

**МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ СВЯЗИ И МАССОВЫХ
КОММУНИКАЦИЙ**

Ордена Трудового Красного Знамени

**Федеральное государственное бюджетное образовательное учреждение
высшего образования**

«Московский технический университет связи и информатики»

Информационные технологии и программирование

Лабораторная работа №1

Выполнил: Студент группы

БПИ-2301

Булацев Давид

Москва 2024

Задания для выполнения лабораторной работы:

Задание 1

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

После установки JDM и IntelliJ IDEA и написания простейшей программы, которая выводит в консоль “Hello, World”, усвоив теорию и базовый синтаксис языка java: Переменные, типы данных, ввод\вывод, циклы for\while, условия и ветвления if\else, функции возвращающие конкретный тип данных (int, String, ...) и невозвратные функции (void), поняв как отличаются public от private, мы можем приступить к выполнению задания 1.

Нас просят найти все простые числа в диапазоне (0, 100)

Для начала я создал папку src/Labs, в этой папке еще две папки:

lab1 и tasks – для лаб. 1 и десяти задач соответственно.

В папке lab1 создадим Java Class – Primes. Обязательно укажем в какой папке мы находимся используя

```
package Labs.lab1;
```

Затем объявим класс с таким же названием, как и название файла:

```
public class Primes
```

Следом объявим main метод, в этой IDEA можно прописать psvm:

```
public static void main(String[] args)
```

Теперь можно решать задачу, благо с алгоритмом поиска простых чисел я был знаком и справился с этой задачей

```

1 package ForMTUCI.lab1;
2
3
4 public class Primes {
5
6
7     public static void main(String[] args) {
8
9         for (int i = 2; i < 100; i++) {
10             if (isPrime(i))
11                 System.out.print(i + " ");
12         }
13     }
14     public static boolean isPrime(int n) { 1 usage
15         for (int i = 2; i < n; i++) {
16             if ((n % i) == 0){
17                 return false;
18             }
19         }
20         return true;
21     }
22 }
23

```

Я создал доп. Функцию, которая определяет есть ли у числа n, делители кроме 1 и его самого, если есть – число составное => true, нет – число простое => true. Затем уже в функции main проходимся циклом от 2 до 100 и вызываем для каждого числа функцию isPrime.

Консоль после запуска выглядит следующим образом:

```

"C:\Program Files\Java\jdk-22\bin\java.exe" "-javaagent:C:\Program File
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
Process finished with exit code 0

```

Задание 2

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

Создадим java class Palindrome, в методе main напишем цикл, который обрабатывает входные данные с командной строки:

```
1 package Labs.lab1;
2
3 public class Palindrome {
4     public static void main(String[] args) {
5         for (String arg : args) {
6             if (isPalindrome(reverseString(arg), arg)) {
7                 System.out.println("The word " + arg + " is a palindrome");
8             }
9             else {
10                System.out.println("The word " + arg + " is NOT a palindrome");
11            }
12        }
13    }
```

Функции isPalindrome и reverseString помогают определить является ли введенная строка палиндромом. reverseString принимает на вход строку и проходит по ней циклом в обратном порядке, записывая в новую переменную реверсированное слово, затем проверяет совпадает ли эта строка со входной.

```
15     public static String reverseString(String s) { 1 usage
16         StringBuilder word = new StringBuilder();
17         for (int i = s.length(); --i >= 0;) {
18             word.append(s.charAt(i));
19         }
20
21         return word.toString();
22     }
```

Здесь я решил использовать объект класса StringBuilder, потому что использовать обычный тип данных String неудобно в некоторых случаях, тем более если к String просто добавить какой-то символ {+=char} IDE подсвечивает, что это неэффективно и предлагает как раз использовать StringBuilder.

Ну и функция isPalindrome принимает на вход начальную строку и перевернутую, которую возвращает ф-я reverseString

```
public static boolean isPalindrome(String str, String newStr) { 1 usage
    return str.equals(newStr);
}
}
```

Ну и в целом код выглядит так:

```
1 package Labs.lab1;
2
3 public class Palindrome {
4     @ public static void main(String[] args) {
5         for (String arg : args) {
6             if (isPalindrome(reverseString(arg), arg)) {
7                 System.out.println("The word " + arg + " is a palindrome");
8             }
9             else {
10                System.out.println("The word " + arg + " is NOT a palindrome");
11            }
12        }
13    }
14
15    @ public static String reverseString(String s) { 1 usage
16        StringBuilder word = new StringBuilder();
17        for (int i = s.length(); --i >= 0;) {
18            word.append(s.charAt(i));
19        }
20
21        return word.toString();
22    }
23
24    @ public static boolean isPalindrome(String str, String newStr) { 1 usage
25        return str.equals(newStr);
26    }
27 }
```

Скомпилировав программу и запустив с некоторыми входными данными получаем:

```
PS C:\Users\Давид\JavaProjects\IT_AND_PROGRAMMING\src\Labs\lab1> java Palindrome.java madam song nginx shalash
The word madam is a palindrome
The word song is NOT a palindrome
The word nginx is NOT a palindrome
The word shalash is NOT a palindrome
PS C:\Users\Давид\JavaProjects\IT_AND_PROGRAMMING\src\Labs\lab1>
```

Как видим слово madam – палиндром.

Ссылка на гитхаб с лабораторными и тасками https://github.com/David-Bulatsev/IT_AND_PROGRAMMING.git

ВЫВОД: изучил основы java, научился решать простейшие задачи, компилировать и запускать программу.