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

Шмаков Никита

Преподаватель:

Сергеева Елизавета Григорьевна

Киров

2023

В отчете должны отображаться:

1. Цель работы
2. Скриншот задания (с вариантом)
3. Описание алгоритма
4. Код программы
5. Результат выполнения программы
6. Вывод

1. Цель работы

Получить представление об организации хранения и использования информации посредством типизированных файлов, закрепить навыки создания пользовательского интерфейса.

2. Скриншот задания (с вариантом)

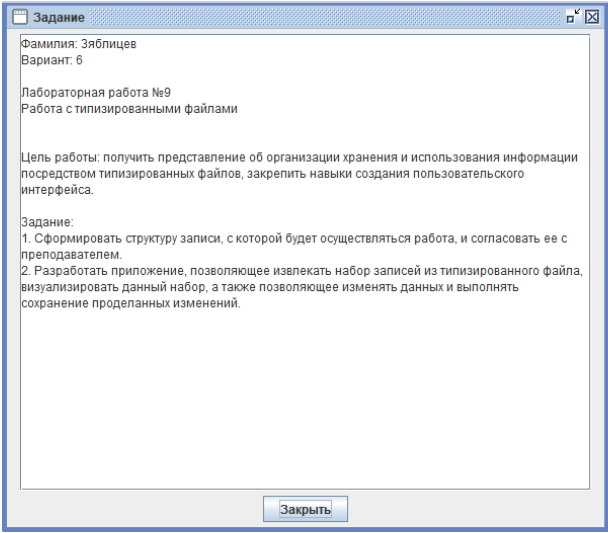


Рис 1 – Скриншот задания

3. Описание алгоритма

Используется база данных SQLite, таблица БД имеет столбцы: Марка, Модель, Мощность двигателя и Стоимость автомобиля.

В программе реализованы возможности вывода информации из базы данных в таблицу, добавление в БД записей, изменение записей, удаление записей и поиск записи по названию модели автомобиля.

4. Код программы

unit Unit1;

{$mode objfpc}{$H+}

interface

uses

Classes, SysUtils, SQLite3Conn, SQLDB, DB, Forms, Controls, Graphics, Dialogs,

StdCtrls, DBGrids, edit;

type

{ TfMain }

TfMain = class(TForm)

ButOpenBd: TButton;

ButAdd: TButton;

ButChange: TButton;

ButDel: TButton;

ButSearch: TButton;

DataSource1: TDataSource;

DBGrid1: TDBGrid;

eSearch: TEdit;

Label1: TLabel;

SQLite3Connection1: TSQLite3Connection;

SQLQuery1: TSQLQuery;

SQLQuery2: TSQLQuery;

SQLTransaction1: TSQLTransaction;

procedure ButOpenBdClick(Sender: TObject);

procedure ButAddClick(Sender: TObject);

procedure ButChangeClick(Sender: TObject);

procedure ButDelClick(Sender: TObject);

procedure ButSearchClick(Sender: TObject);

procedure FormCreate(Sender: TObject);

private

public

end;

var

fMain: TfMain;

implementation

{$R \*.lfm}

{ TfMain }

procedure TfMain.ButOpenBdClick(Sender: TObject);

begin

SQLQuery1.Close;

SQLQuery1.SQL.Text:='select \* from Cars';

SQLQuery1.Open;

SQLQuery2.Close;

SQLQuery2.SQL.Text:='select \* from Cars';

SQLQuery2.Open;

DBGrid1.Columns.Items[0].Width:=30;

DBGrid1.Columns.Items[1].Width:=120;

DBGrid1.Columns.Items[2].Width:=180;

DBGrid1.Columns.Items[3].Width:=120;

DBGrid1.Columns.Items[4].Width:=120;

end;

procedure TfMain.ButAddClick(Sender: TObject);

begin

fEdit.eBrand.Text:= '';

fEdit.eModel.Text:= '';

fEdit.eEngine.Text:= '';

fEdit.ePrice.Text:= '';

//устанавливаем ModalResult редактора в mrNone:

fEdit.ModalResult:= mrNone;

//теперь выводим форму:

fEdit.ShowModal;

//если пользователь ничего не ввел - выходим:

if (fEdit.eBrand.Text= '') or (fEdit.eModel.Text= '') or (fEdit.eEngine.Text= '')

or (fEdit.ePrice.Text= '') then exit;

//если пользователь не нажал "Сохранить" - выходим:

if fEdit.ModalResult <> mrOk then exit;

SQLQuery1.Close; // выключаем компонент

SQLQuery1.SQL.Text := 'insert into cars (brand, model, engine, price) values(:b,:m,:e,:p);'; // добавляем sql запрос для добавления данных

SQLQuery1.ParamByName('b').AsString := fEdit.eBrand.text;// присваиваем записи текстовое значение

SQLQuery1.ParamByName('m').AsString := fEdit.eModel.text;

SQLQuery1.ParamByName('e').AsString := fEdit.eEngine.text;

SQLQuery1.ParamByName('p').AsString := fEdit.ePrice.text;

SQLQuery1.ExecSQL; // выполняем запрос

SQLTransaction1.Commit; //подтверждаем изменения в базе

SQLite3Connection1.DatabaseName := 'C:\Users\ghbdtn\Desktop\dkr\school.db'; // указывает путь к базе

SQLite3Connection1.CharSet := 'UTF8'; // указываем рабочую кодировку

SQLite3Connection1.Transaction := SQLTransaction1; // указываем менеджер транзакций

try // пробуем подключится к базе

SQLite3Connection1.Open;

SQLTransaction1.Active := True;

except // если не удалось то выводим сообщение о ошибке

ShowMessage('Ошибка подключения к базе!');

end;

SQLQuery1.Close;

SQLQuery1.SQL.Text:='select \* from Cars';

SQLQuery1.Open;

DBGrid1.Columns.Items[0].Width:=30;

DBGrid1.Columns.Items[1].Width:=120;

DBGrid1.Columns.Items[2].Width:=180;

DBGrid1.Columns.Items[3].Width:=120;

DBGrid1.Columns.Items[4].Width:=120;

end;

procedure TfMain.ButChangeClick(Sender: TObject);

begin

fEdit.eBrand.Text:= SQLQuery1.Fields.FieldByName('brand').AsString;

fEdit.eModel.Text:= SQLQuery1.Fields.FieldByName('model').AsString;

fEdit.eEngine.Text:= SQLQuery1.Fields.FieldByName('engine').AsString;

fEdit.ePrice.Text:= SQLQuery1.Fields.FieldByName('price').AsString;

//устанавливаем ModalResult редактора в mrNone:

fEdit.ModalResult:= mrNone;

//теперь выводим форму:

fEdit.ShowModal;

//если пользователь не нажал "Сохранить" - выходим:

if fEdit.ModalResult <> mrOk then exit;

///

SQLQuery1.Edit; // открываем процедуру добавления данных

if not (fEdit.eBrand.Text= '') then SQLQuery1.Fields.FieldByName('brand').AsString := fEdit.eBrand.Text;

// присваиваем записи текстовое значение

if not (fEdit.eModel.Text= '') then SQLQuery1.Fields.FieldByName('model').AsString := fEdit.eModel.Text;

if not (fEdit.eEngine.Text= '') then SQLQuery1.Fields.FieldByName('engine').AsString := fEdit.eEngine.Text;

if not (fEdit.eModel.Text= '') then SQLQuery1.Fields.FieldByName('model').AsString := fEdit.eModel.Text;

Sqlquery1.Post; // записываем данные

sqlquery1.ApplyUpdates;// отправляем изменения в базу

SQLTransaction1.Commit; // подтверждаем изменения в базе

///

SQLite3Connection1.DatabaseName := 'C:\Users\ghbdtn\Desktop\dkr\school.db'; // указывает путь к базе

SQLite3Connection1.CharSet := 'UTF8'; // указываем рабочую кодировку

SQLite3Connection1.Transaction := SQLTransaction1; // указываем менеджер транзакций

try // пробуем подключится к базе

SQLite3Connection1.Open;

SQLTransaction1.Active := True;

except // если не удалось то выводим сообщение о ошибке

ShowMessage('Ошибка подключения к базе!');

end;

SQLQuery1.Close;

SQLQuery1.SQL.Text:='select \* from Cars';

SQLQuery1.Open;

DBGrid1.Columns.Items[0].Width:=30;

DBGrid1.Columns.Items[1].Width:=120;

DBGrid1.Columns.Items[2].Width:=180;

DBGrid1.Columns.Items[3].Width:=120;

DBGrid1.Columns.Items[4].Width:=120;

end;

procedure TfMain.ButDelClick(Sender: TObject);

begin

SQLQuery1.Delete;

DBGrid1.Columns.Items[0].Width:=30;

DBGrid1.Columns.Items[1].Width:=120;

DBGrid1.Columns.Items[2].Width:=180;

DBGrid1.Columns.Items[3].Width:=120;

DBGrid1.Columns.Items[4].Width:=120;

end;

procedure TfMain.ButSearchClick(Sender: TObject);

begin

if not (eSearch.text = '') then begin

SQLQuery1.Close;// закрываем датасет

SQLQuery1.SQL.Text := 'select \* from Cars where model = :m'; // добавляем наш запрос

SQLQuery1.ParamByName('m').AsString:= eSearch.text;// присваиваем требуемый параметр

SQLQuery1.Open;// открываем запрос

end

else ShowMessage('Введите id автомобиля для поиска записи!');

DBGrid1.Columns.Items[0].Width:=30;

DBGrid1.Columns.Items[1].Width:=120;

DBGrid1.Columns.Items[2].Width:=180;

DBGrid1.Columns.Items[3].Width:=120;

DBGrid1.Columns.Items[4].Width:=120;

end;

procedure TfMain.FormCreate(Sender: TObject);

begin

fMain.width:=600;

SQLite3Connection1.DatabaseName := 'C:\Users\ghbdtn\Desktop\dkr\school.db'; // указывает путь к базе

SQLite3Connection1.CharSet := 'UTF8'; // указываем рабочую кодировку

SQLite3Connection1.Transaction := SQLTransaction1; // указываем менеджер транзакций

try // пробуем подключится к базе

SQLite3Connection1.Open;

SQLTransaction1.Active := True;

except // если не удалось то выводим сообщение о ошибке

ShowMessage('Ошибка подключения к базе!');

end;

end;

end.

unit edit;

{$mode ObjFPC}{$H+}

interface

uses

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

type

{ TfEdit }

TfEdit = class(TForm)

bSave: TBitBtn;

bCancel: TBitBtn;

eBrand: TEdit;

eEngine: TEdit;

ePrice: TEdit;

eModel: TEdit;

Label1: TLabel;

Label2: TLabel;

Label3: TLabel;

Label4: TLabel;

procedure FormShow(Sender: TObject);

private

public

end;

var

fEdit: TfEdit;

implementation

{$R \*.lfm}

{ TfEdit }

procedure TfEdit.FormShow(Sender: TObject);

begin

eBrand.SetFocus;

end;

end.

5. Результат выполнения программы

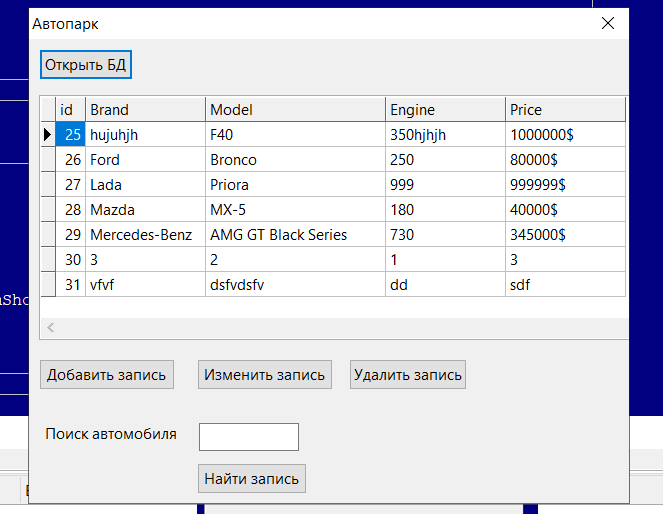


Рис 2 – Результаты выполнения программы

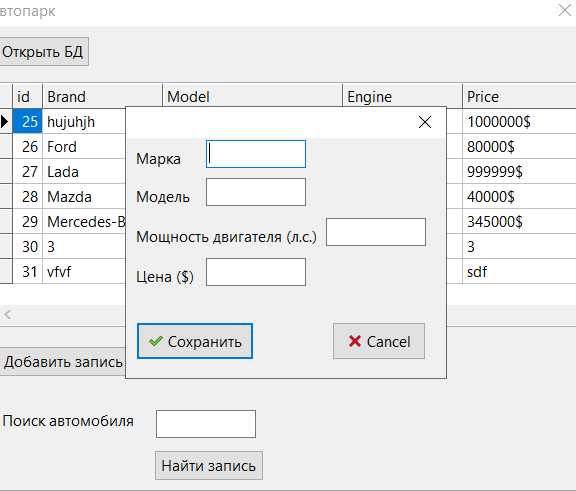


Рис 3 – Результаты выполнения программы

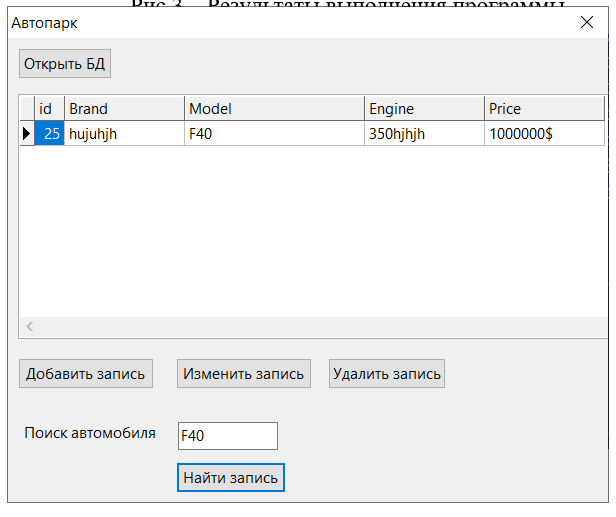


Рис 4 – Результаты выполнения программы

6. Вывод

В ходе данной работы были получены навыки работы с базами данных в среде Lazarus, SQLite и реализованы возможности добавления, удаления, изменения и поиска записей посредством программы, разработанной в Lazarus, в базе данных SQLite.

Благодаря всему вышеперечисленному работа была успешно выполнена.