МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
"МИРЭА - Российский технологический университет"

РТУ МИРЭА

Институт информационных технологий (ИТ)
Кафедра инструментального и прикладного программного обеспечения

ОТЧЁТ
ПО ЛАБОРАТОРНОЙ РАБОТЕ №1
по дисциплине
«Программирование на языке Java»

Тема: Циклы, условия, переменные и массивы в Java

Выполнил студент группы ИКБО-16-20 Пак С.А.

Принял ассистент кафедры ИиППО Русляков А.А.

Практические работы выполнены «_____» 2021г.

«Зачтено» «_____» 2021г

Москва 2021

СОДЕРЖАНИЕ

ТЕОРЕТИЧЕСКОЕ ВВЕДЕНИЕ

1. Переменные

Чтобы объявить переменную, необходимо указать тип переменной и ее имя. Типы переменной могут быть разные: целочисленный(long, int, short, byte), число с плавающей запятой(double, float), логический(boolean), перечисление, объектный(Object). Переменным можно присваивать различные значения с помощью оператора присваивания "=".

Класс String - особый класс в Java, так как ему можно присваивать значение, не создавая экземпляра класса(Java это сделает автоматически). Этот класс предназначен для представления строк. Строковое значение записывается буквами внутри двойных кавычек.

2. Массивы

Для того чтобы создать массив переменных, необходимо указать квадратные скобки при объявлении переменной массива. После чего необходимо создать массив с помощью оператора new. Необходимо указать в квадратных скобках справа размер массива. Например:

```
int[] b = new int[10]
```

3. Условия

Условие - это конструкция, позволяющая выполнять то или другое действие, в зависимости от логического значения, указанного в условии. Синтаксис создания условия следующий:

```
if(a==b) {
   //Если a равно b, то будут выполняться операторы в этой области
}
else {
   //Если a не равно b, то будут выполняться операторы в этой области
}
```

4. Циклы

Цикл - это конструкция, позволяющая выполнять определенную часть кода несколько раз. В Java есть три типа циклов for, while, do while. Цикл for - это цикл со счетчиком, обычно используется, когда известно, сколько раз должна выполниться определенная часть кода. Синтаксис цикла for:

```
for (int i = 0; i < 10; ++i) {
   // Действия в цикле
}
```

Цикл while - это такой цикл, который будет выполняться, пока логическое выражение, указанное в скобках истинно. Синтаксис цикла while:

```
while (logic) {
   // Тело цикла
}
```

Цикл do while - это такой цикл, тело которого выполнится хотя бы один раз. Тело выполнится более одного раза, если условие, указанное в скобках истинно.

```
do {
   // Тело цикла
while (logic);
```

5. Потоки ввода/вывода и строки в Java, класс String

Для ввода данных используется класс Scanner из библиотеки пакетов Этот класс надо импортировать в той программе, где он будет использоваться. Это делается до начала открытого класса в коде программы. В классе есть методы для чтения очередного символа заданного типа со стандартного потока ввода, а также для проверки существования такого символа. Для работы с потоком ввода необходимо создать объект класса Scanner, при создании указав, с каким потоком ввода он будет связан. Стандартный поток ввода (клавиатура) в Java представлен объектом — System.in. А стандартный поток вывода (дисплей) — уже знакомым объектом System.out. Есть ещё стандартный поток для вывода ошибок — System.err.

В классе String существует масса полезных методов, которые можно применять к строкам:

- int length() - возвращает длину строки;
- boolean isEmpty() - проверяет, пустая ли строка;
- String replace(a, b) – возвращает строку, где символ a заменён на символ b;
- String toLowerCase() - возвращает строку, где все символы исходной строки преобразованы к строчным;
- String toUpperCase() - возвращает строку, где все символы строки преобразованы к прописным;
- boolean equals(s) – проверяет, совпадает ли исходная строка с s;
- int indexOf(ch) – возвращает индекс символа ch в строке или -1;

6. Методы

Методы позволяют выполнять блок кода, из любого другого места, где это доступно. Методы определяются внутри классов. Методы могут 13 быть статическими(можно выполнять без создания экземпляра класса), не статическими (не могут выполняться без создания экземпляра класса). Методы могут быть открытыми(public), закрытыми(private). Закрытые методы могут

вызываться только внутри того класса, в котором они определены. Открытые методы можно вызывать для объекта внутри других классов.

РЕШЕНИЕ ЗАДАЧИ

1. Постановка задачи

Вариант: 1.

Задание: вывести на экран сумму чисел массива с помощью циклов for, while, do-while.

2. Программный код

```
package ru.mirea;

import java.util.Random;
import java.util.Scanner;


public class App {
  private static final Random ENG = new Random();
  private static final Scanner IN = new Scanner(System.in);

  public static void main(String[] args) {
    System.out.print("Введите количество элементов: ");
    int length = IN.nextInt();

    System.out.print("Введите наименьший элемент массива: ");
    int min = IN.nextInt();

    System.out.print("Введите наибольший элемент массива: ");
    int max = IN.nextInt();

    int[] array = new int[length];

    fillArray(array, min, max);

    System.out.println();
    System.out.println("Массив: " + arrayToString(array));

    System.out.println();
      System.out.println("Сумма с использованием цикла for: " +
sumFor(array));
     System.out.println("Сумма с использованием цикла while: " +
sumWhile(array));
    System.out.println("Сумма с использованием цикла do-while: " +
sumDoWhile(array));
  }

  /**
```

```
   * Суммирует элементы массива с помощью цикла for
     * @param array            массив, элементы которого будут
суммироваться
   * @return            сумма элементов массива
   */
  private static int sumFor(int[] array) {
    int sum = 0;

    for (int i = 0; i < array.length; ++i) {
      sum += array[i];
    }

    return sum;
  }

  /**
   * Суммирует элементы массива с помощью цикла while
     * @param array            массив, элементы которого будут
суммироваться
   * @return            сумма элементов массива
   */
  private static int sumWhile(int[] array) {
    int i = 0;
    int sum = 0;

    while (i < array.length) {
      sum += array[i++];
    }

    return sum;
  }

  /**
   * Суммирует элементы массива с помощью цикла do-while
     * @param array            массив, элементы которого будут
суммироваться
   * @return            сумма элементов массива
   */
   private static int sumDoWhile(int[] array) {
     int i = 0;
     int sum = 0;

     do {
       sum += array[i++];
     }
     while (i < array.length);
```

```
    return sum;
   }

  /**
   * Возвращает строковое представление массива из целых чисел
   * @param array         массив из целых чисел
   * @return              строка вида [эл1, эл2, ..., элN]
   */
  private static String arrayToString(int[] array) {
    StringBuilder strArray = new StringBuilder("[");

    strArray.append(array[0]);
    for (int i = 1; i < array.length; ++i) {
      strArray.append(", " + array[i]);
    }
    strArray.append("]");

    return strArray.toString();
  }

  /**
     * Наполняет массив случайными числами в заданном диапазоне
чисел
   * @param array         заполняемый массив
   * @param min           наименьшее генерируемое значение
   * @param max           наибольшее генерируемое значение
   */
  private static void fillArray(int[] array, int min, int max) {
    for (int i = 0; i < array.length; ++i) {
      array[i] = ENG.nextInt(max - min) + min;
    }
  }
}
```

3. Вывод программы

Пользователь вводит 3 числа: количество элементов в массиве, минимальное возможное значение элемента массива, максимальное возможное значение элемента массива (рис.1, 2).

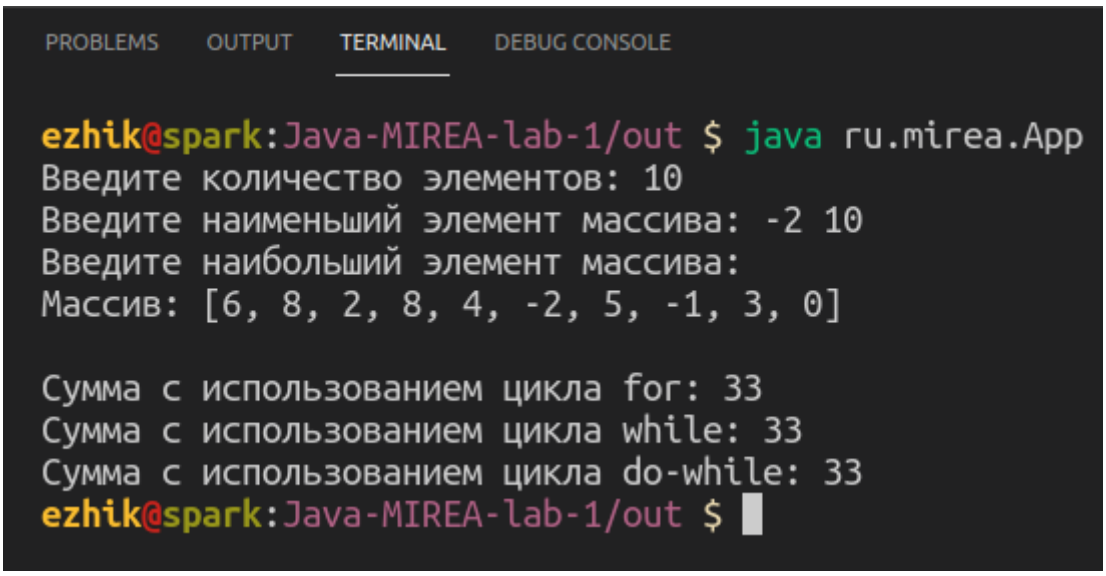

Рис.1 Тест для массива на 10 элементов

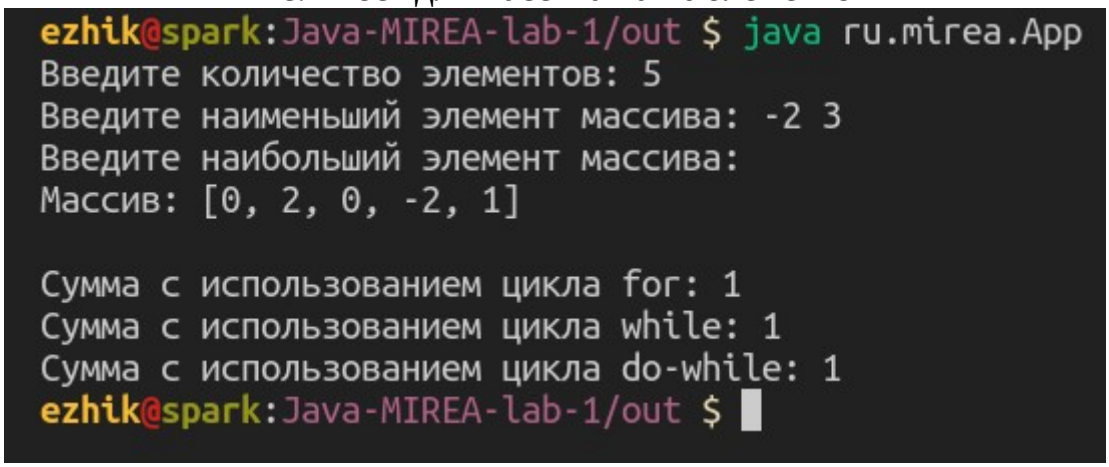

Рис.2 Тест для массива на 5 элементов

ВЫВОД

В результате выполнения работы получил практические навыки разработки программ, изучил синтаксис языка Java, освоил основные конструкции языка Java (циклы, условия, создание переменных и массивов, создание методов, вызов методов), а также научился осуществлять стандартный ввод/вывод данных.