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-204-52-00

Никулин А. И.

Преподаватель:

Сергеева Е. Г.

Киров

2024

1. Цель работы: получить представление об организации хранения и использования информации посредством типизированных файлов, закрепить навыки создания пользовательского интерфейса.
2. Задание для варианта 1:
3. Сформировать структуру записи, с которой будет осуществляться работа, и согласовать её с преподавателем.
4. Разработать приложение, позволяющее извлекать набор записей из типизированного файла, визуализировать данный набор, а также позволяющее изменять данные и выполнять сохранение проделанных изменений.
5. В записи должно быть не менее 5 элементов, среди которых должно быть минимум 3 различных типа данных.
6. Описание алгоритма.

Для выполнения задания было выбрана структура moto содержащая записи о танках (Название – string, год изготовления – integer, тип двигателя бензиновый или нет – boolean, мощность двигателя – integer, цвет – string).

Программа предоставляет возможность добавлять, редактировать и удалять записи, а также сортировать по различным столбцам (при нажатии на их заголовок). Для добавления и редактирования записей используется отдельная экранная форма, на которой отображаются все необходимые поля.

Данные о записях таблицы хранятся в типизированном файле moto.dat.

1. Код основной программы.

unit mainmod;

{$mode objfpc}{$H+}

interface

uses

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

Grids, createRec;

type

{ TmForm }

TmForm = class(TForm)

mAdd: TBitBtn;

mEdit: TBitBtn;

mDel: TBitBtn;

Panel: TPanel;

mRec: TStringGrid;

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

procedure FormCreate(Sender: TObject);

procedure mAddClick(Sender: TObject);

procedure mDelClick(Sender: TObject);

procedure mEditClick(Sender: TObject);

private

public

end;

type

Moto = record

Name: string[100];

Year: integer;

Petrol: boolean;

Power: integer;

Color: string[30];

end;

var

mForm: TmForm;

adres: string;

implementation

{$R \*.lfm}

{ TmForm }

procedure TmForm.FormCreate(Sender: TObject);

var

newMoto: Moto; //для очередной записи

f: file of Moto; //файл данных

i: integer; //счетчик цикла

begin

adres:= ExtractFilePath(ParamStr(0));

mRec.Cells[0, 0]:= 'Название';

mRec.Cells[1, 0]:= 'Год';

mRec.Cells[2, 0]:= 'Бензиновый';

mRec.Cells[3, 0]:= 'Мощность';

mRec.Cells[4, 0]:= 'Цвет';

mRec.ColWidths[0]:= 200;

mRec.ColWidths[1]:= 100;

mRec.ColWidths[2]:= 100;

mRec.ColWidths[3]:= 100;

mRec.ColWidths[4]:= 129;

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

try

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

Reset(f);

while not Eof(f) do begin

Read(f, newMoto);

mRec.RowCount:= mRec.RowCount + 1;

mRec.Cells[0, mRec.RowCount-1]:= newMoto.Name;

mRec.Cells[1, mRec.RowCount-1]:= IntToStr(newMoto.Year);

if (newMoto.Petrol) then mRec.Cells[2, mRec.RowCount-1]:= 'Да'

else mRec.Cells[2, mRec.RowCount-1]:='Нет';

mRec.Cells[3, mRec.RowCount-1]:= FloatToStr(newMoto.Power);

mRec.Cells[4, mRec.RowCount-1]:= newMoto.Color;

end;

finally

CloseFile(f);

end;

end;

procedure TmForm.mAddClick(Sender: TObject);

begin

createRecord.eName.Text:= '';

createRecord.eYear.Text:= '';

createRecord.ePower.Text:= '';

createRecord.ModalResult:= mrNone;

createRecord.ShowModal;

if (createRecord.eName.Text= '') or (createRecord.eYear.Text= '') or (createRecord.ePower.Text= '') then exit;

if createRecord.ModalResult <> mrOk then exit;

mRec.RowCount:= mRec.RowCount + 1;

mRec.Cells[0, mRec.RowCount-1]:= createRecord.eName.Text;

mRec.Cells[1, mRec.RowCount-1]:= createRecord.eYear.Text;

if (createRecord.ePetrol.Checked) then mRec.Cells[2, mRec.RowCount-1]:= 'Да'

else mRec.Cells[2, mRec.RowCount-1]:='Нет';

mRec.Cells[3, mRec.RowCount-1]:= createRecord.ePower.Text;

mRec.Cells[4, mRec.RowCount-1]:= createRecord.eColor.Text;

end;

procedure TmForm.mDelClick(Sender: TObject);

begin

if mRec.RowCount = 1 then exit;

if MessageDlg('Требуется подтверждение',

'Вы действительно хотите удалить запись "' +

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

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

mRec.DeleteRow(mRec.Row);

end;

procedure TmForm.mEditClick(Sender: TObject);

begin

if mRec.RowCount = 1 then exit;

createRecord.eName.Text:= mRec.Cells[0, mRec.Row];

createRecord.eYear.Text:= mRec.Cells[1, mRec.Row];

if (mRec.Cells[2, mRec.RowCount-1] = 'Да') then createRecord.ePetrol.Checked := True

else createRecord.ePetrol.Checked := False;

createRecord.ePower.Text:= mRec.Cells[3, mRec.Row];

createRecord.eColor.Text:= mRec.Cells[4, mRec.Row];

//устанавливаем ModalResult редактора в mrNone:

createRecord.ModalResult:= mrNone;

//теперь выводим форму:

createRecord.ShowModal;

if createRecord.ModalResult = mrOk then begin

mRec.Cells[0, mRec.Row]:= createRecord.eName.Text;

mRec.Cells[1, mRec.Row]:= createRecord.eYear.Text;

if (createRecord.ePetrol.Checked) then mRec.Cells[2, mRec.RowCount-1]:= 'Да'

else mRec.Cells[2, mRec.RowCount-1]:='Нет';

mRec.Cells[3, mRec.Row]:= createRecord.ePower.Text;

mRec.Cells[4, mRec.Row]:= createRecord.eColor.Text;

end;

end;

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

var

newMoto: Moto; //для очередной записи

f: file of Moto; //файл данных

i: integer; //счетчик цикла

begin

if mRec.RowCount = 1 then exit;

try

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

Rewrite(f);

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

newMoto.Name:= mRec.Cells[0, i];

newMoto.Year:= StrToInt(mRec.Cells[1, i]);

if (mRec.Cells[2, mRec.RowCount-1] = 'Да') then newMoto.Petrol := True

else newMoto.Petrol := False;

newMoto.Power:= StrToInt(mRec.Cells[3, i]);

newMoto.Color:= mRec.Cells[4, i];

Write(f, newMoto);

end;

finally

CloseFile(f);

end;

end;

end.

unit createRec;

{$mode ObjFPC}{$H+}

interface

uses

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

type

{ TcreateRec }

TcreateRec = class(TForm)

mCancel: TBitBtn;

mSave: TBitBtn;

eColor: TComboBox;

ePower: TEdit;

ePetrol: TCheckBox;

eYear: TEdit;

eName: TEdit;

motocolor: TLabel;

motopower: TLabel;

petrol: TLabel;

year: TLabel;

moto: TLabel;

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

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

procedure FormShow(Sender: TObject);

private

public

end;

var

createRecord: TcreateRec;

implementation

{$R \*.lfm}

{ TcreateRec }

procedure TcreateRec.FormShow(Sender: TObject);

begin

eName.SetFocus;

ePower.OnKeyPress := @ePowerKeyPress;

eYear.OnKeyPress := @eYearKeyPress;

end;

procedure TcreateRec.ePowerKeyPress(Sender: TObject; var Key: char);

begin

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

begin

Key := #0; // Отменяем ввод, если это не цифра и не backspace

end;

end;

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

begin

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

begin

Key := #0; // Отменяем ввод, если это не цифра и не backspace

end;

end;

end.

1. Результат выполнения программы.

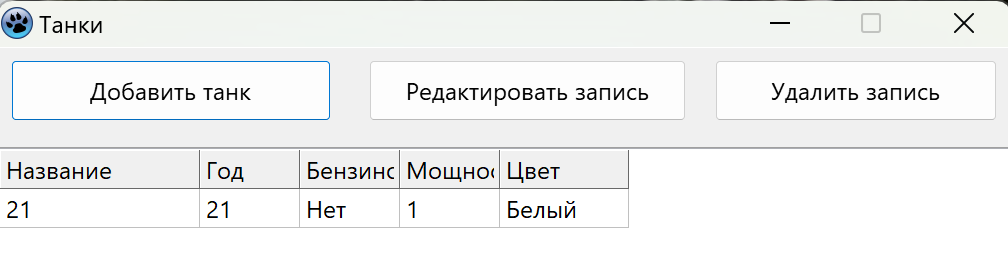


Рисунок 1 - Окно программы

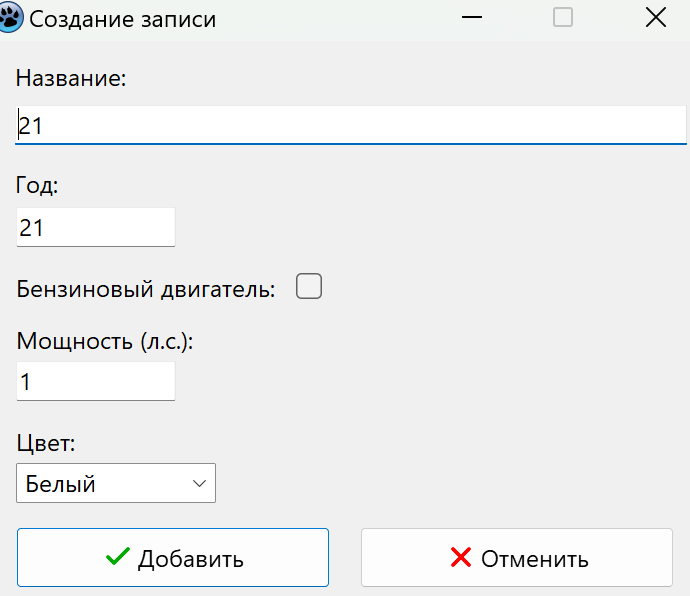


Рисунок 2 - Окно создания/редактирования записи

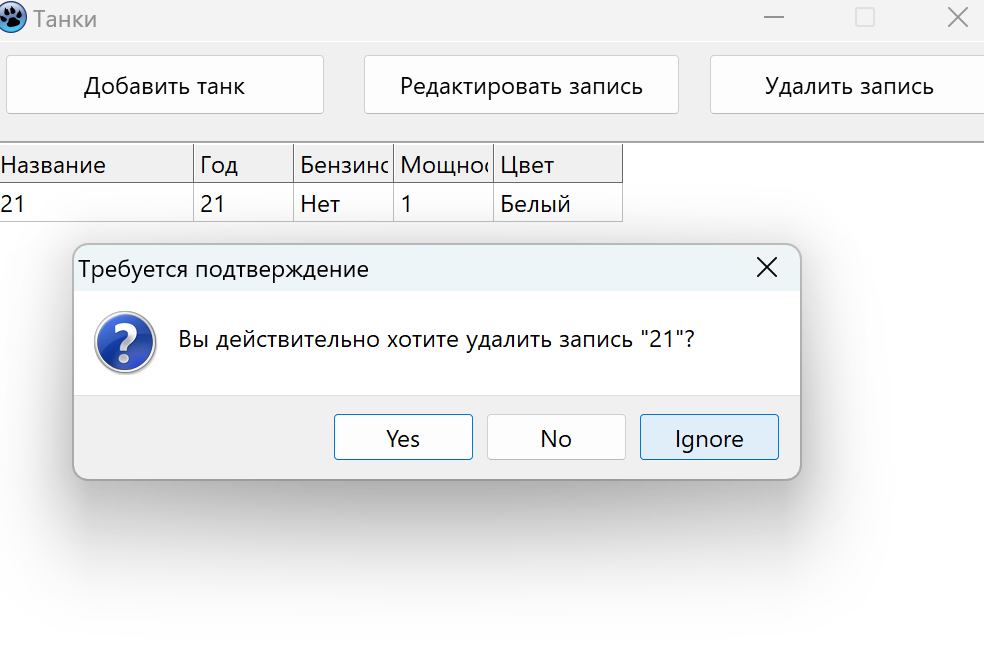


Рисунок 3 - Удаление записи

1. Вывод.

В результате выполнения домашней контрольной работы было получено представление об организации хранения и использования информации посредством типизированных файлов, закреплены навыки создания пользовательского интерфейса.