

МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ, СВЯЗИ И МАССОВЫХ
КОММУНИКАЦИЙ РОССИЙСКОЙ ФЕДЕРАЦИИ
Ордена Трудового Красного Знамени федеральное государственное
бюджетное учреждение высшего образования
«Московский технический университет связи и информатики»

Лабораторная работа №1
по дисциплине
«Информационные технологии и программирование»

Выполнил: студент гр. БВТ2403
Толибов Р.Ш.

Проверил:

Москва, 2025 г.

Цель работы: познакомиться с языком программирования Java, освоить базовый синтаксис.

Ход работы

Задание 1.

Суть задания заключалась в создании программы, находящей простые числа меньше 100.

Для этого я создал файл Primes.java и описал в нём класс

```
public class Prime {  
  
    public static void main(String[] args){  
  
    }  
}
```

Важно: название класса обязательно должно совпадать с названием файла, иначе программа не скомпилируется

Далее я описал метод isPrime(). Он будет определять, является ли число простым или нет. Для его реализации я использовал цикл for с диапазоном от 2 до корня из n, где n – число, которое будет входить в метод isPrime() из метода main().

Затем я описал логику метода: если входящее число n делится без остатка на числа i из диапазона, то это число не простое и метод возвращает false.

```
public static boolean isPrime(int n){  
    for (int i = 2; i <= Math.sqrt(n); i++){  
        if (n % i == 0){  
            return false;  
        }  
    }  
    return true;  
}
```

Чтобы программа начала работать, осталось добавить в метод main() перебирание чисел от 2 до 100. Это также реализуется с помощью цикла for.

Наконец, финальным штрихом будет связь этих двух методов через условный оператор. Если число из цикла в методе `main()` вернет `true`, пройдя цикл из метода `isPrime()`, то программа выведет это число.

Итоговая программа имеет вид:

```
public class Prime {

    public static void main(String[] args){
        for (int i = 2; i <= 100; i++){
            if (isPrime(i)){
                System.out.print(i + " ");
            }
        }
    }

    public static boolean isPrime(int n){
        for (int i = 2; i <= Math.sqrt(n); i++){
            if (n % i == 0){
                return false;
            }
        }
        return true;
    }
}
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\lavah\OneDrive\Рабочий стол\ИТИП\лаб1> java Prime.java
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
PS C:\Users\lavah\OneDrive\Рабочий стол\ИТИП\лаб1> 
```

Задание 2.

Аналогично предыдущему заданию, создал файл, описал класс и метод main()

```
public class Palindrome {  
  
    public static void main(String[] args){  
        for (int i = 0; i < args.length; i++) {  
            String s = args[i];  
        }  
    }  
}
```

Затем я начал описывать метод reverseString(). Его суть заключалась в том, что входящие в него слова «переворачивались». Для его реализации я ввёл пустую переменную reversed и запустил обратный цикл в диапазоне от длины входящего слова, до 0. Этот цикл перебирал символы входящего слова, начиная с конца. Для этого я использовал s.charAt(i).

```
public static String reverseString(String s){  
    String reversed = "";  
    for (int i = s.length() - 1; i >= 0; i--) {  
        reversed = reversed + s.charAt(i);  
    }  
    return reversed;  
}
```

Далее я начал работу над методом isPalindrome(), который будет определять, является ли слово палиндромом. Для этого необходимо, чтобы метод сравнивал 2 переменные: слово и его «перевернутую» версию.

```
public static boolean isPalindrome(String s) {  
    String reversed = reverseString(s);  
    return s.equals(reversed);  
}
```

Для полноты картины необходимо было дополнить метод main(), чтобы тот выводил слова, подписывая, являются ли те палиндромами или нет. Это было довольно просто реализовать через условный оператор.

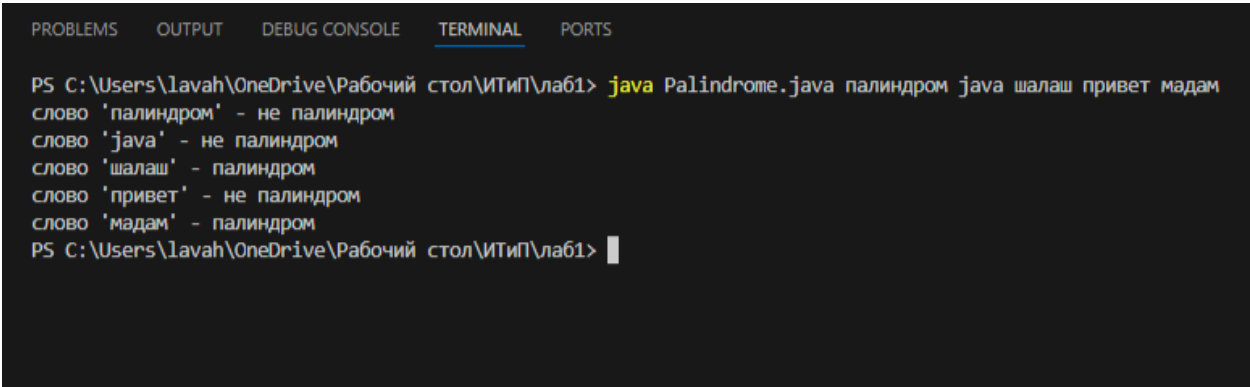
В итоге программа имеет вид:

```
public class Palindrome {

    public static void main(String[] args){
        for (int i = 0; i < args.length; i++){
            String s = args[i];
            if (isPalindrome(s)){
                System.out.println("слово '" + s + "' - палиндром");
            } else {
                System.out.println("слово '" + s + "' - не палиндром");
            }
        }
    }

    public static String reverseString(String s){
        String reverse = "";
        for (int i = s.length() - 1 ; i >= 0; i--){
            reverse = reverse + s.charAt(i);
        }
        return reverse;
    }

    public static boolean isPalindrome(String s){
        String reverse = reverseString(s);
        return s.equals(reverse);
    }
}
```



The screenshot shows a terminal window with tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL, and PORTS. The TERMINAL tab is active, displaying the command `java Palindrome.java палиндром java шалаш привет мадам` and its output. The output consists of five lines, each identifying a word and whether it is a palindrome: 'палиндром' is not, 'java' is not, 'шалаш' is, 'привет' is not, and 'мадам' is. The terminal prompt is `PS C:\Users\lavah\OneDrive\Рабочий стол\ИТиП\лаб1>`.

```
PS C:\Users\lavah\OneDrive\Рабочий стол\ИТиП\лаб1> java Palindrome.java палиндром java шалаш привет мадам
слово 'палиндром' - не палиндром
слово 'java' - не палиндром
слово 'шалаш' - палиндром
слово 'привет' - не палиндром
слово 'мадам' - палиндром
PS C:\Users\lavah\OneDrive\Рабочий стол\ИТиП\лаб1>
```

Вывод: в ходе выполнения работы, я освоил базовый синтаксис языка программирования Java, получил важные теоретические и практические знания, которые пригодятся мне в дальнейших работах.

<https://github.com/TRuslan666/ITiP>