МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования

**«Вятский государственный университет»**

**(ФГБОУ ВО «ВятГУ»)**

Факультет автоматики и вычислительной техники

Кафедра электронных вычислительных машин

Исследование алгоритмов сортировки данных на внешней памяти

Отчет

Лабораторная работа №9 по дисциплине

«Программирование»

Выполнил студент группы ИВТ-11 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/Кудяшев Я.Ю./

Проверил преподаватель\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/Чистяков Г.А./

Киров 2020

**Цель работы**

Познакомиться с организацией стандартных диалоговых окон, изучить принципы работы с данными на внешней памяти, получить навыки работы с типизированными файлами.

**Задание**

1. Разработать программу для сортировки данных, находящихся на жёстком диске в типизированном файле.
2. Размер сортируемых данных должен быть не менее 1ГБ. Объём потребляемой оперативной памяти не должен превышать 10% от размера исходного файла.
3. Исходный и целевой файлы должны выбираться с помощью стандартных диалоговых окон.
4. Структуру сортируемого файла и ключ сортировки согласовать с преподавателем.
5. Помимо основной программы, требуется реализовать две вспомогательные утилиты – для генерации массива данных и проверки упорядоченности.

**Программа**

Генератор: **unit** Unit1;

{$mode objfpc}{$H+}

**interface**

**uses**

Classes, SysUtils, Forms, Controls, Graphics, Dialogs, StdCtrls, Menus, FileUtil, Grids;

**type**

{ TForm1 }

TForm1 = **class**(TForm)

Button1: TButton;

SaveDialog1: TSaveDialog;

**procedure** Button1Click(Sender: TObject);

**private**

**public**

**end**;

**type**

NBA = **record**

player1: string[100];

player2: string[100];

points: 1..55;

rebounds: 1..23;

assists: 1..12;

steals: 0..4;

blocks: 0..6;

combinedStatictics:3..110;

**end**;

**var**

x:NBA;

n: **array** [1..70] **of** string = ('Aaron','Abraham','Adam','Adrian','Aidan',

'Alan','Albert','Alejandro','Alex','Alexander','Alfred','Andrew','Angel',

'Anthony','Antonio','Ashton','Austin','Benjamin','Bernard','Blake','Brandon',

'Brian','Bruce','Bryan','Cameron','Carl','Carlos','Charles','Christopher',

'Cole','Connor','Caleb','Carter','Chase','Christian','Clifford','Cody',

'Colin','Curtis','Cyrus','Daniel','David','Dennis','Devin','Diego','Dominic',

'Donald','Douglas','Dylan','Jack','Jackson','Jacob','Jaden','Jake','James',

'Jason','Jayden','Jeffery','Jeremiah','Jesse','Jesus','John','Jonathan',

'Jordan','Jose','Joseph','Joshua','Juan','Julian','Justin');

nn: **array** [1..62] **of** string = ('Smith','Johnson','Williams','Jones','Brown',

'Davis','Miller','Wilson','Moore','Taylor','Anderson','Thomas','Jackson',

'White','Harris','Martin','Thompson','Garcia','Martinez','Robinson','Clark',

'Rodriguez','Lewis','Lee','Walker','Hall','Allen','Young','Hernandez','King',

'Wright','Lopez','Hill','Scott','Green','Adams','Baker','Gonzalez','Nelson',

'Carter','Mitchell','Perez','Roberts','Turner','Phillips','Campbell',

'Parker','Evans','Edwards','Collins','Stewart','Sanchez','Morris','Rogers',

'Reed','Cook','Morgan','Bell','Murphy','Bailey','Rivera','Cooper');

i,aa,bb:integer;

Form1: TForm1;

namef:string;

**implementation**

{$R \*.lfm}

{ TForm1 }

**procedure** TForm1.Button1Click(Sender: TObject);

**var**

i,j:integer;

myFile:**file of** NBA;

**begin**

Randomize;

**if** SaveDialog1.Execute **then**

**begin**

namef:=SaveDialog1.FileName;

AssignFile(myFile,namef);

rewrite(myFile);

**for** i:=1 **to** 2560000 **do**

**begin**

aa:= Random(70) + 1;

bb:= Random(62) + 1;

x.player1:= n[aa];

x.player2:= nn[bb];

x.points:= random(55)+1;

x.rebounds:= random(23)+1;

x.assists:= random(12)+1;

x.steals:= random(4);

x.blocks:= random(6);

x.combinedStatictics:=x.points+x.rebounds+x.assists+x.steals+x.blocks;

write(myFile,x);

**end**;

closefile(myFile);

**end**;

**end**;

**end**.

Сортировка: **unit** Unit1;

{$mode objfpc}{$H+}

**interface**

**uses**

Classes, SysUtils, Forms, Controls, Graphics, Dialogs, StdCtrls;

**type**

NBA=**record**

player1: string[100];

player2: string[100];

points: 1..55;

rebounds: 1..23;

assists: 1..12;

steals: 0..4;

blocks: 0..6;

combinedStatictics:3..110;

**end**;

fNBA = **file of** NBA;

fi = **file of** integer;

{ TForm1 }

TForm1 = **class**(TForm)

Button1: TButton;

Button2: TButton;

Button3: TButton;

OpenDialog1: TOpenDialog;

SaveDialog1: TSaveDialog;

**procedure** Button1Click(Sender: TObject);

**procedure** Button2Click(Sender: TObject);

**procedure** Button3Click(Sender: TObject);

**private**

**public**

**end**;

**var**

Form1: TForm1;

inputfilename,outputfilename:string;

input,output:fNBA;

buffer:NBA;

**implementation**

{$R \*.lfm}

{ TForm1 }

**procedure** TForm1.Button1Click(Sender: TObject);

**begin**

**if** opendialog1.Execute **then**

**begin**

inputfilename:=OpenDialog1.filename;

assignfile(input,inputfilename);

**end**;

**end**;

**procedure** TForm1.Button2Click(Sender: TObject);

**begin**

**if** savedialog1.Execute **then**

**begin**

outputfilename:=savedialog1.filename;

**end**;

**end**;

**procedure** TForm1.Button3Click(Sender: TObject);

**var**

i,j, i1, j1:integer;

buffmas:**array**[1..40000] **of** NBA;

a,b:NBA;

fla,flb:boolean;

count:integer;

f1,f2,f3:fNBA;

**procedure** QSort(first,last:integer);

**var**

L,R:integer;

c,X:NBA;

**begin**

**if** first>=last **then Exit**;

**if** first<last **then begin**

X:=buffmas[(first+last) **div** 2];

L:=first; R:=last;

**while** L<=R **do begin**

**while** buffmas[L].combinedStatictics < X.combinedStatictics **do** L:=L+1;

**while** buffmas[R].combinedStatictics > X.combinedStatictics **do** R:=R-1;

**if** L<=R **then begin**

c:=buffmas[L]; buffmas[L]:=buffmas[R]; buffmas[R]:=c;

L:=L+1; R:=R-1;

**end**;

**end**;

**end**;

QSort(first,R);

QSort(L,last);

**end**;

**begin**

reset(input);

count:=64;

**for** i:=1 **to** count **do begin**

assignfile(f1,inttostr(i) + '\_1.dat');

rewrite(f1);

**for** j:=1 **to** 40000 **do**

read(input,buffmas[j]);

QSort(1,40000);

**for** j:=1 **to** 40000 **do begin**

write(f1,buffmas[j]);

**end**;

closefile(f1);

**end**;

closefile(input);

**for** i:=1 **to** 6 **do begin**

**for** j:=1 **to** count **div** 2 **do begin**

assignfile(f1,inttostr(j) + '\_' + inttostr(i) + '.dat');

assignfile(f2,inttostr(count+1-j) + '\_' + inttostr(i) + '.dat');

reset(f1);

reset(f2);

read(f1,a);

read(f2,b);

**if** i=6 **then**

assignfile(f3,outputfilename)

**else**

assignfile(f3,inttostr(j) + '\_' + inttostr(i+1) + '.dat');

rewrite(f3);

fla:=true;

flb:=true;

**while** fla **or** flb **do begin**

**if** (a.combinedStatictics<=b.combinedStatictics) **and** fla **or not**(flb) **then begin**

write(f3,a);

**if** eof(f1) **then** fla:=false

**else** read(f1,a);

**end**

**else if** flb **then begin**

write(f3,b);

**if** eof(f2) **then** flb:=false

**else** read(f2,b);

**end**;

**end**;

closefile(f1);

closefile(f2);

closefile(f3);

**end**;

count:=count **div** 2;

**end**;

**end**;

**end**.

Проверка: **unit** Unit1;

{$mode objfpc}{$H+}

**interface**

**uses**

Classes, SysUtils, Forms, Controls, Graphics, Dialogs, StdCtrls;

**type**

NBA=**record**

player1: string[100];

player2: string[100];

points: 1..55;

rebounds: 1..23;

assists: 1..12;

steals: 0..4;

blocks: 0..6;

combinedStatictics:3..110;

**end**;

fNBA = **file of** NBA;

{ TForm1 }

TForm1 = **class**(TForm)

Button1: TButton;

Button2: TButton;

OpenDialog1: TOpenDialog;

**procedure** Button1Click(Sender: TObject);

**procedure** Button2Click(Sender: TObject);

**private**

**public**

**end**;

**var**

Form1: TForm1;

inputfilename:string;

input:fNBA;

**implementation**

{$R \*.lfm}

{ TForm1 }

**procedure** TForm1.Button1Click(Sender: TObject);

**begin**

**if** opendialog1.Execute **then**

**begin**

inputfilename:=OpenDialog1.filename;

assignfile(input,inputfilename);

**end**;

**end**;

**procedure** TForm1.Button2Click(Sender: TObject);

**var**

i:integer;

fl:boolean;

b1,b2:NBA;

**begin**

fl:=true;

reset(input);

read(input,b1);

**for** i:=2 **to** filesize(input) **do begin**

read(input,b2);

**if** b1.combinedStatictics>b2.combinedStatictics **then begin**

fl:=false;

**break**;

**end**;

b1:=b2;

**end**;

**if** fl **then** showmessage('Файл отсортирован верно')

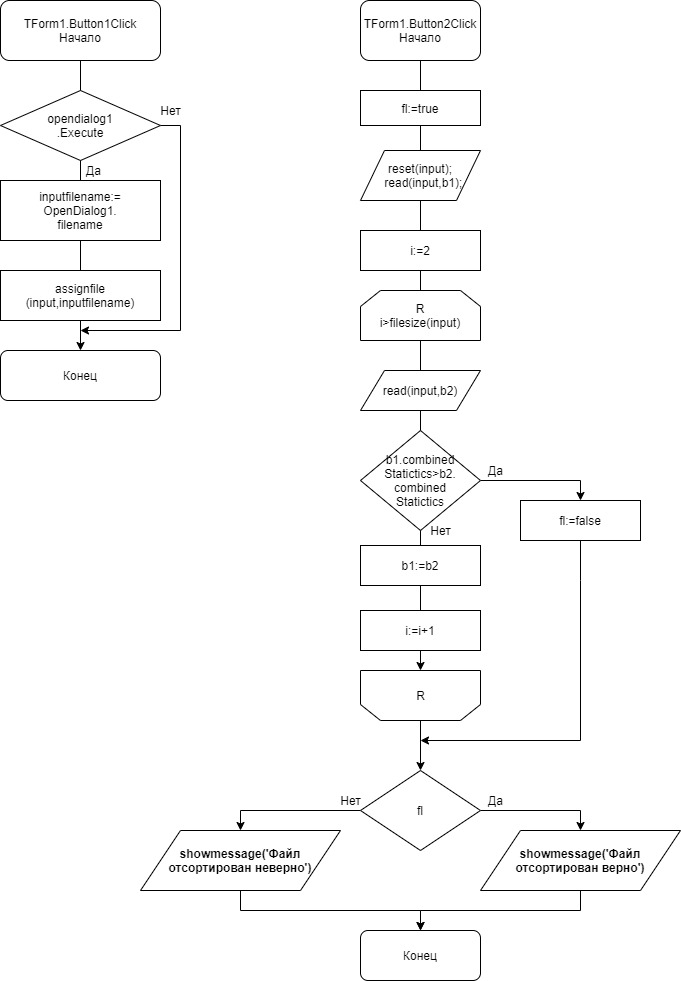
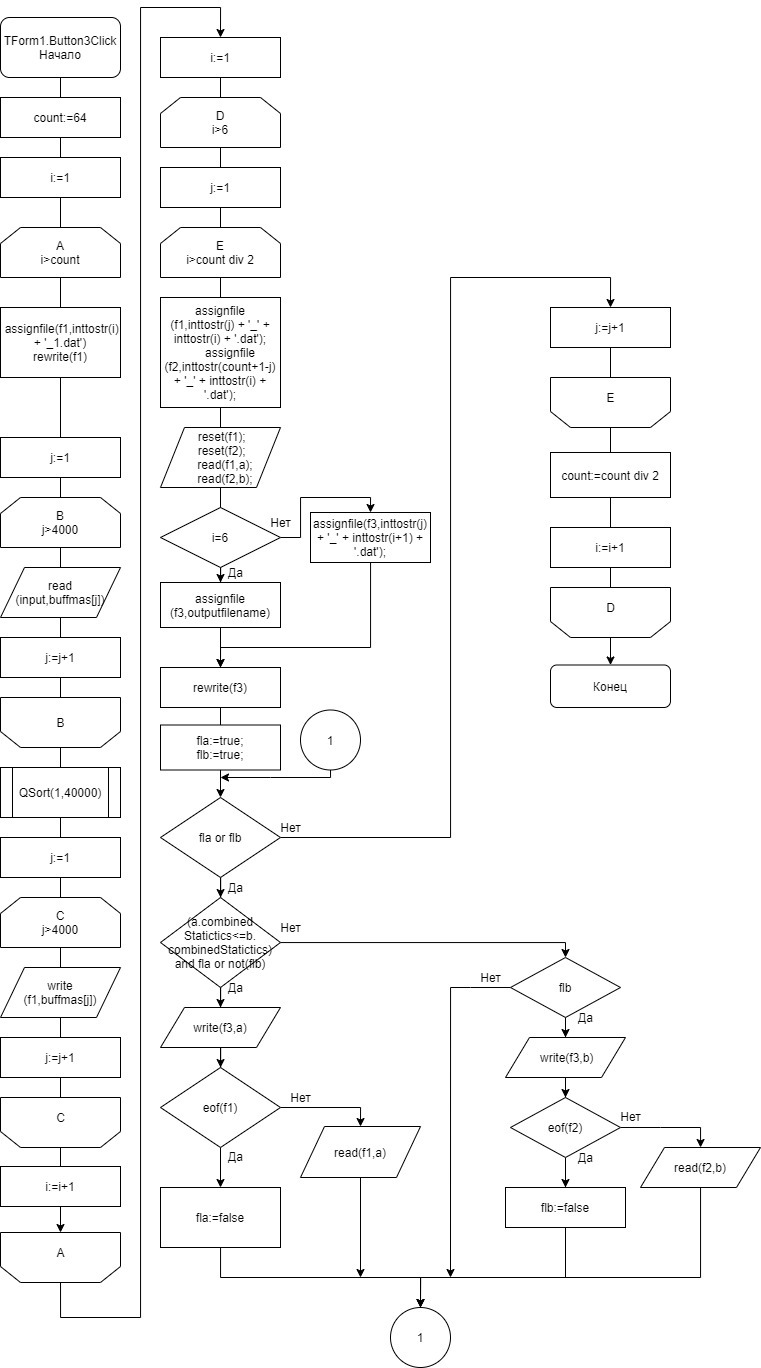
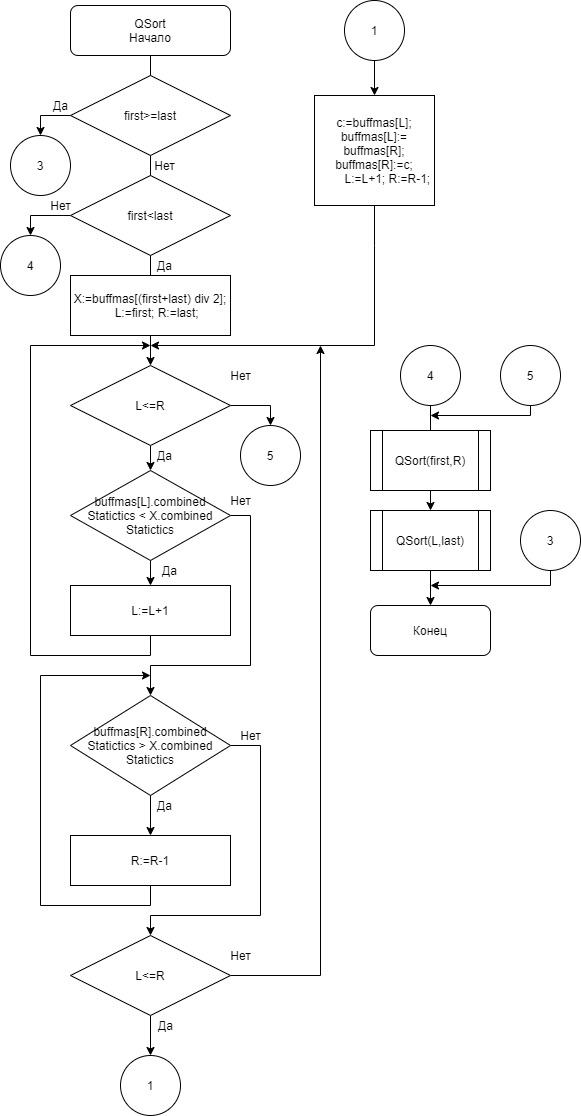
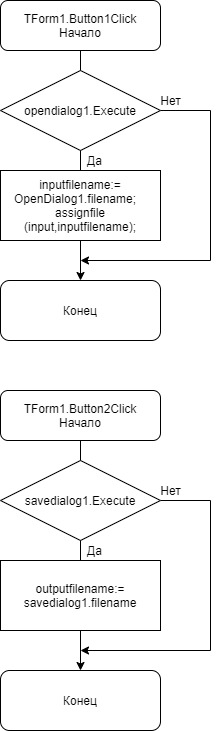
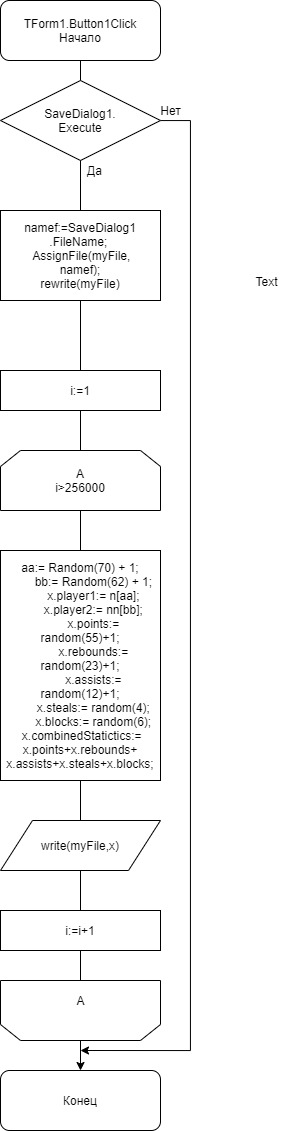
**else** showmessage('Файл отсортирован неверно');

closefile(input);

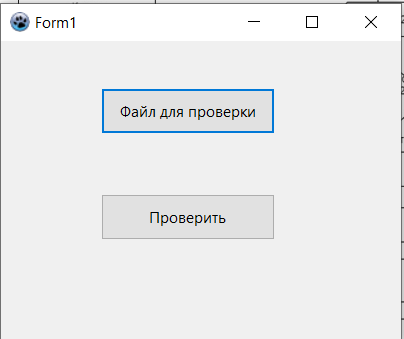
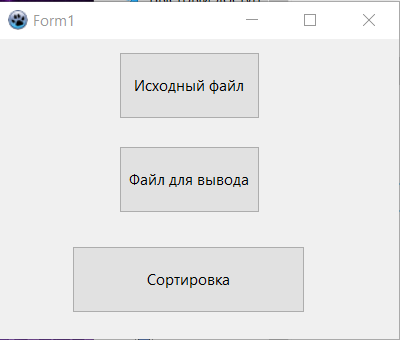
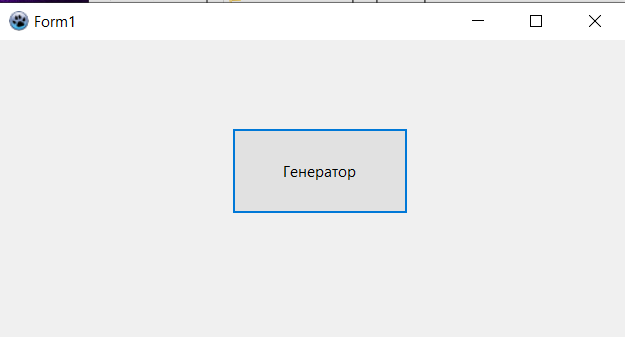
**end**;

**end**.

**Схема**



**Скриншоты программы**



**Вывод**

**Lazarus** — [открытая](https://ru.wikipedia.org/wiki/%D0%9E%D1%82%D0%BA%D1%80%D1%8B%D1%82%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B5_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D0%B5) [среда разработки программного обеспечения](https://ru.wikipedia.org/wiki/%D0%A1%D1%80%D0%B5%D0%B4%D0%B0_%D1%80%D0%B0%D0%B7%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%BA%D0%B8_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B3%D0%BE_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D1%8F) на языке [**Object Pascal**](https://ru.wikipedia.org/wiki/Object_Pascal)для компилятора [**Free Pascal**](https://ru.wikipedia.org/wiki/Free_Pascal) (часто используется сокращение **FPC —**[**Free Pascal Compiler**](https://ru.wikipedia.org/wiki/Free_Pascal), бесплатно распространяемый компилятор языка программирования Pascal). Интегрированная среда разработки предоставляет возможность [кроссплатформенной](https://ru.wikipedia.org/wiki/%D0%9A%D1%80%D0%BE%D1%81%D1%81%D0%BF%D0%BB%D0%B0%D1%82%D1%84%D0%BE%D1%80%D0%BC%D0%B5%D0%BD%D0%BD%D0%BE%D1%81%D1%82%D1%8C) разработки приложений в [**Delphi**](https://ru.wikipedia.org/wiki/Delphi_(%D1%81%D1%80%D0%B5%D0%B4%D0%B0_%D1%80%D0%B0%D0%B7%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%BA%D0%B8))-подобном окружении.

В ходе выполнения данной лабораторной работы были освоены азы работы в среде программирования **Lazarus**. Были созданы 3 программы: для генерации массива, который был придуман лично каждым из студентов, для сортировки данного массива и для проверки отсортированности. Главной особенностью данной лабораторной работы является то, что сортировка должна была проходить на внешней памяти, т.е. без серьезных затрат оперативной памяти. Именно для достижения наименьших затрат памяти была применена сортировка на внешней памяти.

Меню каждой программы не такое сложное и выглядит довольно просто, зато в нём есть все необходимые кнопки для манипуляций.

Суть именно реализованной программы в том, чтобы отсортировать сумму характеристик, привязанных к определённому игроку. Игроки генерируются случайным образом: из заданного массива имён и фамилий. Их статистика также генерируется рандомным образом в заданных пределах. В итоге выявляются игроки с наибольшей суммой показателей. На личном ПК проведение всех процедур в сумме занимает около 2 минут.