**Лабораторная работа №3**

**«ПРОГРАММИРОВАНИЕ АЛГОРИТМОВ ОБРАБОТКИ ОДНОМЕРНЫХ МАССИВОВ НА JAVA»**

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

**1. Вариант задания**

Вариант 30: В одномерном массиве, состоящем из N целых элементов, вычислить:

1) минимальный по модулю элемент массива;

2) сумму модулей элементов массива, расположенных после второго элемента, равного нулю.

Преобразовать массив таким образом, чтобы в первой его половине располагались элементы, стоявшие на нечетных позициях, а во второй половине – элементы, стоявшие на четных позициях. Упорядочить каждую часть массива по возрастанию, используя алгоритм сортировки методом прямого обмена.

**2. Алгоритм программы**

Программа состоит из нескольких циклов , пару циклов вложенных друг в друга для сортировки, условий необходимых для выполнения правильных действий.

Рисунок 1 – Структурная схема программы для решения данной функции(1-я часть)

Рисунок 2 – Структурная схема программы для решения данной функции(2-я часть)

**3.1 Код MAIN**

System.*out*.printf("\n Введите колличество элементов --> ");  
MassivChego\_toTam Massiv = new MassivChego\_toTam(in.nextInt());  
Massiv.RandomGenerate();  
System.*out*.printf("\n Сгененрирован следующий массив ");  
Massiv.PrintMassiv();  
System.*out*.printf("\n Минимальный по модулю элемент--> %d", Massiv.GetElem(Massiv.MinModIndex()));  
System.*out*.printf("\n Сумма модулей элементов массива, расположены после двух нулей ");  
System.*out*.printf("\n %d ",Massiv.SumBeforTwoZero());  
System.*out*.printf("\n Преобразованный массив(первыми стоят с нечет индексами, вторыми части");  
System.*out*.printf("\n Исходный ");  
Massiv.PrintMassiv();  
System.*out*.printf("\n Результирующий");  
Massiv.ChetNechet();  
Massiv.PrintMassiv();

**3.2 Код класса**

package Laba22;  
import java.util.Random;  
  
public class MassivChego\_toTam {  
 int[] element; int lenght;  
 public int GetElem(int index){  
 return this.element[index];  
 }  
 public void RandomGenerate()  
 {  
 Random r = new Random();  
 for (int i =0;i<lenght ;i++){  
 this.element[i] = r.nextInt(200) - 100;  
 }  
 }  
  
 public MassivChego\_toTam (int lenght){  
 this.element = new int[lenght];  
 this.lenght = lenght;  
 }  
  
 public void SortChast()  
 {  
 int center = lenght / 2; int j,i,tema;  
 for (j = center ; j > 1; j--) //сортировка левой части  
 for (i = 0; i < j - 1; i++)// сравнение двух соседних элементов  
 if (element[i] > element[i + 1]) {  
 tema = element[i]; // перестановка элементов  
 element[i] = element[i + 1];  
 element[i + 1] = tema;  
 }  
 for (j = lenght; j > 1; j--) //сортировка правой части  
 for (i = center; i < j - 1; i++)// сравнение двух соседних элементов  
 if (element[i] > element[i + 1]) {  
 tema = element[i]; // перестановка элементов  
 element[i] = element[i + 1];  
 element[i + 1] = tema;  
 }  
 }  
  
 public void ChetNechet(){  
 int center = lenght / 2;  
 int[] massiv = element;  
 int counter = 0;  
 for (int i = 0; i < lenght; i++) //перенос данных согласно условию, слева то что стояло на нечетном месте , справа то что стояло на четной позиции  
 {  
 if (i % 2 == 1) // если позиция не кратна двум , то ставим его в начало  
 {  
 element[counter] = massiv[i];  
 counter++;  
 }  
 else // если позиция кратна двум , то элементы ставим от центра  
 {  
 element[center] = massiv[i];  
 center++;  
 }  
 }  
 }  
  
 public int MinModIndex(){  
 int minMod = Math.*abs*(element[0]);  
 int index = 0;  
 for (int i = 1; i < lenght; i++) //поиск минимального по модулю элемента  
 {  
 if (minMod > Math.*abs*(element[i]))  
 {  
 minMod = Math.*abs*(element[i]);  
 index = i;  
 }  
 }  
 return index;  
 }  
  
 public int SumBeforTwoZero()  
 {  
 int sum = 0,count = 0;  
 for (int i =0;i<lenght;i++){  
 if (element[i] == 0)  
 {  
 count++;  
 }  
 if (count >= 2) // если два нулевых элемента было , сумируем остальные  
 {  
 sum += Math.*abs*(element[i]);  
 }  
 }  
 return sum;  
 }  
  
 public void PrintMassiv()  
 {  
 System.*out*.println();  
 for (int i =0;i<lenght;i++){  
 System.*out*.printf("%d ",element[i]);  
 }  
 System.*out*.println();  
 }  
}

**4.Результаты тестирования программы**

Для проверки правильности работы программы , для начала , отключим полностью случайные значения.

Выполним первичную генерацию массива и посмотрим что выведет программа.

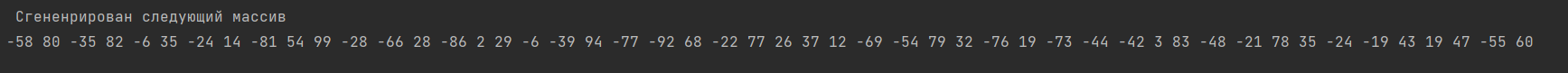


Рисунок 3 – Сгенерированный программой массив из целых рандомных значений.

Программа создала массив с такими переменными: --58 80 -35 82 -6 35 -24 14 -81 54 99 -28 -66 28 -86 2 29 -6 -39 94 -77 -92 68 -22 77 26 37 12 -69 -54 79 32 -76 19 -73 -44 -42 3 83 -48 -21 78 35 -24 -19 43 19 47 -55 60

В данном массиве минимальный по модулю элемент = 2, сумма элементов по модулю после двух 0 равна 0 , так как второго ноля в массиве нет.

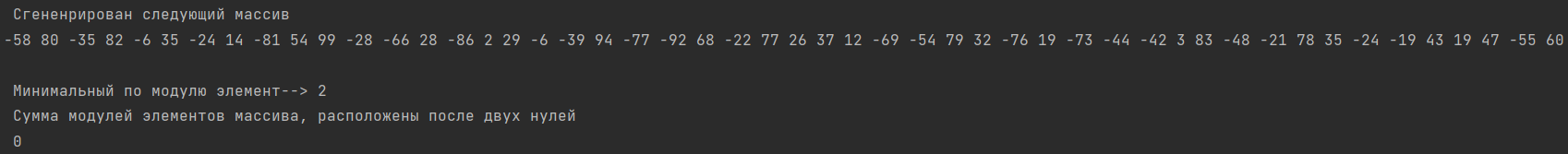


Рисунок 4 – Найденные минимальный по модулю элемент и сумма элементов по модулю после двух элементов равных 0.

Преобразованный по правилу массив должен быть таким : 80 82 35 14 54 -28 28 2 -6 94 -92 -22 -58 -6 -81 -66 29 -77 77 -24 -86 68 99 -35 -39 -58 -35 -6 -24 -81 99 -66 -86 29 -39 -77 68 77 -35 -24 99 -86 -39 68 -35 99 -39 -35 -39 -39

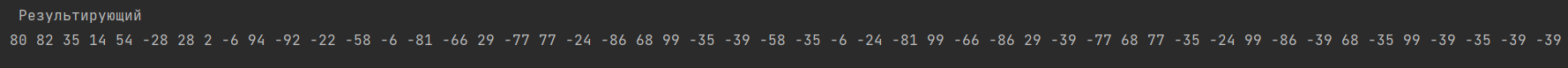


Рисунок 5 – Массив , полученные в результате перестановки элементов по правилу из задания

Данный массив надо отсортировать методом пузырька, при этом левая и правая сторона сортируются отдельно. В результате получим такой массив:

-92 -86 -81 -77 -66 -58 -39 -35 -28 -24 -22 -6 -6 2 14 28 29 35 54 68 77 80 82 94 99 -86 -86 -81 -77 -66 -58 -39 -39 -39 -39 -39 -35 -35 -35 -35 -24 -24 -6 29 68 68 77 99 99 99

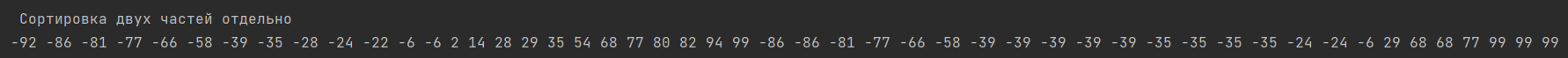


Рисунок 6 – Две половинки массива отсортированные по возрастанию элементов.

В результате тестирования, видно, что программа выполняет действия правильно , верно находит сумму элементов , минимальный по модулю элемент. Правильно переставляет элементы местами и сортирует массив методом прямого обмена(пузырька).

**Вывод**

В результате выполнения данной лабораторной работы были изучены принципы работы с массивами в JAVA, задание рандомных значения для элементов массива, вывод элементов массива в цикле в консоль. Так же были получены знания по применению на практике различных методы сортировки одномерных массивов, был реализован свой класс для работы с массивом.