

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

УТВЕРЖДЕНО

**Директор высшей школы
программной инженерии
А.В. Малеев**

	Рабочая программа дисциплины (модуля)
по дисциплине:	Программирование на языке Java
по направлению:	Программная инженерия
профиль подготовки:	Разработка программно-информационных систем высшая школа программной инженерии высшая школа программной инженерии
курс:	1
квалификация:	бакалавр

Семестры, формы промежуточной аттестации:

1 (осенний) - Дифференцированный зачет
2 (весенний) - Экзамен

Аудиторных часов: 120 всего, в том числе:

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

Самостоятельная работа: 30 час.

Подготовка к экзамену: 30 час.

Всего часов: 180, всего зач. ед.: 5

Количество контрольных работ, заданий: 2

Программу составил: А.В. Созыкин, канд. техн. наук

Программа обсуждена на заседании высшей школы программной инженерии 06.06.2022

Аннотация

В рамках модуля «Программирование на языке Java» рассматриваются темы, которые позволят слушателям сформировать систему понятий, знаний, умений и навыков в области современного программирования, включающего в себя методы проектирования, анализа и создания программных продуктов на языке Java, основанные на использовании объектно-ориентированной методологии. Целью освоения данного модуля является получение профессиональных компетенций, необходимых для создания приложений на языке Java.

1. Цели и задачи

Цель дисциплины

- формирование у обучающихся прикладных знаний и навыков по работе с языком программирования Java, ключевым его инструментарием и особенностями его использования в разработке.

Задачи дисциплины

- сформировать у студентов целостное представление о принципах построения и функционирования Java;
- освоение студентами навыков разработки и проектирования приложений на языке java с использованием инструментария веб-разработки и объектно-ориентированного программирования;
- формирование у студентов навыков использования передовых инструментов разработки для решения практических задач.

2. Перечень формируемых компетенций

Освоение дисциплины направлено на формирование следующих компетенций:

Код и наименование компетенции	Индикаторы достижения компетенции
УК-1 Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач	УК-1.2 Находит, критически анализирует и выбирает информацию, необходимую для решения поставленной задачи
	УК-1.5 Определяет и оценивает практические последствия возможных вариантов решения задачи
ОПК-3 Способен решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности	ОПК-3.1 Обладает навыками разработки архитектуры программных систем и компонентов с учетом требований к производительности, надежности и безопасности
	ОПК-3.2 Умеет применять информационно-коммуникационные технологии для поиска и анализа профессиональной информации, выделения в ней главного, структурирования, оформления и представления в виде аналитических обзоров с обоснованными выводами и рекомендациями
	ОПК-3.3 Знает основы информационной безопасности и методы защиты программного обеспечения от угроз и атак
ОПК-6 Способен разрабатывать алгоритмы и программы, пригодные для практического использования, применять основы информатики и программирования к проектированию, конструированию и тестированию программных продуктов	ОПК-6.1 Знает алгоритмы решения типовых задач, области и способы их применения
	ОПК-6.2 Умеет применять языки программирования для решения прикладных задач
	ОПК-6.3 Знает методы тестирования программного кода на ошибки и способен проводить тестирование на различных уровнях (модульное, интеграционное, системное)
	ОПК-6.4 Имеет навыки программирования и тестирования программных продуктов

ПК-4 Способен разрабатывать тесты, подготавливать тестовые данные, проводить тестирование, разрабатывать документы для тестирования	ПК-4.2 Умеет оценивать важность различных тестов на основе приоритетов пользователя, проектных задач и рисков возникновения ошибки
---	--

3. Перечень планируемых результатов обучения по дисциплине (модулю)

В результате освоения дисциплины обучающиеся должны

знать:

- основные информационные источники, содержащие термины и понятия, относящиеся к языку программирования Java;
- основной инструментарий программирования на языке Java, в частности: типы данных языка Java, управление потоком выполнения в Java, иерархию классов стандартной библиотеки, правила работы с исключениями, возможности Java Reflection API.

уметь:

- оперировать понятиями типов, синтаксиса, основных конструкций;
- грамотно использовать типы, конструкции языка при решении задач.

владеть:

- навыками разработки программ на языке программирования Java.

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	Многопоточность в языке Java.	15	15		6
2	Инструменты языка программирования Java.	15	15		6
3	Многопоточность в языке Java.	15	15		8
4	Экосистема современного программирования.	15	15		10
Итого часов		60	60		30
Подготовка к экзамену		30 час.			
Общая трудоёмкость		180 час., 5 зач.ед.			

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

Семестр: 1 (Осенний)

1. Многопоточность в языке Java.

- Основные понятия Java - JDK, JRE, Java SE/EE/ME
- Написание, компиляция и запуск первого Java-приложения из командной строки. Упаковка в JAR файл
- Знакомство со средой разработки IntelliJ IDEA

2. Инструменты языка программирования Java.

VM. Прimitivesкие типы. Интерфейсы. Иерархия классов. Исключения. Ввод-вывод (java.io, java.nio). Инициализация объектов. Enum. Лямбды. Коллекции. Сборка мусора. JUnit. Java Reflection API. Аннотации. Dynamic proxy. Сериализация.

Семестр: 2 (Весенний)

3. Многопоточность в языке Java.

Прimitivesкие языка: Thread, synchronized, volatile, wait/notify, final.

Семантика многопоточных программ: Java Memory Model.

Java.util.concurrent: atomic-типы, локи (в т. ч. ReadWriteLock), семафоры, thread-safe коллекции, ThreadLocal.

4. Экосистема современного программирования.

Форматы XML, JSON, CSV.

Кодировки текстовых данных.

Распределенная система контроля версий git.

Стек TCP/IP, протоколы HTTP/HTTPS, HTTP API, JSON API.

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

Компьютер, мультимедийное оборудование (проектор).

6. Перечень рекомендуемой литературы

Основная литература

1. Философия JAVA , Электрон. версия печ. публикации / Б. Эккель. — Санкт-Петербург, Питер, 2019

Дополнительная литература

Рекомендуемая литература для самостоятельного изучения:

Java 8 : полное руководство : пер. с англ., Шилдт, Г., 2015.

Java без сбоев: обработка исключений, тестирование, отладка [Текст] / С. Стелтинг ; [пер. с англ. В. И. Казаченко] .— М. : КУДИЦ-ОБРАЗ, 2005 .— 464 с.

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

Не требуются

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 программного обеспечения:

Office 365 EDUA5 ShrdSvr ALNG

SubsVL MVL PerUsr B Faculty EES

Microsoft Windows 8.1 Pro 64-bit

RUS OLP NL Acdmc

Eclipse IDE for Java EE Developers

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

Программа курса в разделе «самостоятельная работа» обозначает минимально необходимое время для работы студента над темой.

Самостоятельная работа включает в себя:

- проработку учебного материала (по учебной и научной литературе), подготовку ответов на вопросы, предназначенных для самостоятельного изучения, доказательство отдельных утверждений, свойств;
- выполнение домашних заданий.

ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ПО ДИСЦИПЛИНЕ (МОДУЛЮ)

по направлению:	Программная инженерия
профиль подготовки:	Разработка программно-информационных систем высшая школа программной инженерии высшая школа программной инженерии
курс:	<u>1</u>
квалификация:	бакалавр

Семестры, формы промежуточной аттестации:

- 1 (осенний) - Дифференцированный зачет
- 2 (весенний) - Экзамен

Разработчик: А.В. Созыкин, канд. техн. наук

1. Компетенции, формируемые в процессе изучения дисциплины

Код и наименование компетенции	Индикаторы достижения компетенции
УК-1 Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач	УК-1.2 Находит, критически анализирует и выбирает информацию, необходимую для решения поставленной задачи
	УК-1.5 Определяет и оценивает практические последствия возможных вариантов решения задачи
ОПК-3 Способен решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности	ОПК-3.1 Обладает навыками разработки архитектуры программных систем и компонентов с учетом требований к производительности, надежности и безопасности
	ОПК-3.2 Умеет применять информационно-коммуникационные технологии для поиска и анализа профессиональной информации, выделения в ней главного, структурирования, оформления и представления в виде аналитических обзоров с обоснованными выводами и рекомендациями
	ОПК-3.3 Знает основы информационной безопасности и методы защиты программного обеспечения от угроз и атак
ОПК-6 Способен разрабатывать алгоритмы и программы, пригодные для практического использования, применять основы информатики и программирования к проектированию, конструированию и тестированию программных продуктов	ОПК-6.1 Знает алгоритмы решения типовых задач, области и способы их применения
	ОПК-6.2 Умеет применять языки программирования для решения прикладных задач
	ОПК-6.3 Знает методы тестирования программного кода на ошибки и способен проводить тестирование на различных уровнях (модульное, интеграционное, системное)
	ОПК-6.4 Имеет навыки программирования и тестирования программных продуктов
ПК-4 Способен разрабатывать тесты, подготавливать тестовые данные, проводить тестирование, разрабатывать документы для тестирования	ПК-4.2 Умеет оценивать важность различных тестов на основе приоритетов пользователя, проектных задач и рисков возникновения ошибки

2. Показатели оценивания компетенций

В результате изучения дисциплины «Программирование на языке Java» обучающийся должен:

знать:

- основные информационные источники, содержащие термины и понятия, относящиеся к языку программирования Java;
- основной инструментарий программирования на языке Java, в частности: типы данных языка Java, управление потоком выполнения в Java, иерархию классов стандартной библиотеки, правила работы с исключениями, возможности Java Reflection API.

уметь:

- оперировать понятиями типов, синтаксиса, основных конструкций;
- грамотно использовать типы, конструкции языка при решении задач.

владеть:

- навыками разработки программ на языке программирования Java.

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

Текущий контроль

1. Ввести n строк с консоли, найти самую короткую строку. Вывести эту строку и ее длину.

2. Ввести n строк с консоли. Упорядочить и вывести строки в порядке возрастания их длин, а также (второй приоритет) значений этих их длин.
3. Ввести n строк с консоли. Вывести на консоль те строки, длина которых меньше средней, также их длины.
4. В каждом слове текста k-ю букву заменить заданным символом. Если k больше длины слова, коррекцию не выполнять.
5. В русском тексте каждую букву заменить ее номером в алфавите. В одной строке печатать текст с двумя пробелами между буквами, в следующей строке внизу под каждой буквой печатать ее номер.
6. Из небольшого текста удалить все символы, кроме пробелов, не являющиеся буквами. Между последовательностями подряд идущих букв оставить хотя бы один пробел.
7. Из текста удалить все слова заданной длины, начинающиеся на согласную букву.
8. В тексте найти все пары слов, из которых одно является обращением другого.
9. Найти и напечатать, сколько раз повторяется в тексте каждое слово.
10. Найти, каких букв, гласных или согласных, больше в каждом предложении текста.

Лабораторные работы

Контейнеры, IO потоки, классы Object и String.

Написать программу, которая будет принимать в качестве аргумента имя текстового файла, и выводить файл с колонками:

1. Слово.
2. Частота.
3. Частота (в %).

Файл должен быть упорядочен по убыванию частоты, то есть самые частые слова должны идти в начале.

Разделителями считать все символы кроме букв и цифр.

Методические указания:

- Использовать класс `java.lang.StringBuilder` для построения слов.
- Для чтения из файла удобно использовать: `java.io.InputStreamReader`

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

Вопросы дифференцированного зачета

1. Цикл разработки программных средств с использованием Java.
2. Минимальный комплект для разработки программ на Java.
3. Средства разработки и системные требования.
4. Основные технические возможности NetBeans.
5. Переменные, константы, область видимости Java.
6. Примитивные типы данных.
7. Ссылочные типы данных.
8. Логические операторы.
9. Арифметические операторы.
10. Преобразования типов. Оболочечные классы.
11. Класс Math.
12. Сложные типы данных. Массивы.
13. Сложные типы данных. Строки.
14. Управляющие конструкции. Оператор If, Switch.
15. Операторы цикла.
16. Операторы перехода.
17. Основные принципы ООП. Инкапсуляция.
18. Основные принципы ООП. Наследование.
19. Основные принципы ООП. Полиморфизм.
20. Понятие объекта и работа с ним.

21. Типы отношений между классами. Агрегация, ассоциация.
22. Достоинства и недостатки ООП.
23. Класс. Описание полей класса. New.
24. Описание методов класса. Модификаторы доступа.
25. Конструкторы.
26. Ссылка This. Перегрузка методов.
27. Наследование. Super. Замещение методов (override).
28. Динамическое назначение методов.
29. Ввод вывод в Java.
30. Элементы графического интерфейса.

Вопросы к экзамену:

1. Технология Java, ее возникновение, современное состояние и развитие.
2. Структура программы Java. Библиотеки стандартных классов. Основные пакеты . Пример простой программы.
Примитивные типы данных.
3. Массивы. Многомерные массивы.
4. Классы. Модификаторы доступа (инкапсуляция). Разграничение доступа в Java.
Объявление класса (заголовок + тело). Объявление полей, методов. Объявление конструкторов.
Создание
объекта. Инициализаторы.
5. Параметры методов. Преобразование и приведение типов. Присвоение значений.
Перегрузка методов.
6. Статические элементы. Ключевые слова this и super. Особенности их применения. Ключевое слово abstract.
7. Интерфейсы. Объявление, реализация, применение. Полиморфизм (суть, условия применения).
8. Пакет java.lang. Класс Object и его основные методы. Класс Class
9. Классы-оболочки: Integer и другие числовые классы. Их методы. Классы-оболочки Character, Boolean. Их
методы. Класс Math.
10. Класс String. Конструкторы, методы. Класс StringBuffer.
11. Исключения. Иерархия исключений в Java. Генерация исключений. Создание классов исключений.
12. Перехват исключений. Вложенные блоки try. Повторная генерация исключений. Блок finally.
Try с ресурсами
Как использовать исключения.
13. Классы Arrays и Collections и их методы. Интерфейсы Comparator и Comparable и их применение.
14. Интерфейс List и классы его реализующие.
15. Интерфейс Set и классы его реализующие.
16. Интерфейс Map и классы его реализующие.
17. Принципы построения графического интерфейса. Парадигма MVC. Контейнеры. Иерархия компонентов и
контейнеров. Компоненты. Атрибуты компонентов. Размер и позиция компонентов.
18. Структура фрейма. Создание простого фрейма. Двумерные фигуры. Иерархия фигур.
Определение цвета,
выбор шрифта. Вывод графических изображений.
19. Основные менеджеры компоновки.
20. Компоненты фрейма (метки, кнопки и пр.), их описание и использование.
21. Текстовые области и поля. Меню на Java.
22. Многопоточность. Состояние потока. Класс Thread и интерфейс Runnable. Создание потока.
Создание
нескольких потоков. Завершение потока.
23. Приоритеты потоков. Синхронизация. Взаимодействие потоков. Потоки-демоны.

24. Система ввода/вывода. Потоки данных. Байтовый ввод/вывод. Иерархия классов входных и выходных потоков.

Файловый ввод и вывод. Классы фильтров ввода и вывода. Сериализация.

25. Потоки ввода символов. Консольный ввод и вывод (байтовый и символьный). Файлы произвольного доступа.

Примеры экзаменационных билетов:

Билет 1:

1. Расскажите про термины JVM, JRE, JDK, JIT;
2. Суть использования стримов;

Билет 2:

1. Опишите внутреннее строение контейнеров стандартной библиотеки и временную сложность операций с ними;
2. Опишите принцип работы сборки мусора в Java;

Билет 3:

1. Что такое кодировки, использующиеся при хранении текстовых данных (ASCII, Windows-1250/1251, UTF-8, UTF-16);
2. Какие есть форматы хранения данных XML, JSON, CSV, средства стандартной библиотеки для работы с ними;

Билет 4:

1. Расскажите про принципы сетевого взаимодействия с использованием стека TCP/IP;
2. Каковы возможности протоколов HTTP/HTTPS.

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

Дифференцированный зачет и экзамен проводятся по итогам текущей успеваемости и сдачи заданий и других видов работ, предусмотренных программой дисциплины и (или) путем организации специального опроса, проводимого в устной и (или) письменной форме.