

**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»**

**Отчет по лабораторной работе №1**

**по дисциплине «Информационные технологии и программирование»**

Выполнил: студент группы БПИ2401

Юлдашев Всеволод

Проверил: Харрасов Камиль  
Раисович

Москва

2025

1. **Цель работы:** Освоение принципов объектно-ориентированного программирования на Java через создание методов и классов. Получение практических навыков в работе с циклами, условными операторами, строками и методами в языке Java.

2. **Ход работы:**

**Hello World!**

```
public class JavaHelloWorldProgram {  
    Run | Debug  
    public static void main(String args[]){  
        System.out.println("Hello World");  
    }  
}
```

```
> java JavaHelloWorldProgram  
Hello World
```

**Задание 1:**

```
public class Primes {  
    Run | Debug  
    public static void main(String args[]){  
    }  
}
```

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

```

public class Primes {
    Run|Debug
    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<= Math.sqrt(n); i++) {
            if (n%i == 0) {
                return false;
            }
        }
        return true;
    }
}

```

```
> 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

```

**Задание 2:**

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

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

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

```

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.println(s + " ПАЛИНДРОМ");
            }
            else {
                System.out.println(s + " НЕ ПАЛИНДРОМ");
            }
        }
    }

    public static String reverseString(String s) {

        String reversed = "";

        for (int i = s.length()-1; i >= 0; i--) {
            reversed += s.charAt(i);
        }

        return reversed;
    }

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

```

```

> java Palindrome madam racecar apple kayak song noon
madam ПАЛИНДРОМ
racecar ПАЛИНДРОМ
apple НЕ ПАЛИНДРОМ
kayak ПАЛИНДРОМ
song НЕ ПАЛИНДРОМ
noon ПАЛИНДРОМ

```

### Контрольные вопросы:

1. Исходный код компилируется, а потом интерпретируется в JVM.
2. JVM виртуальная машина, которая выполняет байт-код Java.

3. Написание кода - компиляция в байт-код (.class) - загрузка в JVM - выполнение JVM - завершение работы.
4. Примитивные (int, char, boolean и др.) и ссылочные (классы, массивы, интерфейсы).
5. Примитивные типы данных хранят данные непосредственно, а ссылочные содержат ссылку на объект в памяти.
6. Расширение (автоматически) — byte - int; сужение (требуется явное приведения) — int - byte.
7. Байт-код — промежуточный код, который выполняется JVM. Позволяет запускать программу на любой ОС с установленной JVM.
8. Тип char. Символы хранятся в памяти как 16-битные беззнаковые числа.
9. Литералы — явные значения в коде. Примеры: 123 (int), 3.14 (double), 'A' (char), "hello" (String), true (boolean).
10. Потому что все переменные имеют определенный тип, который проверяется на этапе компиляции, и неявные преобразования ограничены.
11. Потеря данных (при сужении), переполнение, неожиданные результаты вычислений, ошибки округления.

3.     **Вывод:** В ходе лабораторной работы освоены основы программирования на Java: создание классов, методов, работа с циклами и строками. Разработаны программы для поиска простых чисел и проверки палиндромов. Приобретены практические навыки отладки кода и тестирования программ.

<https://github.com/zevy3/java-labs/tree/main/java-1>