

**МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ СВЯЗИ И МАССОВЫХ  
КОММУНИКАЦИЙ**

**Ордена Трудового Красного Знамени**

**Федеральное государственное бюджетное образовательное учреждение  
высшего образования**

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

**Кафедра «Математическая Кибернетика и Информационные технологии»**

Лабораторная работа №1

Знакомство с языком java

Выполнил: Студент группы

БВТ2303

Кунецкий Владислав

Москва

2024

Цели работы:

- написать программу, которая выводит простые числа
- написать программу, которая определяет является ли число палиндромом

Ход работы:

```
package lab1;
import java.util.Scanner;

public class Primes {
    Run | Debug
    public static void main(String args[]) {
        Scanner input = new Scanner(System.in);
        System.out.print(s:"Введите число: ");
        int number = input.nextInt();

        for (int i = 0; i < number; i++) {
            if (isPrime(i))
                System.out.println(i);
        }

        input.close();
    }

    public static boolean isPrime(int n) {
        if (n <= 1)
            return false;
        if (n <= 3)
            return true;
        if (n % 2 == 0 || n % 3 == 0)
            return false;

        int sqrtN = (int) Math.sqrt(n);
        for (int i = 5; i <= sqrtN; i += 6) {
            if (n % i == 0 || n % (i + 2) == 0)
                return false;
        }
        return true;
    }
}
```

Скрин 1 – Программа, выводящая простые числа

Сначала я определил главный класс Primes, а внутри него стандартную точку входа в программу - публичный статический метода main(). Далее описал в методе isPrime самый эффективный алгоритм определения простоты числа. В главном методе создал экземпляр класса Scanner и для считывания числа вызвал его метод nextInt(). Прошелся циклом по всем числам до введенного, передавая их в наш ранее определенный метод. Если метод возвращает true, то мы печатаем число, в ином случае ничего не происходит.

В конце метода закрываем поток ввода с помощью метода close(). Как я понял, сборщик мусора не закрывает потоки сам, поэтому это необходимо

```
package lab1;
public class Palindrome {
    Run | Debug
    public static void main(String args[]) {
        for (int i = 0; i < args.length; i++) {
            String s = args[i];
            if (isPalindrome(s)){
                System.out.printf(format: "%s является палиндромом\n", s);    "явд
            } else System.out.printf(format: "%s не является палиндромом\n", s);
        }
    }

    public static String reverseString(String str) {
        StringBuilder ret_string = new StringBuilder();
        for (int i = str.length() - 1; i >= 0; i--)
            ret_string.append(str.charAt(i));
        return ret_string.toString();
    }

    public static boolean isPalindrome(String str) {
        String reverse_str = reverseString(str);
        if (reverse_str.equals(str))
            return true;
        return false;
    }
}
```

Скрин 2 – Программа, определяющая палиндром

Во-втором задании я определил 2 метода( помимо main()) reverseString() и isPalindrom(). Первый метод переворачивает строку, как можно догадаться по названию. Так как строки в java являются иммутабельными, мне пришлось использовать класс StringBuilder, который оказался достаточно удобным решением. Экземпляр StringBuilder создает массив char, который можно изменять. Я воспользовался методом append() и циклом, для того чтобы заполнить массив char из исходной строки а потом вернул строку, применив метод toString().

Во втором методе я использовал метод reverseString для создания перевернутой копии подаваемой в функцию строки и с помощью метода equals() определял, являются ли строка и ее перевертыш одинаковыми.