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**

**«Работа с типизированными файлами»**

**ПО МДК 05.02 РАЗРАБОТКА КОДА ИНФОРМАЦИОННЫХ СИСТЕМ**

Выполнил: студент учебной группы

ИСПк-203-52-00

Цветкова Кристина Андреевна

Преподаватель:

Сергеева Елизавета Григорьевна

Киров

2024

1. **Цель домашней контрольной работы:**

Цель работы: получение представлений об организации хранения и использования информации посредством типизированных файлов, закрепить навыки создания пользовательского интерфейса.

1. **Формулировка задания**

1. Сформировать структуру записи, с которой будет осуществляться работа, и согласовать ее с преподавателем. (Мультфильмы)

2. Разработать приложение, позволяющее извлекать набор записей из типизированного файла, визуализировать данных набор, а также позволяющее изменять данные и выполнять сохранение проделанных изменений.

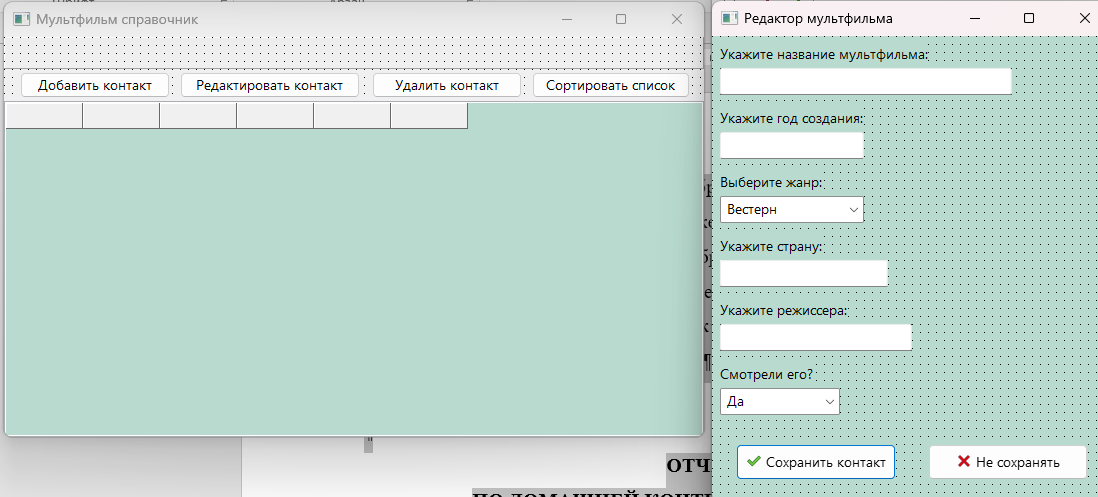


Рисунок 1 – Вид программы

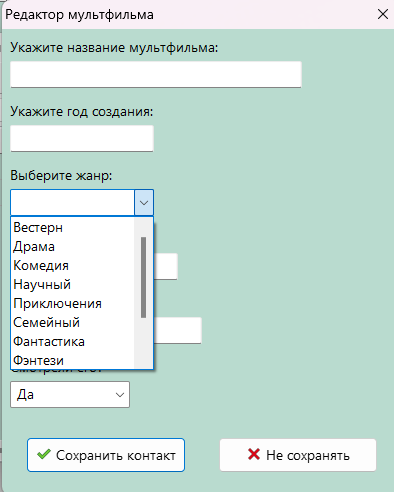


Рисунок 2 – Выбор жанра

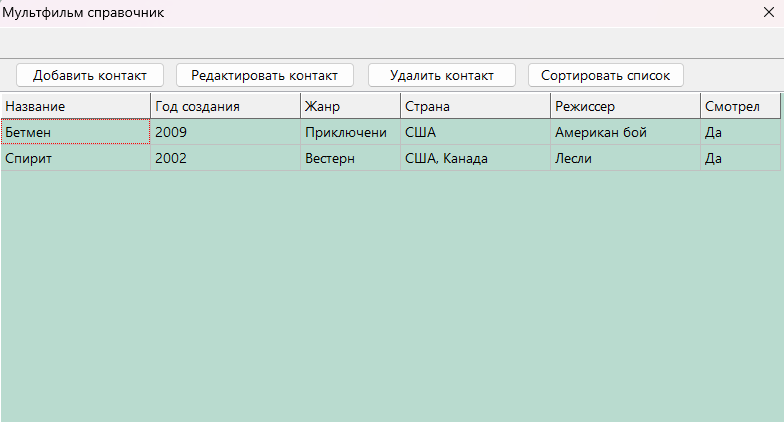


Рисунок 3 – Результат программы

4. Код программы.

Edit––––––––––––––––––––––––––––––––––––––––––––––––––––

unit Edit;

{$mode ObjFPC}{$H+}

interface

uses

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

type

{ TfEdit }

TfEdit = class(TForm)

bCancel: TBitBtn;

bSave: TBitBtn;

CBNote: TComboBox;

CBYesNo: TComboBox;

eCountry: TEdit;

eRegh: TEdit;

eGod: TEdit;

eName: TEdit;

Label1: TLabel;

Label2: TLabel;

Label3: TLabel;

Label4: TLabel;

Label5: TLabel;

Label6: TLabel;

procedure eReghChange(Sender: TObject);

procedure FormShow(Sender: TObject);

private

public

end;

var

fEdit: TfEdit;

implementation

{$R \*.lfm}

{ TfEdit }

procedure TfEdit.FormShow(Sender: TObject);

begin

eName.SetFocus;

end;

procedure TfEdit.eReghChange(Sender: TObject);

begin

end;

end.

Main––––––––––––––––––––––––––––––––––––––––––––––

unit Main;

{$mode objfpc}{$H+}

interface

uses

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

Buttons, edit, Grids, StdCtrls;

type

Contacts = record

Name: string[100];

God: string[20];

Note: string[20];

Country: string[40];

Regh: string[70];

YesNo: string[10];

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.eGod.Text:= '';

fEdit.eCountry.Text:= '';

fEdit.eRegh.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.eGod.Text= '') or (fEdit.eCountry.Text= '') or (fEdit.eRegh.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.eGod.Text;

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

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

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

SG.Cells[5, SG.RowCount-1]:= fEdit.CBYesNo.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 контакт "' +

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

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

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

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

fEdit.CBYesNo.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.eGod.Text;

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

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

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

SG.Cells[5, SG.Row]:= fEdit.CBYesNo.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 + '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.God:= SG.Cells[1, i];

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

MyCont.Country:= SG.Cells[3, i];

MyCont.Regh:= SG.Cells[4, i];

MyCont.YesNo:= 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]:= 'Год создания';

SG.Cells[2, 0]:= 'Жанр';

SG.Cells[3, 0]:= 'Страна';

SG.Cells[4, 0]:= 'Режиссер';

SG.Cells[5, 0]:= 'Смотрел';

SG.ColWidths[0]:= 150;

SG.ColWidths[1]:= 150;

SG.ColWidths[2]:= 100;

SG.ColWidths[3]:= 150;

SG.ColWidths[4]:= 150;

SG.ColWidths[5]:= 80;

//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.God;

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

SG.Cells[3, SG.RowCount-1]:= MyCont.Country;

SG.Cells[4, SG.RowCount-1]:= MyCont.Regh;

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

end;

finally

CloseFile(f);

end;

end;

end.