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**

**«Основы событийно-ориентированного программирования»**

**ПО «МДК 05.02 РАЗРАБОТКА КОДА ИНФОРМАЦИОННЫХ СИСТЕМ»**

Выполнил: студент учебной группы

ИСПк-204-52-00

Вейс Роман Михайлович

Преподаватель:

Сергеева Елизавета Григорьевна

Киров

2024

1. Цель работы

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

2. Формулировка задания.

Вариант 1.

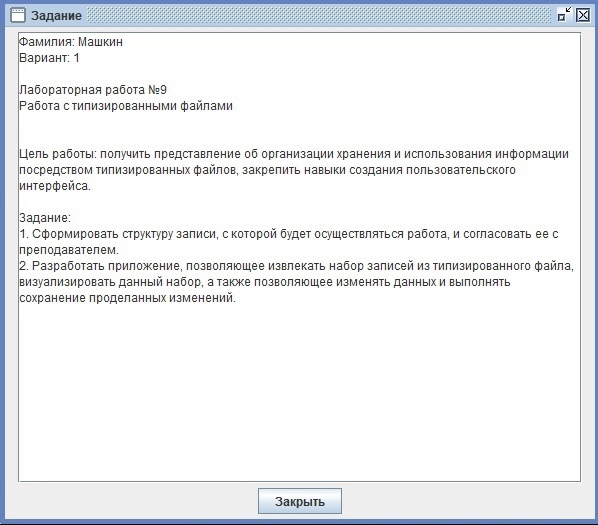


Рисунок 1 – Вариант работы.

3. Описание алгоритма.

Данный алгоритм предназначен для отрисовки программы по справочнику заказов.

1. Нажатие на кнопку Добавить программа откроет дополнительное окно с редактором заказа, где можно указать название напитка, его цену, выбрать один из 5 допингов, указать тип растительного молоки и указать что оплата картой, после нажатия на кнопку сохранить, в таблице добавляется новая строка куда вносятся данные, где и будут храниться.
2. Нажатие на кнопку редактирования заказа, открывает окно с редактированием заказа с данными последнего внесенного заказа.
3. Нажатие на кнопку Удалить, удаляет последний заказ и его строку.
4. Нажатие на кнопку Сортировать список, сортировка выполняется методом пузырька.

4. Код программы.

Задание 1.

Основной модуль.

unit Main;

{$mode objfpc}{$H+}

interface

uses

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

Buttons, edit;

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);

procedure Panel1Click(Sender: TObject);

private

public

end;

type

Contacts = record

Name: string[100];

Telephon: string[20];

Note: string[20];

DLC:Boolean;

Year:string[20];

end;

var

fMain: TfMain;

adres: string;

implementation

{$R \*.lfm}

{ TfMain }

procedure TfMain.bAddClick(Sender: TObject);

begin

fEdit.eName.Text := '';

fEdit.eTelephone.Text := '';

fEdit.eName.Text := '';

fEdit.eYear.Text := '';

fEdit.ModalResult := mrNone;

fEdit.ShowModal;

if (fEdit.eName.Text = '') or (fEdit.eTelephone.Text = '') then exit;

if fEdit.ModalResult <> mrOk then Exit;

SG.RowCount := SG.RowCount + 1;

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

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

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

if fEdit.CheckBox1.checked then SG.Cells[3,SG.RowCount-1] := 'Да' else SG.Cells[3,SG.RowCount-1] := 'Нет';

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

end;

procedure TfMain.bDelClick(Sender: TObject);

begin

if SG.RowCount = 1 then exit;

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

if SG.RowCount = 1 then exit;

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

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

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

fEdit.ModalResult:= mrNone;

fEdit.ShowModal;

if fEdit.ModalResult = mrOk then begin

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

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

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

if fEdit.CheckBox1.checked then SG.Cells[3,SG.RowCount-1] := 'Да' else SG.Cells[3,SG.RowCount-1] := 'Нет';

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

end;

end;

procedure TfMain.bSortClick(Sender: TObject);

begin

if SG.RowCount = 1 then exit;

SG.SortColRow(true, 0);

end;

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

var MyCont: Contacts;

f: file of Contacts;

i: integer;

begin

if SG.RowCount = 1 then exit;

try

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

Rewrite(f);

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

begin

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

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

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

if SG.Cells[3,i] = 'Да' then MyCont.DLC:=true else MyCont.DLC:=false;

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

Write(f,MyCont);

end;

finally

CloseFile(f);

end;

end;

procedure TfMain.FormCreate(Sender: TObject);

var

MyCont: Contacts;

f: file of Contacts;

i: integer;

begin

adres:= ExtractFilePath(ParamStr(0));

SG.Cells[0, 0]:= 'Название';

SG.Cells[1, 0]:= 'Цена';

SG.Cells[2, 0]:= 'Доппинг';

SG.Cells[3, 0]:= 'Оплата картой';

SG.Cells[4, 0]:= 'Растительное молоко';

SG.ColWidths[0]:= 365;

SG.ColWidths[1]:= 150;

SG.ColWidths[2]:= 150;

SG.ColWidths[3]:= 100;

SG.ColWidths[4]:= 150;

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

try

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

Reset(f);

while not Eof(f) do begin

Read(f, MyCont);

SG.RowCount:= SG.RowCount + 1;

SG.Cells[0, SG.RowCount-1]:= MyCont.Name;

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

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

if MyCont.DLC then SG.Cells[3, SG.RowCount-1]:= 'Да' else SG.Cells[3, SG.RowCount-1] := 'Нет';

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

end;

finally

CloseFile(f);

end;

end;

procedure TfMain.Panel1Click(Sender: TObject);

begin

end;

end.

Модуль дополнительного окна:

unit edit;

{$mode objfpc}{$H+}

interface

uses

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

type

{ TfEdit }

TfEdit = class(TForm)

bSave: TBitBtn;

bCancel: TBitBtn;

CBNote: TComboBox;

CheckBox1: TCheckBox;

eName: TEdit;

eTelephone: TEdit;

eYear: TEdit;

Label1: TLabel;

Label2: TLabel;

Label3: TLabel;

Label4: TLabel;

procedure bCancelClick(Sender: TObject);

procedure bSaveClick(Sender: TObject);

procedure FormCreate(Sender: TObject);

procedure FormShow(Sender: TObject);

procedure Label3Click(Sender: TObject);

procedure Label4Click(Sender: TObject);

private

public

end;

var

fEdit: TfEdit;

implementation

{$R \*.lfm}

{ TfEdit }

procedure TfEdit.FormCreate(Sender: TObject);

begin

end;

procedure TfEdit.bSaveClick(Sender: TObject);

begin

end;

procedure TfEdit.bCancelClick(Sender: TObject);

begin

end;

procedure TfEdit.FormShow(Sender: TObject);

begin

eName.SetFocus;

end;

procedure TfEdit.Label3Click(Sender: TObject);

begin

end;

procedure TfEdit.Label4Click(Sender: TObject);

begin

end;

end.

5. Вывод.

Выполняя домашнюю контрольную работу, мы смогли достичь поставленной цели: узнали, как работать с приложениями с графическим интерфейсом пользователя на основе событийно-ориентированной парадигмы.

В ходе работы над задание мы изучили принципы работы с IDLE Lazarus.