Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Вятский государственный университет»

Колледж ВятГУ

**ОТЧЕТ**

**ПО ДОМАШНЕЙ КОНТРОЛЬНОЙ РАБОТЕ №9**

**«РАБОТА С ТИПИЗИРОВАННЫМИ ДАННЫМИ»**

**ПО МДК 05.02 РАЗРАБОТКА КОДА ИНФОРМАЦИОННЫХ СИСТЕМ**

Выполнил: студент учебной группы

ИСПк 203-52-00

Калинин Максим Константинович

Преподаватель:

Сергеева Елизавета Григорьевна

Киров

2023

1. **Цель работы:** получение навыков реализации алгоритмов с рекурсивными вычислениями, знакомство с фракталами.
2. **Формулировка задания**
3. Сформировать структуру записи, с которой будет осуществляться работа, и согласовать ее с преподавателем. (Компьютеры)
4. Разработать приложение, позволяющее извлекать набор записей из типизированного файла, визуализировать данных набор, а также позволяющее изменять данные и выполнять сохранение проделанных изменений.
5. **Внешний вид программы**

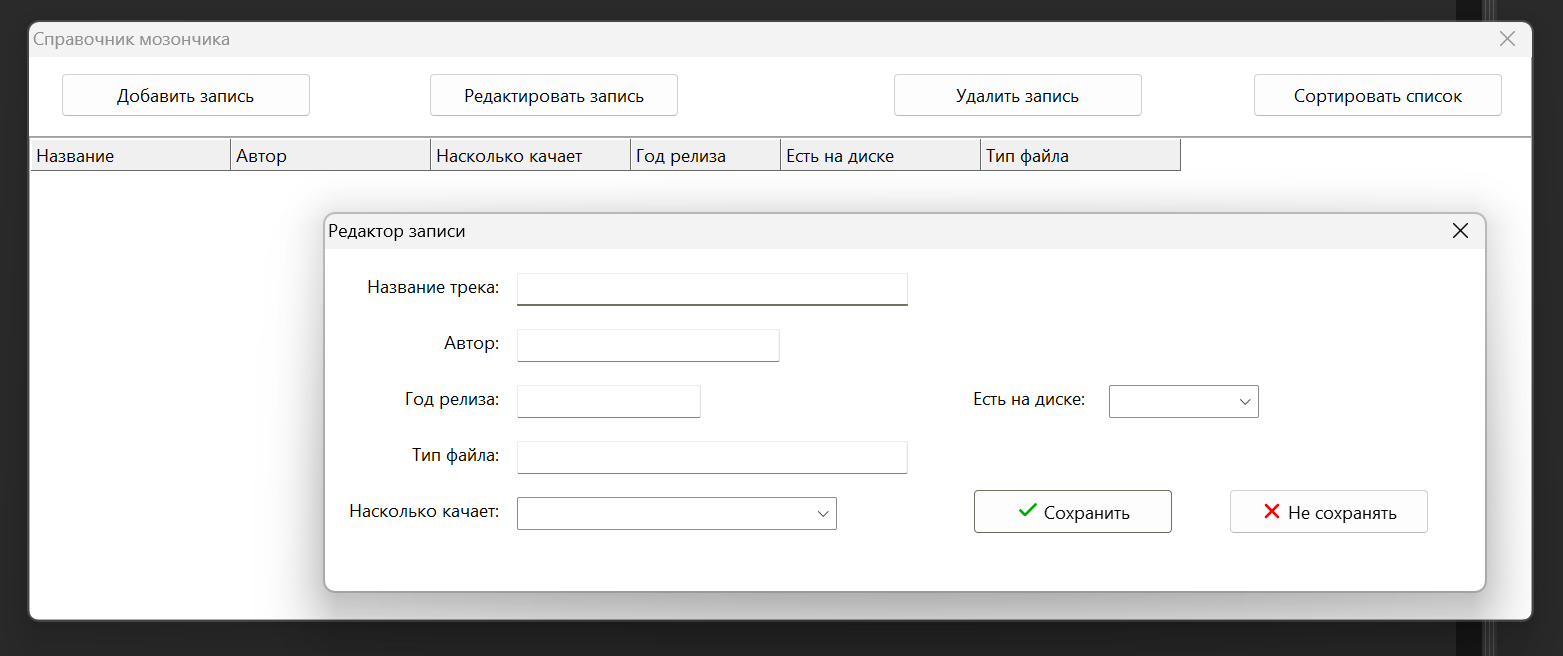


Рис.1 – Вид программы

1. **Код**

unit Edit;

{$mode ObjFPC}{$H+}

interface

uses

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

type

{ TfEdit }

TfEdit = class(TForm)

bCancel: TBitBtn;

bSave: TBitBtn;

CBSmotr: TComboBox;

CBNote: TComboBox;

eAuthor: TEdit;

eYear: TEdit;

Regh: TEdit;

eName: TEdit;

Lname: TLabel;

Author: TLabel;

Bass: TLabel;

Year: TLabel;

Disk: TLabel;

Fyletypee: TLabel;

procedure bSaveClick(Sender: TObject);

procedure eDlitChange(Sender: TObject);

procedure eYearChange(Sender: TObject);

procedure FormShow(Sender: TObject);

procedure AuthorClick(Sender: TObject);

procedure LnameClick(Sender: TObject);

procedure ReghChange(Sender: TObject);

procedure eYearKeyPress(Sender: TObject; var Key: char);

private

public

end;

var

fEdit: TfEdit;

implementation

{$R \*.lfm}

{ TfEdit }

procedure TfEdit.eYearKeyPress(Sender: TObject; var Key:char);

begin

if not (Key in ['0'..'9', #8]) then Key := #0;

end;

end.

unit Main;

{$mode objfpc}{$H+}

interface

uses

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

Buttons, edit, Grids;

type

Contacts = record

Name: string[100];

Country: string[80];

Genre: string[80];

Year: integer;

Watched: boolean;

Director: string[100];

end; //record

type

{ TfMain }

TfMain = class(TForm)

Panel1: TPanel;

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

procedure FormCreate(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.eAuthor.Text:= '';

fEdit.eYear.Text := '';

fEdit.Regh.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.eAuthor.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.eAuthor.Text;

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

SG.Cells[3, SG.RowCount-1]:= fEdit.eYear.Text;

SG.Cells[4, SG.RowCount-1]:= fEdit.CBSmotr.Text;

SG.Cells[5, SG.RowCount-1]:= fEdit.Regh.Text;

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 мультфильм "' +

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

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

SG.DeleteRow(SG.Row);

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

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

fEdit.eYear.Text := SG.Cells[3, SG.Row];

fEdit.CBSmotr.Text := SG.Cells[4, SG.Row];

fEdit.Regh.Text := SG.Cells[5, 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.eAuthor.Text;

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

SG.Cells[3, SG.Row]:= fEdit.eYear.Text;

SG.Cells[4, SG.Row]:= fEdit.CBSmotr.Text;

SG.Cells[5, SG.Row]:= fEdit.Regh.Text;

end;

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;

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 + 'Muzik.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.Country:= SG.Cells[1, i];

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

MyCont.Year := StrToInt(SG.Cells[3, i]);

MyCont.Watched := (SG.Cells[4, i] = 'Да'); //Преобразование в булево значение

MyCont.Director:= SG.Cells[5, 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]:= 'Название';

SG.Cells[1, 0]:= 'Автор';

SG.Cells[2, 0]:= 'Насколько качает';

SG.Cells[3, 0]:= 'Год релиза';

SG.Cells[4, 0]:= 'Есть на диске';

SG.Cells[5, 0]:= 'Тип файла';

SG.ColWidths[0]:= 200;

SG.ColWidths[1]:= 200;

SG.ColWidths[2]:= 200;

SG.ColWidths[3]:= 150;

SG.ColWidths[4]:= 200;

SG.ColWidths[5]:= 200;

//если файла данных нет, просто выходим:

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

//иначе файл есть, открываем его для чтения и

//считываем данные в сетку:

try

AssignFile(f, adres + 'Muzik.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.Country;

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

SG.Cells[3, SG.RowCount-1]:= IntToStr(MyCont.Year);

SG.Cells[4, SG.RowCount-1]:= BoolToStr(MyCont.Watched, True);

SG.Cells[5, SG.RowCount-1]:= MyCont.Director;

end;

finally

CloseFile(f);

end;

end;

end.

**Вывод**

В результате выполненной работы была выполнена программа для работы со списками. При выполнении данной работы были получены дополнительные навыки работы с Lazarus и его компонентами. В программе введены ограничения в каждое поле, для того чтобы реализовать корректную работу программы.