Министерство образования и науки Российской Федерации Федеральное агентство по образованию Федеральное государственное бюджетное образовательное учреждение высшего образования «Вятский государственный университет»

Факультет автоматики и вычислительной техники

Кафедра электронных вычислительных машин

Лабораторная работа №9

по курсу «Программирование»

**Исследование алгоритмов сортировки на внешней памяти**

Выполнил студент группы ИВТ-11\_\_\_\_\_\_\_\_\_\_\_\_\_\_/Птахова А.М/

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

Киров 2021

**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 жестком диске в типизированном файле.

2. Размер сортируемых данных должен быть не менее 1 Гб. Объем 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.

**Структура:**

TRecord = record

Surname: string[30];

Name: string[30];

Picture: string[30];

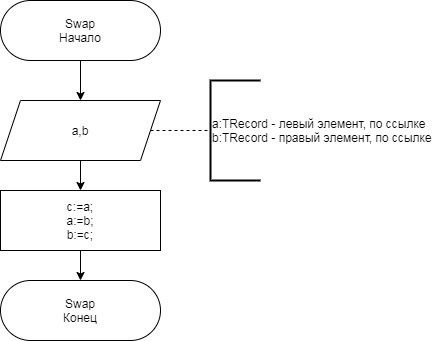
Price: LongWord;

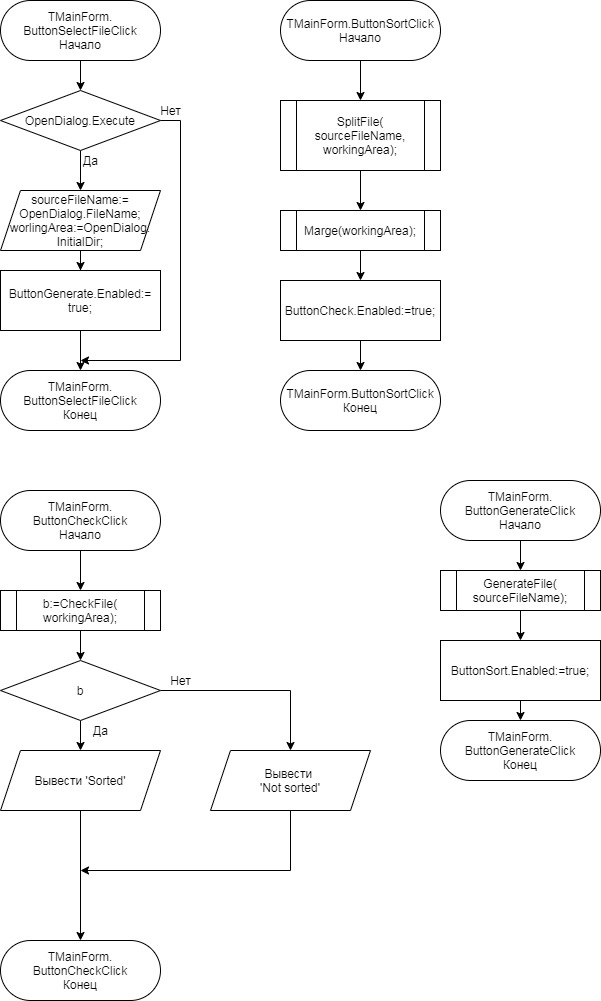
Years: LongWord;

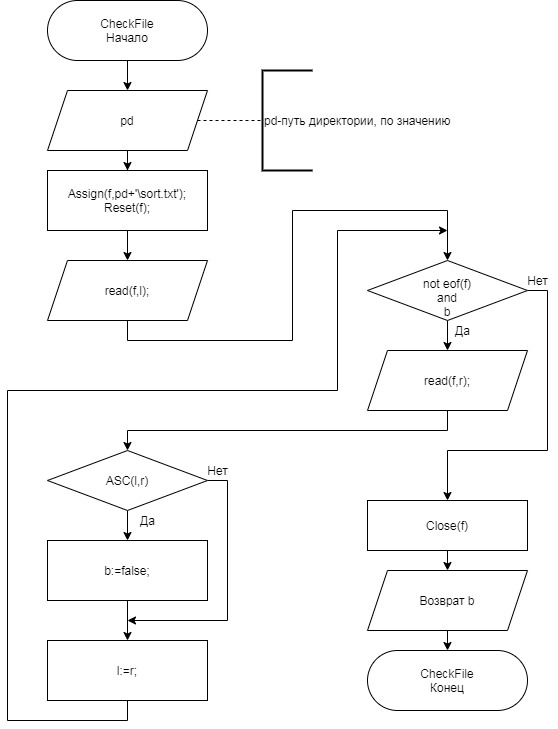
Country: string[20];

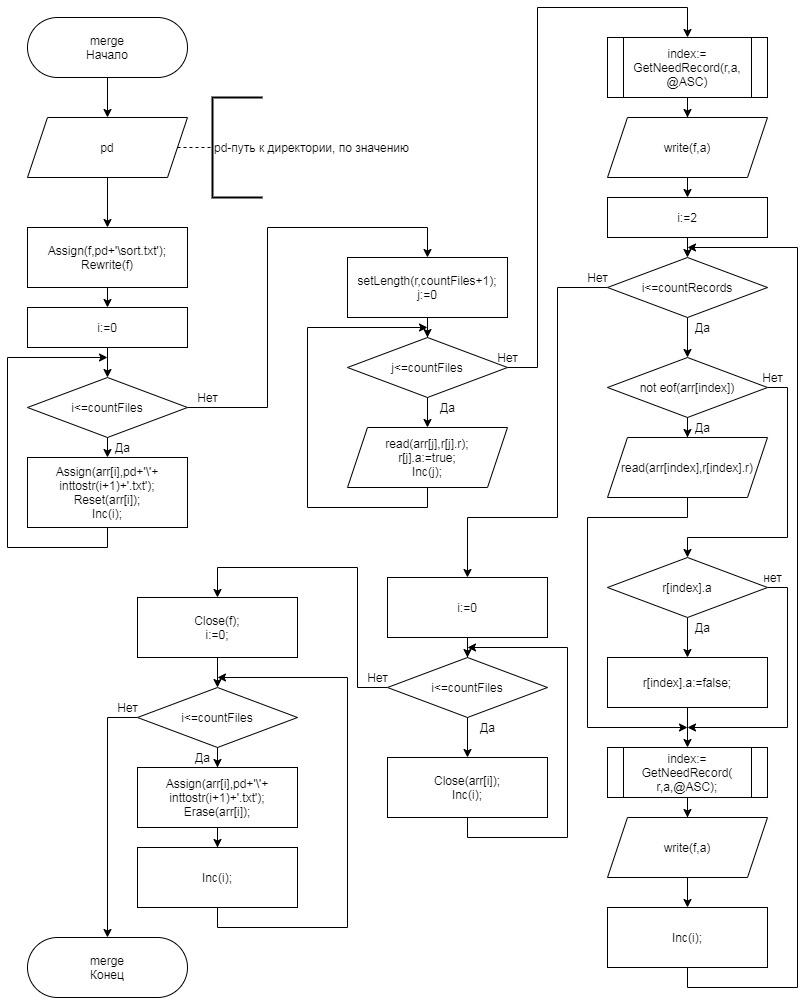
End;

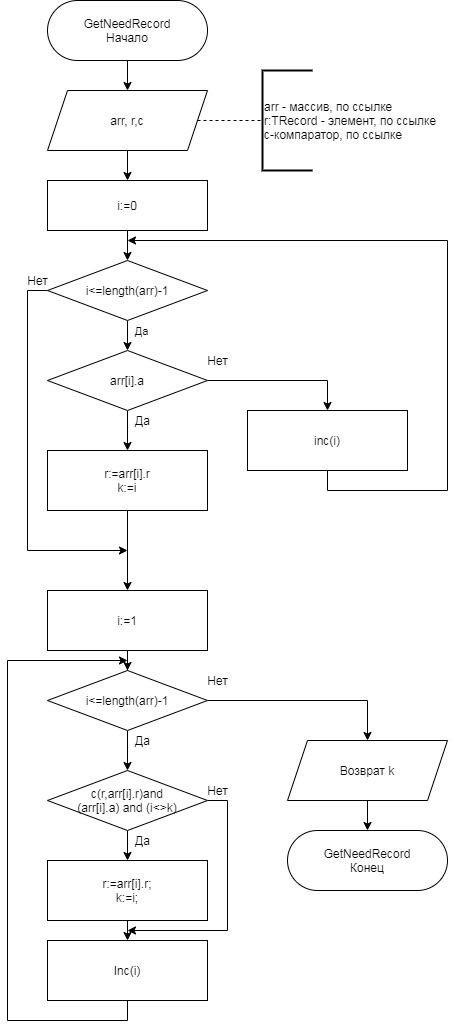
**Схемы алгоритмов:**

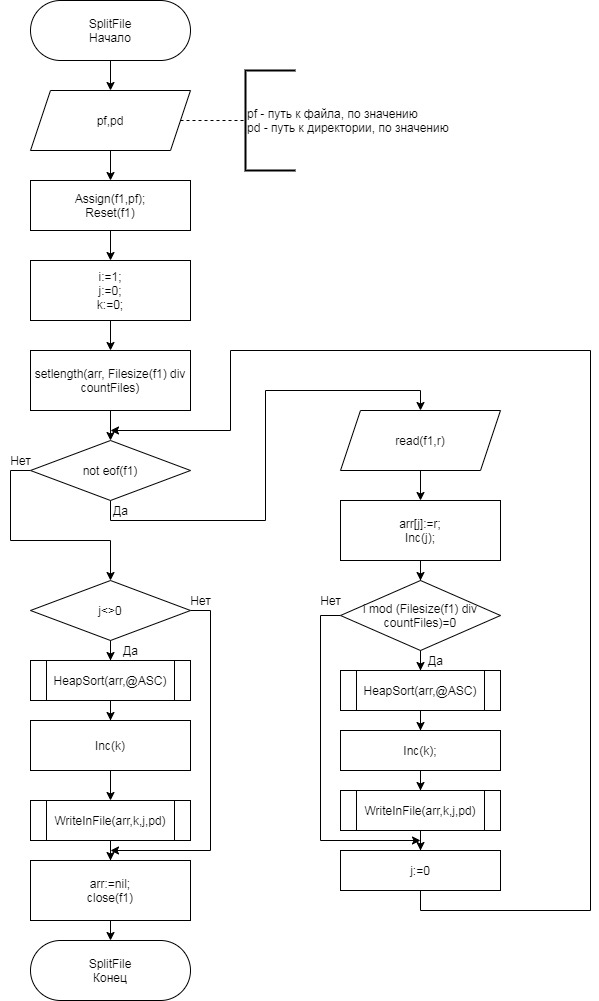
****

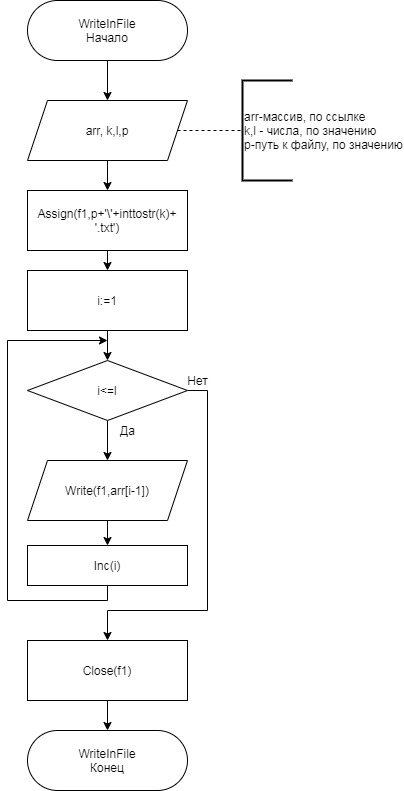
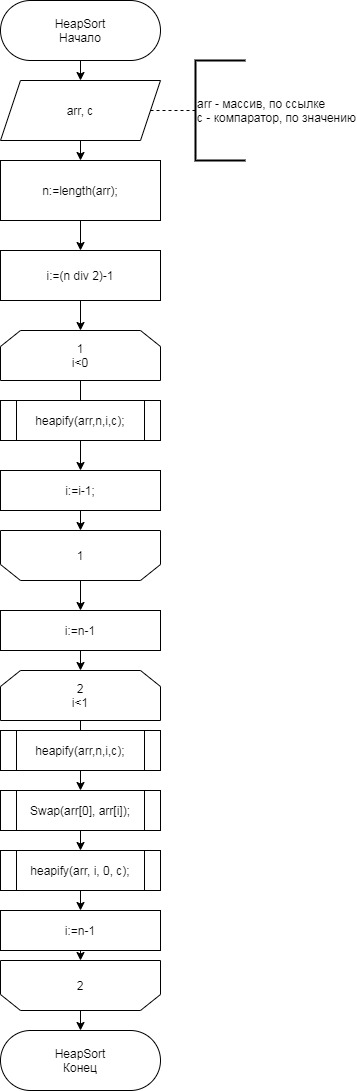
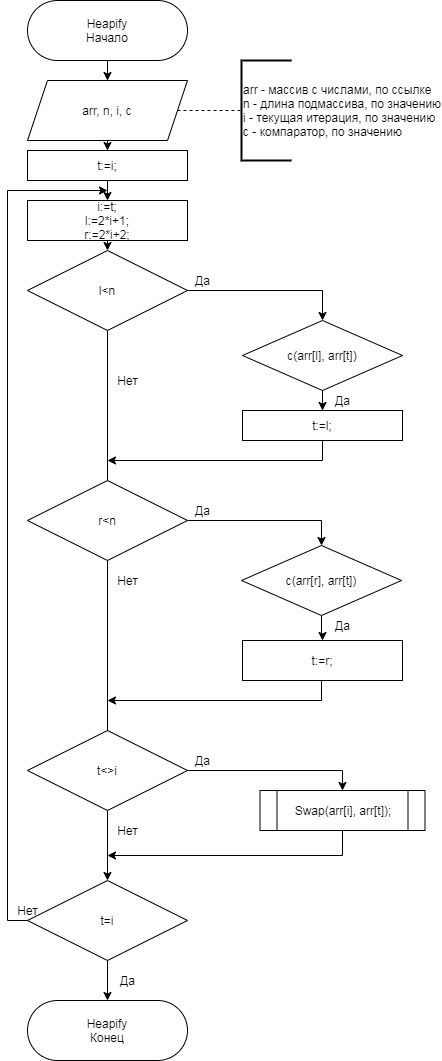
****

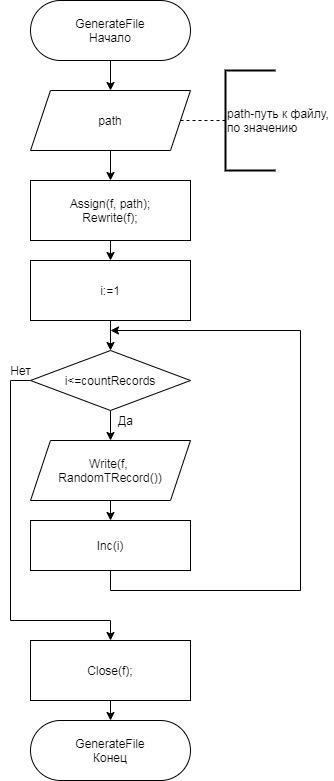
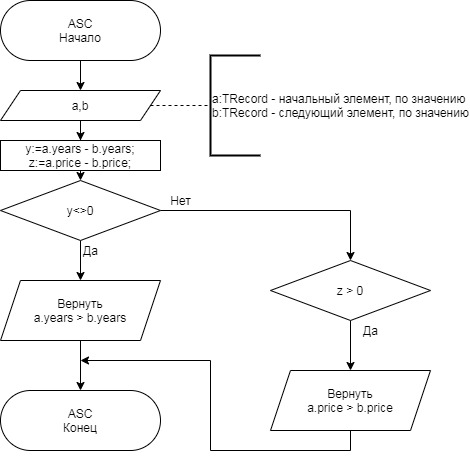
****

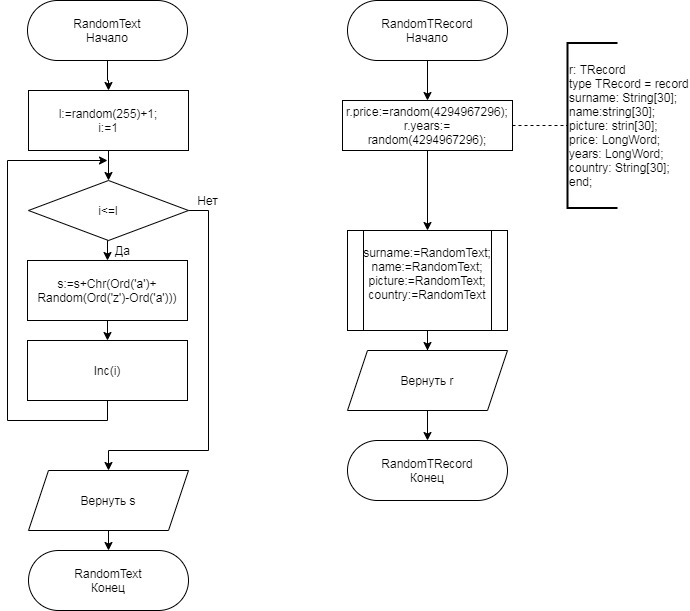
****

****

****

****

** **

****

**Листинг программы:**

unit Main;

{$mode objfpc}{$H+}

interface

uses

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

Generate;

type

{ TForm1 }

{ TMainForm }

TMainForm = class(TForm)

ButtonSort: TButton;

ButtonSelectFile: TButton;

ButtonCheck: TButton;

LabelCheck: TLabel;

OpenDialog: TOpenDialog;

ButtonGenerate: TButton;

procedure ButtonCheckClick(Sender: TObject);

procedure ButtonGenerateClick(Sender: TObject);

procedure ButtonSelectFileClick(Sender: TObject);

procedure ButtonSortClick(Sender: TObject);

private

public

sourceFileName: String;

workingArea: String;

end;

var

MainForm: TMainForm;

implementation

{$R \*.lfm}

{ TForm1 }

{ TMainForm }

procedure TMainForm.ButtonSelectFileClick(Sender: TObject);

begin

if OpenDialog.Execute then

begin

sourceFileName := OpenDialog.FileName;

workingArea := OpenDialog.InitiaLDir;

ButtonGenerate.Enabled:=true;

end;

end;

procedure TMainForm.ButtonSortClick(Sender: TObject);

begin

SplitFile(sourceFileName,workingArea);

Merge(workingArea);

ButtonCheck.Enabled:=true;

end;

procedure TMainForm.ButtonCheckClick(Sender: TObject);

var b: boolean;

begin

b:=CheckFile(workingArea);

if b then

begin

labelCheck.caption:='Sorted';

end

else

begin

labelCheck.caption:='Not sorted';

end;

end;

procedure TMainForm.ButtonGenerateClick(Sender: TObject);

begin

GenerateFile(sourceFileName);

ButtonSort.Enabled:=true;

end;

end.

unit Sort;

{$mode objfpc}{$H+}

interface

uses

Classes, SysUtils, Generate;

const

countFiles=6;

type records=array of TRecord;

comparator=function(a,b:TRecord):boolean;

URecord=record

r:TRecord;//сама запись

a:boolean;//актуальность

end;

urecords=array of URecord;

procedure HeapSort(var arr:records; c:comparator);

procedure SplitFile(pf:string;pd:string);

procedure merge(pd:string);

function CheckFile(pd:string):boolean;

implementation

function ASC(a,b:TRecord):boolean;

var y,z:LongWord;

begin

y:=a.years - b.years;

z:=a.price - b.price;

if y <> 0 then

begin

Exit(a.years > b.years);

end

else if z > 0 then

begin

Exit(a.price < b.price);

end ;

{else if a.price <> b.price then

begin

Exit(a.price < b.price);

end;}

end;

procedure Swap(var a,b:TRecord);

var c: TRecord;

begin

c:=a;

a:=b;

b:=c;

end;

//Сортировка пиромидой

procedure heapify(var arr:records; n,i:longint; c:comparator);

var l,r,t:longint;

begin

t:=i;

repeat

i:=t;

l:=2\*i+1;

r:=2\*i+2;

if l < n then begin

if c(arr[l], arr[t]) then

begin

t:=l;

end;

end;

if r < n then begin

if c(arr[r], arr[t]) then

begin

t:=r;

end;

end;

if t<>i then begin

Swap(arr[i], arr[t]);

//heapify(arr,n,t,c);

end;

until t=i;

end;

procedure HeapSort(var arr:records; c:comparator);

var n,i:longint;

begin

n:=length(arr);

for i:=(n div 2)-1 downto 0 do

begin

heapify(arr,n,i,c);

end;

for i:=n-1 downto 1 do

begin

Swap(arr[0],arr[i]);

heapify(arr,i,0,c);

end;

end;

procedure WriteInFile(var arr:records;k,l:longint;p:string);

var f1: File of TRecord;

i: integer;

begin

Assign(f1,p+'\'+inttostr(k)+'.txt');

Rewrite(f1);

for i:=1 to l do

begin

Write(f1,arr[i-1]);

end;

Close(f1);

end;

procedure SplitFile(pf:string;pd:string);

var i,j,k: longint;

f1: File of TRecord;

arr: records;

r:TRecord;

begin

Assign(f1,pf);

Reset(f1);

i:=1;

j:=0;//количество элементов в массиве

k:=0;//номер файла

setlength(arr,Filesize(f1) div countFiles);

while not eof(f1) do begin

read(f1,r);

arr[j]:=r;

Inc(j);

if i mod (Filesize(f1) div countFiles)=0 then

begin

HeapSort(arr,@ASC);

Inc(k);

WriteInFile(arr,k,j,pd);

j:=0;

//arr:=nil;

end;

Inc(i);

end;

if j<>0 then begin

HeapSort(arr,@ASC);

Inc(k);

WriteInFile(arr,k,j,pd);

end;

arr:=nil;

close(f1);

end;

function GetNeedRecord(var arr:urecords;var r:TRecord;c:comparator):integer;

var i,k:integer;

begin

for i:=0 to length(arr)-1 do

begin

if arr[i].a then

begin

r:=arr[i].r;

k:=i;

break;

end;

end;

for i:=1 to length(arr)-1 do

begin

if c(r,arr[i].r) and (arr[i].a) and (i<>k) then

begin

r:=arr[i].r;

k:=i;

end;

end;

GetNeedRecord:=k;

end;

procedure merge(pd:string{G:TGenerateForm});

var arr:array[0..countFiles] of File of TRecord;

r:Urecords;

a:TRecord;

f:File of TRecord;

i,j,k,index:longint;

begin

Assign(f,pd+'\sort.txt');

Rewrite(f);

for i:=0 to countFiles do

begin

Assign(arr[i],pd+'\'+inttostr(i+1)+'.txt');

Reset(arr[i]);

end;

setlength(r,countFiles+1);

for j:=0 to countFiles do

begin

read(arr[j],r[j].r);

r[j].a:=true;

end;

index:=GetNeedRecord(r,a,@ASC);

write(f,a);

for i:=2 to countRecords do

begin

if not eof(arr[index]) then

begin

read(arr[index],r[index].r);

end

else if r[index].a then

begin

r[index].a:=false;

end;

index:=GetNeedRecord(r,a,@ASC);

write(f,a);

end;

for i:=0 to countFiles do

begin

Close(arr[i]);

end;

Close(f);

for i:=0 to countFiles do

begin

Assign(arr[i],pd+'\'+inttostr(i+1)+'.txt');

Erase(arr[i]);

end;

end;

function CheckFile(pd:string{G:TGenerateForm}):boolean;

var b:boolean = true;

f: file of Trecord;

l,r:Trecord;

begin

Assign(f,pd+'\sort.txt');

Reset(f);

read(f,l);

While not eof(f) and b do

begin

read(f,r);

if ASC(l,r) then

begin

b:=false;

end;

l:=r;

end;

Close(f);

Exit(b);

end;

end.

unit Generate;

{$mode objfpc}{$H+}

interface

uses

Classes, SysUtils{, statusunit};

type

TRecord = record

surname: string[30];

name: string[30];

picture:string[30] ;//+

price: LongWord; //+

years: LongWord;//+

country: string[20];

end;

const

countRecords = 2033602;

var f: File of TRecord;

procedure GenerateFile(path:string);

implementation

function RandomText:string;

var

l,i:integer;

s:string='';

begin

l:=random(255)+1;

for i:=1 to l do

begin

s:=s+Chr(Ord('a') + Random(Ord('z') - Ord('a')));

end;

RandomText:=s;

end;

function RandomTRecord:TRecord;

var

r:TRecord;

begin

with r do

begin

surname:=RandomText;

name:=RandomText;

picture:=RandomText;

price:=random(4294967296);

years:=random(4294967296);

country:=RandomText;

end;

Exit(r);

end;

procedure GenerateFile(path:string);

var

i:integer;

begin

Assign(f, path);

Rewrite(f);

for i:=1 to countRecords do

begin

Write(f, RandomTRecord());

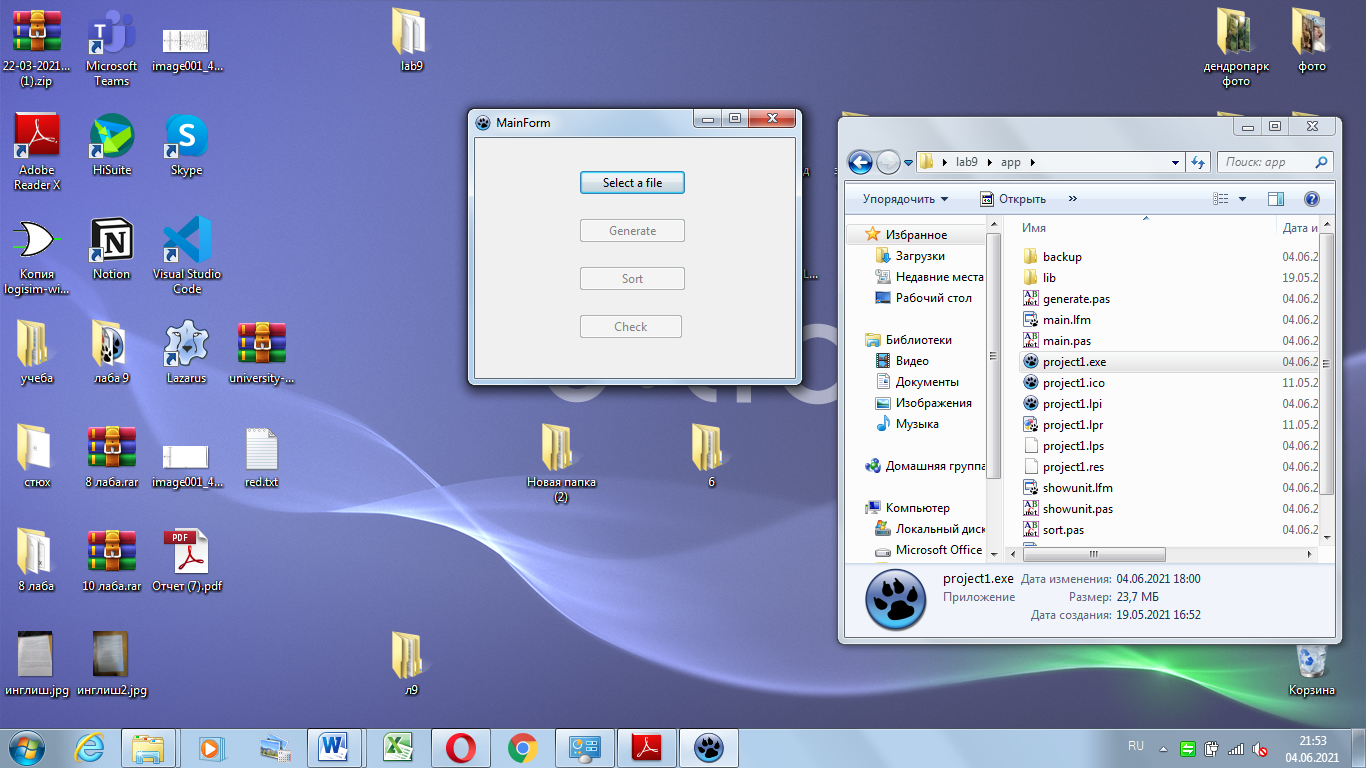
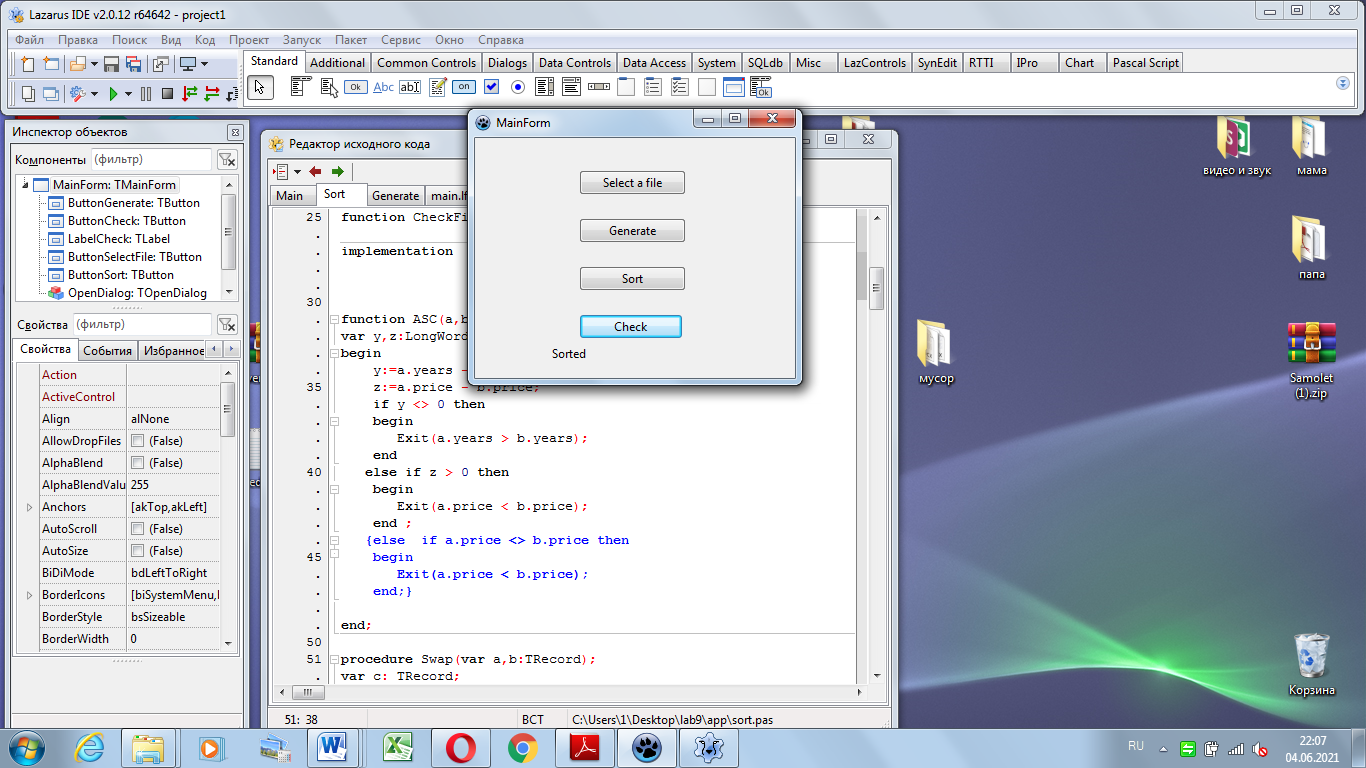
end;

Close(f);

end;

end.

**Экранные формы:**

**Вывод:** В ходе выполнения работы была изучена организация стандартных диалоговых окон, изучены принципы работы с данными на внешней памяти, получены навыки работы с типизированными файлами.