**ЛАБОРАТОРНАЯ РАБОТА №8**

**«ПРОГРАММИРОВАНИЕ АЛГОРИТМОВ ОБРАБОТКИ**

**ОДНОМЕРНЫХ МАССИВОВ НА JAVA»**

**Вариант 9**

**8.1 Цель работы**

Изучить способы представления массивов в памяти ЭВМ, получить практические навыки реализации алгоритмов обработки одномерных массивов на языке Java.

**8.2 Вариант задания**

В одномерном массиве, состоящем из N вещественных элементов, вычислить:

1) максимальный по модулю элемент массива;

2) сумму элементов массива, расположенных между первым и вторым положительными элементами.

Преобразовать массив таким образом, чтобы элементы, большие 1, располагались после всех остальных. Упорядочить каждую часть массива по убыванию, используя алгоритм сортировки методом прямого выбора.

**8.3 Ход работы**

Были разработаны структурные схемы для функций программы (рисунки М.1 – М.8 приложение М).

Была разработана программа на языке программирования Java, которая решала поставленную задачу (листинг 8.1).

Листинг 8.1 – Код программы

import java.util.Scanner;

import static java.lang.Math.\*;

public class laba8 {

public static void main(String[] args) {

Scanner scn = new Scanner(System.in);

Array mass = new Array(scn);

System.out.println("Absolute max value: " + mass.absMaxElement());

System.out.println("Sum between first two positive elements: " + mass.sumBetweenPositive());

int i = mass.strangeSort();

mass.normalSort(0, i);

mass.normalSort(i);

mass.printArray();

scn.close();

return;

}

}

class Array {

private int amountOfElemts;

private double[] array;

public Array(Scanner scn) {

System.out.print("Amount of elements: ");

this.amountOfElemts = scn.nextInt();

this.array = new double[amountOfElemts];

getNumbers(scn);

}

private void getNumbers(Scanner scn) {

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

System.out.print("a[" + i + "] = ");

array[i] = scn.nextInt();

}

return;

}

protected double absMaxElement() {

double maxElement = array[0];

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

if (abs(array[i]) > abs(maxElement))

maxElement = abs(array[i]);

}

return maxElement;

}

protected double sumBetweenPositive() {

double sum = 0;

int i = 0;

while (i < amountOfElemts && array[i] <= 0)

i++;

i++;

while (i < amountOfElemts && array[i] <= 0) {

sum += array[i];

i++;

}

return sum;

}

protected void printArray() {

for (int i = 0; i < amountOfElemts; i++)

System.out.println("a[" + i + "] = " + array[i]);

return;

}

protected int strangeSort() {

int count = 0;

for (int i = 0; i < amountOfElemts - 1; i++) {

double temp = array[i];

int imin = i;

for (int j = i + 1; j < amountOfElemts; j++) {

if (array[j] < 1) {

imin = j;

temp = array[j];

}

}

if (temp < 1) {

count++;

}

array[imin] = array[i];

array[i] = temp;

}

return count;

}

protected void normalSort(int start, int stop) {

for (int i = start; i < stop - 1; i++) {

double temp = array[i];

int imax = i;

for (int j = i + 1; j < stop; j++) {

if (array[j] > temp) {

imax = j;

temp = array[j];

}

}

array[imax] = array[i];

array[i] = temp;

}

}

protected void normalSort(int start) {

int stop = amountOfElemts;

for (int i = start; i < stop - 1; i++) {

double temp = array[i];

int imax = i;

for (int j = i + 1; j < stop; j++) {

if (array[j] > temp) {

imax = j;

temp = array[j];

}

}

array[imax] = array[i];

array[i] = temp;

}

}

}

Был проведён тест программы (рисунки 8.1).

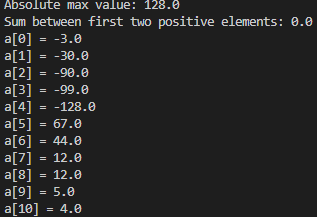


Рисунок 8.1 – Тест программы

**8.4 Вывод**

В ходе лабораторной работы были изучены способы представления массивов в памяти ЭВМ, также были получены практические навыки реализации алгоритмов обработки одномерных массивов на языке Java.

**Приложение М**

**Структурные схемы**

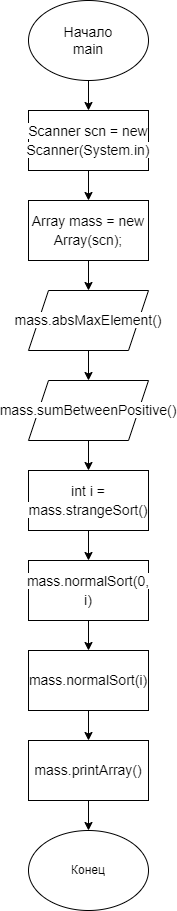


Рисунок М.1 – Структурная схема main

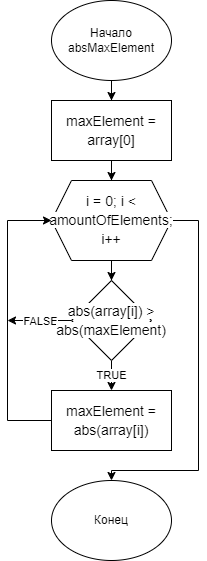


Рисунок M.2 – Структурная схема absMaxElement

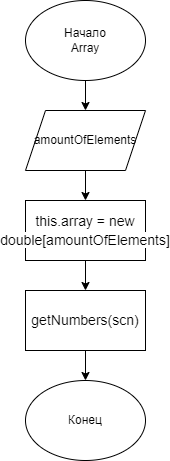


Рисунок M.3 – Структурная схема Array

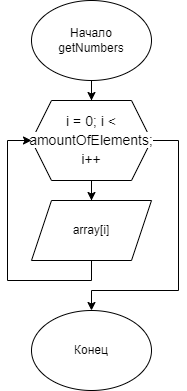


Рисунок M.4 – Структурная схема getNumbers

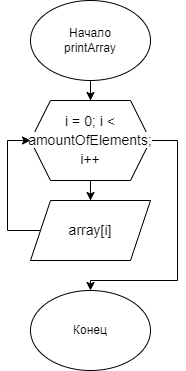


Рисунок M.5 – Структурная схема printArray

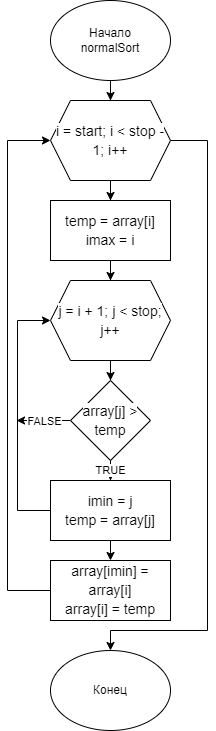


Рисунок M.6 – Структурная схема normalSort

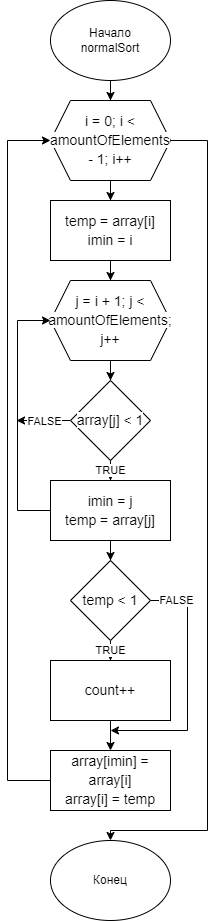


Рисунок M.7 – Структурная схема strangeSort

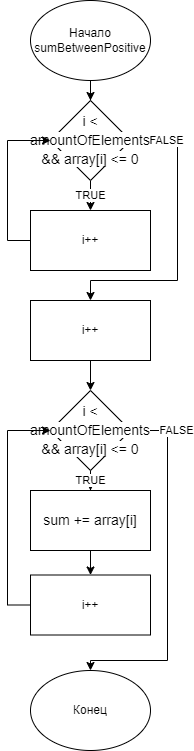


Рисунок M.8 – Структурная схема sumBetweenPositive