

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

**УТВЕРЖДЕНО**

**Директор физтех-школы  
прикладной математики и  
информатики**

**А.М. Райгородский**

	<b>Рабочая программа дисциплины (модуля)</b>
<b>по дисциплине:</b>	Программирование на Java
<b>по направлению:</b>	Прикладная математика и информатика
<b>профиль подготовки:</b>	Прикладная математика, компьютерные науки и инженерия Физтех-школа Прикладной Математики и Информатики кафедра алгоритмов и технологий программирования
<b>курс:</b>	2
<b>квалификация:</b>	бакалавр

Семестр, формы промежуточной аттестации: 3 (осенний) - Дифференцированный зачет

Аудиторных часов: 60 всего, в том числе:

лекции: 30 час.

семинары: 30 час.

лабораторные занятия: 0 час.

Самостоятельная работа: 30 час.

Всего часов: 90, всего зач. ед.: 2

Количество контрольных работ, заданий: 2

Программу составил: О.Н. Ивченко, заведующий кафедрой

Программа обсуждена на заседании кафедры алгоритмов и технологий программирования 10.02.2025

## Аннотация

Данный курс посвящен изучению языка программирования Java – одного из наиболее популярных и востребованных языков в мире. Курс предоставит студентам фундаментальные знания основ программирования на Java и объектно-ориентированного программирования (ООП). В ходе обучения студенты освоят синтаксис Java, научатся работать с различными типами данных, структурами управления потоком выполнения, массивами и коллекциями.

Особое внимание будет уделено принципам объектно-ориентированного программирования: инкапсуляции, наследованию, полиморфизму и абстракции. Студенты научатся создавать классы, объекты, использовать интерфейсы и абстрактные классы, а также разбираться в механизмах наследования и перегрузки методов.

### 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;

### 2. Перечень формируемых компетенций

Освоение дисциплины направлено на формирование следующих компетенций:

Код и наименование компетенции	Индикаторы достижения компетенции
ОПК-1 Способен применять фундаментальные знания, полученные в области физико-математических и (или) естественных наук и использовать их в профессиональной деятельности	ОПК-1.1 Способен анализировать поставленную задачу, намечать пути ее решения
	ОПК-1.2 Способен строить математические модели, производить количественные расчеты и оценки
	ОПК-1.3 Способен определять границы применимости полученных результатов
ПК-1 Способен ставить, формализовывать и решать задачи, в том числе разрабатывать и исследовать математические модели изучаемых явлений и процессов, системно анализировать научные проблемы, получать новые научные результаты	ПК-1.1 Способен находить, анализировать и обобщать информацию об актуальных результатах исследований в рамках тематической области своей профессиональной деятельности
	ПК-1.2 Способен выдвигать гипотезы, строить математические модели для описания изучаемых явлений и процессов, оценивать качество разработанной модели
	ПК-1.3 Способен применять теоретические и (или) экспериментальные методы исследований к конкретной научной задаче и интерпретировать полученные результаты

### 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

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

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	5		5
2	Java Core	5	5		5
3	Работа с базами данных	5	5		5
4	Работа на сервере	5	5		5

5	Работа с фреймворками	5	5		5
6	Микросервисы и паттерны проектирования ПО	5	5		5
Итого часов		30	30		30
Подготовка к экзамену		0 час.			
Общая трудоёмкость		90 час., 2 зач.ед.			

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

##### Семестр: 3 (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

##### 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

## 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)

Учебная аудитория, оснащенная компьютером и мультимедийным оборудованием (проектор, звуковая система).

## 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)**

Не используются

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

На лекционных занятиях используются мультимедийные технологии, включая демонстрацию презентаций.

Для контроля и коррекции знаний обучающиеся могут использовать компьютерное тестирование.

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

Студент, изучающий дисциплину, должен с одной стороны, овладеть общим понятийным аппаратом, а с другой стороны, должен научиться применять теоретические знания на практике.

Успешное освоение дисциплины требует:

- посещения студентом всех видов аудиторных занятий;
- ведения конспекта в ходе аудиторных занятий;
- качественной самостоятельной подготовки к практическим занятиям, активной работы на них;
- активной самостоятельной и аудиторной работы студента;
- своевременной сдачи преподавателю заданий по аудиторным видам работ.

**ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ПО ДИСЦИПЛИНЕ (МОДУЛЮ)**

<b>по направлению:</b>	Прикладная математика и информатика
<b>профиль подготовки:</b>	Прикладная математика, компьютерные науки и инженерия Физтех-школа Прикладной Математики и Информатики кафедра алгоритмов и технологий программирования
<b>курс:</b>	<u>2</u>
<b>квалификация:</b>	бакалавр

Семестр, формы промежуточной аттестации: 3 (осенний) - Дифференцированный зачет

**Разработчик:** О.Н. Ивченко, заведующий кафедрой

## 1. Компетенции, формируемые в процессе изучения дисциплины

Код и наименование компетенции	Индикаторы достижения компетенции
ОПК-1 Способен применять фундаментальные знания, полученные в области физико-математических и (или) естественных наук и использовать их в профессиональной деятельности	ОПК-1.1 Способен анализировать поставленную задачу, намечать пути ее решения
	ОПК-1.2 Способен строить математические модели, производить количественные расчеты и оценки
	ОПК-1.3 Способен определять границы применимости полученных результатов
ПК-1 Способен ставить, формализовывать и решать задачи, в том числе разрабатывать и исследовать математические модели изучаемых явлений и процессов, системно анализировать научные проблемы, получать новые научные результаты	ПК-1.1 Способен находить, анализировать и обобщать информацию об актуальных результатах исследований в рамках тематической области своей профессиональной деятельности
	ПК-1.2 Способен выдвигать гипотезы, строить математические модели для описания изучаемых явлений и процессов, оценивать качество разработанной модели
	ПК-1.3 Способен применять теоретические и (или) экспериментальные методы исследований к конкретной научной задаче и интерпретировать полученные результаты

## 2. Показатели оценивания компетенций

В результате изучения дисциплины «Программирование на Java» 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

### 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

#### **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**

1. Что такое JVM (Java Virtual Machine) и как она работает?
2. Объясните разницу между типами данных int, Integer, float и Double.
3. Что такое автоупаковка и распаковка в Java? Приведите примеры.
4. Опишите назначение модификаторов доступа public, private, protected и default.
5. Что такое перегрузка методов (method overloading) и переопределение методов (method overriding)? В чем разница?
6. Объясните концепцию наследования в Java. Что такое интерфейсы и абстрактные классы?
7. Как работает механизм исключений (exceptions) в Java? Какие типы исключений вы знаете?
8. Объясните, что такое finally блок в конструкции try-catch-finally.
9. Что такое полиморфизм и как он реализуется в Java?
10. Объясните принцип инкапсуляции в контексте Java.

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

1. Что такое абстрактный класс и зачем он нужен? В чем его отличие от интерфейса?
2. Объясните концепцию статических членов класса (static).
3. Опишите основные интерфейсы коллекций в Java (List, Set, Map). Приведите примеры их реализации.
4. В чем разница между ArrayList и LinkedList? Когда лучше использовать каждый из них?
5. Что такое итератор (Iterator) и как он используется для обхода коллекций?
6. Что такое потоки (Threads) в Java? Как создать и запустить новый поток?
7. Как обеспечить синхронизацию доступа к общим ресурсам между несколькими потоками? Какие механизмы синхронизации вы знаете?
8. Как осуществляется чтение и запись данных в файлы в Java?
9. Что такое Generics в Java и зачем они нужны?
10. Напишите небольшой фрагмент кода на Java, демонстрирующий использование лямбда-выражений.

#### **Критерии оценивания**

- оценка «отлично (10)» выставляется студенту, показавшему 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 «отлично (9)» выставляется студенту, показавшему всесторонние, систематизированные, глубокие знания учебной программы дисциплины и умение применять их на практике при решении конкретных задач, свободное и правильное обоснование принятых решений
- оценка «отлично (8)» выставляется студенту, показавшему всесторонние систематизированные, глубокие знания учебной программы дисциплины и умение применять их на практике при решении конкретных задач, и правильное обоснование принятых решений
- оценка «хорошо (7)» выставляется студенту, если он твердо знает материал, грамотно и по существу излагает его, умеет применять полученные знания на практике, но допускает в ответе или в решении задач некоторые неточности;
- оценка «хорошо (6)» выставляется студенту, если он знает материал, грамотно и по существу излагает его, умеет применять полученные знания на практике, но допускает в ответе или в решении задач некоторые неточности;
- оценка «хорошо (5)» выставляется студенту, если он знает материал, и по существу излагает его, умеет применять полученные знания на практике, но допускает в ответе или в решении задач некоторые неточности;
- оценка «удовлетворительно (4)» выставляется студенту, показавшему фрагментарный, разрозненный характер знаний, недостаточно правильные формулировки базовых понятий, нарушения логической последовательности в изложении программного материала, но при этом он владеет основными разделами учебной программы, необходимыми для дальнейшего обучения и может применять полученные знания по образцу в стандартной ситуации;
- оценка «удовлетворительно (3)» выставляется студенту, показавшему фрагментарный, разрозненный характер знаний, недостаточно правильные формулировки базовых понятий, нарушения логической последовательности в изложении программного материала, но при этом он владеет фрагментарно основными разделами учебной программы, необходимыми для дальнейшего обучения и может применять полученные знания по образцу в стандартной ситуации;
- оценка «неудовлетворительно (2)» 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
- оценка «неудовлетворительно (1)» выставляется студенту, который не знает формулировок основных понятий дисциплины

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

При проведении зачета обучающемуся предоставляется 30 минут на подготовку. Опрос обучающегося не должен превышать одного астрономического часа.

Во время проведения зачета обучающиеся могут пользоваться программой дисциплины, а также справочной литературой, конспектами лекций или другими материалами.