Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Вятский государственный университет»

Колледж ВятГУ

**ОТЧЕТ**

**ПО ДОМАШНЕЙ КОНТРОЛЬНОЙ РАБОТЕ №8**

**«ОСНОВЫ СОБЫТИЙНО-ОРИЕНТИРОВАННОГО ПРОГРАММИРОВАНИЯ»**

**ПО МДК 05.02 РАЗРАБОТКА КОДА ИНФОРМАЦИОННЫХ СИСТЕМ**

Выполнил: студент учебной группы

ИСПк-203-52-00

Исакова Кристина Валерьевна

Преподаватель:

Сергеева Елизавета Григорьевна

Киров

2023

**Цель работы:** получить представление об организации хранения и использования информации посредством типизированных файлов, закрепить навыки создания пользовательского интерфейса.

**Задание:**

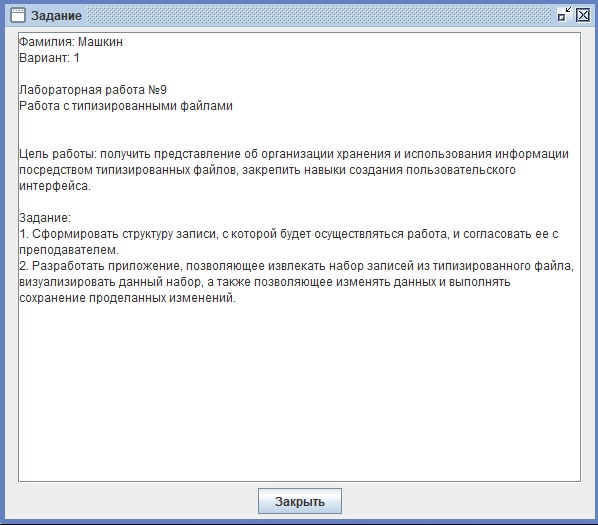


Рисунок 1 – Задание

**Код программы:**

unit Unit1;

{$mode objfpc}{$H+}

interface

uses

Classes, SysUtils, Forms, Controls, Graphics, Dialogs, ExtCtrls, Buttons, Unit2,

Grids;

type

Contacts = record

Name: string[100];

Telephon: string[20];

Note: string[20];

end; //record

{ TfMain }

TfMain = class(TForm)

bAdd: TSpeedButton;

bEdit: TSpeedButton;

bDel: TSpeedButton;

bSort: TSpeedButton;

SG: TStringGrid;

procedure bAddClick(Sender: TObject);

procedure bDelClick(Sender: TObject);

procedure bEditClick(Sender: TObject);

procedure bSortClick(Sender: TObject);

procedure FormClick(Sender: TObject);

procedure FormClose(Sender: TObject; var CloseAction: TCloseAction);

procedure FormCreate(Sender: TObject);

procedure Image1Click(Sender: TObject);

private

public

end;

var

fMain: TfMain;

adres: string; //адрес, откуда запущена программа

implementation

{$R \*.lfm}

{ TfMain }

procedure TfMain.bAddClick(Sender: TObject);

begin

//очищаем поля, если там что-то есть:

fEdit.eName.Text:= '';

fEdit.eTelephone.Text:= '';

//устанавливаем ModalResult редактора в mrNone:

fEdit.ModalResult:= mrNone;

//теперь выводим форму:

fEdit.ShowModal;

//если пользователь ничего не ввел - выходим:

if (fEdit.eName.Text= '') or (fEdit.eTelephone.Text= '') then exit;

//если пользователь не нажал "Сохранить" - выходим:

if fEdit.ModalResult <> mrOk then exit;

//иначе добавляем в сетку строку, и заполняем её:

SG.RowCount:= SG.RowCount + 1;

SG.Cells[0, SG.RowCount-1]:= fEdit.eName.Text;

SG.Cells[1, SG.RowCount-1]:= fEdit.eTelephone.Text;

SG.Cells[2, SG.RowCount-1]:= fEdit.CBNote.Text;

end;

procedure TfMain.FormClick(Sender: TObject);

begin

end;

procedure TfMain.FormClose(Sender: TObject; var CloseAction: TCloseAction);

var

MyCont: Contacts; //для очередной записи

f: file of Contacts; //файл данных

i: integer; //счетчик цикла

begin

//если строки данных пусты, просто выходим:

if SG.RowCount = 1 then exit;

//иначе открываем файл для записи:

try

AssignFile(f, adres + 'telephones.dat');

Rewrite(f);

//теперь цикл - от первой до последней записи сетки:

for i:= 1 to SG.RowCount-1 do begin

//получаем данные текущей записи:

MyCont.Name:= SG.Cells[0, i];

MyCont.Telephon:= SG.Cells[1, i];

MyCont.Note:= SG.Cells[2, i];

//записываем их:

Write(f, MyCont);

end;

finally

CloseFile(f);

end;

end;

procedure TfMain.FormCreate(Sender: TObject);

var

MyCont: Contacts; //для очередной записи

f: file of Contacts; //файл данных

i: integer; //счетчик цикла

begin

//сначала получим адрес программы:

adres:= ExtractFilePath(ParamStr(0));

//настроим сетку:

SG.Cells[0, 0]:= 'Имя Персонажа';

SG.Cells[1, 0]:= 'Способности Персонажа';

SG.Cells[2, 0]:= 'Сезон';

SG.ColWidths[0]:= 200;

SG.ColWidths[1]:= 200;

SG.ColWidths[2]:= 150;

//если файла данных нет, просто выходим:

if not FileExists(adres + 'telephones.dat') then exit;

//иначе файл есть, открываем его для чтения и

//считываем данные в сетку:

try

AssignFile(f, adres + 'telephones.dat');

Reset(f);

//теперь цикл - от первой до последней записи сетки:

while not Eof(f) do begin

//считываем новую запись:

Read(f, MyCont);

//добавляем в сетку новую строку, и заполняем её:

SG.RowCount:= SG.RowCount + 1;

SG.Cells[0, SG.RowCount-1]:= MyCont.Name;

SG.Cells[1, SG.RowCount-1]:= MyCont.Telephon;

SG.Cells[2, SG.RowCount-1]:= MyCont.Note;

end;

finally

CloseFile(f);

end;

end;

procedure TfMain.Image1Click(Sender: TObject);

begin

end;

procedure TfMain.bEditClick(Sender: TObject);

begin

//если данных в сетке нет - просто выходим:

if SG.RowCount = 1 then exit;

//иначе записываем данные в форму редактора:

fEdit.eName.Text:= SG.Cells[0, SG.Row];

fEdit.eTelephone.Text:= SG.Cells[1, SG.Row];

fEdit.CBNote.Text:= SG.Cells[2, SG.Row];

//устанавливаем ModalResult редактора в mrNone:

fEdit.ModalResult:= mrNone;

//теперь выводим форму:

fEdit.ShowModal;

//сохраняем в сетку возможные изменения,

//если пользователь нажал "Сохранить":

if fEdit.ModalResult = mrOk then begin

SG.Cells[0, SG.Row]:= fEdit.eName.Text;

SG.Cells[1, SG.Row]:= fEdit.eTelephone.Text;

SG.Cells[2, SG.Row]:= fEdit.CBNote.Text;

end;

end;

procedure TfMain.bDelClick(Sender: TObject);

begin

//если данных нет - выходим:

if SG.RowCount = 1 then exit;

//иначе выводим запрос на подтверждение:

if MessageDlg('Требуется подтверждение',

'Вы действительно хотите удалить персонажа "' +

SG.Cells[0, SG.Row] + '"?',

mtConfirmation, [mbYes, mbNo, mbIgnore], 0) = mrYes then

SG.DeleteRow(SG.Row);

end;

procedure TfMain.bSortClick(Sender: TObject);

begin

//если данных в сетке нет - просто выходим:

if SG.RowCount = 1 then exit;

//иначе сортируем список:

SG.SortColRow(true, 0);

end;

end.

**Результаты выполнения программы:**

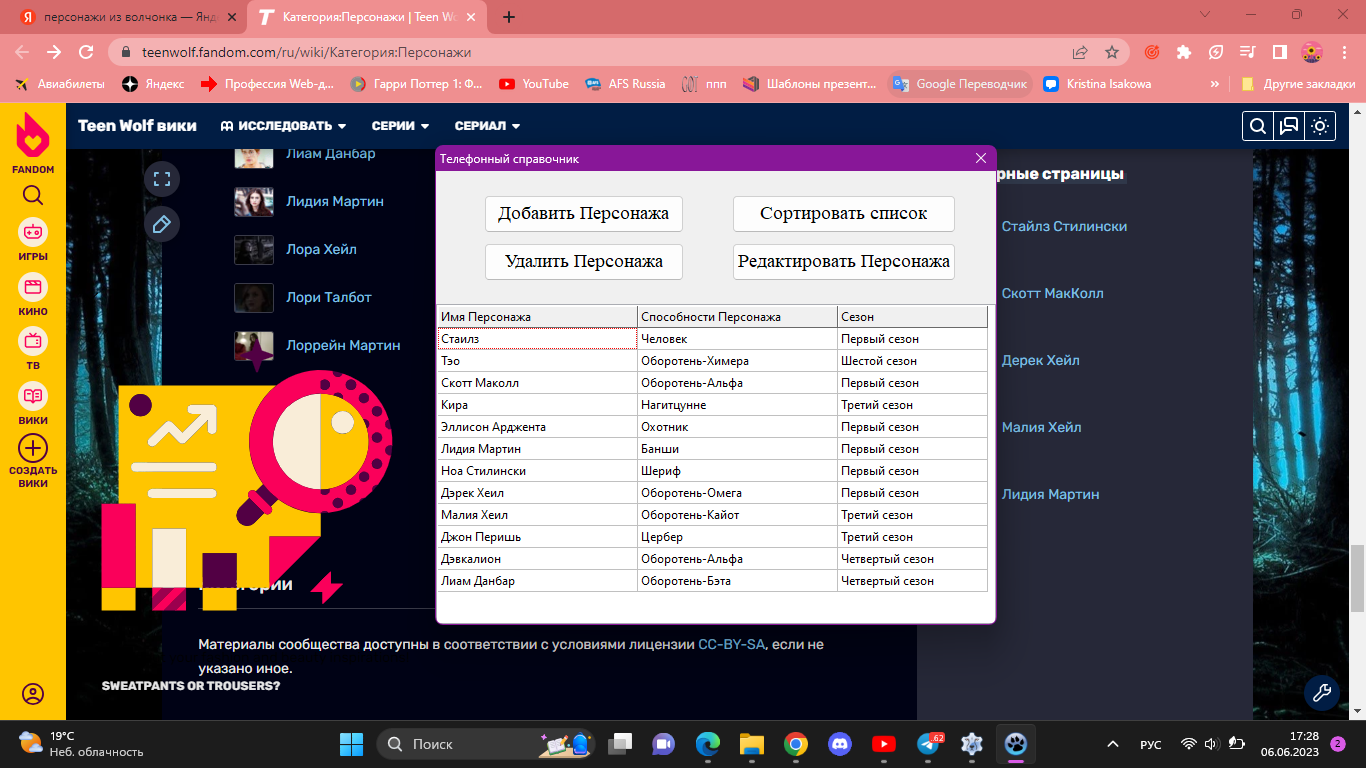
****

Рисунок 2 – Приложение Персонажи TeenWolf

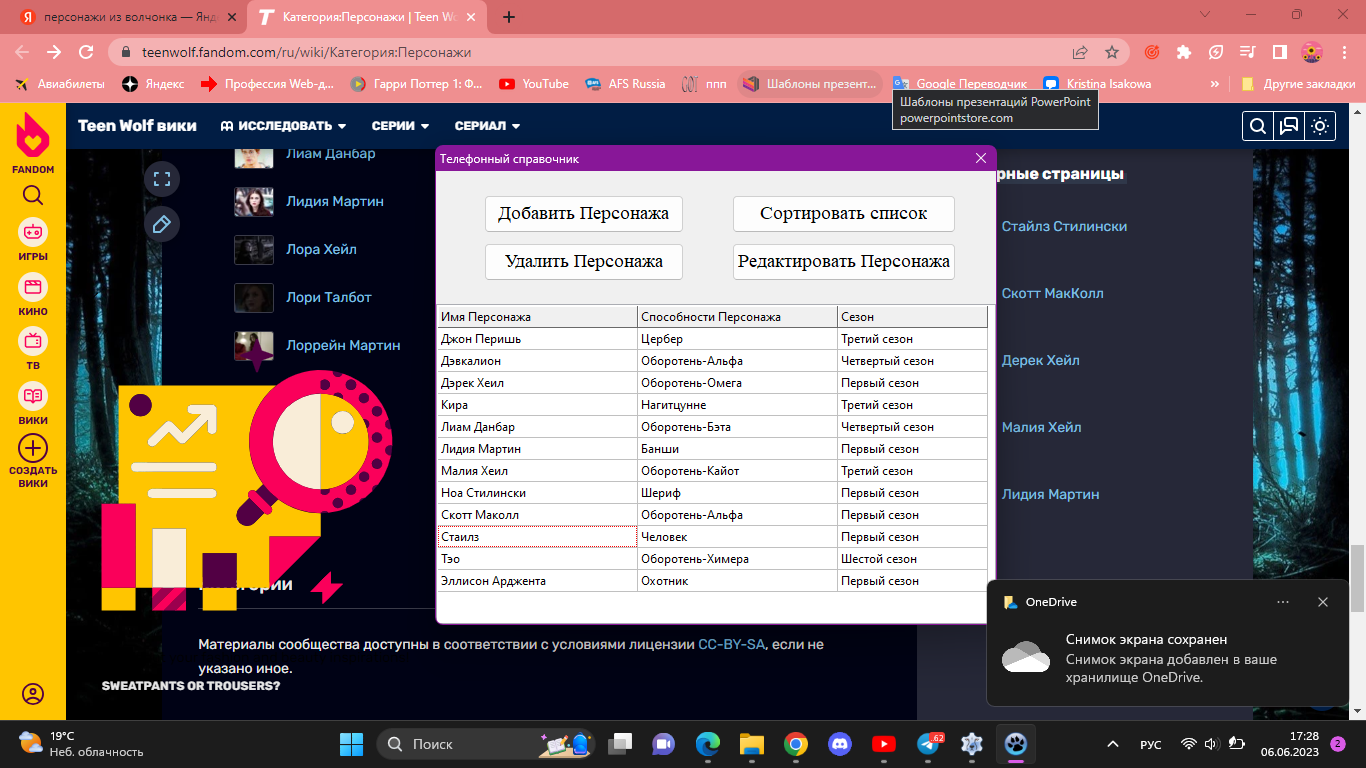
****

Рисунок 3 – Приложение Персонажи TeenWolf

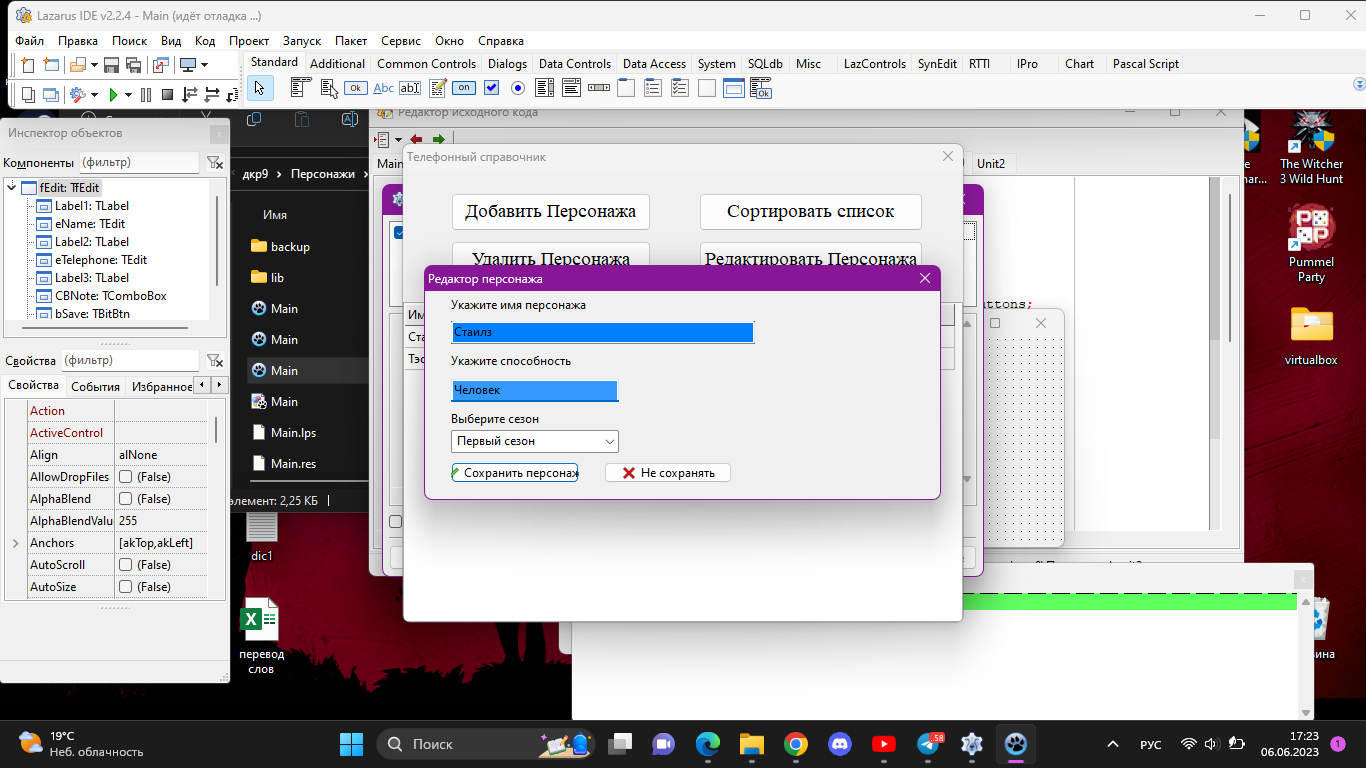


Рисунок 4 – Приложение Персонажи TeenWolf

Вывод: при выполнении данной домашней контрольной работы, мы получили представление об организации хранения и использования информации посредством типизированных файлов, закрепили навыки создания пользовательского интерфейса.

**Вывод**

В ходе выполнения домашней контрольной работы № 9 была написана программа «Персонажи TeenWolf». Необходимо было вспомнить базовые конструкции языка программирования Pascal и Lzarus. В коде программы были использованы: процедуры. Процедуры представляет собой последовательность операторов, которая имеет имя, список параметров и может быть вызвана из различных частей программы.

При выполнении данной домашней контрольной работы, мы получили представление об организации хранения и использования информации посредством типизированных файлов, закрепили навыки создания пользовательского интерфейса. При выполнении данной домашней контрольной работы №9 возникли некоторые трудности, но они были разрешены.

Знания, полученные на лекциях и практических занятиях по предмету «Разработка кода информационных сетей», помогли в написании данной работы. В результате работы были достигнуты все поставленные цели.