JDK и структура исходного файла; - метод main и вывод в консоль; - компиляция и запуск из командной строки.
5	Переменные и примитивные типы данных Рассматриваемые вопросы: - объявление и инициализация переменных; - целочисленные, вещественные, символьный и логический типы; - преобразование типов и область видимости.
6	Операторы, выражения и ветвления Рассматриваемые вопросы: - арифметические операторы, сравнение и логические операции; - приоритет вычисления выражений; - условные конструкции if, else и switch.
7	Циклы Рассматриваемые вопросы: - циклы while, do-while и for; - счётчики, условия продолжения и выхода; - вложенные циклы и типовые циклические алгоритмы.
8	Массивы Рассматриваемые вопросы: - объявление, заполнение и обход массива; - поиск, суммирование и выборка значений; - типовые ошибки при работе с индексами.
9	Строки и консольный ввод-вывод Рассматриваемые вопросы: - чтение данных с консоли; - методы String и разбор текста; - форматирование вывода результата.
10	Методы и декомпозиция программы Рассматриваемые вопросы: - объявление методов, параметры и возвращаемые значения; - передача данных и область видимости локальных переменных; - разбиение алгоритма на отдельные методы.

№ п/п	Тематика лекционных занятий / краткое содержание
11	Классы, объекты и инкапсуляция Рассматриваемые вопросы: - поля, методы и конструкторы; - состояние и поведение объектов; - инкапсуляция и инварианты состояния.
12	Наследование, полиморфизм и интерфейсы Рассматриваемые вопросы: - отношения между классами; - переопределение методов и полиморфизм; - интерфейсы как контракты поведения.
13	Пакеты и модификаторы доступа Рассматриваемые вопросы: - пакеты и импорт классов; - модификаторы доступа и управление видимостью; - организация исходного кода программы.
14	Исключения и надёжная обработка ошибок Рассматриваемые вопросы: - проверяемые и непроверяемые исключения; - try-catch-finally и try-with-resources; - понятные сообщения об ошибках ввода.
15	Даты и точные вычисления Рассматриваемые вопросы: - классы java.time и арифметика дат и интервалов; - ограничения double в точных расчётах; - BigDecimal и явные режимы округления.
16	Стиль кода, документирование и репозиторий Рассматриваемые вопросы: - соглашения об именовании и форматировании; - комментарии, Javadoc и README; - создание репозитория и коммиты.
17	Коллекции Java: списки, множества и отображения Рассматриваемые вопросы: - интерфейсы List, Set и Map и их реализации; - добавление, поиск и удаление элементов; - выбор коллекции под характер задачи.
18	Сравнение объектов и сортировка Рассматриваемые вопросы: - контракты equals и hashCode; - естественный порядок через Comparable; - настраиваемое сравнение через Comparator.
19	Обобщённые типы Рассматриваемые вопросы: - параметризация классов и методов; - ограничения параметров типов; - типобезопасность и приведение типов.
20	Лямбда-выражения и функциональные интерфейсы Рассматриваемые вопросы: - синтаксис лямбда-выражений; - стандартные функциональные интерфейсы; - ссылки на методы.

№ п/п	Тематика лекционных занятий / краткое содержание
21	Потоки данных Stream API Рассматриваемые вопросы: - создание потоков из коллекций и массивов; - промежуточные операции filter, map и sorted; - терминальные операции и сбор результатов.
22	Перечисления и классы-записи Рассматриваемые вопросы: - перечисления enum и фиксированные наборы значений; - классы-записи record для хранения данных; - выбор между классом, записью и перечислением.
23	Работа с файлами и текстовыми данными Рассматриваемые вопросы: - пути и файловые операции в java.nio.file; - чтение и запись текстовых файлов; - обработка ошибок ввода-вывода.
24	Регулярные выражения и обработка текста Рассматриваемые вопросы: - синтаксис регулярных выражений; - классы Pattern и Matcher; - проверка и разбор текстовых данных.
25	Алгоритмическая сложность и структуры данных Рассматриваемые вопросы: - оценка сложности алгоритмов в нотации Big O; - стек, очередь и дерево как базовые структуры; - эмпирическое сравнение структур данных.
26	Тестирование программ и JUnit 5 Рассматриваемые вопросы: - контрольные примеры и классы эквивалентности; - аннотации, утверждения и параметризованные тесты; - организация автотестов в программе.
27	Отладка и диагностика дефектов Рассматриваемые вопросы: - воспроизведение и локализация дефекта; - точки останова и пошаговое выполнение; - логирование и диагностические сообщения.
28	Сборка приложения и автотесты в Maven Рассматриваемые вопросы: - назначение Maven и стандартная структура каталогов; - файл pom.xml, зависимости и плагины; - сборка приложения и воспроизводимый запуск автотестов.
29	Форматы данных CSV и JSON Рассматриваемые вопросы: - структура и применение формата CSV; - синтаксис JSON: объекты, массивы и значения; - чтение и запись данных в учебных программах.
30	Базы данных и JDBC: краткое введение Рассматриваемые вопросы: - реляционная модель и назначение SQL в общих чертах; - подключение к PostgreSQL или Postgres Pro через JDBC; - выполнение простого параметризованного запроса.

№ п/п	Тематика лекционных занятий / краткое содержание
31	Класс Optional и отсутствующие значения Рассматриваемые вопросы: - проблема нулевых ссылок; - создание и разворачивание Optional; - типовые приёмы и антипаттерны.
32	Обзор стандартной библиотеки Java Рассматриваемые вопросы: - утилитные классы Math, Objects и Random; - форматирование чисел, дат и сообщений; - навигация по документации Java API.

4.2. Занятия семинарского типа.

Лабораторные работы

№ п/п	Наименование лабораторных работ / краткое содержание
1	Установка и проверка среды разработки Студент устанавливает Axiom JDK в среде Astra Linux, ALT Linux или РЕД ОС и проверяет версию Java. Он создаёт каталог приложения и настраивает среду разработки. После проверки команд компиляции и запуска он оформляет протокол версий и команд. Итогом становится готовое рабочее место для разработки.
2	Первая программа и вывод данных Студент создаёт исходный файл с методом main и выводит приветствие и данные учебной задачи. Он компилирует программу и выполняет её из командной строки. После серии запусков он исправляет ошибки синтаксиса. Итогом становится первая работающая программа.
3	Переменные и типы данных Студент объявляет переменные для данных учебной задачи и выбирает подходящие примитивные типы. Он проверяет инициализацию, преобразования типов и область видимости. После запуска он сверяет промежуточные значения с ручным расчётом. Итогом становится проверенный фрагмент вычислений.
4	Операторы и выражения Студент записывает вычисления с арифметическими и логическими операторами. Он проверяет приоритет операций на контрольных выражениях. После сравнения результатов с эталоном он исправляет расхождения. Итогом становится программа с проверенными выражениями.
5	Ветвления по шагам алгоритма Студент реализует условные конструкции для шагов утверждённого алгоритма. Он проверяет каждую ветвь на нормальных и граничных примерах. После обнаружения неверных условий он исправляет их и повторяет запуск. Итогом становится программа, воспроизводящая ветвление алгоритма.
6	Циклы и обработка серий примеров Студент реализует повторяемую обработку набора примеров с помощью циклов. Он накапливает промежуточные результаты и контролирует условие завершения. После запуска на эталонной ведомости он устраняет расхождения. Итогом становится программа циклической обработки.
7	Массивы и сводка результатов Студент хранит серию значений в массиве и выполняет обход, поиск и суммирование. Он формирует сводку результатов по набору примеров. После проверки границ массива он устраняет ошибки индексирования. Итогом становится программа обработки массива данных.

№ п/п	Наименование лабораторных работ / краткое содержание
8	Строки и консольный ввод Студент реализует чтение полей задачи с консоли и разбор строк. Он преобразует введённые значения и выводит понятное сообщение при ошибке. После запуска корректного и ошибочного наборов он приводит поведение к заданному формату. Итогом становится программа с проверенным вводом.
9	Методы и декомпозиция программы Студент выделяет шаги расчёта в отдельные методы с параметрами и возвращаемыми значениями. Он собирает программу из ранее проверенных фрагментов и запускает её на нормальных примерах. Обнаруженные расхождения с эталонными результатами он исправляет и повторяет запуск. Итогом становится программа, разбитая на методы.
10	Классы и инкапсуляция Студент переносит данные учебной модели в отдельные классы. Он закрывает состояние объектов, добавляет конструкторы и проверяет инварианты в методах. После компиляции и запуска он устраняет ошибки переноса без изменения результата. Итогом становится объектная версия программы.
11	Наследование и полиморфизм Студент реализует иерархию классов учебной модели и переопределяет методы. Он обрабатывает объекты разных классов через общий интерфейс. После запуска контрольных примеров он устраняет ошибки полиморфной обработки. Итогом становится иерархия классов с полиморфным поведением.
12	Пакеты и модификаторы доступа Студент распределяет классы по пакетам и настраивает их видимость. Он проверяет доступность полей и методов между пакетами. После компиляции и запуска он устраняет ошибки организации кода. Итогом становится структурированная версия программы.
13	Исключения и сообщения об ошибках Студент добавляет обработку исключительных ситуаций ввода и расчёта. Он формирует проверяемые сообщения об ошибках и освобождает ресурсы корректно. После запуска ошибочных сценариев он добивается понятной реакции программы. Итогом становится версия с надёжной обработкой ошибок.
14	Даты и точные вычисления Студент применяет java.time для работы с датами и интервалами и BigDecimal для точных расчётов. Он задаёт явный режим округления и проверяет границы интервалов. После полного прогона эталонных примеров он устраняет расхождения. Итогом становится технически устойчивая версия программы.
15	Стиль кода и документация Студент приводит код к контрольному перечню стиля и добавляет Javadoc для публичных элементов. Он оформляет README с инструкцией сборки и запуска. После самопроверки он устраняет нарушения оформления. Итогом становится документированная версия программы.
16	Репозиторий и коммиты Студент создаёт репозиторий в GitFlic или GitVerse и загружает код. Он выполняет небольшие коммиты с понятными сообщениями и проверяет историю. После получения копии репозитория он проверяет сборку и запуск по README. Итогом становится опубликованная рабочая версия.
17	Списки, множества и отображения Студент хранит учебный набор записей в списке, множестве и отображении. Он выполняет добавление, поиск и удаление элементов и сравнивает поведение структур. После прогона контрольных примеров он оформляет вывод о выборе коллекции. Итогом становится программа обработки данных в коллекциях.
18	Сравнение и сортировка объектов Студент реализует equals и hashCode для учебных записей. Он задаёт естественный порядок через

№ п/п	Наименование лабораторных работ / краткое содержание
	Comparable и варианты сортировки через Comparator. После прогона примеров он устраняет расхождения порядка. Итогом становится сортировка объектов по нескольким критериям.
19	Обобщённые классы и методы Студент создаёт обобщённый класс и обобщённые методы для учебных данных. Он применяет ограничения параметров типов. После компиляции и запуска он устраняет ошибки приведения типов. Итогом становится типобезопасный обобщённый компонент.
20	Лямбда-выражения и функциональные интерфейсы Студент реализует фильтрацию и преобразование учебных данных лямбда-выражениями. Он применяет стандартные функциональные интерфейсы и ссылки на методы. После прогона примеров он сверяет результаты с ожидаемыми. Итогом становится обработка данных лямбда-выражениями.
21	Конвейерная обработка в Stream API Студент строит конвейер из промежуточных и терминальных операций. Он формирует сводку по учебному набору записей. После сравнения с эталонным результатом он устраняет расхождения. Итогом становится аналитическая сводка средствами Stream API.
22	Перечисления и классы-записи Студент заменяет строковые коды на перечисления и простые классы данных на записи. Он применяет перечисления в выборе ветвей программы. После компиляции и прогона примеров он устраняет ошибки перестройки. Итогом становится уточнённая модель данных.
23	Чтение и запись текстовых файлов Студент реализует чтение учебных данных из текстового файла средствами java.nio.file. Он разбирает записи и обрабатывает ошибки формата и отсутствие файла. После запуска на корректном и повреждённом файлах он приводит сообщения к понятному виду. Итогом становится файловый обмен данными.
24	Регулярные выражения Студент компилирует шаблоны Pattern и проверяет учебные текстовые данные. Он разбирает корректные значения и отклоняет ошибочные с понятными сообщениями. После прогона набора примеров он уточняет шаблоны. Итогом становится проверка и разбор текста регулярными выражениями.
25	Сравнение структур данных Студент реализует стек и очередь на базе стандартных классов. Он выполняет компьютерный эксперимент по замеру времени типовых операций. После обработки замеров он обосновывает выбор структур. Итогом становится эмпирическое сравнение структур данных.
26	Контрольные примеры и автотесты JUnit 5 Студент подключает JUnit 5 и переносит контрольные примеры в автотесты. Он применяет утверждения и параметризованные тесты. После запуска он исправляет найденные ошибки программы или ожиданий. Итогом становится успешный прогон автотестов.
27	Отладка и диагностика дефектов Студент воспроизводит внесённый преподавателем дефект и локализует причину в отладчике. Он проверяет гипотезы по журналу отладки и вносит минимальное исправление. После повторного прогона он подтверждает устранение дефекта. Итогом становится исправленная программа и журнал отладки.
28	Сборка приложения в Maven Студент создаёт pom.xml и переносит исходные файлы и тесты в стандартные каталоги. Он подключает зависимости и выполняет сборку из командной строки. После устранения ошибок конфигурации он повторяет сборку из чистой копии. Итогом становится воспроизводимая сборка с автотестами.
29	Обмен данными в форматах CSV и JSON Студент реализует чтение и запись учебных данных в формате CSV. Он формирует и разбирает

№ п/п	Наименование лабораторных работ / краткое содержание
	простые документы JSON. После прогона примеров он проверяет соответствие данных эталону. Итогом становится обмен данными в стандартных форматах.
30	Простые запросы через JDBC Студент подключает учебную базу PostgreSQL или Postgres Pro через драйвер JDBC. Он выполняет чтение справочной таблицы и вставку записи параметризованным запросом. После запуска он проверяет результат выборкой. Итогом становится программа с простым обменом данными с базой.
31	Безопасная работа с отсутствующими значениями Студент заменяет нулевые ссылки на Optional в учебных функциях. Он применяет разворачивание значений и реакции по умолчанию. После прогона примеров он убеждается в отсутствии ошибок нулевых ссылок. Итогом становится устойчивая обработка неполных данных.
32	Утилиты стандартной библиотеки Студент применяет Math, Objects и Random в учебных задачах. Он настраивает форматирование чисел, дат и сообщений. После прогона примеров он сверяет вывод с эталоном. Итогом становится программа, использующая стандартную библиотеку.

Практические занятия

№ п/п	Тематика практических занятий/краткое содержание
1	Постановка учебной задачи и построение алгоритма Студент анализирует учебные задачи и выделяет входные данные, результаты и ограничения. Он строит блок-схему и псевдокод без привязки к синтаксису Java. Для нормальных и граничных примеров он выполняет ручную трассировку. Итогом становится утверждённый алгоритм и эталонные результаты для проверки программы.
2	Подготовка первой программы и выбор типов данных Студент планирует структуру первой программы и выбирает переменные с подходящими примитивными типами. Он определяет, какие вычисления и сообщения вывода требуются. В табличном редакторе он сопоставляет данные задачи и типы Java. Итогом становится задание на создание и запуск первой программы.
3	Проектирование ветвлений и циклов Студент выделяет условия выбора и повторяемые действия в учебных задачах. Он определяет порядок проверок и условия завершения циклов. В офисном редакторе он оформляет таблицу условий и ожидаемых результатов. Итогом становится задание на реализацию ветвлений и циклов.
4	Проектирование обработки массивов и строк Студент определяет, какие серии значений хранятся в массивах и какие поля читаются с консоли. Он выбирает операции обхода, поиска и суммирования. В табличном редакторе он формирует наборы примеров и ожидаемые сводки. Итогом становится задание на обработку массивов и строк.
5	Декомпозиция программы на методы и классы Студент разбивает программу на именованные методы с параметрами и возвращаемыми значениями. Он выделяет классы учебной модели и определяет их поля и инварианты. В офисном редакторе он оформляет схему методов и классов. Итогом становится задание на объектную структуру программы.
6	Проектирование иерархии классов и пакетов Студент проектирует наследование и интерфейсы для учебной модели. Он распределяет классы по пакетам и выбирает модификаторы доступа. В офисном редакторе он оформляет схему иерархии и структуру пакетов. Итогом становится задание на перестройку программы без изменения результата.
7	Проектирование надёжной обработки данных Студент определяет реакцию программы на ошибочный ввод и состав понятных сообщений. Он выбирает представление дат и денежных величин и фиксирует правила округления. В офисном

№ п/п	Тематика практических занятий/краткое содержание
	редакторе он оформляет таблицу решений о надёжности. Итогом становится задание на усиление устойчивости программы.
8	Оформление кода и подготовка репозитория Студент формирует контрольный перечень именования, форматирования и комментариев. Он определяет состав README, требования к Javadoc и правила сообщений коммитов. В офисном редакторе он готовит перечень публикации кода в GitFlic или GitVerse. Итогом становится регламент оформления и версионирования программы.
9	Выбор коллекций для учебных данных Студент сравнивает списки, множества и отображения по операциям и ограничениям. Он выбирает структуры для учебных наборов записей. В табличном редакторе он оформляет сравнительную таблицу коллекций. Итогом становится задание на работу с коллекциями.
10	Проектирование сравнения и сортировки объектов Студент определяет правила равенства и порядка для учебных записей. Он выбирает естественный и настраиваемый порядок сортировки. В табличном редакторе он оформляет критерии сравнения и ожидаемый порядок. Итогом становится задание на сортировку объектов.
11	Проектирование обобщённых компонентов Студент выделяет повторяющийся код, который можно обобщить по типу. Он определяет параметры типов и их ограничения. В офисном редакторе он оформляет схему обобщённых классов и методов. Итогом становится задание на обобщённые компоненты.
12	Проектирование обработки с лямбда-выражениями Студент выбирает операции фильтрации и преобразования учебных данных. Он определяет, какие функциональные интерфейсы требуются. В табличном редакторе он оформляет таблицу операций и ожидаемых результатов. Итогом становится задание на применение лямбда-выражений.
13	Проектирование конвейера обработки данных Студент проектирует цепочку операций Stream API для учебной сводки. Он определяет источник данных, промежуточные и терминальные операции. В табличном редакторе он оформляет схему конвейера и эталонный результат. Итогом становится задание на конвейерную обработку.
14	Проектирование перечислений и записей Студент выделяет фиксированные наборы значений и простые носители данных в учебной модели. Он выбирает между классом, записью и перечислением. В офисном редакторе он оформляет таблицу выбора типов. Итогом становится задание на перестройку модели данных.
15	Проектирование файлового обмена данными Студент определяет формат текстового файла с учебными данными, порядок чтения записей и реакцию программы на ошибки файла. Он выбирает, какие данные программа сохраняет в файл результата. В офисном редакторе он оформляет спецификацию формата файлов. Итогом становится задание на чтение и запись файлов.
16	Проектирование проверки текста регулярными выражениями Студент выделяет форматы учебных текстовых данных и правила их проверки. Он составляет регулярные выражения и наборы корректных и ошибочных примеров. В табличном редакторе он оформляет таблицу шаблонов и ожиданий. Итогом становится задание на проверку и разбор текста.
17	Оценка сложности и выбор структур данных Студент оценивает сложность типовых операций в нотации Big O. Он выбирает структуры данных для учебных сценариев с обоснованием. В табличном редакторе он оформляет план компьютерного эксперимента. Итогом становится задание на сравнение структур данных.
18	Проектирование контрольных примеров и автотестов Студент выделяет классы эквивалентности и граничные значения для учебных функций. Он проектирует автотесты и ожидаемые результаты. В табличном редакторе он формирует матрицу тестов. Итогом становится задание на автоматизированную проверку программы.

№ п/п	Тематика практических занятий/краткое содержание
19	Планирование отладки и диагностики Студент анализирует учебный дефект и формулирует проверяемые гипотезы. Он определяет контрольные точки и наблюдаемые переменные. В офисном редакторе он готовит журнал отладки. Итогом становится план локализации и устранения ошибки.
20	Подготовка приложения к сборке в Maven Студент сопоставляет текущую структуру каталогов приложения со стандартной структурой Maven. Он выбирает координаты, версию Java и состав файлов, которые требуется перенести. В офисном редакторе он оформляет план переноса и риски потери работоспособности. Итогом становится техническое задание на перевод приложения на сборку Maven.
21	Проектирование обмена данными в CSV и JSON Студент выбирает структуру учебных данных в форматах CSV и JSON. Он определяет правила чтения, записи и проверки значений. В табличном редакторе он оформляет примеры файлов и ожидаемые результаты. Итогом становится задание на обмен данными в стандартных форматах.
22	Знакомство с базой данных и JDBC Студент изучает предоставленный преподавателем эталонный пример подключения к базе данных. Он разбирает строку подключения, параметризованный запрос и обработку результата. В офисном редакторе он оформляет краткое описание учебной таблицы. Итогом становится задание на выполнение простых запросов через JDBC.
23	Проектирование безопасной работы с отсутствующими значениями Студент выделяет ситуации, где учебные данные могут отсутствовать. Он выбирает приёмы Optional вместо нулевых ссылок. В табличном редакторе он оформляет таблицу случаев и реакций программы. Итогом становится задание на устранение нулевых ссылок.
24	Применение стандартной библиотеки Java Студент выбирает утилитные классы стандартной библиотеки для учебных задач. Он определяет форматирование чисел, дат и сообщений в программе. В офисном редакторе он оформляет перечень применяемых классов. Итогом становится задание на доработку программы средствами стандартной библиотеки.

4.3. Самостоятельная работа обучающихся.

№ п/п	Вид самостоятельной работы
1	Изучение рекомендованной литературы.
2	Подготовка к лабораторным и практическим работам.
3	Подготовка к промежуточной аттестации.
4	Подготовка к текущему контролю.

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

№ п/п	Библиографическое описание	Место доступа
1	Курбатова И. В., Печкуров А. В. Основы программирования на языке Java. 2-е изд., стер. СПб.: Лань, 2026.	Лань : электронно-библиотечная система. – URL: https://e.lanbook.com/book/511511 (дата обращения: 30.08.2026).

	348 с. ISBN 978-5-507-51392-5. – Текст : электронный	
2	Карпович Е. Е. Основы алгоритмизации и программирования. СПб.: Лань, 2026. 144 с. ISBN 978-5-507-54192-8. – Текст : электронный	Лань : электронно-библиотечная система. – URL: https://e.lanbook.com/book/515057 (дата обращения: 30.08.2026).
3	Гуськова О. И. Объектно-ориентированное программирование в Java. М.: МПГУ, 2018. 240 с. ISBN 978-5-4263-0648-6. – Текст : электронный	Лань : электронно-библиотечная система. – URL: https://e.lanbook.com/book/122311 (дата обращения: 30.08.2026).
4	Старолетов С. М. Основы тестирования программного обеспечения. 3-е изд., стер. СПб.: Лань, 2023. 192 с. ISBN 978-5-507-45434-1. – Текст : электронный	Лань : электронно-библиотечная система. – URL: https://e.lanbook.com/book/269900 (дата обращения: 30.08.2026).
5	Рощин П. Г. Командная разработка программного обеспечения с помощью системы контроля версий Git: конспект лекций. М.: НИЯУ МИФИ, 2022. 108 с. ISBN 978-5-7262-2846-4. – Текст : электронный	Лань : электронно-библиотечная система. – URL: https://e.lanbook.com/book/355550 (дата обращения: 30.08.2026).
6	Моргунов Е. П. PostgreSQL. Основы языка SQL. СПб.: БХВ-Петербург, 2018. 336 с. ISBN 978-5-9775-4022-3. – Текст : электронный	URL: https://postgrespro.ru/education/books/sqlprimer; свободный PDF: https://edu.postgrespro.ru/sql_primer.pdf (дата обращения: 30.08.2026).

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

- ЭБС «Лань» — основные учебники дисциплины в электронном виде. URL: https://e.lanbook.com (дата обращения: 30.08.2026).

- eLIBRARY.RU — поиск научных публикаций и индексов цитирования. URL: https://elibrary.ru (дата обращения: 30.08.2026).

- ЭБС Znanium — учебная и научная литература для вузов. URL: https://znanium.ru (дата обращения: 30.08.2026).

- ЭБС IPRbooks — учебники и профессиональная литература. URL: https://www.iprbookshop.ru (дата обращения: 30.08.2026).

- Фонд «Защита национальных стандартов» — официальный банк ГОСТ, включая требования к программной документации. URL: https://protect.gost.ru (дата обращения: 30.08.2026).

J- ava SE 21 Documentation — документация платформы Java 21 (LTS). URL: https://docs.oracle.com/en/java/javase/21/ (дата обращения: 30.08.2026).

- Java SE 21 API Specification — нормативный справочник стандартной библиотеки. URL: https://docs.oracle.com/en/java/javase/21/docs/api/index.html (дата обращения: 30.08.2026).

- The Java Language Specification, Java SE 21 Edition — формальная спецификация языка. URL: https://docs.oracle.com/javase/specs/jls/se21/html/index.html (дата обращения: 30.08.2026).

- JEP 512: Compact Source Files and Instance Main Methods — компактные исходные файлы и упрощённая точка входа. URL: https://openjdk.org/jeps/512 (дата обращения: 30.08.2026).

- Axiom JDK — российская среда разработки и исполнения Java, руководство по установке. URL: https://axiomjdk.ru/products/axiom-jdk (дата обращения: 30.08.2026).

- Apache Maven — руководство по началу работы и справочник POM. URL: https://maven.apache.org/guides/getting-started/; https://maven.apache.org/pom.html (дата обращения: 30.08.2026).

- JUnit 5 User Guide — аннотации, утверждения, параметризованные тесты. URL: https://junit.org/junit5/docs/current/user-guide/ (дата обращения: 30.08.2026).

- Git Documentation и книга Pro Git на русском языке — команды, ветвление, слияние. URL: https://git-scm.com/doc; https://git-scm.com/book/ru/v2 (дата обращения: 30.08.2026).

- Документация GitFlic — репозитории, коммиты, совместная работа. URL: https://docs.gitflic.ru (дата обращения: 30.08.2026).

- Документация GitVerse — репозитории, задачи, образовательные сценарии. URL: https://gitverse.ru/docs/ (дата обращения: 30.08.2026).

- PostgreSQL Documentation — официальная документация по SQL и транзакциям. URL: https://www.postgresql.org/docs/ (дата обращения: 30.08.2026).

- Документация PostgreSQL на русском языке от Postgres Pro. URL: https://postgrespro.ru/docs/postgresql/16/ (дата обращения: 30.08.2026).

- Документация Astra Linux. URL: https://wiki.astralinux.ru (дата обращения: 30.08.2026).

- Документация ALT Linux. URL: https://docs.altlinux.org (дата обращения: 30.08.2026).

- Документация и база знаний РЕД ОС. URL: https://redos.red-soft.ru/support/docs/ (дата обращения: 30.08.2026).

- Центр поддержки Р7-Офис. URL: https://support.r7-office.ru/ (дата обращения: 30.08.2026).

- Документация МойОфис. URL: https://docs.myoffice.ru (дата обращения: 30.08.2026).

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

- Операционные системы: Astra Linux Common Edition «Орёл» (основная, бесплатна для образовательных организаций) / ALT Linux, Simply Linux / РЕД ОС.

- Платформа Java: Axiom JDK 21 (LTS) – российский дистрибутив на базе OpenJDK, свободен для учебной разработки.

- Среды разработки: IntelliJ IDEA (открытая сборка, Apache 2.0) / Eclipse IDE (EPL-2.0) / Apache NetBeans (Apache 2.0).

- Система сборки: Apache Maven (Apache 2.0).

- Модульное тестирование: JUnit 5 (EPL-2.0); встроенная СУБД H2 (MPL 2.0/EPL 1.0) для учебных тестов доступа к данным.

- Система контроля версий и хостинги кода: Git (GPLv2); российские платформы GitFlic и GitVerse (бесплатные тарифы, публикация репозиториев, ИИ-ассистенты отключены и при оценивании не используются).

- СУБД и SQL-клиенты: PostgreSQL и консольный клиент psql (PostgreSQL License) / Postgres Pro Standard (академические лицензии для вузов); графический клиент pgAdmin 4 или DBeaver Community Edition (Apache 2.0).

- Доступ к базам данных из Java: JDBC API в составе JDK и PostgreSQL JDBC Driver (BSD-2-Clause), без ORM.

- Документирование и качество кода: Javadoc (в составе JDK), Checkstyle (LGPL 2.1+).

- Офисные пакеты для отчётов: МойОфис Образование (бесплатен для образовательных организаций) / Р7-Офис (академические лицензии).

- Построение схем: draw.io / diagrams.net (Apache 2.0, основной) / PlantUML (дополнительный).

- Сравнение и слияние файлов (опционально): Meld (GPLv2).

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

Учебные аудитории для проведения учебных занятий, оснащенные компьютерной техникой и наборами демонстрационного оборудования.

Для практических и лабораторных занятий – наличие персональных компьютеров вычислительного класса.

9. Форма промежуточной аттестации:

Экзамен в 1, 2 семестрах.

10. Оценочные материалы.

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

Авторы:

старший преподаватель кафедры
«Цифровые технологии управления
транспортными процессами»
Института железнодорожного
транспорта И.С. Разживайкин

Согласовано:

Заведующий кафедрой ЦТУТП В.Е. Нутович

Директор ЦС IT - программ ВИШ О.Б. Проневич

Председатель учебно-методической
комиссии Н.А. Андриянова