**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 лабораторной работе № 1.

Выполнил: студент группы БПИ2401

Костиль Антон Валерьевич

Проверил: Харрасов Камиль Раисович

Москва, 2025

Цель работы: освоить базовый синтаксис языка программирования Java, закрепить работу с типами данных, управляющими конструкциями.

Ход работы:

**Задание 1**. Создайте программу, которая находит и выводит все простые числа меньше 100.

Шаг 1. Создаем класс, в котором будем описывать методы:

public class Task1 {

    public static void main(String[] args) {

    }

Шаг 2. Описываем метод IsPrime, который определяет, является ли аргумент простым числом

public static boolean isPrime(int n) {

        for (int i = 2; i\*i <= n; i++) {

            if (n % i == 0) {

                return false;

            }

        }

        return true;

    }

Шаг 3. Вызовем его в методе main. Полный код программы:

public class Task1 {

    public static void main(String[] args) {

        for (int i =2; i <= 100; i++) {

            if (isPrime(i)) {

                System.out.println(i);

            }

        }

    }

    public static boolean isPrime(int n) {

        for (int i = 2; i\*i <= n; i++) {

            if (n % i == 0) {

                return false;

            }

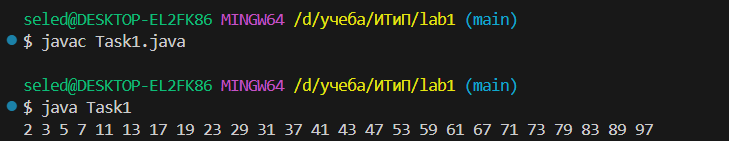
        }

        return true;

    }

}

Шаг 4. Компиляция и запуск



Вывод программы

**Задание 2**. Создайте программу, которая определяет, является ли введенная строка палиндромом.

Шаг 1. Задаем класс

public class Task2 {

    public static void main(String[] args) {

}

Шаг 2. Опишем метод, который будет переворачивать исходную строку.

public static String reverseSting(String s) {

        String rev = "";

        for (int i = s.length() - 1; i >= 0; i--) {

            rev += s.charAt(i);

        }

        return rev;

    }

Шаг 3. Опишем метод, который будет сравнивать исходную и перевернутую строки.

public static boolean isPalindrome(String s) {

        return s.equals(reverseSting(s));

    }

}

Шаг 4. Вызовем эти методы в main(). Полный код программы:

public class Task2 {

    public static void main(String[] args) {

        for (int i = 0; i < args.length; i++) {

            String s = args[i];

            if (isPalindrome(s)) {

                System.out.println(s + " - is a palindrome");

            } else {

                System.out.println(s + " - is not a palindrome");

            }

        }

    }

    public static String reverseSting(String s) {

        String rev = "";

        for (int i = s.length() - 1; i >= 0; i--) {

            rev += s.charAt(i);

        }

        return rev;

    }

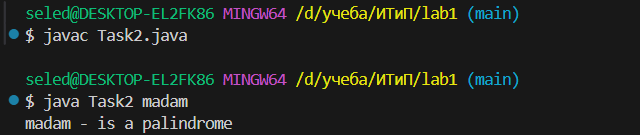
    public static boolean isPalindrome(String s) {

        return s.equals(reverseSting(s));

    }

}

Шаг 5. Компиляция и запуск



Вывод программы

**Контрольные вопросы:**

**1.Java является компилируемым или интерпретируемым языком?**

Java и компилируемый, и интерпретируемый: код (.java) компилируется в байт-код (.class). Байт-код исполняется JVM

**2. Что такое JVM и для чего предназначается?**

JVM (Java Virtual Machine) — виртуальная машина, которая выполняет байт-код Java, обеспечивает выполнение Java-программ на любой платформе (Windows, Linux, Mac), управляет памятью, интерпретирует байт-код

**3. Каков жизненный цикл программы на языке Java?**

1.Написание исходного кода в .java файле

2.Компиляция через javac

3.Запуск. JVM интерпретирует байт-код и выполняет программу

4.Работа программы

5.Завершение программы, сборка мусора

**4. Какие виды типов данных есть в языке Java?**

Примитивные типы (byte, short, int, long, float, double, char, Boolean)

Ссылочные типы – объекты (массивы, классы, String)

**5. Чем примитивные типы данных отличаются от ссылочных?**

Примитивные типы хранят значение напрямую, легкие, не могут быть null, а ссылочные типы хранят ссылку на объект, могут быть null

**6. Как происходит преобразование примитивных типов в Java?**

Неявное преобразование – из маленького типа в большой (например из int в long)  
Явное – из большого типа в маленький (например из double в int)

**7. Что такое байт-код в Java, и почему он важен для платформенной независимости?**

Байт-код — это промежуточный код, который создаётся компилятором Java. Это не нативный код, поэтому JVM на любой платформе может его выполнить.

**8. Какой тип данных используется для хранения символов в Java? Как представляются символы в памяти?**

Для хранения символа используется тип char, весит 2 байта. Символы в памяти представлены как код UTF-16

**9. Что такое литералы в Java? Приведите примеры литералов для разных типов данных.**

Литералы — это константные значения, которые пишутся в коде.

Примеры:

* Целые числа: 10, 0xFF
* Дробные числа: 3.14, 2.0f
* Символы: 'A', 'Ж'
* Строки: "Hello, world!"
* Булевы: true, false

**10. Почему Java считается строго типизированным языком?**

Каждая переменная имеет жёстко определённый тип. Компилятор проверяет типы на этапе компиляции. Строгость предотвращает ошибки во время выполнения.

**11. Какие проблемы могут возникнуть при использовании неявного преобразования типов?**

* Потеря точности или данных при явном преобразовании.
* Потенциальные непредсказуемые результаты при операциях с разными типами.

**Вывод:** в ходе выполнения работы были освоены базовые конструкции языка программирования Java и принципы построения простых алгоритмов. Работа размещена на GitHub - <https://github.com/Antosha044/ITiP>