УО «Белорусский государственный университет информатики и радиоэлектроники»

Кафедра ПОИТ

Отчет по лабораторной работе №3.1

по предмету

Основы алгоритмизации и программирования

Вариант 14

Выполнил:

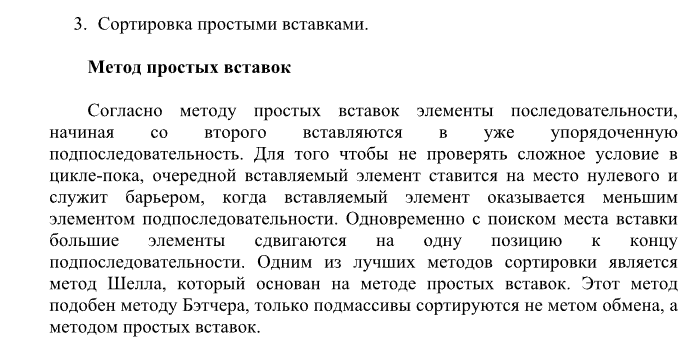
Крутько А.А.

Проверила:

Данилова Г.В.

Группа 251004

Минск 2022

Задание:  


Код программы на **Delphi**:

program LabThirdBlockThird;

uses

System.SysUtils;

Type

TArrOI = Array of Integer;

Function ChooseOption(): String;

Var

Input: String;

IsRight: Boolean;

Begin

Writeln('Введите console, если хотите использовать консоль, file, если файл');

Repeat

IsRight := True;

Readln(Input);

Input := LowerCase(Input);

If (Input <> 'console') and (Input <> 'file') then

Begin

IsRight := False;

Writeln('Повторите ввод!');

End;

Until IsRight;

ChooseOption := Input;

End;

Function ChooseConsoleWayToFill(): String;

Var

Input: String;

IsRight: Boolean;

Begin

Writeln('Введите console, если хотите сами ввести данные, random, чтобы   
 сгенерировать случайную размерность массива и случайные числа');

Repeat

IsRight := True;

Readln(Input);

Input := LowerCase(Input);

If (Input <> 'console') and (Input <> 'random') then

Begin

IsRight := False;

Writeln('Повторите ввод!');

End;

Until IsRight;

ChooseConsoleWayToFill := Input;

End;

Function InputSizeOfArray(): Integer;

Const

MIN\_NUM = 2;

Var

N: Integer;

IsCorrect: Boolean;

Begin

Write('Введите размерность массива: ');

Repeat

Try

IsCorrect := True;

Readln(N);

Except

IsCorrect := False;

Writeln('Повторите ввод');

End;

If IsCorrect and (N < MIN\_NUM) Then

Begin

IsCorrect := False;

Writeln('Повторите ввод');

End;

Until IsCorrect;

InputSizeOfArray := N;

End;

Function TakeCellIntoArray(): Integer;

Var

N: Integer;

IsCorrect: Boolean;

Begin

Repeat

Try

IsCorrect := True;

Readln(N);

Except

IsCorrect := False;

Writeln('Повторите ввод');

End;

Until IsCorrect;

TakeCellIntoArray := N;

End;

Function FillArray(N: Integer): TArrOI;

Var

Arr: TArrOI;

I: Integer;

Begin

SetLength(Arr, N);

For I := 0 to High(Arr) do

Begin

Write('Введите элемент: ', I + 1, ' ');

Arr[I] := TakeCellIntoArray();

End;

FillArray := Arr;

End;

Function ShellsSort(Arr: TArrOI): TArrOI;

Var

I, J, Temp, Gap: Integer;

Begin

Gap := (High(Arr) + 1) div 2;

While (Gap > 0) do

Begin

For I := 0 to (High(Arr) - Gap) do

Begin

J := I;

Temp := Arr[J + Gap];

While (J >= 0) and (Arr[J] > Temp) do

Begin

Arr[J + Gap] := Arr[J];

Arr[J] := Temp;

J := J - Gap;

End;

End;

Gap := Gap div 2;

End;

ShellsSort := Arr;

End;

Procedure WriteFirstArrayOfNumbers(Arr: TArrOI);

Var

I: Integer;

Begin

Writeln('Первоначальный массив: ');

For I := 0 to High(Arr) do

Write(Arr[I]:7);

Writeln;

End;

Procedure WriteNewArrayOfNumbers(Arr: TArrOI);

Var

I: Integer;

Begin

Arr := ShellsSort(Arr);

Writeln('Новый массив: ');

For I := 0 to High(Arr) do

Write(Arr[I]:7);

End;

Function TakeRandomSizeOfArray(): Integer;

Var

N: Integer;

Begin

Randomize;

N := Random(100) + 2;

TakeRandomSizeOfArray := N;

End;

Function TakeRandomElementsOfArray(N: Integer): TArrOI;

Var

Arr: TArrOI;

I: Integer;

Begin

SetLength(Arr, N);

For I := 0 to High(Arr) do

Arr[I] := Random(1000) - Random(1000);

TakeRandomElementsOfArray := Arr;

End;

Function ConsoleInput(InputChoice: String): TArrOI;

Var

N: Integer;

Arr: TArrOI;

Begin

If InputChoice = 'console' Then

Begin

N := InputSizeOfArray();

Arr := FillArray(N);

End

else

Begin

N := TakeRandomSizeOfArray();

Arr := TakeRandomElementsOfArray(N);

End;

ConsoleInput := Arr;

End;

Function InputSizeOfArrayFromFile(Var F: TextFile): Integer;

Const

MIN\_NUM = 2;

Var

N: Integer;

Begin

Readln(F, N);

InputSizeOfArrayFromFile := N;

End;

Function FillArrayFromFile(Var F: TextFile; N: Integer): TArrOI;

Var

Arr: TArrOI;

I: Integer;

Begin

SetLength(Arr, N);

For I := 0 to High(Arr) do

Read(F, Arr[I]);

FillArrayFromFile := Arr;

End;

Function InputFromFile(Way: String): TArrOI;

Var

F: TextFile;

N: Integer;

Arr: TArrOI;

Begin

AssignFile(F, Way);

Reset(F);

N := InputSizeOfArrayFromFile(F);

Arr := FillArrayFromFile(F, N);

Close(F);

InputFromFile := Arr;

End;

Function IsCorrectSizeOfArrayInFile(Var F: TextFile; Var N: Integer): Boolean;

Const

MIN\_NUM = 2;

Var

IsCorrect: Boolean;

Begin

IsCorrect := True;

If Not Eof(F) Then

Begin

Try

Readln(F, N);

Except

IsCorrect := False;

Writeln('Повторите ввод в файле размерности вашего массива');

End;

End

Else

Begin

Writeln('Нет Размерности массива');

IsCorrect := False;

End;

If IsCorrect and (N < MIN\_NUM) Then

IsCorrect := False;

IsCorrectSizeOfArrayInFile := IsCorrect;

End;

Function IsCorrectElementsOfArrayInFile(Var F: TextFile; N: Integer): Boolean;

Var

I, IndexRow, Elem: Integer;

IsCorrect: Boolean;

Begin

IsCorrect := True;

I := 0;

If Not Eof(F) Then

Begin

While (I < N) and (IsCorrect) do

Begin

If Not Eof(F) Then

Begin

Try

Read(F, Elem);

Except

IsCorrect := False;

IndexRow := I;

Writeln('Ошибка! Проверьте элемент под номером',   
 IndexRow + 1, 'на правильность ввода');

End;

End

Else

Begin

IndexRow := I;

IsCorrect := False;

Writeln('Данные для считывания матрицы закончились на   
 элементе ', IndexRow + 1, ', Осталось еще   
 элементов не считанных: ', N - IndexRow);

End;

Inc(I);

End;

End

Else

Begin

IsCorrect := False;

Writeln('Нет данных для матрицы');

End;

IsCorrectElementsOfArrayInFile := IsCorrect;

End;

Function TakeFileWay(Flag: String): String;

Var

Way: String;

F: TextFile;

IsCorrect: Boolean;

N: Integer;

Begin

Repeat

IsCorrect := True;

Writeln('Введите путь к файлу');

Readln(Way);

AssignFile(F, Way);

If FileExists(Way) Then

Begin

If Flag = 'In' Then

Begin

Reset(F);

IsCorrect := IsCorrectSizeOfArrayInFile(F, N);

If (IsCorrect) Then

IsCorrect := IsCorrectElementsOfArrayInFile(F, N);

If (IsCorrect and Not Eof(F)) Then

Begin

IsCorrect := False;

Writeln('уберите лишние данные');

End;

Close(F);

End;

End

Else

Begin

Writeln('Файла в заданном пути нет');

IsCorrect := False;

End;

Until IsCorrect;

TakeFileWay := Way;

End;

Function TakeFinalArrInformation(InputChoice: String): TArrOI;

Var

Arr: TArrOI;

Way: String;

Begin

If InputChoice = 'console' Then

Begin

InputChoice := ChooseConsoleWayToFill();

Arr := ConsoleInput(InputChoice);

End

else

Begin

Way := TakeFileWay('In');

Arr := InputFromFile(Way);

Writeln('Данные считаны успешно');

End;

TakeFinalArrInformation := Arr;

End;

Procedure WriteFirstArrayOfNumbersInFile(Var F: TextFile; Arr: TArrOI);

Var

I: Integer;

Begin

Writeln(F, 'Первоначальный массив: ');

For I := 0 to High(Arr) do

Write(F, Arr[I]:7);

Writeln(F, #13#10);

End;

Procedure WriteNewArrayOfNumbersInFile(Var F: TextFile; Arr: TArrOI);

Var

I: Integer;

Begin

Arr := ShellsSort(Arr);

Writeln(F, 'Новый Массив: ');

For I := 0 to High(Arr) do

Write(F, Arr[I]:7);

Writeln('Данные успешно выведены в файл');

End;

Procedure OutputAnswer(OutputChoice: String; Arr: TArrOI);

Var

Way: String;

F: TextFile;

Begin

If OutputChoice = 'console' Then

Begin

WriteFirstArrayOfNumbers(Arr);

WriteNewArrayOfNumbers(Arr);

End

Else

Begin

Way := TakeFileWay('Out');

AssignFile(F, Way);

ReWrite(F);

WriteFirstArrayOfNumbersInFile(F, Arr);

WriteNewArrayOfNumbersInFile(F, Arr);

Close(F);

End;

End;

Var

Arr: TArrOI;

InputChoice, OutputChoice: String;

Begin

Writeln('Данная программа сортирует массив методом Шелла');

InputChoice := ChooseOption();

Arr := TakeFinalArrInformation(InputChoice);

OutputChoice := ChooseOption();

OutputAnswer(OutputChoice, Arr);

Readln;  
End.

Код программы на **C++**:

#include <iostream>

#include <iomanip>

#include <string>

#include <fstream>

#include <algorithm>

using namespace std;

int inputSizeOfArray() {

int n;

bool isIncorrect;

const int MIN\_NUM = 2;

cout << "Введите размерность массива" << endl;

do {

isIncorrect = false;

cin >> n;

if (cin.fail())

{

cin.clear();

while (cin.get() != '\n');

cerr << "Проверьте правильность ввода данных: " << endl;

isIncorrect = true;

}

if (!isIncorrect && cin.get() != '\n')

{

cin.clear();

while (cin.get() != '\n');

cerr << "Проверьте правильность ввода данных: " << endl;

isIncorrect = true;

}

if (!isIncorrect && n < MIN\_NUM) {

cerr << "Число > 1" << endl;

isIncorrect = true;

}

} while (isIncorrect);

return n;

}

string chooseOption()

{

bool isIncorrect;

string input;

cout << "Введите console, если хотите использовать консоль, file, если файл" << endl;

do

{

isIncorrect = false;

cin >> input;

transform(input.begin(), input.end(), input.begin(), ::tolower);

if (input != "console" && input != "file")

{

cerr << "Повторите ввод" << endl;

isIncorrect = true;

}

} while (isIncorrect);

return input;

}

string chooseConsoleWayToFill()

{

bool isIncorrect;

string input;

cout << "Введите console, если хотите ввести данные сами, random, если хотите   
 сгенерировать случайно" << endl;

do

{

isIncorrect = false;

cin >> input;

transform(input.begin(), input.end(), input.begin(), ::tolower);

if (input != "console" && input != "random")

{

cerr << "Повторите ввод" << endl;

isIncorrect = true;

}

} while (isIncorrect);

return input;

}

int takeCellIntoArray()

{

bool isIncorrect;

int n;

do {

isIncorrect = false;

cin >> n;

if (cin.fail())

{

cin.clear();

while (cin.get() != '\n');

cerr << "Проверьте правильность ввода данных: " << endl;

isIncorrect = true;

}

if (!isIncorrect && cin.get() != '\n')

{

cin.clear();

while (cin.get() != '\n');

cerr << "Проверьте правильность ввода данных: " << endl;

isIncorrect = true;

}

} while (isIncorrect);

return n;

}

int\* fillArray(int n)

{

int\* arr = new int[n];

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

{

cout << "Введите элемент " << (i + 1) << endl;

arr[i] = takeCellIntoArray();

}

return arr;

}

int takeRandomSizeOfArray()

{

int n;

n = rand() % 200 + 2;

return n;

}

int\* takeRandomElementsInArray(int n)

{

int\* arr = new int[n];

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

{

arr[i] = rand() % 1000 - rand() % 1000;

}

return arr;

}

int\* shellsSort(int\* arr, int n)

{

int gap = n / 2;

int temp, j;

while (gap > 0) {

for (int i = 0; i < n - gap; i++) {

j = i;

temp = arr[j + gap];

while (j >= 0 && arr[j] > temp) {

arr[j + gap] = arr[j];

arr[j] = temp;

j -= gap;

}

}

gap /= 2;

}

return arr;

}

bool checkExtension(string path)

{

bool isInCorrect = true;

if ((path[path.length() - 1] == 't') && (path[path.length() - 2] == 'x') &&   
 (path[path.length() - 3] == 't') && (path[path.length() - 4] == '.'))

isInCorrect = false;

return isInCorrect;

}

bool isFileCorrectForSize(string way)

{

int n;

const int MIN\_NUM = 2;

bool isIncorrect = false;

ifstream fin(way);

if (!fin.eof())

{

fin >> n;

if (fin.fail())

{

isIncorrect = true;

cerr << "Проверьте правильность ввода размерности массива" << endl;

}

if (!isIncorrect && n < MIN\_NUM)

{

isIncorrect = true;

cerr << "Число > 1" << endl;

}

}

else

{

isIncorrect = true;

cerr << "Нет данных для размерности" << endl;;

}

fin.close();

return isIncorrect;

}

bool isArrayCorrectInFile(string way)

{

bool isIncorrect = false;

int n, elem;

ifstream fin(way, ios::beg);

fin >> n ;

for (int i = 0; i < n && !isIncorrect; i++)

{

if (!fin.eof())

{

fin >> elem;

if (fin.fail())

{

isIncorrect = true;

cerr << "Проверьте ваши данные на " << (i + 1) << " элементе массива" <<   
 endl;

}

}

else

{

isIncorrect = true;

cerr << "Закончились данные для матрицы на " << (i + 1) << " элементе" <<   
 endl;

}

}

if (!fin.eof() && !isIncorrect)

{

isIncorrect = true;

cerr << "Уберите лишние данные" << endl;

}

fin.close();

return isIncorrect;

}

string takeFileWay(string temp)

{

string way;

bool isIncorrect;

do

{

isIncorrect = false;

cout << "Введите путь к файлу" << endl;

cin >> way;

if (temp == "InSize")

{

ifstream fin(way);

if (!fin.is\_open() && checkExtension)

{

cout << "Проверьте правильность и нахождение вашего файла по заданному   
 пути" << endl;

isIncorrect = true;

}

else

{

isIncorrect = isFileCorrectForSize(way);

}

fin.close();

}

else if (temp == "InArray")

{

ifstream fin(way);

if (!fin.is\_open() && checkExtension)

{

cout << "Проверьте правильность и нахождение вашего файла по заданному   
 пути" << endl;

isIncorrect = true;

}

else

{

isIncorrect = isArrayCorrectInFile(way);

}

fin.close();

}

else

{

ifstream fout(way);

if (!fout.is\_open() && checkExtension)

{

cout << "Проверьте правильность и нахождение вашего файла по заданному   
 пути" << endl;

isIncorrect = true;

}

fout.close();

}

} while (isIncorrect);

return way;

}

int consoleInputSize(string inputChoice)

{

int n;

n = (inputChoice == "console") ? inputSizeOfArray() : takeRandomSizeOfArray();

return n;

}

int\* consoleInputArray(string inputChoice, int n)

{

int\* arr;

arr = (inputChoice == "console") ? fillArray(n) : takeRandomElementsInArray(n);

return arr;

}

int takeSizeOfArrayFromFile(string way)

{

int n;

ifstream fin(way, ios::beg);

fin >> n;

fin.close();

return n;

}

int\* takeElementsOfArrayFromFile(string way, int n)

{

string skipline;

int\* arr = new int[n];

ifstream fin(way, ios::beg);

getline(fin, skipline);

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

{

fin >> arr[i];

}

fin.close();

return arr;

}

int takeFinalInformationForSize(string inputChoice)

{

int n;

string way;

if (inputChoice == "console")

{

inputChoice = chooseConsoleWayToFill();

n = consoleInputSize(inputChoice);

}

else

{

way = takeFileWay("InSize");

n = takeSizeOfArrayFromFile(way);

}

return n;

}

int\* takeFinalInformationForArray(string inputChoice, int n)

{

int\* arr = nullptr;

string way;

if (inputChoice == "console")

{

inputChoice = chooseConsoleWayToFill();

arr = consoleInputArray(inputChoice, n);

}

else

{

way = takeFileWay("InArray");

arr = takeElementsOfArrayFromFile(way, n);

}

return arr;

}

void outputFirstArray(int\* arr, int n) {

cout << "Первоначальный массив: " << endl;

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

{

cout << arr[i] << '\t';

}

cout << '\n';

}

void outputNewArray(int\* arr, int n) {

arr = shellsSort(arr, n);

cout << "Новый массив:" << endl;

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

{

cout << arr[i] << '\t';

}

}

void outputFirstArrayInFile(int\* arr, string way, int n)

{

ofstream fout(way);

fout << "Первоначальный массив: " << endl;

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

{

fout << arr[i] << '\t';

}

fout << '\n';

fout.close();

}

void outputNewArrayInFile(int\* arr, string way, int n)

{

arr = shellsSort(arr, n);

ofstream fout(way, ios::app);

fout << "Новый массив: " << endl;

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

{

fout << arr[i] << '\t';

}

fout.close();

}

void outputFinalInformation(int\* arr, string outputChoice, int n) {

string way;

if (outputChoice == "console")

{

outputFirstArray(arr, n);

outputNewArray(arr, n);

}

else

{

way = takeFileWay("Out");

outputFirstArrayInFile(arr, way, n);

outputNewArrayInFile(arr, way, n);

}

}

int main()

{

setlocale(LC\_ALL, "RUSSIAN");

int\* arr;

int n;

string inputChoice, outputChoice;

cout << "Данная программа сортирует массив методом Шелла" << endl;

cout << "Ввод размерности:" << endl;

inputChoice = chooseOption();

n = takeFinalInformationForSize(inputChoice);

cout << "Ввод элементов массива" << endl;

inputChoice = chooseOption();

arr = takeFinalInformationForArray(inputChoice, n);

outputChoice = chooseOption();

outputFinalInformation(arr, outputChoice, n);

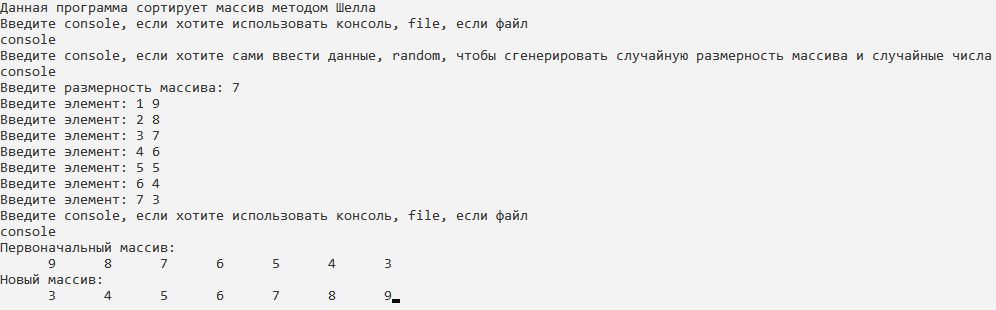
delete[] arr;

}

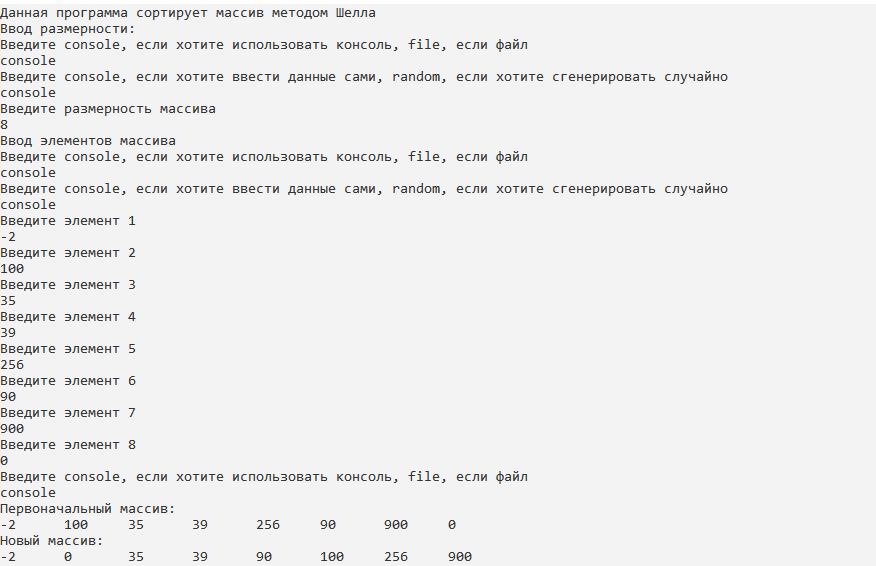
Код программы на **Java**:

import java.util.Scanner;  
import java.io.\*;  
  
public class LabThirdBlockThird {  
 static Scanner scan = new Scanner(System.in);  
  
 public static int inputSizeOfArray() {  
 int n = 0;  
 boolean isIncorrect;  
 final int MIN\_NUM = 2;  
 System.out.println("Введите размерность массива");  
 do {  
 isIncorrect = false;  
 try {  
 n = Integer.parseInt(scan.nextLine());  
 } catch (NumberFormatException E) {  
 System.err.println("Повторите ввод:\s");  
 isIncorrect = true;  
 }  
 if (!isIncorrect && n < MIN\_NUM) {  
 System.err.println("Размерность должна быть > 1:\s");  
 isIncorrect = true;  
 }  
 } while (isIncorrect);  
 return n;  
 }  
  
 public static int takeCellIntoArray() {  
 int temp = 0;  
 boolean isIncorrect;  
 do {  
 isIncorrect = false;  
 try {  
 temp = Integer.parseInt(scan.nextLine());  
 } catch (NumberFormatException E) {  
 System.err.println("Повторите ввод");  
 isIncorrect = true;  
 }  
 } while (isIncorrect);  
 return temp;  
 }  
 public static int[] FillArray() {  
 int n = inputSizeOfArray();  
 int[] arr = new int [n];  
 for (int i = 0; i < arr.length; i++) {  
 System.out.println("Введите элемент:\s" + (i+1));  
 arr[i] = takeCellIntoArray();  
 }  
 return arr;  
 }  
 public static String chooseOption(){  
 boolean isIncorrect;  
 String input;  
 System.out.println("Введите console, если хотите использовать   
 консоль, file, если файл");  
 do {  
 isIncorrect = false;  
 input = scan.nextLine();  
 if (!input.equalsIgnoreCase("console") && !input.equalsIgnoreCase("file")){  
 System.err.println("Повторите ввод:\s");  
 isIncorrect = true;  
 }  
 } while (isIncorrect);  
 return input;  
 }  
  
 public static String chooseConsoleWayToFill() {  
 boolean isIncorrect;  
 String input;  
 System.out.println("Введите console, если хотите ввести данные с консоли,   
 random, чтобы сгенерировать случайный массив");  
 do {  
 isIncorrect = false;  
 input = scan.nextLine();  
 if (!input.equalsIgnoreCase("console") &&   
 !input.equalsIgnoreCase("random")) {  
 System.err.println("Повторите ввод:\s");  
 isIncorrect = true;  
 }  
 } while (isIncorrect);  
 return input;  
 }  
  
 public static int takeRandomSizeOfArray() {  
 int n;  
 n = (int) Math.round((Math.random() \* 200) + 2);  
 return n;  
 }  
  
 public static int[] takeRandomElementsOfArray() {  
 int n = takeRandomSizeOfArray();  
 int[] arr = new int [n];  
 for (int i = 0; i < arr.length; i++) {  
 arr[i] = (int) (Math.round(Math.random() \* 1000) -   
 Math.round(Math.random() \* 1000));  
 }  
 return arr;  
 }  
  
 public static int[] shellsSort(int[] arr) {  
 int gap = arr.length / 2;  
 int temp, j;  
 while (gap > 0) {  
 for (int i = 0; i < arr.length - gap; i++) {  
 j = i;  
 temp = arr[j + gap];  
 while (j >= 0 && arr[j] > temp) {  
 arr[j + gap] = arr[j];  
 arr[j] = temp;  
 j -= gap;  
 }  
 }  
 gap /= 2;  
 }  
 return arr;  
 }  
  
 public static boolean isElemsOfArrayInFileCorrect(String tempStr, int n) {  
 String[] elems = tempStr.split("\s");  
 int temp = 0;  
 boolean isIncorrect = false;  
 if (elems.length == n) {  
 for (int i = 0; i < n && !isIncorrect; i++) {  
 try {  
 temp = Integer.parseInt(elems[i]);  
 } catch (NumberFormatException E) {  
 isIncorrect = true;  
 System.err.println("Число под индексом\s" + (i + 1) +   
 "\sнеправильное");  
 }  
 }  
 } else {  
 isIncorrect = true;  
 System.err.println("Закончились данные для считывания размерности матрицы   
 на элементе\s" + elems.length);  
 }  
 return isIncorrect;  
 }  
 public static boolean isTakeInformationFromFile(String way) {  
 int n = 0, temp = 0;  
 String inputSize, inputElemsOfArray;  
 final int MIN\_NUM = 2;  
 boolean isIncorrect = false;  
 try (BufferedReader br = new BufferedReader(new FileReader(way))) {  
 if ((inputSize = br.readLine()) != null) {  
 try {  
 n = Integer.parseInt(inputSize);  
 } catch(NumberFormatException E) {  
 isIncorrect = true;  
 System.err.println("Проверьте соответствие размерности к целому   
 числу");  
 }  
 if (!isIncorrect & n < MIN\_NUM) {  
 isIncorrect = true;  
 System.err.println("Число должно быть > 1");  
 }  
 } else {  
 isIncorrect = true;  
 System.err.println("В файле нет данных размерности массива");  
 }  
 if ((inputElemsOfArray = br.readLine()) != null) {  
 if (!isIncorrect) {  
 isIncorrect = isElemsOfArrayInFileCorrect(inputElemsOfArray, n);  
 }  
 } else {  
 isIncorrect = true;  
 System.err.println("Нет данных для элементов массива");  
 }  
 if (!isIncorrect & br.readLine() != null) {  
 isIncorrect = true;  
 System.out.println("Уберите лишние данные");  
 }  
 } catch (IOException e) {  
 System.err.println("Ошибка файла");  
 System.err.println(e.getMessage());  
 isIncorrect = true;  
 }  
 return isIncorrect;  
 }  
  
 public static String takeFileWay(String temp) {  
 String way;  
 boolean isIncorrect;  
 do {  
 isIncorrect = false;  
 System.out.println("Введите путь к файлу");  
 way = scan.nextLine();  
 File file = new File(way);  
 if (!file.exists() || !file.canWrite() || !file.canRead() ||   
 !way.endsWith(".txt") || file.isDirectory()) {  
 isIncorrect = true;  
 System.err.println("Файл не найден, или неверный формат файла;   
 повторите попытку ввода");  
 } else if (temp.equals("In")) {  
 isIncorrect = isTakeInformationFromFile(way);  
 }  
 } while (isIncorrect);  
 return way;  
 }  
  
 public static int[] consoleInput(String inputChoice) {  
 int[] arr;  
 arr = (inputChoice.equalsIgnoreCase("console")) ? FillArray() :   
 takeRandomElementsOfArray();  
 return arr;  
 }  
  
 public static int takeFromFileArraySize(String way) {  
 int n = 0;  
 try (BufferedReader br = new BufferedReader(new FileReader(way))) {  
 n = Integer.parseInt(br.readLine());  
 } catch (IOException e) {  
 System.err.println("Ошибка файла");  
 System.err.println(e.getMessage());  
 }  
 return n;  
 }  
  
 public static int[] takeFromFileArray(String way) {  
 int n = takeFromFileArraySize(way);  
 int[] arr = new int[n];  
 try (BufferedReader br = new BufferedReader(new FileReader(way))) {  
 br.readLine();  
 String[] elems = br.readLine().split("\s");  
 for (int i = 0; i < n; i++) {  
 arr[i] = Integer.parseInt(elems[i]);  
 }  
 } catch (IOException e) {  
 System.err.println("Ошибка файла");  
 System.err.println(e.getMessage());  
 }  
 return arr;  
 }  
 public static int[] takeFinalInformaton(String inputChoice) {  
 int[] arr;  
 String way;  
 if (inputChoice.equalsIgnoreCase("console")) {  
 inputChoice = chooseConsoleWayToFill();  
 arr = consoleInput(inputChoice);  
 } else {  
 way = takeFileWay("In");  
 arr = takeFromFileArray(way);  
 }  
 return arr;  
 }  
  
 public static void outputFirstArray(int[] arr) {  
 System.out.println("Первоначальный массив: ");  
 for (int i : arr) { //int i = 0; i < arr.length; i++  
 System.out.printf("%5d", i);  
 }  
 System.out.println('\n');  
 }  
  
 public static void outputNewArray(int[] arr) {  
 System.out.println("Новый массив:\s");  
 for (int i : arr) {  
 System.out.printf("%5d", i);  
 }  
 }  
  
 public static void outputFirstArrayInFile(int[] arr, String way){  
 try (PrintWriter pw = new PrintWriter(new FileWriter(way, false))) {  
 pw.write("Первоначальный массив:\s\n");  
 for (int i : arr) {  
 pw.printf("%5d", i);  
 }  
 pw.write('\n');  
 } catch (IOException e) {  
 System.err.println("Ошибка файла");  
 System.err.println(e.getMessage());  
 }  
 }  
  
 public static void outputNewArrayInFile(int[] arr, String way){  
 try (PrintWriter pw = new PrintWriter(new FileWriter(way, true))) {  
 pw.write("Новый массив:\s\n");  
 for (int i : arr) {  
 pw.printf("%5d", i);  
 }  
 } catch (IOException e) {  
 System.err.println("Ошибка файла");  
 System.err.println(e.getMessage());  
 }  
 }  
  
 public static void outputFinalInformation(int[] arr, String outputChoice) {  
 String way;  
 if (outputChoice.equalsIgnoreCase("console")) {  
 outputFirstArray(arr);  
 arr = shellsSort(arr);  
 outputNewArray(arr);  
 } else {  
 way = takeFileWay("Out");  
 outputFirstArrayInFile(arr, way);  
 arr = shellsSort(arr);  
 outputNewArrayInFile(arr, way);  
 }  
 }  
  
 public static void main(String[] args) {  
 String inputChoice, outputChoice;  
 int[] arr;  
 System.out.println("Данная программа сортирует массив методом Шелла");  
 inputChoice = chooseOption();  
 arr = takeFinalInformaton(inputChoice);  
 outputChoice = chooseOption();  
 outputFinalInformation(arr, outputChoice);  
 }  
}

Результат на **Delphi**:



Результат на **C++**:

****

Результат на **Java**:

