УО «Белорусский государственный университет информатики и радиоэлектроники»

Кафедра ПОИТ

Отчет по лабораторной работе №3.3

по предмету «Основы алгоритмизации и программирования»

Вариант 15

Выполнила:

Городко К. Е.

Гр. 351005

Проверила:

Данилова Г. В.

Минск 2023

**ЗАДАНИЕ:**

Массив записей содержит фамилии участников соревнования и количество набранных баллов. Упорядочить записи в порядке убывания количества баллов. Если некоторые участники набрали одинаковое количество баллов, то их фамилии вывести в алфавитном порядке.

**КОД ПРОГРАММЫ DELPHI:**

Program LAB3\_3;

Uses

System.SysUtils;

Const

MINSIZE = 3;

MAXSIZE = 20;

MINPOINT = 0;

MAXPOINT = 100;

MAXSURNAMELENGTH = 30;

Type

TParticipant = Record

Surname: String[30];

Points: 0 .. 100;

End;

TList = Array Of TParticipant;

Procedure WriteCondition();

Begin

Writeln('Данная программа упорядочивает список участников с их баллами на

соревновании.');

End;

Function IsNumRangeCorrect(Const MIN, MAX: Integer; Num: Integer; ErrorMessage:

String; Var IsCorrect: Boolean): Boolean;

Begin

If IsCorrect And (Num > MAX) Or (Num < MIN) Then

Begin

IsCorrect := False;

Writeln(ErrorMessage);

End;

IsNumRangeCorrect := IsCorrect;

End;

Function InputNum(Const MIN, MAX: Integer; OutputMessage: String): Integer;

Var

Num: Integer;

IsCorrect: Boolean;

Begin

Repeat

IsCorrect := True;

Write(OutputMessage);

Try

Readln(Num);

Except

IsCorrect := False;

Writeln('Введенное значение должно быть целым числом! Повторите

попытку.');

End;

IsCorrect := IsNumRangeCorrect(MIN, MAX, Num, 'Диапазон введенного числа не

соответствует условию! Повторите попытку.',

IsCorrect);

Until (IsCorrect);

InputNum := Num;

End;

Function InputSurname(I: Integer): String;

Var

Surname: String;

IsCorrect: Boolean;

Begin

Repeat

IsCorrect := True;

Write('Введите фамилию ', I + 1, ' участника: ');

Readln(Surname);

If (Surname = '') Or (Length(Surname) > MAXSURNAMELENGTH) Then

Begin

Writeln('Фамилия должна содержать от 1 до 30 символов! Повторите

попытку.');

IsCorrect := False;

End;

Until IsCorrect;

InputSurname := Surname;

End;

Function GetListFromConsole(ListSize: Integer): TList;

Var

I, Check: Integer;

PointsString: String;

List: TList;

IsCorrect: Boolean;

Begin

SetLength(List, ListSize);

Writeln('Условия ввода: фамилия участника должна быть не длиннее 30 символов,

количество баллов - целое число от 0 до 100.');

For I := 0 To High(List) Do

Begin

List[I].Surname := InputSurname(I);

List[I].Points := InputNum(MINPOINT, MAXPOINT, 'Введите количество баллов

участника: ');

End;

GetListFromConsole := List;

End;

Function IsFilePathCorrect(Path: String): Boolean;

Var

IsCorrect: Boolean;

Begin

IsCorrect := True;

If Not FileExists(Path) Then

Begin

Writeln('Введенного файла не существует. Повторите попытку.');

IsCorrect := False;

End

Else

If ExtractFileExt(Path) <> '.txt' Then

Begin

Writeln('Введенный Вами файл не является текстовым. Повторите попытку.');

IsCorrect := False;

End;

IsFilePathCorrect := IsCorrect;

End;

Function CheckFileInputPath(): String;

Var

Path: String;

Begin

Writeln;

Repeat

Writeln('Условия: в первой строке должно быть записано количество участников

(от 3 до 20).');

Writeln('В строках с участниками сначала записывается фамилия (до 30

символов), затем количество набранных баллов (целое число от 0 до

100).');

Writeln('Введите путь к файлу, содержащему список участников.');

Readln(Path);

Until IsFilePathCorrect(Path);

CheckFileInputPath := Path;

End;

Function ReadSurnameFromFile(Var IsCorrect: Boolean; Var FIn: TextFile): String;

Var

Symbol: Char;

Surname: String;

Begin

Surname := '';

If IsCorrect Then

Begin

Read(FIn, Symbol);

Repeat

Surname := Surname + Symbol;

Read(FIn, Symbol);

Until (Symbol = ' ') Or (Length(Surname) = MAXSURNAMELENGTH);

If (Surname = '') Or (Symbol <> ' ') Then

Begin

Writeln('Фамилии, записанные в файле, не соответствуют диапазону.

Повторите попытку.');

IsCorrect := False;

End;

End;

ReadSurnameFromFile := Surname;

End;

Function GetListFromFile(Path: String): TList;

Var

I, ListSize: Integer;

List: TList;

IsCorrect: Boolean;

FIn: TextFile;

Begin

Repeat

IsCorrect := True;

Try

Try

Assign(FIn, Path);

Reset(FIn);

Readln(FIn, ListSize);

IsCorrect := IsNumRangeCorrect(MINSIZE, MAXSIZE, ListSize,'Размер,

записанный в файле, не соответствует

диапазону. Повторите попытку.',

IsCorrect);

SetLength(List, ListSize);

I := 0;

While IsCorrect And (I < (High(List) + 1)) Do

Begin

If Eof(FIn) Then

Begin

Writeln('Количество участников не совпадает с записанным в

файле размером. Повторите попытку.');

IsCorrect := False;

End;

List[I].Surname := ReadSurnameFromFile(IsCorrect, FIn);

Readln(FIn, List[I].Points);

IsCorrect := IsNumRangeCorrect(MINPOINT, MAXPOINT,List[I].Points,

'Баллы, записанные в файле, не

соответстуют диапазону. Повторите

попытку.', IsCorrect);

Inc(I);

End;

If IsCorrect And Not Eof(FIn) Then

Begin

Writeln('Количество участников не совпадает с записанным в файле

размером. Повторите попытку.');

IsCorrect := False;

End;

Finally

CloseFile(FIn);

End;

Except

Writeln('Данные выбранного файла не соответствуют условию. Повторите

попытку.');

IsCorrect := False;

End;

If Not IsCorrect Then

Path := CheckFileInputPath();

Until IsCorrect;

Writeln('Данные из файла успешно считаны.');

GetListFromFile := List;

End;

Function InputList(): TList;

Var

Choice, ListSize: Integer;

List: TList;

FInPath: String;

Begin

Choice := InputNum(0, 1, 'Если Вы хотите вводить данные в консоль, введите 0.

Если использовать файл, введите 1: ');

If Choice = 0 Then

Begin

ListSize := InputNum(MINSIZE, MAXSIZE, 'Введите количество участников (от 3

до 20): ');

List := GetListFromConsole(ListSize);

End

Else

Begin

FInPath := CheckFileInputPath();

List := GetListFromFile(FInPath);

End;

InputList := List;

End;

Function SwapElements(List: TList; I: Integer): TList;

Var

Temp: TParticipant;

Begin

Temp := List[I];

List[I] := List[I - 1];

List[I - 1] := Temp;

SwapElements := List;

End;

Function SortList(List: TList): TList;

Var

I, J: Integer;

Begin

For I := 1 To High(List) Do

For J := 1 To High(List) Do

If List[J].Points > List[J - 1].Points Then

List := SwapElements(List, J)

Else

If (List[J].Points = List[J - 1].Points) And (AnsiUpperCase

(List[J].Surname) < AnsiUpperCase(List[J - 1].Surname)) Then

List := SwapElements(List, J);

SortList := List;

End;

Procedure OutputList(List: TList; ListName: String);

Var

I: Integer;

Begin

Writeln;

Writeln(ListName);

For I := 0 To High(List) Do

Writeln(List[I].Surname, ' ', List[I].Points);

WriteLn;

End;

Function CheckFileOutputPath(): String;

Var

Path: String;

IsCorrect: Boolean;

Begin

Repeat

IsCorrect := True;

Writeln('Введите путь к файлу, в который нужно записать результат.');

Readln(Path);

IsCorrect := IsFilePathCorrect(Path);

If IsCorrect And FileIsReadOnly(Path) Then

Begin

Writeln('Введенный Вами файл доступен только для чтения. Повторите

попытку.');

IsCorrect := False;

End;

Until IsCorrect;

CheckFileOutputPath := Path;

End;

Procedure WriteResultIntoFile(List: TList; Path: String);

Var

I: Integer;

FOut: TextFile;

IsCorrect: Boolean;

Begin

Repeat

IsCorrect := True;

Assign(FOut, Path);

Try

Try

Rewrite(FOut);

For I := 0 To High(List) Do

Writeln(FOut, List[I].Surname, ' ', List[I].Points);

Finally

CloseFile(FOut);

End;

Except

Writeln('Произошла ошибка. Повторите попытку.');

IsCorrect := False;

Path := CheckFileOutputPath();

End;

Until IsCorrect;

Writeln('Результат записан.');

End;

Procedure OutputResult(SortedList: TList);

Var

Choice: Integer;

FOutPath: String;

Begin

Choice := InputNum(0, 1, 'Если нужно вывести результат в консоль, введите 0. Если

в файл, введите 1: ');

If Choice = 0 Then

OutputList(SortedList, 'Отсортированный список:')

Else

Begin

FOutPath := CheckFileOutputPath();

WriteResultIntoFile(SortedList, FOutPath);

End;

End;

Var

List, SortedList: TList;

Begin

WriteCondition();

List := InputList();

OutputList(List, 'Введенный список:');

SortedList := SortList(List);

OutputResult(SortedList);

End.

**КОД ПРОГРАММЫ С++:**

#include <iostream>

#include <fstream>

#include <locale>

#include <Windows.h>

using namespace std;

const int MINSIZE = 3;

const int MAXSIZE = 20;

const int MINPOINT = 0;

const int MAXPOINT = 100;

const int MAXSURNAMELENGTH = 30;

struct participant {

string surname;

int points;

};

void writeCondition() {

cout << "Данная программа упорядочивает список участников с их баллами на

соревновании.\n";

}

bool isNumRangeIncorrect(const int MIN, const int MAX, int num, string errorMessage,

bool& isIncorrect) {

if (!isIncorrect && ((num < MIN) || (num > MAX))) {

cout << errorMessage;

isIncorrect = true;

}

return isIncorrect;

}

int inputNum(const int MIN, const int MAX, string outputMessage) {

int num;

num = 0;

bool isIncorrect;

do {

isIncorrect = false;

cout << outputMessage;

cin >> num;

if (cin.get() != '\n') {

cin.clear();

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

isIncorrect = true;

cout << "Введенное значение должно быть целым числом! Повторите

попытку.\n";

}

isIncorrect = isNumRangeIncorrect(MIN, MAX, num, "Диапазон введенного числа

не соответствует условию! Повторите

попытку.\n", isIncorrect);

} while (isIncorrect);

return num;

}

bool isSurnameIncorrect(string surname) {

bool isIncorrect;

isIncorrect = false;

if ((surname == "") || (surname.size() > MAXSURNAMELENGTH)) {

isIncorrect = true;

cout << "Фамилия должна содержать от 1 до 30 символов! Повторите попытку.\n";

}

return isIncorrect;

}

string inputSurname(int i) {

string surname;

do {

cout << "Введите фамилию " << i + 1 << " участника: ";

cin >> surname;

} while (isSurnameIncorrect(surname));

return surname;

}

participant\* getListFromConsole(int listSize) {

participant\* list = new participant[listSize];

cout << "Условия ввода: фамилия участника должна быть не длиннее 30 символов,

количество баллов - целое число от 0 до 100.\n";

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

list[i].surname = inputSurname(i);

list[i].points = inputNum(MINPOINT, MAXPOINT, "Введите количество баллов

участника: ");

}

return list;

}

bool isFileExtIncorrect(string path, bool isIncorrect) {

if (!isIncorrect) {

if (path.substr(path.size() - 4) == ".txt")

isIncorrect = false;

else

cout << "Введенный Вами файл не является текстовым. Повторите

попытку.\n";

}

return isIncorrect;

}

string checkFileInputPath() {

string path;

bool isIncorrect;

do {

isIncorrect = false;

cout << "Условия: в первой строке должно быть записано количество участников

(от 3 до 20).\n";

cout << "В строках с участниками сначала записывается фамилия (до 30

символов), затем количество набранных баллов (целое число от 0 до

100).\n";

cout << "Введите путь к файлу, содержащему список участников.\n";

cin >> path;

ifstream fin(path);

fin.open(path);

if (!fin.is\_open()) {

cout << "Произошла ошибка. Повторите попытку.\n";

isIncorrect = true;

}

isIncorrect = isFileExtIncorrect(path, isIncorrect);

fin.close();

} while (isIncorrect);

return path;

}

participant\* getListFromFile(string path, int& listSize) {

participant\* list = new participant();

bool isIncorrect;

do {

isIncorrect = false;

ifstream fin(path);

try {

fin >> listSize;

isIncorrect = isNumRangeIncorrect(MINSIZE, MAXSIZE, listSize, "Размер,

записанный в файле, не соответствует

диапазону. Повторите попытку.",

isIncorrect);

list = new participant[listSize];

int i = 0;

while ((!isIncorrect) && (i < (listSize))) {

fin >> list[i].surname;

isIncorrect = isSurnameIncorrect(list[i].surname);

fin >> list[i].points;

isIncorrect = isNumRangeIncorrect(MINPOINT, MAXPOINT, list[i].points,

"Баллы, записанные в файле, не

соответстуют диапазону. Повторите

попытку.", isIncorrect);

i++;

}

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

cout << "Количество участников не совпадает с записанным в файле

размером. Повторите попытку.\n";

isIncorrect = true;

}

}

catch (string errorMessage) {

cout << "Данные выбранного файла не соответствуют условию. Повторите

попытку.\n";

isIncorrect = true;

}

fin.close();

if (isIncorrect)

path = checkFileInputPath();

} while (isIncorrect);

cout << "Данные из файла успешно считаны.\n";

return list;

}

participant\* inputList(int& listSize) {

int choice;

choice = 0;

participant\* list;

choice = inputNum(0, 1, "Если Вы хотите вводить данные в консоль, введите 0. Если

использовать файл, введите 1: ");

if (choice == 0) {

listSize = inputNum(MINSIZE, MAXSIZE, "Введите количество участников (от 3 до

20): ");

list = getListFromConsole(listSize);

}

else {

string finPath;

finPath = checkFileInputPath();

list = getListFromFile(finPath, listSize);

}

return list;

}

string upcaseWord(string word) {

locale locRus("Russian");

for (int i = 0; i < (word.size() + 1); i++)

word[i] = toupper(word[i], locRus);

return word;

}

participant\* sortList(participant\* list, int listSize) {

for (int i = 1; i < listSize; i++)

for (int j = 1; j < listSize; j++)

if (list[j].points > list[j - 1].points)

swap(list[j - 1], list[j]);

else

if ((list[j].points == list[j - 1].points) && ((upcaseWord

(list[j].surname) < upcaseWord(list[j - 1].surname)))

swap(list[j - 1], list[j]);

return list;

}

void outputList(participant\* list, string listName, int listSize) {

cout << "\n" << listName;

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

cout << list[i].surname << " " << list[i].points << "\n";

cout << "\n";

}

string checkFileOutputPath() {

string path;

bool isIncorrect;

do {

isIncorrect = false;

cout << "Введите путь к файлу, в который нужно записать результат.\n";

cin >> path;

ofstream fout(path);

fout.open(path);

if (!fout.is\_open()) {

cout << "Произошла ошибка. Повторите попытку.\n";

isIncorrect = true;

}

isIncorrect = isFileExtIncorrect(path, isIncorrect);

fout.close();

} while (isIncorrect);

return path;

}

void writeResultIntoFile(participant\* list, string path, int listSize) {

bool isIncorrect;

do {

isIncorrect = false;

ofstream fout(path);

try {

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

fout << list[i].surname << " " << list[i].points << "\n";

}

catch (string errorMessage) {

cout << "Произошла ошибка. Повторите попытку.\n";

isIncorrect = true;

}

fout.close();

if (isIncorrect)

path = checkFileOutputPath();

} while (isIncorrect);

cout << "Результат записан.\n";

}

void outputResult(participant\* sortedList, int listSize) {

int choice;

choice = 0;

choice = inputNum(0, 1, "Если нужно вывести результат в консоль, введите 0. Если

в файл, введите 1: ");

if (choice == 0)

outputList(sortedList, "Отсортированный список:\n", listSize);

else {

string foutPath;

foutPath = checkFileOutputPath();

writeResultIntoFile(sortedList, foutPath, listSize);

}

}

int main() {

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

int listSize;

participant\* list;

participant\* sortedList;

listSize = 0;

writeCondition();

list = inputList(listSize);

outputList(list, "Введенный список:\n", listSize);

sortedList = sortList(list, listSize);

outputResult(sortedList, listSize);

delete[] list;

delete[] sortedList;

return 0;

}

**КОД ПРОГРАММЫ JAVA:**

import java.util.Scanner;

import java.io.\*;

public class Main {

static final int MINSIZE = 3;

static final int MAXSIZE = 20;

static final int MINPOINT = 0;

static final int MAXPOINT = 100;

static final int MAXSURNAMELENGTH = 30;

record participant (String surname, int points) {};

private static void writeCondition() {

System.out.println("Данная программа упорядочивает список участников с их

баллами на соревновании.");

}

private static boolean isNumRangeIncorrect(final int MIN, final int MAX, int num,

String errorMessage) {

boolean isIncorrect;

isIncorrect = false;

if ((num < MIN) || (num > MAX)) {

System.out.println(errorMessage);

isIncorrect = true;

}

return isIncorrect;

}

private static int inputNum(final int MIN, final int MAX, String outputMessage,

Scanner scan) {

int num;

num = 0;

boolean isIncorrect;

do {

isIncorrect = false;

System.out.print(outputMessage);

try {

num = Integer.parseInt(scan.nextLine());

}

catch (NumberFormatException e){

System.out.println("Введенное значение должно быть целым числом!

Повторите попытку.");

isIncorrect = true;

}

if (!isIncorrect)

isIncorrect = isNumRangeIncorrect(MIN, MAX, num, "Диапазон введенного

числа не соответствует условию!

Повторите попытку.");

} while (isIncorrect);

return num;

}

private static boolean isSurnameIncorrect(String surname) {

boolean isIncorrect;

isIncorrect = false;

if (surname.isEmpty() || (surname.length() > MAXSURNAMELENGTH)) {

isIncorrect = true;

System.out.println("Фамилия должна содержать от 1 до 30 символов!

Повторите попытку.");

}

return isIncorrect;

}

private static String inputSurname(int i, Scanner scan) {

String surname;

do {

System.out.print("Введите фамилию " + i + 1 + " участника: ");

surname = scan.nextLine();

} while (isSurnameIncorrect(surname));

return surname;

}

private static participant[] getListFromConsole(int listSize, Scanner scan) {

participant[] list = new participant[listSize];

String surname;

int points;

points = 0;

System.out.println("Условия ввода: фамилия участника должна быть не длиннее

30 символов, количество баллов - целое число от 0 до

100.");

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

surname = inputSurname(i, scan);

points = inputNum(MINPOINT, MAXPOINT, "Введите количество баллов

участника: ", scan);

list[i] = new participant(surname, points);

}

return list;

}

private static boolean isFilePathIncorrect(String path, File file) {

boolean isIncorrect;

isIncorrect = false;

if (!file.exists()) {

System.out.println("Введенного файла не существует. Повторите попытку.");

isIncorrect = true;

}

else

if (!path.endsWith(".txt")) {

System.out.println("Введенный Вами файл не является текстовым.

Повторите попытку.");

isIncorrect = true;

}

return isIncorrect;

}

private static String checkFileInputPath(Scanner scan) {

String path;

File file;

do {

System.out.println("\nУсловия: в первой строке должно быть записано

количество участников (от 3 до 20).");

System.out.println("В строках с участниками сначала записывается фамилия

(до 30 символов), затем количество набранных баллов

(целое число от 0 до 100).");

System.out.println("Введите путь к файлу, содержащему список

участников.");

path = scan.nextLine();

file = new File(path);

} while (isFilePathIncorrect(path, file));

return path;

}

private static participant[] getListFromFile(String path, Scanner scan) {

int listSize;

listSize = 0;

int points;

points = 0;

String surname;

participant[] list = new participant[0];

boolean isIncorrect;

do {

isIncorrect = false;

try (Scanner scanFile = new Scanner(new FileReader(path))) {

listSize = scanFile.nextInt();

isIncorrect = isNumRangeIncorrect(MINSIZE, MAXSIZE, listSize,

"Размер, записанный в файле, не

соответствует диапазону. Повторите

попытку.");

list = new participant[listSize];

int i = 0;

while ((!isIncorrect) && (i < listSize)) {

surname = scanFile.next();

isIncorrect = isSurnameIncorrect(surname);

if (!isIncorrect) {

points = scanFile.nextInt();

isIncorrect = isNumRangeIncorrect(MINPOINT, MAXPOINT, points,

"Баллы, записанные в файле,

не соответстуют диапазону.

Повторите попытку.");

}

list[i] = new participant(surname, points);

i++;

}

if (!isIncorrect && scanFile.hasNext()) {

System.out.println("Количество участников не совпадает с

записанным в файле размером. Повторите

попытку.");

isIncorrect = true;

}

} catch (Exception e){

System.out.println("Данные выбранного файла не соответствуют условию.

Повторите попытку.");

isIncorrect = true;

}

if (isIncorrect)

path = checkFileInputPath(scan);

} while (isIncorrect);

System.out.println("Данные из файла успешно считаны.");

return list;

}

private static participant[] inputList(Scanner scan) {

int choice;

int listSize;

choice = 0;

listSize = 0;

participant[] list;

choice = inputNum(0, 1, "Если Вы хотите вводить данные в консоль, введите 0.

Если использовать файл, введите 1: ", scan);

if (choice == 0) {

listSize = inputNum(MINSIZE, MAXSIZE, "Введите количество участников (от

3 до 20): ", scan);

list = getListFromConsole(listSize, scan);

}

else {

String finPath;

finPath = checkFileInputPath(scan);

list = getListFromFile(finPath, scan);

}

return list;

}

private static participant[] swapElements(participant[] list, int i) {

participant temp;

temp = list[i];

list[i] = list[i - 1];

list[i - 1] = temp;

return list;

}

private static participant[] sortList(participant[] list) {

for (int i = 1; i < list.length; i++)

for (int j = 1; j < list.length; j++)

if (list[j].points > list[j - 1].points)

list = swapElements(list, j);

else

if (list[j].points == list[j - 1].points) {

int compare;

compare = list[j].surname.compareToIgnoreCase(list[j –

1].surname);

if (compare < 0)

list = swapElements(list, j);

}

return list;

}

private static void outputList(participant[] list, String listName) {

System.out.println("\n" + listName);

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

System.out.println(list[i].surname + " " + list[i].points);

System.out.println();

}

private static String checkFileOutputPath(Scanner scan) {

String path;

boolean isIncorrect;

do {

isIncorrect = false;

System.out.println("Введите путь к файлу, в который нужно записать

результат.");

path = scan.nextLine();

File file = new File(path);

isIncorrect = isFilePathIncorrect(path, file);

if (!isIncorrect & !file.canWrite()) {

System.out.println("Введенный Вами файл доступен только для чтения.

Повторите попытку.");

isIncorrect = true;

}

} while (isIncorrect);

return path;

}

private static void writeResultIntoFile(participant[] list, String path, Scanner

scan) {

boolean isIncorrect;

do {

isIncorrect = false;

try (FileWriter fw = new FileWriter(new File(path))) {

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

fw.write(list[i].surname + " " + list[i].points + "\n");

} catch (Exception e) {

System.out.println("Произошла ошибка. Повторите попытку.");

isIncorrect = true;

}

if (isIncorrect)

path = checkFileOutputPath(scan);

} while (isIncorrect);

System.out.println("Результат записан.");

}

private static void outputResult(participant[] sortedList, Scanner scan) {

int choice;

choice = 0;

choice = inputNum(0, 1, "Если нужно вывести результат в консоль, введите 0.

Если в файл, введите 1: ", scan);

if (choice == 0)

outputList(sortedList, "Отсортированный список:");

else {

String foutPath;

foutPath = checkFileOutputPath(scan);

writeResultIntoFile(sortedList, foutPath, scan);

}

}

public static void main(String[] args) {

Scanner scan = new Scanner(System.in);

participant[] list;

participant[] sortedList;

writeCondition();

list = inputList(scan);

outputList(list, "Введенный список:");

sortedList = sortList(list);

outputResult(sortedList, scan);

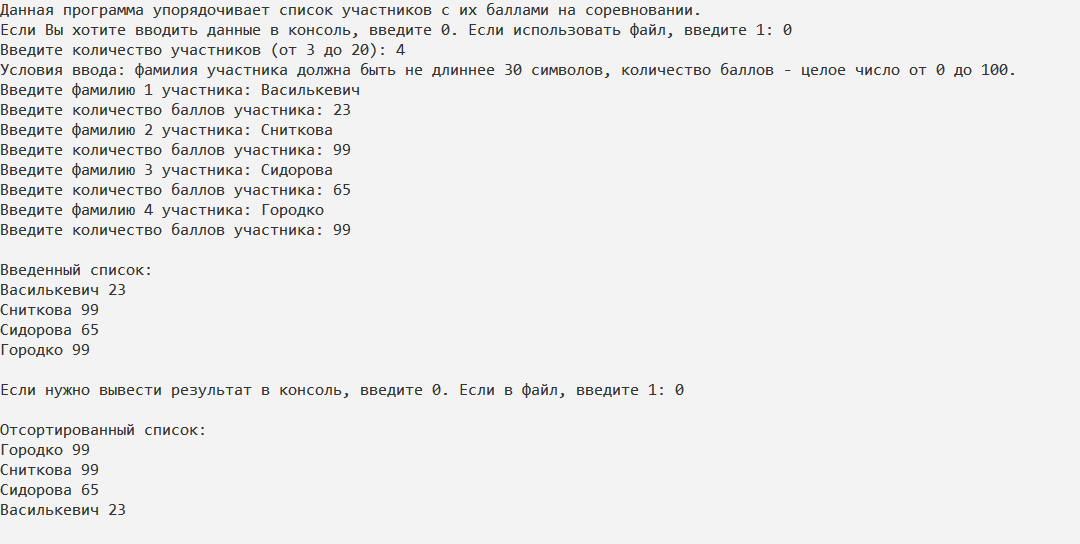
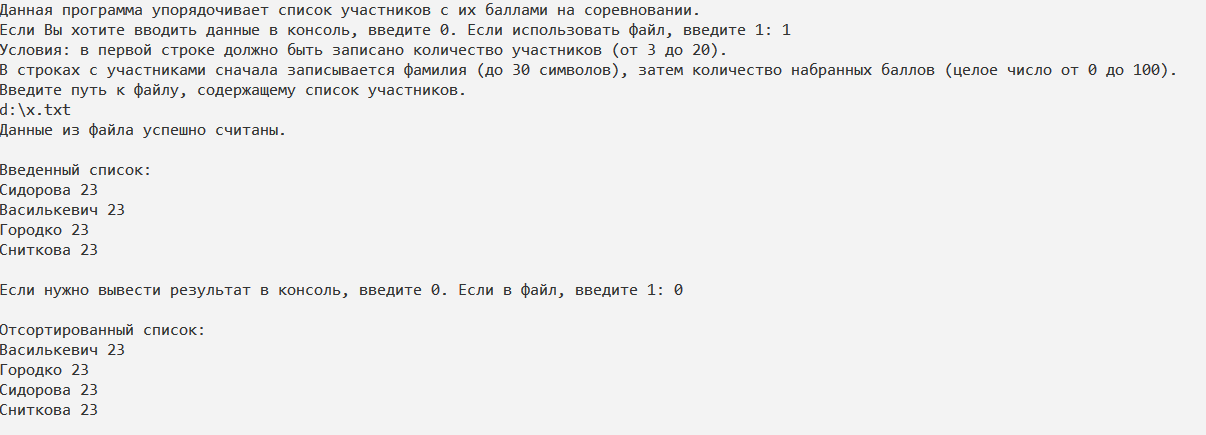
scan.close();

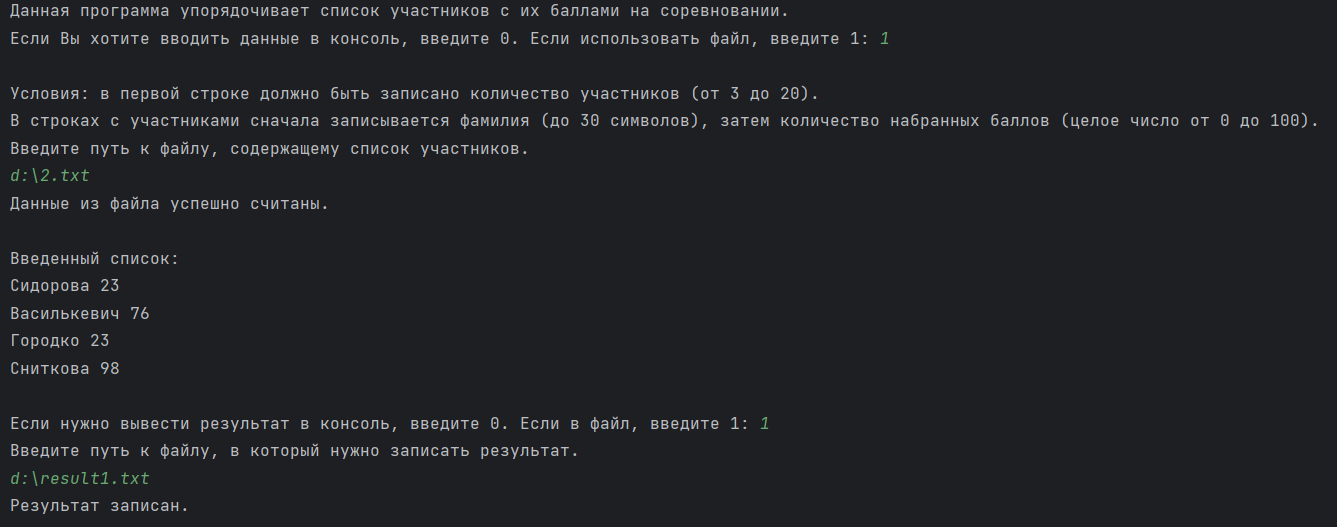
}

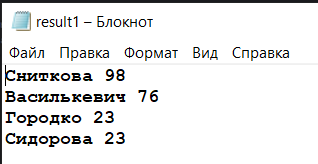
}

**СКРИНШОТЫ:**

**Delphi:**

**C++:****Java:**





**БЛОК-СХЕМА:**









