МИНИСТЕРСТВО СВЯЗИ И ИНФОРМАТИЗАЦИИ РЕСПУБЛИКИ БЕЛАРУСЬ

УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ «БЕЛОРУССКАЯ ГОСУДАРСТВЕННАЯ АКАДЕМИЯ СВЯЗИ»

ОБОСОБЛЕННОЕ ПОДРАЗДЕЛЕНИЕ

«КОЛЛЕДЖ ИНФОРМАЦИОННО-КОММУНИКАЦИОННЫХ ТЕХНОЛОГИЙ»

Отчет по лабораторной работе № 4

“ Работа с массивами в Java”

Выполнила учащаяся гр. ТО-011

Ясюкевич Полина

Проверили

Янович Н.И.

Рогалевич П.И.

Минск, 2022

**Цель:** сформировать знания об описании и инициализации, порядке

размещения в памяти одно- и двумерных массивов, о доступе к элементам

массива, и их обработке в языке Java; сформировать навыки решения задач с

использованием массивов на языке Java.

**Практические задания и методические указания**

1. Изучить теоретические сведения по теме «Одномерные массивы»,

выполнить приведенные там примеры.

2. Выполнить индивидуальное задание № 1.

3. \*Модифицировать программный код п.2 индивидуального задания:

добавить возможность ввода данных из текстового файла

4. Изучить теоретические сведения по теме «Многомерные массивы»,

выполнить приведенные там примеры.

5. Создать отчет, включив в него описание работы по пункту 2 – 4 ответы на

контрольные вопросы.

**Задание 2.1**

Дан одномерный массив, который содержит не более 50 целых чисел. Найти среднее арифметическое элементов, которые стоят на нечетных местах.

public class Zadanie1 {

public static void main (String [] args){

// Scanner in=new Scanner(System.in);

int sum=0;

double k=0;

int c=0;

int [] mas = {1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20};

int count = mas.length;

for (int i:mas) {

if (i % 2 != 0) {

sum += mas[i];

c++;

System.out.println(i);

} }

k=sum/(c+1);

System.out.println("Среднее арифметическое элементов," +

" стоящих на нечетных позициях = "+k);}}

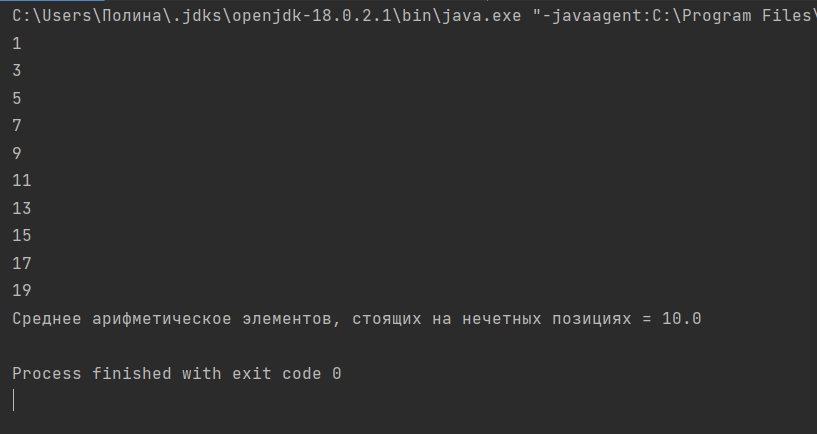


Рисунок 1 – Результат выполнения кода задания 2.1

**Задание 2.2**

Дано длинное целое число, содержащее не более 100 цифр. Рассматривать цифры числа, как массив символов. Определить, есть ли в данном числе четные цифры и сколько их.

import java.util.Scanner;

public class Zadanie2 {

public static void main (String [] args) {

Scanner in = new Scanner(System.in);

int c = 0;

String numb = in.nextLine();

int[] newStr = new int[numb.length()];

for (int i = 0; i < numb.length(); i++) {

newStr[i] = numb.charAt(i) - '0';

System.out.print(newStr[i] + " ");}

for (int i : newStr) {

if (i % 2 == 0) {c = c + 1;}}

System.out.println("Есть, всего их: " + c);}}

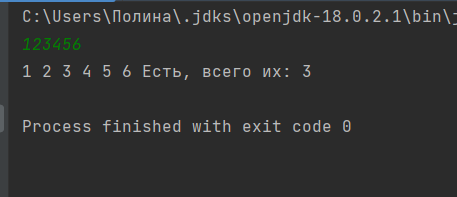


Рисунок 2 – Результат выполнения кода задания 2.2

**Задание 3**

import java.io.File;

import java.io.FileNotFoundException;

import java.util.Scanner;

public class Zadanie3 {

public static void main(String[] args) throws FileNotFoundException {

Scanner file = new Scanner(new File("Zadanie3.txt"));

//Scanner in = new Scanner(System.in);

int c = 0;

String numb = file.nextLine();

file.close();

int[] newStr = new int[numb.length()];

for (int i = 0; i < numb.length(); i++) {

newStr[i] = numb.charAt(i) - '0';

System.out.print(newStr[i] + " ");}

for (int i : newStr) {

if (i % 2 == 0) {

c = c + 1;}}

System.out.println("Есть, всего их: " + c);}}

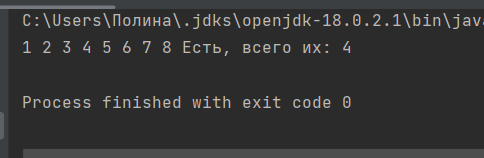


Рисунок 3 – Результат выполнения кода задания 3

**Задание 4**

**Пример 1**

public class Primer2 {

public static void main(String[] args) {

int[] marks = {74, 43, 58, 60, 90, 64, 70};

int sum = 0;

int count = marks.length;

for (int i=0; i<count; ++i) {

sum += marks[i];

}

double mean = sum/count;

System.out.printf("Mean is %.2f%n", mean);

}

}

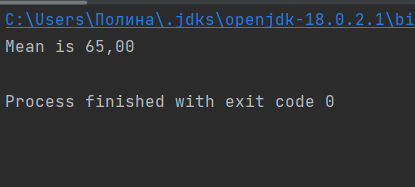


Рисунок 1 – результат выполнения программы

**Пример 2**

public class Primer2 {

public static void main(String[] args) {

int[] marks = {8, 2, 6, 4, 3};

int sum = 0;

for (int mark : marks) { // for each int number in int[] numbers

sum += mark;}System.out.println("The sum is " + sum);}}

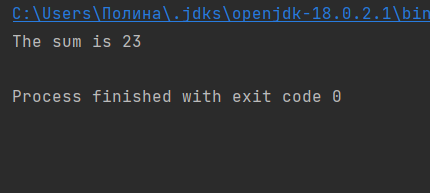
****

Рисунок 2 – результат выполнения программы

**Пример 3**

public class Primer3 {

public static void main(String[] args) {

int[][] grid = {{1, 2},{3, 4, 5},{6, 7, 8, 9}};

// Вывод массива grid

for (int y = 0; y < grid.length; ++y) {

for (int x = 0; x < grid[y].length; ++x) {

System.out.printf("%2d", grid[y][x]);}System.out.println();}

int[][] grid1 = new int[3][];

grid1[0] = new int[2];

grid1[1] = new int[3];

grid1[2] = new int[4];

// Вывод массива grid1

for (int y = 0; y < grid1.length; ++y) {

for (int x = 0; x < grid1[y].length; ++x) {

System.out.printf("%2d", grid1[y][x]);}

System.out.println();}}

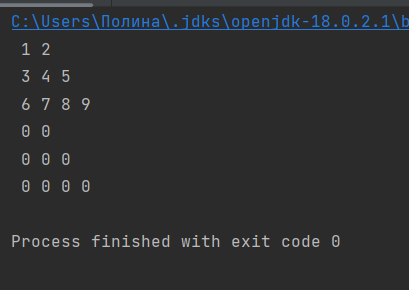
****

Рисунок 3 – Пример ввода данных из файла

**Контрольные вопросы**

1. Раскройте суть понятия массив в языке Java.

Массив – это упорядоченная коллекция элементов одного типа, идентифицируемых []. Чтобы использовать массив в программе его необходимо объявить, указав имя и тип, и определить – выделить память под его элементы, используя оператор new. В качестве имен массивов рекомендуется использовать существительные во множественном числе.

2. С какого значения начинается индексация массива в языке Java.

Индексация массивов в Java начинается с 0.

3. Имеется ли в языке Java возможность контроля выхода за границы массива.

В отличии от языков C/C++, в Java во время исполнения программы выполняется проверка выхода за границы массива. Другими словами, для каждой ссылки на элемент массива, индекс проверяется на соответствие длине массива. Если индекс находится за пределами диапазона [0, arrayName.legnth-1], JVM генерирует исключение ArrayIndexOutOfBoundException.

4. Приведите синтаксис объявления одномерного массива в языке Java.

int[] factors = new int[20];

5. Какие разновидности операторов циклов используются для перебора элементов массива в языке Java?

Для перебора элементов массива в Java используются циклы for и for-each.

6. Как интерпретируется двумерный массив в языке Java.

Многомерные массивы в Java рассматриваются как массив, элементами которого являются массивы, причем каждый из элементов-массивов может иметь разную длину.

7. Приведите синтаксис объявления и доступа к элементам много мерного массива в языке Java.

Синтаксис объявления:

int grid[][] = new int[3][4];

Синтаксис доступа к элементам:

grid[0][0] = 8;

8. Назовите особенности работы с массивами в языке Java по сравнению с другими языками программирования

1) Массивы в Java создаются только динамическими

2) В Java существует механизм генерации исключения при выходе за границы массива

3) Массивы в Java имеют набор методов для упрощения работы с ними, в том числе поле length.

У двумерных массивов элементы-массивы могут иметь разную длину.