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 файлами»**

**ПО «МДК 05.02. Разработка кода»**

Выполнил: студент учебной группы

ИСПк-204-52-00

Гордеева В. Д.

Преподаватель:

Сергеева Е. Г.

Киров

2025

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

**Задание:**

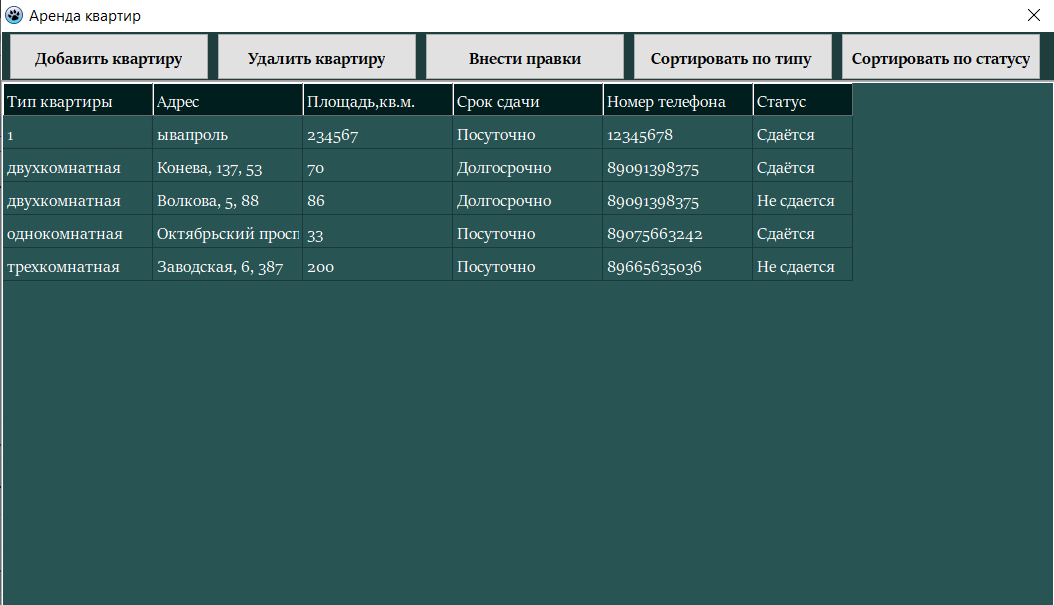
1. Сформировать структуру записи, с которой будет осуществляться работа, и согласовать ее с преподавателем.

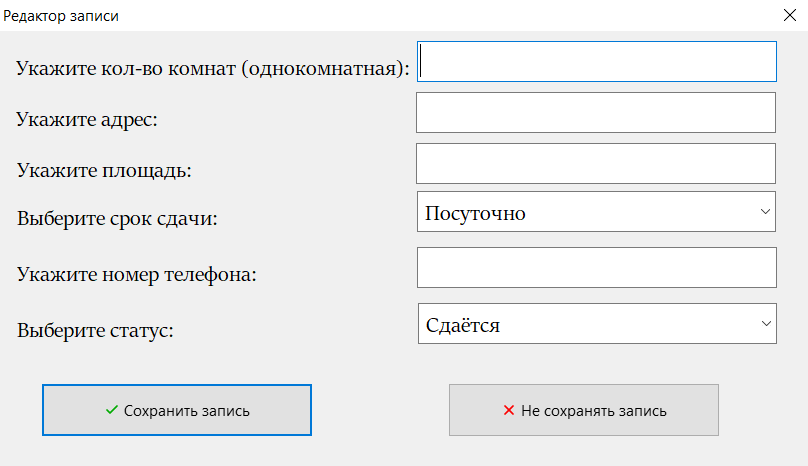
2. Разработать приложение, позволяющее извлекать набор записей из типизированного файла, визуализировать данный набор, а также позволяющее изменять данных и выполнять сохранение проделанных изменений.

**Описание программы:**

Программа представляет собой простое приложение для управления списком записей о квартирах. При запуске программы создается главное окно, содержащее таблицу (StringGrid) для отображения данных. Программа позволяет добавлять новые записи через форму редактирования, изменять существующие записи, удалять их, и сортировать записи по типу квартиры или статусу. Данные вводятся и редактируются через модальную форму, а при закрытии программы все данные из StringGrid сохраняются в файл, который автоматически загружается при следующем запуске, обеспечивая сохранение информации между сеансами работы.

**Результат выполнения:**

****Рисунок 1 – Скриншот формы Main

Рисунок 2 – Скриншот формы fedit

**Код программы:**

**Форма Main:**

unit Main;

{$mode objfpc}{$H+}

interface

uses

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

Grids, edit1;

type

Contacts = record

Adress: string[100];

Nomber: string[20];

S: string[20];

Tip: string[30];

Time: string[30];

Status: string[30];

end; //record

type

{ TfMain }

TfMain = class(TForm)

Panel1: TPanel;

SG: TStringGrid;

bDel: TSpeedButton;

bAdd: TSpeedButton;

bChange: TSpeedButton;

bSorttype: TSpeedButton;

bSortstatus: TSpeedButton;

procedure bAddClick(Sender: TObject);

procedure bChangeClick(Sender: TObject);

procedure bDelClick(Sender: TObject);

procedure bSortstatusClick(Sender: TObject);

procedure bSorttypeClick(Sender: TObject);

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

procedure FormCreate(Sender: TObject);

procedure SGDblClick(Sender: TObject);

private

public

end;

var

fMain: TfMain;

adres: string;

implementation

{$R \*.lfm}

{ TfMain }

procedure TfMain.bAddClick(Sender: TObject);

begin

fEdit.eAdress.Text:= '';

fEdit.eNomber.Text:= '';

fEdit.eS.Text:= '';

fEdit.eTip.Text:= '';

fEdit.ModalResult:= mrNone;

fEdit.ShowModal;

if (fEdit.eAdress.Text= '') or (fEdit.eNomber.Text= '') or (fEdit.eTip.text='') or (fEdit.eS.text='') then exit;

if fEdit.ModalResult <> mrOk then exit;

SG.RowCount:= SG.RowCount + 1;

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

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

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

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

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

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

end;

procedure TfMain.bChangeClick(Sender: TObject);

begin

//если данных в сетке нет - просто выходим:

if SG.RowCount = 1 then exit;

//иначе записываем данные в форму редактора:

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

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

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

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

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

fEdit.CBStatus.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.eTip.Text;

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

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

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

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

SG.Cells[5, SG.Row]:= fEdit.CBStatus.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 контакт "' +

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

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

SG.DeleteRow(SG.Row);

end;

procedure TfMain.bSortstatusClick(Sender: TObject);

begin

//если данных в сетке нет - просто выходим:

if SG.RowCount = 1 then exit;

//иначе сортируем список:

SG.SortColRow(true, 5);

end;

procedure TfMain.bSorttypeClick(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);

Rewrite(f);

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

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

//получаем данные текущей записи:

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

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

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

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

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

MyCont.Status:= 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)) + 'Apartament.dat';

//настроим сетку:

SG.Cells[0, 0]:= 'Тип квартиры';

SG.Cells[1, 0]:= 'Адрес';

SG.Cells[2, 0]:= 'Площадь,кв.м.';

SG.Cells[3, 0]:= 'Срок сдачи';

SG.Cells[4, 0]:= 'Номер телефона';

SG.Cells[5, 0]:= 'Статус';

SG.ColWidths[0]:= 150;

SG.ColWidths[1]:= 150;

SG.ColWidths[2]:= 150;

SG.ColWidths[3]:= 150;

SG.ColWidths[4]:= 150;

SG.ColWidths[5]:= 100;

//если файла данных нет, просто выходим:

if not FileExists(adres) then exit;

//иначе файл есть, открываем его для чтения и

//считываем данные в сетку:

try

AssignFile(f, adres);

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.Tip;

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

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

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

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

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

end;

finally

CloseFile(f);

end;

end;

procedure TfMain.SGDblClick(Sender: TObject);

begin

end;

end.

**Форма Edit:**

unit edit1;

{$mode ObjFPC}{$H+}

interface

uses

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

type

{ TfEdit }

TfEdit = class(TForm)

bSave: TBitBtn;

bCancel: TBitBtn;

CBTime: TComboBox;

CBStatus: TComboBox;

eAdress: TEdit;

eS: TEdit;

eTip: TEdit;

eNomber: TEdit;

Label1: TLabel;

Label2: TLabel;

Label3: TLabel;

Label4: TLabel;

Label5: TLabel;

Label6: TLabel;

procedure eAdressChange(Sender: TObject);

procedure FormCreate(Sender: TObject);

procedure FormShow(Sender: TObject);

procedure Label1Click(Sender: TObject);

private

public

end;

var

fEdit: TfEdit;

implementation

{$R \*.lfm}

{ TfEdit }

procedure TfEdit.FormCreate(Sender: TObject);

begin

end;

procedure TfEdit.FormShow(Sender: TObject);

begin

eTip.SetFocus;

end;

procedure TfEdit.Label1Click(Sender: TObject);

begin

end;

procedure TfEdit.eAdressChange(Sender: TObject);

begin

end;

end.

**Вывод:**

В ходе работы, благодаря доступной необходимой теоретической информации, не возникло особых сложностей. Я смогла глубже понять изучаемую тему и применить полученные знания на практике. Процесс работы был увлекательным и познавательным, что позволило не только улучшить навыки анализа и критического мышления, но и развить творческий подход к решению задач.

Кроме того, сотрудничество с одногруппниками и обмен идеями способствовали более глубокому осмыслению материала и расширению горизонтов. Я уверена, что полученный опыт будет полезен в будущем, и с нетерпением жду возможности применить его в новых проектах. В целом, проделанная работа стала важным этапом в моем профессиональном развитии.