

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»  
(МТУСИ)

Факультет  
Информационные технологии  
Искусственный интеллект и машинное обучение

по теме:  
Основы алгоритмизации и программирования в Java: работа с числами и  
строками

Студент:  
*БВТ 2401*

*А.И. Меланич*

Предподаватель:

Москва 2025

## СОДЕРЖАНИЕ

ЦЕЛЬ ЛАБОРАТОРНОЙ РАБОТЫ.....	3
ЗАДАНИЕ 1 .....	4
ЗАДАНИЕ 2 .....	7
ВЫВОД.....	9

## ЦЕЛЬ ЛАБОРАТОРНОЙ РАБОТЫ

Основная цель данной лабораторной работы заключается в практическом ознакомлении с базовыми возможностями языка программирования Java и формировании первичных навыков разработки простых приложений.

Конкретные задачи, которые необходимо решить в ходе выполнения работы:

- Изучить основы синтаксиса языка Java
- Освоить принципы работы с основными типами данных
- Получить практические навыки создания и компиляции Java-программ
- Научиться работать с переменными и операторами
- Познакомиться с базовыми конструкциями управления потоком выполнения программы
- Развить понимание принципов объектно-ориентированного программирования на начальном уровне
- Сформировать навыки отладки и тестирования простых Java-приложений

Данная лабораторная работа направлена на создание прочной базы для дальнейшего изучения более сложных концепций и технологий Java-разработки.

## ЗАДАНИЕ 1

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

Так как обычный перебор в цикле является слишком прямым подходом к проблеме, предлагается расширить возможности алгоритма применив опыт из следующего видео: [ссылка на видео](#)

```
import java.math.BigInteger;
import java.util.InputMismatchException;
import java.util.Random;
import java.util.Scanner;

public class Primes {

    public static void main(String[] args) {
        Long number = 1L;
        Scanner scanner = new Scanner(System.in);

        for (int i = 1; i <= 100; i++) {
            boolean result = isPrime(i, 10);

            if (result) {
                System.out.print(i + " ");
            }
        }
        System.out.println("");

        while (number > 0L) {
            System.out.print("Введите число: ");

            try {
                number = scanner.nextLong();

                if (number > 0) {
                    boolean result = isPrime(number, 10);

                    if (result) {
                        System.out.println(number + " - оно простое\n");
                    } else {
                        System.out.println(number + " - оно НЕ простое\n");
                    }
                }
            } catch (InputMismatchException e) {
                System.out.println("Ошибка! Введите корректное целое число.\n");
                scanner.nextLine();
            }
        }
        scanner.close();
    }

    /**
     * Проверяет, является ли число простым с помощью теста Миллера-Рабина
     */
}
```

```

*
* @param number проверяемое число
* @param iterations количество итераций (точность теста)
* @return true, если число вероятно простое, false если составное
*/
public static boolean isPrime(long number, int iterations) {
    if (number <= 2) {
        return number == 2;
    }
    if (number == 3) {
        return true;
    }

    long s = 0;
    long d = number - 1;
    while ((d & 1) == 0) {
        d >>= 1;
        s++;
    }

    Random generator = new Random();

    for (int i = 0; i < iterations; i++) {
        long a = 2 + (long) (generator.nextDouble() * (number - 3));

        BigInteger bigA = BigInteger.valueOf(a);
        BigInteger bigN = BigInteger.valueOf(number);
        BigInteger x = bigA.modPow(BigInteger.valueOf(d), bigN);

        if (x.equals(BigInteger.ONE) || x.equals(bigN.subtract(BigInteger.ONE))) {
            continue;
        }

        boolean composite = true;
        for (long r = 1; r < s; r++) {
            x = x.modPow(BigInteger.valueOf(2), bigN);
            if (x.equals(BigInteger.ONE)) {
                return false;
            }
            if (x.equals(bigN.subtract(BigInteger.ONE))) {
                composite = false;
                break;
            }
        }
        if (composite) {
            return false;
        }
    }
    return true;
}
}

```

Вывод программы:

```

// javac Primes.java && java Primes
// 2 3 5 7 11 13 17 19 23 29 31 37 41 43 47 53 59 61 67 71 73 79 83 89 97
// Введите число: 12987

```

```
// 12987 - оно НЕ простое
//
// Введите число: 7
// 7 - оно простое
//
// Введите число:
```

## ЗАДАНИЕ 2

Создайте программу, которая определяет, является ли введенная строка палиндромом.

```
import java.util.Scanner;

public class Palindrome {

    public static void main(String[] args) {
        String str = "not empty";
        Scanner scanner = new Scanner(System.in);

        for (int i = 0; i < args.length; i++) {
            try {
                checkString(args[i]);
            } catch (Exception e) {
                System.out.println(e);
            }
        }

        while (!str.equals("0")) {
            System.out.print("Введите строку: ");

            try {
                str = scanner.next();
                checkString(str);
            } catch (Exception e) {
                System.out.println("Ошибка!\n" + e);
                scanner.nextLine();
            }
        }

        scanner.close();
    }

    /**
     * Делает все необходимые проверки и выводит результат
     * в консоль
     *
     * @param str строка для проверки
     */
    public static void checkString(String str) {
        if (str.equals("0"))
            return;

        boolean isPalindrome = isPalindrome(str);

        if (isPalindrome) {
            System.out.println("Палиндром: " + str);
        } else {
            System.out.print("Не палиндром: " + str + " ");
            System.out.println(reverseString(str));
        }
    }
}
```

```

}

/**
 * Переворачивает строку
 *
 * @param str входная строка
 * @return возвращает перевернутую строку
 */
public static String reverseString(String str) {
    int lng = str.length();
    StringBuilder sb = new StringBuilder(str);
    sb.setLength(lng);
    for (int i = lng - 1; i >= 0; i--) {
        sb.setCharAt(lng - i - 1, str.charAt(i));
    }

    return sb.toString();
}

/**
 * Проверяет является ли строка Палиндромом
 *
 * @param str входная строка
 * @return true если строка Палиндром, false если нет
 */
public static boolean isPalindrome(String str) {
    int lng = str.length();
    for (int i = 0; i < lng / 2; i++) {
        if (str.charAt(i) != str.charAt(lng - i - 1))
            return false;
    }

    return true;
}
}

```

Вывод программы:

```

// javac Primes.java && java Palindrome 123 asdfdsa asddsa
// Не палиндром: 123 321
// Палиндром: asdfdsa
// Палиндром: asddsa
// Введите строку: fzf
// Палиндром: fzf
// Введите строку:

```



## ВЫВОД

В ходе выполнения лабораторной работы были успешно решены две задачи по программированию на языке Java, направленные на закрепление навыков работы с базовыми конструкциями языка и алгоритмами.

Первая задача была посвящена поиску простых чисел. В процессе решения:

- Создан класс Primes с методом isPrime для проверки простоты числа
- Реализован эффективный алгоритм через тест Миллера-Рабина
- Организован вывод простых чисел
- Отработаны навыки работы с циклами и условными операторами

Вторая задача касалась работы со строками и проверки палиндромов. В ходе её выполнения:

- Разработан метод reverseString для переворота строки
- Создан метод isPalindrome для проверки палиндрома
- Отработаны навыки работы с методами length() и charAt()
- Получен опыт обработки аргументов командной строки

В результате выполнения работы достигнуты следующие цели:

- Закреплены знания по работе с базовыми типами данных в Java
- Получены практические навыки создания и использования методов
- Отработаны алгоритмы проверки чисел и строк
- Приобретен опыт разработки консольных приложений

Обе программы успешно скомпилированы и протестированы, что подтверждает корректность реализованных алгоритмов и правильность их реализации. Полученные навыки являются фундаментальными для дальнейшего изучения программирования и разработки более сложных приложений.

[Ссылка на git](#)