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**

**информационных систем»**

Выполнил: студент учебной группы

ИСПк-203-52-00

Четвериков Д.А.

Преподаватель:

Сергеева Е.Г.

Киров

2025

**Цель:** получение базовых навыков реализации приложений с графическим интерфейсом пользователя на основе событийно-ориентированного программирования.

**Задание (Вариант 0):**

1. Сформировать структуру записи, с которой будет осуществляться работа, и согласовать её 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 алгоритма:**

Main

1. Вводятся переменные и их тип
2. Объявление процедур
3. Настройка таблицы
4. Процедура добавления:

//очищаем поля, если там что-то есть:

fEdit.eName.Text:= '';

fEdit.eNation.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.eNation.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.eNation.Text;

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

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

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

1. Процедура редактирования : редактирует уже написанный текст и обновляет данные
2. Процедура удаления: удаляет выделенный столбец
3. Процедура загрузки загружает данные в типизированный файл
4. Процедура TForm1.Save считывает данные с файла и сохраняет новые записи
5. Сортировка работает по методу пузырька и перезаписывает файлы в новом порядке
6. Перевод значений в строку и обратно

Unit 2.

1. TfrmEdit.FormCreate процедура для выбора из списка
2. Процедура сохранения считывает всю написанную информацию и сохраняет в типизированный файл
3. Процедура отметы завершает редактирование/добавление без результата

Описание элементов

TLabel - используется для подписей полей, в которые записываются значения.

Tbutton - используется для создания кнопок для рассчета итоговой суммы, очистки полей, вывода окна с информацией.

TEdit - используется для ввода значений.

TCombobox - используется как список, в котором можно выбрать по какому типу процентов будет происходить вычисление

TPanel – необходим для размещения других элементов

TStringGrid – используется для ввода табличной информации

Код программы

unit Main;

{$mode objfpc}{$H+}

interface

uses

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

StdCtrls, Grids, Edit;

type

Tank = record

Name: string[50];

Nation: string[50];

Price: integer;

BR: string[10];

People: integer;

end;

{ TfMain }

TfMain = class(TForm)

cbSort: TComboBox;

Panel1: TPanel;

dAdd: TSpeedButton;

bEdit: TSpeedButton;

bDel: TSpeedButton;

bSort: TSpeedButton;

SG: TStringGrid;

procedure bDelClick(Sender: TObject);

procedure bEditClick(Sender: TObject);

procedure bAddClick(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;

implementation

{$R \*.lfm}

{ TfMain }

procedure TfMain.bAddClick(Sender: TObject);

begin

//очищаем поля, если там что-то есть:

fEdit.eName.Text:= '';

fEdit.eNation.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.eNation.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.eNation.Text;

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

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

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

end;

procedure TfMain.bSortClick(Sender: TObject);

begin

if SG.RowCount = 1 then exit;

if cbSort.ItemIndex = 0 then

SG.SortColRow(true, 0);

if cbSort.ItemIndex = 1 then

SG.SortColRow(true, 1);

if cbSort.ItemIndex = 2 then

SG.SortColRow(true, 2);

if cbSort.ItemIndex = 3 then

SG.SortColRow(true, 3);

if cbSort.ItemIndex = 4 then

SG.SortColRow(true, 4);

end;

procedure TfMain.bEditClick(Sender: TObject);

begin

// если нет выбранной строки или только заголовок — выходим

if (SG.RowCount = 1) or (SG.Row = 0) then exit;

// заполняем форму текущими значениями из таблицы

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

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

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

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

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

// устанавливаем ModalResult редактора в mrNone:

fEdit.ModalResult := mrNone;

// показываем форму редактирования

fEdit.ShowModal;

// если пользователь нажал "Сохранить", сохраняем изменения

if fEdit.ModalResult = mrOk then

begin

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

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

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

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

SG.Cells[4, SG.Row] := fEdit.ePeople.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 данную модель? "' +

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

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

SG.DeleteRow(SG.Row);

end;

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

var

MyCont: Tank; //для очередной записи

f: file of Tank; //файл данных

i: integer; //счетчик цикла

begin

//если строки данных пусты, просто выходим:

if SG.RowCount = 1 then exit;

//иначе открываем файл для записи:

AssignFile(f, adres + 'tanks.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.Nation := SG.Cells[1, i];

MyCont.Price := StrToIntDef(SG.Cells[2, i], 0);

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

MyCont.People := StrToIntDef(SG.Cells[4, i], 0);

//записываем их:

Write(f, MyCont);

end;

CloseFile(f);

end;

procedure TfMain.FormCreate(Sender: TObject);

var

MyCont: Tank; //для очередной записи

f: file of Tank; //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]:= 'Нация';

SG.Cells[2, 0]:= 'Цена';

SG.Cells[3, 0]:= 'Боевой рейтинг';

SG.Cells[4, 0]:= 'Экипаж';

SG.ColWidths[0] := 200;

SG.ColWidths[1] := 100;

SG.ColWidths[2] := 100;

SG.ColWidths[3] := 150;

SG.ColWidths[4] := 120;

//если файла данных нет, просто выходим:

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

//иначе файл есть, открываем его для чтения и

//считываем данные в сетку:

try

AssignFile(f, adres + 'tanks.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.Nation;

SG.Cells[2, SG.RowCount - 1] := IntToStr(MyCont.Price);

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

SG.Cells[4, SG.RowCount - 1] := IntToStr(MyCont.People);

end;

finally

CloseFile(f);

end;

end;

end.unit Unit2;

{$mode objfpc}{$H+}

interface

uses

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

ExtCtrls,DateTimePicker;

type

{ TfrmEditAgent }

TfrmEditAgent = class(TForm)

btnCancel: TButton;

btnSave: TButton;

cbUnlocked: TCheckBox;

cmbRole: TComboBox;

dtpRelease: TDateTimePicker;

edtWinRate: TEdit;

edtName: TEdit;

edtOrigin: TEdit;

edtKDA: TEdit;

edtPickRate: TEdit;

edtUltimateCost: TEdit;

edtUltimateName: TEdit;

Label1: TLabel;

Label10: TLabel;

Label11: TLabel;

Label2: TLabel;

Label3: TLabel;

Label4: TLabel;

Label5: TLabel;

Label6: TLabel;

Label8: TLabel;

Label9: TLabel;

procedure btnCancelClick(Sender: TObject);

procedure btnSaveClick(Sender: TObject);

procedure FormCreate(Sender: TObject);

procedure Label10Click(Sender: TObject);

procedure Label5Click(Sender: TObject);

private

{ private declarations }

public

{ public declarations }

end;

var

frmEditAgent: TfrmEditAgent;

implementation

{$R \*.lfm}

{ TfrmEditAgent }

procedure TfrmEditAgent.FormCreate(Sender: TObject);

begin

cmbRole.Items.Add('Duelist');

cmbRole.Items.Add('Controller');

cmbRole.Items.Add('Sentinel');

cmbRole.Items.Add('Initiator');

dtpRelease.Date := Now();

end;

procedure TfrmEditAgent.Label10Click(Sender: TObject);

begin

end;

procedure TfrmEditAgent.Label5Click(Sender: TObject);

begin

end;

procedure TfrmEditAgent.btnSaveClick(Sender: TObject);

begin

if edtName.Text = '' then

begin

ShowMessage('Введите имя агента!');

edtName.SetFocus;

Exit;

end;

if cmbRole.ItemIndex = -1 then

begin

ShowMessage('Выберите роль агента!');

cmbRole.SetFocus;

Exit;

end;

ModalResult := mrOk;

end;

procedure TfrmEditAgent.btnCancelClick(Sender: TObject);

begin

ModalResult := mrCancel;

end;

end.

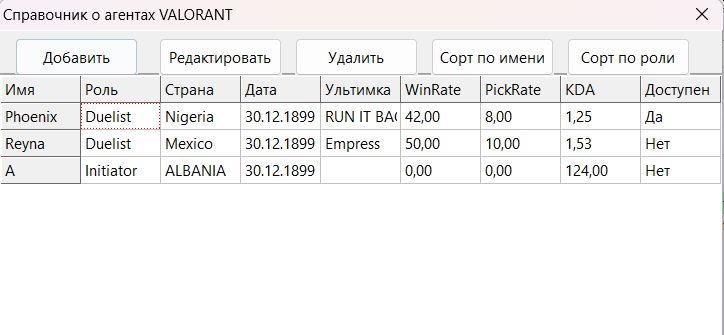


Рисунок 1 - Результат работы программы .

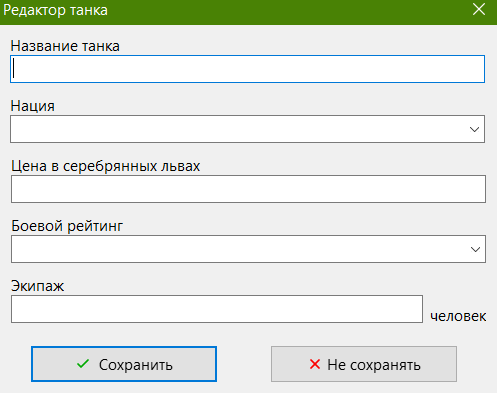


Рисунок 2 - Результат работы программы (Редактирование)

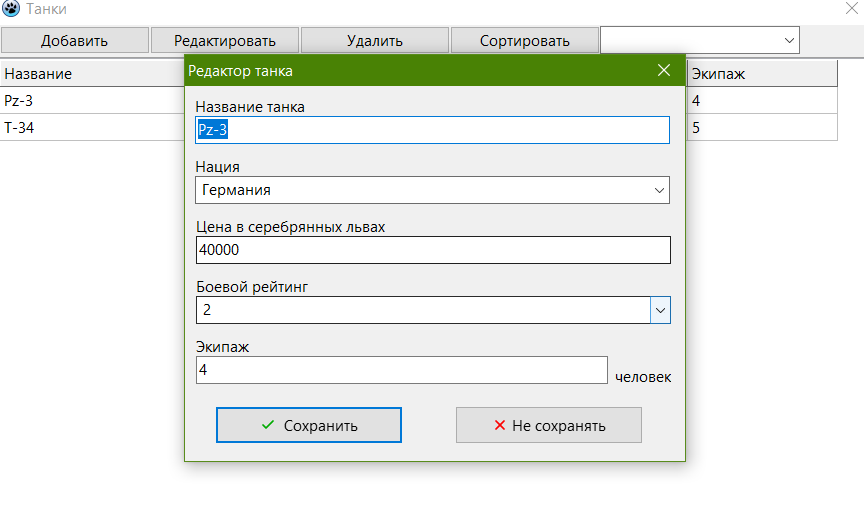


Рисунок 3 - Результат работы программы (Добавление)

**Вывод:** в ходе данной работы были получены навыки создания таблиц для работы с типизированными файлами. Было изучено как записывать данные с разным типом в файл , а также было разработано приложение, позволяющее извлекать набор записей из типизированного файла, а также была добавлена возможность сохранять , редактировать , удалять и сортировать эти данные.