**МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ, СВЯЗИ И МАССОВЫХ КОММУНИКАЦИЙ РОССИЙСКОЙ ФЕДЕРАЦИИ**

**Ордена трудового Красного Знамени федеральное государственное бюджетное**

**образовательное учреждение высшего образования**

**«Московский технический университет связи и информатики»**

Кафедра Математическая кибернетика и информационные технологии

Отчет по \_1\_ лабораторной работе

«История языка программирования Java»

Выполнил: студент группы \_БПИ2401\_

Баранова Полина Николаевна

ФИО

Руководитель: \_Мосева Марина Сергеевна\_

Москва, 2025

**Цель работы:**

Разобраться в основах Java: как он устроен, как появился, как работает с типами данных и компиляцией. Написать, скомпилировать и запустить первые программы, чтобы понять на практике, как это работает.

**Основная часть:**

Разобрала историю и основы Java: методы, типы данных, циклы, компиляцию и запуск программ через командную строку.

Задание 1. Поиск простых чисел

Написала программу Primes.java, которая ищет простые числа от 2 до 100. Реализовала метод isPrime(int n) — он проверяет число на простоту перебором делителей до корня из n. В main() цикл перебирает числа и выводит те, что прошли проверку.

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Содержимое, созданное искусственным интеллектом, может быть неверным.

Задание 2. Проверка палиндромов

Сделала Palindrome.java, которая определяет, палиндром строка или нет.

- reverseString(String s) переворачивает строку

- isPalindrome(String s) сравнивает исходную и перевернутую

- программа берет аргументы из командной строки, но т.к. я работаю в IntelliJ IDEA, настроила конфигурацию запуска и ввела аргументы там

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Содержимое, созданное искусственным интеллектом, может быть неверным.

Обе программы скомпилировались успешно. Алгоритмы работают корректно, концепции Java применены верно.

**Ответы на контрольные вопросы:**

1. Java является компилируемым или интерпретируемым

языком?

И компилируемым и интерпретируемым. Исходный код компилируется в байт-код, который потом интерпретируется JVM с JIT-оптимизацией.

2. Что такое JVM и для чего предназначается?

JVM – виртуальная машина Java. Она выполняет байт-код, превращая его в машинные команды под конкретную платформу. Это основа кроссплатформенности.

3. Каков жизненный цикл программы на языке Java?

Пишем код, компилируем в байт-код (javac), JVM интерпретирует его с JIT, программа работает.

4. Какие виды типов данных есть в языке Java?

Два вида: примитивные (int, boolean, char и т.д.) и ссылочные (классы, интерфейсы, массивы).

5. Чем примитивные типы данных отличаются от ссылоч

ных?

Примитивные хранят значение прямо в стеке, ссылочные – содержат указатель на объект в куче. У примитивных нет методов, у ссылочных – есть.

6. Как происходит преобразование примитивных типов в

Java?

Преобразование может быть неявным (автоматически, когда типы совместимы) и явным – через приведение, с риском потери точности.

7. Что такое байт-код в Java, и почему он важен для плат

форменной независимости?

Байт-код – промежуточный код, который понимает JVM. Именно он позволяет Java быть платформенно независимой: один раз скомпилировал –работает везде, где есть JVM.

8. Какой тип данных используется для хранения символов в

Java? Как представляются символы в памяти?

В char, 16-битная кодировка Unicode. В памяти – как беззнаковое целое.

9. Что такое литералы в Java? Приведите примеры литера

лов для разных типов данных.

Литералы – это явные значения в коде. Например:

- целые: 100, 0xFF

- с плавающей точкой: 3.14f, 2.5e-4

- символ: 'A'

- строка: "Hello"

- булевы: true, false

10. Почему Java считается строго типизированным языком?

Потому что каждая переменная обязана иметь тип, который проверяется и при компиляции, и при выполнении. Нельзя работать с данными без явного указания их типа.

11. Какие проблемы могут возникнуть при использовании

неявного преобразования типов?

Потеря точности (например double в int), переполнение, неочевидное поведение в выражениях со смешанными типами. Всё это ведёт к ошибкам, которые компилятор не всегда поймает.

**Заключение:**

Разобралась в основах java и истории его появления, написала первые программы, увидела, как в данном языке работают циклы и условия, изучила основы синтаксиса.