

**Федеральное государственное автономное образовательное  
учреждение высшего образования  
«Московский физико-технический институт  
(национальный исследовательский университет)»**

**УТВЕРЖДЕНО**  
**Заместитель директора**

**А.И. Рыбакова**

	<b>Рабочая программа дисциплины (модуля)</b>
<b>по дисциплине:</b>	Программирование на Java: создание современных веб-приложений
<b>по направлению:</b>	Прикладные математика и физика
<b>профиль подготовки:</b>	Термоядерные и плазменные технологии Физтех-школа природоподобных, плазменных и ядерных технологий им. И.В. Курчатова центр дополнительного, дополнительного профессионального и онлайн-образования "Пуск"
<b>курс:</b>	1
<b>квалификация:</b>	магистр

Семестры, формы промежуточной аттестации:

1 (осенний) - Зачет  
2 (весенний) - Зачет

Аудиторных часов: 90 всего, в том числе:

лекции: 30 час.  
семинары: 60 час.  
лабораторные занятия: 0 час.

Самостоятельная работа: 180 час.

Всего часов: 270, всего зач. ед.: 6

Количество контрольных работ, заданий: 2

Программу составили:

Д.А. Басов, преподаватель  
Е.А. Решетникова, методист

Программа обсуждена на заседании центра дополнительного, дополнительного профессионального и онлайн-образования "Пуск" 29.01.2025

## Аннотация

Дисциплина "Программирование на Java: создание современных веб-приложений" ориентирована на изучение основ языка программирования Java и его использования для разработки веб-приложений. В рамках курса рассматриваются ключевые аспекты создания серверных компонентов, работа с базами данных, использование фреймворков Spring и Hibernate, а также основы фронтенда с применением HTML/CSS/JavaScript. Особое внимание уделяется практическим аспектам разработки, таким как внедрение CI/CD-процессов, тестирование кода и обеспечение безопасности приложений.

Дисциплина состоит из 6 модулей:

Модуль 1. Java. Основы

Модуль 2. Java Core

Модуль 3. Работа с базами данных

Модуль 4. Работа на сервере

Модуль 5. Работа с фреймворками

Модуль 6. Микросервисы и паттерны проектирования ПО

## 1. Цели и задачи

### Цель дисциплины

заключается в освоении фундаментальных принципов классической теории объектно-ориентированного программирования, а также изучении возможностей объектно-ориентированного и обобщенного программирования на языке Java.

### Задачи дисциплины

ознакомить студентов с основами языка Java и установить среду разработки;  
научить работать с типами данных, переменными, операторами и выражениями;  
сформировать умение использование условных операторов и циклов;  
научиться создавать и использовать методы;  
изучить основы ООП: классы, объекты, наследование;  
изучить работу с одномерными и многомерными массивами;  
освоить основы работы с коллекциями (List, Set, Map), их реализации и базовые операции;  
сформировать умение работы с механизмом обработки исключений в Java;  
научиться работать с файлами и потоками ввода/вывода в Java;  
научиться основам работы с многопоточности в Java;  
понять основы сетевого программирования в Java;  
научиться использовать лямбда-выражения для создания более выразительного и краткого кода;  
изучить основы реляционных баз данных и SQL;  
сформировать умение подключать Java-приложений к базам данных с использованием JDBC;  
научиться выполнять основные операции с базами данных через JDBC;  
научиться работать с транзакциями и их управлением;  
понять основы ORM (Object-Relational Mapping) и использование Hibernate;  
освоить принципы проектирования баз данных и создание слоя доступа к данным (DAO);  
изучить базовые и углубленные принципы клиент-серверной архитектуры и протокол HTTP;  
научить разработке веб-приложений с использованием сервлетов;  
научить использовать JSP и JavaBeans в веб-приложениях;  
научить разработке RESTful API на Java;  
научиться управлять сессиями и реализовывать аутентификацию пользователей;  
сформировать понимание архитектуры и основных принципов работы Spring;  
научиться управлять бинами и контекстом приложения в Spring;  
сформировать умение работать с упрощенной разработкой на Spring с использованием Spring Boot;  
научиться создавать RESTful сервисов с использованием Spring Boot;  
сформировать умение работать с базами данных с помощью Spring Data JPA;  
научиться обрабатывать и трансформировать данные внутри Spring приложений.

## 2. Перечень формируемых компетенций

Освоение дисциплины направлено на формирование следующих компетенций:

Код и наименование компетенции	Индикаторы достижения компетенции
УК-3 Способен организовывать и руководить работой команды, вырабатывая командную стратегию для достижения поставленной задачи	УК-3.1 Организует и координирует работу участников проекта, способствует конструктивному преодолению возникающих разногласий и конфликтов
	УК-3.4 Способен планировать командную работу, распределять поручения членам команды, организовать обсуждение разных идей и мнений
ОПК-4 Способен успешно реализовывать решение поставленной задачи, провести анализ результата и представить выводы, применяя знания и навыки в области физико-математических наук и информационно-коммуникационных технологий	ОПК-4.1 Способен применять знания и навыки по использованию информационно-коммуникационных технологий для поиска и изучения научной литературы, применения прикладных программных продуктов

## 3. Перечень планируемых результатов обучения по дисциплине (модулю)

В результате освоения дисциплины обучающиеся должны

знать:

Основные особенности языка Java, установка и настройка JDK и IDE.  
Понятия переменных, типов данных, литералов и приведение типов.  
Основные операторы и выражения, используемые в Java.  
Принципы работы инкремента и декремента, различия между постфиксной и префиксной формами, сокращенные операции присваивания.  
Логические операторы и их применение в условиях и выражениях.  
Принципы работы условных конструкций if-else, вложенных операторов if и тернарного оператора.  
Принцип работы оператора switch и его особенности по сравнению с if-else.  
Основные принципы работы циклов for, while, do-while, понимание бесконечных циклов и способы их предотвращения.  
Использование операторов break и continue в циклах для управления их выполнением.  
Основные понятия и предназначение методов в Java.  
Принципы объявления и вызова методов.  
Понятия параметров и аргументов методов, способы их передачи.  
Понятие области видимости переменных и их классификация (локальная, глобальная и т.д.).  
Принцип работы инструкции return и её использование в методах.  
Основы рекурсии, её преимущества и отличие от циклов.  
Основы обработки исключений, виды исключений и способы их обработки.  
Основы объектно-ориентированного программирования, включая понятия класс и объект.  
Принципы создания экземпляров классов и работа оператора new.  
Назначение и использование конструкторов, а также принципы работы сборщика мусора в Java.  
Различия между передачей по ссылке и передачей по значению.  
Использование объектов в качестве аргументов методов.  
Принципы и применение ключевого слова this в методах и конструкторах.  
Основные понятия массивов и их использование в Java.  
Методы инициализации массивов и доступ к их элементам.  
Принципы работы циклов при обходе массивов, в том числе использование цикла for-each.  
Концепции многомерных массивов и работы с массивами строк.  
Базовые операции с массивами, такие как копирование, сравнение и сортировка.  
Способы передачи массивов в методы и особенности работы с массивами внутри методов.

уметь:

Устанавливать JDK и настраивать интегрированную среду разработки (IDE).  
 Писать и запускать элементарные программы на Java.  
 Работать с переменными, выполнять типизацию и преобразования типов данных.  
 Использовать основные операторы и выражения для работы с данными.  
 Понимать и применять инкремент/декремент, сокращенные операции присваивания.  
 Использовать логические операторы для создания сложных условий и выражений.  
 Использовать управляющие конструкции if-else для ветвления программы.  
 Работать с оператором switch, понимать его отличия от if-else.  
 Работать с циклами for, while, do-while, осознавать риски бесконечных циклов и уметь их избегать.  
 Применять вложенные циклы.  
 Использовать операторы управления ходом циклов break и continue.  
 Понимать, что такое методы и для чего они нужны.  
 Объявлять и вызывать методы в программе.  
 Использовать параметры и аргументы методов.  
 Различать области видимости переменных в программе.  
 Применять инструкцию return для возврата значений из методов.  
 Понимать и использовать рекурсивные методы.  
 Применять механизм обработки исключений для повышения надежности и устойчивости приложения.  
 Понимать основные понятия объектно-ориентированного программирования (ООП), такие как класс и объект.  
 Создавать экземпляры классов с помощью оператора new.  
 Использовать конструкторы для инициализации объектов и понимать принцип работы сборщика мусора.  
 Различать передачу по ссылке и передачу по значению.  
 Передавать объекты в методы и использовать объекты как аргументы.  
 Использовать ключевое слово this для обращения к текущему экземпляру объекта.  
 Создавать и инициализировать массивы.  
 Получать доступ к элементам массива и изменять их.  
 Выполнять обход массива с использованием циклов, включая цикл for-each.  
 Работать с многомерными массивами и массивами строк.  
 Выполнять базовые операции с массивами (копирование, сравнение, сортировка).  
 Передавать массивы в методы и работать с ними внутри методов.

владеть:

- методами работы с основными структурами данных (массивы, списки, карты).
- навыками создания веб-приложений с использованием сервлетов и JSP (JavaServer Pages).
- опытом использования популярных фреймворков для разработки веб-приложений (Spring Framework, Spring MVC, Spring Security).
- навыками работы с библиотеками для обработки HTTP-запросов (Apache HttpClient, OkHttp).
- навыками реализации механизмов защиты данных (шифрование, хеширование паролей).

#### 4. Содержание дисциплины (модуля), структурированное по темам (разделам) с указанием отведенного на них количества академических часов и видов учебных занятий

##### 4.1. Разделы дисциплины (модуля) и трудоемкости по видам учебных занятий

№	Тема (раздел) дисциплины	Трудоемкость по видам учебных занятий, включая самостоятельную работу, час.			
		Лекции	Семинары	Лаборат. работы	Самост. работа
1	Java. Основы	5	10		30
2	Java Core	5	10		30
3	Работа с базами данных	5	10		30
4	Работа на сервере	5	10		30
5	Работа с фреймворками	5	10		30

6	Микросервисы и паттерны проектирования ПО	5	10		30
Итого часов		30	60		180
Подготовка к экзамену		0 час.			
Общая трудоёмкость		270 час., 6 зач.ед.			

#### 4.2. Содержание дисциплины (модуля), структурированное по темам (разделам)

##### Семестр: 1 (Осенний)

###### 1. Java. Основы

Цель: Заложить фундаментальные знания и навыки работы с языком Java, включая синтаксис, базовые концепции ООП и работу с простыми программами.

Введение в Java и базовый синтаксис

Управляющие конструкции (условные операторы, switch, циклы)

Методы и передача параметров

Введение в ООП

Работа с массивами

Практика: примеры и показ кода

###### 2. Java Core

Цель: развитие понимания и навыков работы с более сложными структурами данных и ключевыми возможностями Java, такими как коллекции, потоки ввода/вывода и исключения.

Введение в коллекции

Продвинутое ООП

Работа с файлами и потоками

Введение в многопоточность

Лямбда-выражения и Stream API

Практика: примеры и показ кода

###### 3. Работа с базами данных

Цель: научить студентов работать с базами данных с использованием Java, включая подключение, выполнение запросов и обработку данных.

Введение в реляционные базы данных и SQL

Подключение к базе данных с использованием JDBC

Выполнение CRUD операций через JDBC

Транзакции и управление транзакциями

ORM и работа с Hibernate

Проектирование базы данных и создание DAO

Практика: примеры и показ кода

Проект 1.

##### Семестр: 2 (Весенний)

###### 4. Работа на сервере

Цель: подготовить студентов к разработке серверных приложений на Java, включая основы веб-программирования, работу с сервлетами и разработку RESTful API.

Основы клиент-серверной архитектуры и HTTP

Введение в сервлеты

Разработка простого веб-приложения  
Введение в JSP и JavaBeans  
Создание RESTful API  
Управление сессиями и аутентификация  
Практика: примеры и показ кода  
Проект 2.

## 5. Работа с фреймворками

Цель: научить студентов разработке корпоративных приложений с использованием фреймворка Spring.  
Введение в Spring Framework  
Spring Core: Бины и контексты  
Введение в Spring Boot  
Разработка RESTful сервисов с Spring Boot  
Spring Data JPA: Работа с базами данных  
Обработка данных и трансформации  
Практика: примеры и показ кода

## 6. Микросервисы и паттерны проектирования ПО

Цель: дать студентам знания и навыки для разработки сложных программных систем с использованием микросервисной архитектуры и применения лучших практик проектирования программного обеспечения.  
Введение в микросервисы  
Коммуникации между микросервисами  
Управление состоянием и данными в микросервисах  
Обеспечение надежности и масштабируемости  
Паттерны проектирования ПО

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

Система дистанционного обучения:  
Обучающемуся необходимо наличие доступа в сеть интернет, компьютер.  
Преподавателю курса необходимо наличие доступа администратора курса и оборудование для проведения дистанционных семинаров (вебинаров), качественный отказоустойчивый доступ в сеть интернет.

## 6.Перечень рекомендуемой литературы

### Основная литература

Литература из средств кафедры:

1. Блох, Д. Java Эффективное программирование / Д. Блох. - М.: Лори, 2016. - 440 с.
2. Герберт, Шилдт Java 8. Руководство для начинающих / Шилдт Герберт. - М.: Диалектика / Вильямс, 2015. - 899 с.
3. Java: руководство для начинающих, 9-е изд./Герберт Шилдт; пер. с англ . Ю.Н. Артеменко. —Киев. : “Науковий Світ”, 2023. — 752 с. : ил . —Парал . тит. англ .
4. Хорстманн Кей С. Java. Библиотека профессионала. Том 1. Основы / Хорстманн Кей С.. - М.: Диалектика / Вильямс, 2019 г. - 864 с.
5. Хорстманн Кей С. Java. Библиотека профессионала. Том 2. Расширенные средства программирования, 11-е изд.: Пер с англ. - СПб.: / Диалектика, 2020г. - 864 с.

### Дополнительная литература

Дополнительная литература из средств кафедры:

1. Дубаков А.А. Введение в объектно-ориентированное программирование на Java: учебное пособие [Электронный ресурс]. – СПб, Университет ИТМО, 2016. – 250 с. – Режим доступа: <http://books.ifmo.ru/file/pdf/2139.pdf> свободный (дата обращения 12.01.2017).
2. Дэвид, Флэнаган JavaScript. Подробное руководство / Флэнаган Дэвид. - М.: Символ-плюс, 2017. - 555 с.
3. Перри, Брюс У. Java сервлеты и JSP: сб. рецептов
4. 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
5. Хорстманн, Кей С. Java SE 8. Вводный курс / Хорстманн Кей С.. - М.: Диалектика / Вильямс, 2014. - 898 с.

## **7. Перечень ресурсов информационно-телекоммуникационной сети "Интернет", необходимых для освоения дисциплины (модуля)**

1. Руководство по языку программирования Java <https://metanit.com/java/tutorial/>
2. <https://www.codewars.com/>
3. Java и Вы [Электронный ресурс]. – Режим доступа: <https://www.java.com/ru/> .– Загл. с экрана. (дата обращения: 12.01.2017)

## **8. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине (модулю), включая перечень необходимого программного обеспечения и информационных справочных систем (при необходимости)**

Перечень лицензионного и свободно распространяемого программного обеспечения, в том числе отечественного производства, необходимого для освоения программы

Операционные системы:

РЕД ОС

Windows

Mac

Linux

Astra Linux

Среда разработки (IDE):

IntelliJ IDEA Ultimate

Eclipse

NetBeans

Программы: JDK (Java Development Kit). Рекомендуемая версия: Oracle JDK версия 23. Java 23.

Сервер приложений или веб-сервер: Apache Tomcat.

Фреймворки для разработки веб-приложений: Spring Boot.

Базы данных: PostgreSQL.

Системы сборки:

Maven

Gradle

Система управления версиями:

Git

Для репозиторий: GitHub, GitLab, Bitbucket.

Инструменты тестирования:

JUnit

TestNG

Mockito

## **9. Методические указания для обучающихся по освоению дисциплины (модуля)**

Самостоятельная работа подразделяется на аудиторную и внеаудиторную. Аудиторную самостоятельную работу составляют практические задания, которые выполняются студентами во время учебных занятий, результаты ее выполнения проверяются и оцениваются преподавателем в учебном процессе.

Внеаудиторная самостоятельная работа включает формы: изучение дополнительной литературы, подготовка итоговых проектов по модулям, подготовка проекта.

Основными критериями качества организации самостоятельной работы служит наличие контроля результатов самостоятельной работы.

Основными современными формами организации самостоятельной работы являются творческие работы и работа с информационными компьютерными технологиями.



**ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ПО ДИСЦИПЛИНЕ (МОДУЛЮ)**

<b>по направлению:</b>	Прикладные математика и физика
<b>профиль подготовки:</b>	Термоядерные и плазменные технологии Физтех-школа природоподобных, плазменных и ядерных технологий им. И.В. Курчатова центр дополнительного, дополнительного профессионального и онлайн-образования "Пуск"
<b>курс:</b>	<u>1</u>
<b>квалификация:</b>	магистр

Семестры, формы промежуточной аттестации:

- 1 (осенний) - Зачет
- 2 (весенний) - Зачет

**Разработчики:**

Д.А. Басов, преподаватель  
Е.А. Решетникова, методист

## 1. Компетенции, формируемые в процессе изучения дисциплины

Код и наименование компетенции	Индикаторы достижения компетенции
УК-3 Способен организовывать и руководить работой команды, вырабатывая командную стратегию для достижения поставленной задачи	УК-3.1 Организует и координирует работу участников проекта, способствует конструктивному преодолению возникающих разногласий и конфликтов
	УК-3.4 Способен планировать командную работу, распределять поручения членам команды, организовать обсуждение разных идей и мнений
ОПК-4 Способен успешно реализовывать решение поставленной задачи, провести анализ результата и представить выводы, применяя знания и навыки в области физико-математических наук и информационно-коммуникационных технологий	ОПК-4.1 Способен применять знания и навыки по использованию информационно-коммуникационных технологий для поиска и изучения научной литературы, применения прикладных программных продуктов

## 2. Показатели оценивания компетенций

В результате изучения дисциплины «Программирование на Java: создание современных веб-приложений» обучающийся должен:

### знать:

Основные особенности языка Java, установка и настройка JDK и IDE.  
Понятия переменных, типов данных, литералов и приведение типов.  
Основные операторы и выражения, используемые в Java.  
Принципы работы инкремента и декремента, различия между постфиксной и префиксной формами, сокращенные операции присваивания.  
Логические операторы и их применение в условиях и выражениях.  
Принципы работы условных конструкций if-else, вложенных операторов if и тернарного оператора.  
Принцип работы оператора switch и его особенности по сравнению с if-else.  
Основные принципы работы циклов for, while, do-while, понимание бесконечных циклов и способы их предотвращения.  
Использование операторов break и continue в циклах для управления их выполнением.  
Основные понятия и предназначение методов в Java.  
Принципы объявления и вызова методов.  
Понятия параметров и аргументов методов, способы их передачи.  
Понятие области видимости переменных и их классификация (локальная, глобальная и т.д.).  
Принцип работы инструкции return и её использование в методах.  
Основы рекурсии, её преимущества и отличие от циклов.  
Основы обработки исключений, виды исключений и способы их обработки.  
Основы объектно-ориентированного программирования, включая понятия класс и объект.  
Принципы создания экземпляров классов и работа оператора new.  
Назначение и использование конструкторов, а также принципы работы сборщика мусора в Java.  
Различия между передачей по ссылке и передачей по значению.  
Использование объектов в качестве аргументов методов.  
Принципы и применение ключевого слова this в методах и конструкторах.  
Основные понятия массивов и их использование в Java.  
Методы инициализации массивов и доступ к их элементам.  
Принципы работы циклов при обходе массивов, в том числе использование цикла for-each.  
Концепции многомерных массивов и работы с массивами строк.  
Базовые операции с массивами, такие как копирование, сравнение и сортировка.  
Способы передачи массивов в методы и особенности работы с массивами внутри методов.

### уметь:

Устанавливать JDK и настраивать интегрированную среду разработки (IDE).  
 Писать и запускать элементарные программы на Java.  
 Работать с переменными, выполнять типизацию и преобразования типов данных.  
 Использовать основные операторы и выражения для работы с данными.  
 Понимать и применять инкремент/декремент, сокращенные операции присваивания.  
 Использовать логические операторы для создания сложных условий и выражений.  
 Использовать управляющие конструкции if-else для ветвления программы.  
 Работать с оператором switch, понимать его отличия от if-else.  
 Работать с циклами for, while, do-while, осознавать риски бесконечных циклов и уметь их избегать.  
 Применять вложенные циклы.  
 Использовать операторы управления ходом циклов break и continue.  
 Понимать, что такое методы и для чего они нужны.  
 Объявлять и вызывать методы в программе.  
 Использовать параметры и аргументы методов.  
 Различать области видимости переменных в программе.  
 Применять инструкцию return для возврата значений из методов.  
 Понимать и использовать рекурсивные методы.  
 Применять механизм обработки исключений для повышения надежности и устойчивости приложения.  
 Понимать основные понятия объектно-ориентированного программирования (ООП), такие как класс и объект.  
 Создавать экземпляры классов с помощью оператора new.  
 Использовать конструкторы для инициализации объектов и понимать принцип работы сборщика мусора.  
 Различать передачу по ссылке и передачу по значению.  
 Передавать объекты в методы и использовать объекты как аргументы.  
 Использовать ключевое слово this для обращения к текущему экземпляру объекта.  
 Создавать и инициализировать массивы.  
 Получать доступ к элементам массива и изменять их.  
 Выполнять обход массива с использованием циклов, включая цикл for-each.  
 Работать с многомерными массивами и массивами строк.  
 Выполнять базовые операции с массивами (копирование, сравнение, сортировка).  
 Передавать массивы в методы и работать с ними внутри методов.

#### **владеть:**

- методами работы с основными структурами данных (массивы, списки, карты).
- навыками создания веб-приложений с использованием сервлетов и JSP (JavaServer Pages).
- опытом использования популярных фреймворков для разработки веб-приложений (Spring Framework, Spring MVC, Spring Security).
- навыками работы с библиотеками для обработки HTTP-запросов (Apache HttpClient, OkHttp).
- навыками реализации механизмов защиты данных (шифрование, хеширование паролей).

### **3. Перечень типовых (примерных) вопросов, заданий, тем для подготовки к текущему контролю**

Задания на программирование по теме: Введение в Java и базовый синтаксис.

#### **1. Приветствие**

Напишите программу, которая выводит в консоль строку «Привет, мир!»

Ввод	Результат
-	Привет, мир!

#### **2. Приветствие пользователя.**

Напишите программу, которая будет приветствовать пользователя в формате «Привет, пользователь!». Имя должно передаваться при обращении к классу через командную строку, например: java Main.java Сергей

Ввод	Результат
Иван	Привет, Иван!

Анна   Привет, Анна!  
1234   Привет, 1234!

### 3. Программа о себе.

Напишите программу, которая выводит в консоль ваше имя, возраст, страну и любимое занятие. Данные должны передаваться при обращении к классу через командную строку, например: java Main.java Сергей 22 Россия Плавать

Ввод	Результат
Сергей 22 Россия Плавать	Имя: Сергей Возраст: 22 Страна: Россия Хобби: Плавать
Анна 12 Италия Гулять	Имя: Анна Возраст: 12 Страна: Италия Хобби: Гулять
Павел 44 Япония Спать	Имя: Павел Возраст: 44 Страна: Япония Хобби: Спать

Задания на программирование по теме: Работа с массивами.

#### 1. Создание и вывод массива чисел.

Создайте массив из пяти целых чисел от 1 до 5 и выведите его на экран.

Ввод	Результат
-	1 2 3 4 5

#### 2. Инициализация массива и доступ к элементам.

Создайте массив из трех строк, заполните его значениями "Яблоко", "Банан", "Апельсин". Выведите в консоль «Яблоко Апельсин»

Ввод	Результат
-	Яблоко Апельсин

#### 3. Вывод элементов массива с помощью for-each.

Создайте массив чисел 5, 10, 15, 20. С помощью for-each выведите значения массива в консоль.

Ввод	Результат
-	10 20 30 40 50

### 4. Перечень типовых (примерных) вопросов и тем для проведения промежуточной аттестации обучающихся

Итоговый тест по теме: Введение в Java и базовый синтаксис (Задание с автопроверкой).

#### 1. Какую роль выполняет JDK (Java Development Kit) в процессе разработки на Java?

- a) Он выполняет программу
- b) Он компилирует и запускает Java-программы
- c) Он используется только для написания кода
- d) Он предназначен для работы с базами данных

#### 2. Какие основные особенности выделяют язык Java? (выберите несколько вариантов)

- a) Платформонезависимость
- b) Низкий уровень безопасности
- c) Автоматическое управление памятью
- d) Сложность освоения для начинающих

3. Для чего используется IDE (среда разработки)?

- a) Для выполнения Java-программ
- b) Для упрощения написания, отладки и запуска программ
- c) Только для компиляции кода
- d) Для создания и запуска программ только на C++

4. Какой результат выполнения следующей программы?

```
public class HelloWorld {  
    public static void main(String[] args) {  
        System.out.println("Hello, Java!");  
    }  
}
```

- a) Ошибка компиляции
- b) Ошибка времени выполнения
- c) "Hello, Java!"
- d) "Hello World!"

5. Какой из этих типов данных в Java используется для хранения десятичных чисел?

- a) int
- b) double
- c) char
- d) boolean

Проект 1: Разработка новостного блога на Java с использованием PostgreSQL и работы с файлами

Цель: Создать приложение для управления и публикации новостного блога, взаимодействующее с пользователем через текстовые файлы. Новости и категории хранятся в базе данных PostgreSQL, либо другой по выбору.

Основное назначение приложения:

- Позволяет администраторам добавлять, редактировать и удалять статьи и категории через файлы ввода.
- Выводит результаты выполнения операций в файлы вывода.
- Пользователи могут просматривать статьи и категории, экспортируемые в файлы.

Функциональные требования

1. Файлы ввода:

input\_articles.txt — для операций с новостями.

input\_categories.txt — для операций с категориями.

2. Файлы вывода:

output\_articles.txt — список статей после выполнения операций.

output\_categories.txt — список категорий после выполнения операций.

3. Каждая операция в файле ввода оформляется строкой в формате:

- Для статей:

ADD|Title|Summary|Content|YYYY-MM-DD|CategoryName

UPDATE|ID|Title|Summary|Content|YYYY-MM-DD|CategoryName

DELETE|ID

- Для категорий:

ADD|CategoryName

UPDATE|ID|NewCategoryName

DELETE|ID

4. Программа выполняет операции получения всех новостей из БД с дальнейшей записью в файл.
5. Программа выполняет операции получения всех категорий из БД с дальнейшей записью в файл.
6. Программа выполняет операции получения одной новости по ID из БД с дальнейшей записью в файл.
7. Программа выполняет операции получения одной категории по ID из БД с дальнейшей записью в файл.
8. Программа выполняет операции обновления данных новости в БД.
9. Программа выполняет операции обновления данных категории в БД.
10. Программа выполняет операции удаления новости из БД.
11. Программа выполняет операции удаления категории из БД.
12. Программа выполняет операции добавления новости в БД.
13. Программа выполняет операции добавления категории в БД.

#### Технические требования

1. Язык разработки: Java 21 или новее.
2. База данных: PostgreSQL.
3. Обработка файлов: Используйте java.nio.file для работы с файлами.

#### Этапы разработки:

##### Этап 1: Настройка окружения

- Установите PostgreSQL и создайте базу данных.
- Создайте таблицы Category и Article в базе данных так, чтобы они могли работать с данными из файлов.

##### Этап 2. Реализация модели (Model): создайте классы сущностей и реализуйте в них запросы к СУБД.

- Сущность Category.
- Сущность Article.

##### Этап 3. Слой DAO: создайте классы для реализации логики функциональных требований проекта

- ArticleDAO
- CategoryDAO

##### Этап 4. Работа с файлами: создайте исполняемые файлы под каждое функциональное требование проекта.

##### Этап 5. Документация: Напишите проектную документацию по выполненному проекту.

#### Проект 2. Разработка интернет-магазина на Java с использованием PostgreSQL и веб-интерфейса

Цель: Создать веб-приложение для управления и работы интернет-магазина. Товары и категории хранятся в базе данных PostgreSQL, а взаимодействие с пользователем осуществляется через веб-интерфейс.

#### Основное назначение приложения

##### Административный функционал:

- Добавление, редактирование и удаление товаров и категорий через веб-страницы.

##### Пользовательский функционал:

- Просмотр всех товаров и категорий.
- Просмотр конкретного товара с подробным описанием.
- Фильтрация товаров по категориям.
- Добавление товаров в корзину.

#### Функциональные требования

## 1. Основные страницы веб-приложения:

### Главная страница:

- Отображает список всех товаров с краткой информацией (название, цена, описание).
- Ссылки на просмотр полного описания товара и фильтрацию по категориям.

### Страница управления категориями:

- Форма для добавления категории.
- Список всех категорий с возможностью редактирования и удаления.

### Страница управления товарами:

- Форма для добавления и редактирования товара (название, описание, цена, количество, категория).
- Список всех товаров с кнопками "Редактировать" и "Удалить".

### Страница товара:

- Полное отображение выбранного товара (название, описание, цена, категория, количество на складе).
- Кнопка "Добавить в корзину".

### Страница корзины:

- Список товаров, добавленных в корзину, с подсчетом итоговой стоимости.
- Возможность удаления товаров из корзины.

## 2. API действий через веб-интерфейс:

- Добавление, редактирование, удаление товара.
- Добавление, редактирование, удаление категории.
- Фильтрация товаров по категориям.
- Управление корзиной (добавление/удаление товаров).

## Технические требования

1. Язык разработки: Java 21 или новее.
2. Фреймворк: Spring Boot.
3. База данных: PostgreSQL.
4. Фронтенд: HTML, CSS, Bootstrap (для стилей).

## Этапы разработки

### Этап 1: Настройка окружения:

- Установите PostgreSQL и настройте базу данных:
- Создайте таблицы для всех сущностей.

### Этап 2: Реализация модели (Model)

- Создайте сущность Category.
- Создайте сущность Product.

### Этап 3: Реализация контроллеров (Controller)

- CategoryController
- ProductController
- CartController

### Этап 4: Реализация представлений (View)

- Шаблоны:

templates/categories.html: страница управления категориями.

templates/products.html: страница управления товарами.

templates/product.html: страница просмотра одного товара.

templates/cart.html: страница просмотра корзины.

- Используйте Bootstrap для создания удобного интерфейса.

## Критерии оценивания

### Критерии оценивания проекта 1.

#### 1. Программа запускается

- Все необходимые библиотеки и зависимости корректно установлены и настроены – 10 баллов
- Приложение корректно обрабатывает ошибки и исключения – 10 баллов

## 2. Реализована требуемая функциональность

- Добавление, просмотр, редактирование и удаление категорий расходов – 4 балла
- Добавление, просмотр, редактирование и удаление записей о расходах – 4 балла
- Каждая запись о расходе относится к одной категории – 4 балла
- Отслеживание расходов за день/неделю/месяц – 4 балла
- Возможность ограничения бюджета на день/неделю/месяц – 4 балла

## 3. Корректное оформление структуры БД и связи в таблицах

- Таблица Category создана и имеет правильную структуру – 8.
- Таблица Article создана и имеет правильную структуру – 8.
- Таблицы связаны между собой – 4.

## 4. Следование заданной архитектуре

- Работа с таблицами ведется в моделях – 10 баллов.
- Логика вынесена в контроллеры – 10 баллов.

## 5. Общая оценка качества кода

- Все элементы кода имеют понятные и информативные названия – 8 балла
- Документация содержит описание каждого метода и класса – 5 балла
- Код написан таким образом, что его легко читать и понимать, без лишних усложнений и запутанности – 7 балла

## Критерии оценивания проекта 2:

### 1. Программа запускается

#### Правильная настройка окружения (10 баллов):

- Все необходимые библиотеки и зависимости корректно установлены и настроены.
- Правильно настроено подключение к базе данных PostgreSQL.
- Приложение успешно запускается на локальном сервере.

#### Обработка ошибок и исключений (10 баллов):

- Программа корректно обрабатывает исключения при работе с базой данных (например, ошибки связи с БД, ошибки запроса).
- Предусмотрены проверки на корректность данных, введенных пользователем.

### 2. Реализована требуемая функциональность (20 баллов)

#### Добавление, редактирование и удаление категорий товаров (4 баллов):

- Администратор может добавить, редактировать и удалять категории товаров через веб-страницу.
- Все изменения сохраняются в базе данных.

#### Добавление, редактирование и удаление товаров (4 баллов):

- Администратор может добавлять, редактировать и удалять товары через веб-интерфейс.
- Все изменения сохраняются в базе данных.

#### Отображение списка товаров с возможностью фильтрации по категориям (4 баллов):

- На главной странице отображается список товаров.
- Пользователь может фильтровать товары по категориям.

#### Просмотр подробной информации о товаре (3 баллов):

- Пользователь может выбрать товар и просмотреть его подробное описание, цену, количество и другие характеристики.

#### Функциональность корзины (3 баллов):

- Пользователь может добавлять товары в корзину.
- Пользователь может удалять товары из корзины.
- Корзина отображает список выбранных товаров и их колич

### 3. Структура базы данных и связи между таблицами (20 баллов)

#### Таблица Category создана и имеет правильную структуру (8 баллов):

- Таблица категории содержит все необходимые поля (например, ID и название категории).
- Таблица имеет правильные ограничения (например, уникальность имени категории).

#### Таблица Product создана и имеет правильную структуру (8 баллов):

- Таблица товаров содержит все необходимые поля (например, ID, название, описание, цена, количество, внешний ключ для категории).
- Таблица правильно связана с таблицей категорий через внешний ключ.



Таблицы связаны между собой (4 балла):

- Взаимосвязь между таблицами реализована через внешний ключ в таблице товаров.
- Обеспечена целостность данных (например, удаление категории должно привести к удалению всех товаров, относящихся к ней).

4. Следование заданной архитектуре (20 баллов)

Работа с данными ведется через модели (10 баллов):

- Данные обрабатываются в моделях.
- Каждая модель отвечает за работу с одной таблицей и выполняет весь цикл работы.

Использование Spring MVC для реализации контроллеров (10 баллов):

- Приложение использует архитектуру Spring MVC.
- Логика представлений отделена от бизнес-логики, с использованием контроллеров, сервисов и репозитория.

5. Качество кода и документации (10 баллов)

Читабельность и структура кода (4 баллов):

- Все элементы кода имеют понятные и информативные названия (переменные, методы, классы).
- Код легко читается и структурирован (правильное форматирование, комментарии).

Документация (4 баллов):

- В проекте есть документация для каждого класса и метода.
- Описание методов и их функционала.
- Описание процесса установки и запуска приложения.

Отсутствие избыточности и усложнений в коде (2 балла):

- Код написан с учетом принципов DRY (Don't Repeat Yourself) и KISS (Keep It Simple, Stupid).
- Логика приложения не излишне усложнена.

6. Пользовательский интерфейс (10 баллов)

Интуитивно понятный и удобный интерфейс (5 баллов):

- Веб-интерфейс удобен для пользователя и легко воспринимается.
- Навигация по страницам проста, оформление соответствует стандартам веб-дизайна.

Корректная работа с фронтенд-частью (5 баллов):

- Страницы корректно отображаются в популярных браузерах.

Максимальная сумма, которую можно набрать, успешно выполнив все контрольные мероприятия, составляет 100 баллов. Для получения положительной оценки «зачтено» необходимо набрать не менее 30 баллов.

Оценка «зачтено» выставляется студенту, если он показал всесторонние, систематизированные, глубокие знания учебной программы дисциплины и умение уверенно применять их на практике при решении конкретных задач, свободное и правильное обоснование принятых решений.

Оценка «не зачтено» выставляется студенту, который не знает большей части основного содержания учебной программы дисциплины, допускает грубые ошибки в формулировках основных понятий дисциплины и не умеет использовать полученные знания при решении типовых практических задач.

Составляющие процесса обучения, которые оцениваются в ходе обучения, и их вклад в итоговый зачет представлены ниже:

Модуль	Вклад в зачет, %
Оценка за модуль 1	20
Оценка за модуль 2	20
Оценка за модуль 3	20
Зачет 1	40
Оценка за модуль 4	20
Оценка за модуль 5	20
Оценка за модуль 6	20
Зачет 2	40

## **5. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности**

Во время проведения зачета обучающиеся могут пользоваться программой дисциплины.