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-101-51-00

Бекмансуров Рустам Мхадтисович

Преподаватель:

Сергеева Елизавета Григорьевна

Киров

2024

**Цель:** Получить представления об организации хранения и использования информации посредством типизированных файлов, закрепить навыки создания пользовательского интерфейса.

**Формулировка задания:**

**Вариант 22**

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.

**Кода программы:**

**unit Main;**

**{$mode objfpc}{$H+}**

**interface**

**uses**

**Classes, SysUtils, SQLDB, DB, mysql55conn, mysql80conn, MSSQLConn, Forms,**

**Controls, Graphics, Dialogs, ExtCtrls, Buttons, Grids, edit;**

**type**

**{ TfMain }**

**TfMain = class(TForm)**

**ButtonPanel: 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);**

**private**

**public**

**end;**

**type**

**Contacts = record**

**Name: string[100];**

**Price: string[10];**

**Factory: string[100];**

**Calory: string[10];**

**HowMany: string[10];**

**FoodType: string[20];**

**end;**

**var**

**fMain: TfMain;**

**adres: string; //адрес, откуда запущена программа**

**implementation**

**{$R \*.lfm}**

**{ TfMain }**

**procedure TfMain.bAddClick(Sender: TObject);**

**begin**

**//очищаем поля, если там что-то есть:**

**fEdit.fName.Text:= '';**

**fEdit.fPrice.Text:= '';**

**fEdit.factoryName.Text:= '';**

**fEdit.Calory.Text:= '';**

**fEdit.HowMany.Text:= '';**

**fEdit.fType.Text:= '';**

**//устанавливаем ModalResult редактора в mrNone:**

**fEdit.ModalResult:= mrNone;**

**//теперь выводим форму:**

**fEdit.ShowModal;**

**//если пользователь ничего не ввел - выходим:**

**if (fEdit.fName.Text= '') or (fEdit.fPrice.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.fName.Text;**

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

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

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

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

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

**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 продукт "' +**

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

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

**SG.DeleteRow(SG.Row);**

**end;**

**procedure TfMain.bEditClick(Sender: TObject);**

**begin**

**//если данных в сетке нет - просто выходим:**

**if SG.RowCount = 1 then exit;**

**//иначе записываем данные в форму редактора:**

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

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

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

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

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

**fEdit.fType.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.fName.Text;**

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

**SG.Cells[2, SG.Row]:= fEdit.factoryName.text;**

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

**SG.Cells[4, SG.Row]:= fEdit.HowMany.text;**

**SG.Cells[5, SG.Row]:= fEdit.fType.Text;**

**end;**

**end;**

**procedure TfMain.bSortClick(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 + 'food.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.Price:= SG.Cells[1, i];**

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

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

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

**MyCont.FoodType:= 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));**

**//настроим сетку:**

**SG.Cells[0, 0]:= 'Наименование';**

**SG.Cells[1, 0]:= 'Стоимость';**

**SG.Cells[2, 0]:= 'Производитель';**

**SG.Cells[3, 0]:= 'Калорийность';**

**SG.Cells[4, 0]:= 'Количество';**

**SG.Cells[5, 0] := 'Тип продукта';**

**SG.ColWidths[0]:= 360;**

**SG.ColWidths[1]:= 100;**

**SG.ColWidths[2]:= 150;**

**SG.ColWidths[3]:= 110;**

**SG.ColWidths[4]:= 100;**

**SG.ColWidths[5]:= 130;**

**//если файла данных нет, просто выходим:**

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

**//иначе файл есть, открываем его для чтения и**

**//считываем данные в сетку:**

**try**

**AssignFile(f, adres + 'food.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.Price;**

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

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

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

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

**end;**

**finally**

**CloseFile(f);**

**end;**

**end;**

**end.**

**unit Edit;**

**{$mode ObjFPC}{$H+}**

**interface**

**uses**

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

**MaskEdit;**

**type**

**{ TfEdit }**

**TfEdit = class(TForm)**

**bCancel: TBitBtn;**

**bSave: TBitBtn;**

**Calory: TEdit;**

**factoryName: TEdit;**

**fPrice: TEdit;**

**fType: TComboBox;**

**fName: TEdit;**

**fNameLabel: TLabel;**

**fPriceLabel: TLabel;**

**fTypeLabel: TLabel;**

**factoryNameLabel: TLabel;**

**CaloryLabel: TLabel;**

**HowMany: TEdit;**

**HowManyLabel: TLabel;**

**procedure CaloryKeyPress(Sender: TObject; var Key: char);**

**procedure FormShow(Sender: TObject);**

**procedure fPriceKeyPress(Sender: TObject; var Key: char);**

**procedure HowManyKeyPress(Sender: TObject; var Key: char);**

**private**

**public**

**end;**

**var**

**fEdit: TfEdit;**

**implementation**

**{$R \*.lfm}**

**{ TfEdit }**

**procedure TfEdit.FormShow(Sender: TObject);**

**begin**

**fName.SetFocus;**

**end;**

**procedure TfEdit.CaloryKeyPress(Sender: TObject; var Key: char);**

**begin**

**// проверяем нажатую клавишу**

**case Key of**

**// цифры разрешаем**

**'0'..'9': key:=key;**

**// разрешаем десятичный разделитель (только точку)**

**'.', ',': key:='.';**

**// разрешаем BackSpace**

**#8: key:=key;**

**// все прочие клавиши "гасим"**

**else key:=#0;**

**end;**

**end;**

**procedure TfEdit.fPriceKeyPress(Sender: TObject; var Key: char);**

**begin**

**// проверяем нажатую клавишу**

**case Key of**

**// цифры разрешаем**

**'0'..'9': key:=key;**

**// разрешаем десятичный разделитель (только точку)**

**'.', ',': key:='.';**

**// разрешаем BackSpace**

**#8: key:=key;**

**// все прочие клавиши "гасим"**

**else key:=#0;**

**end;**

**end;**

**procedure TfEdit.HowManyKeyPress(Sender: TObject; var Key: char);**

**begin**

**// проверяем нажатую клавишу**

**case Key of**

**// цифры разрешаем**

**'0'..'9': key:=key;**

**// разрешаем десятичный разделитель (только точку)**

**'.', ',': key:='.';**

**// разрешаем BackSpace**

**#8: key:=key;**

**// все прочие клавиши "гасим"**

**else key:=#0;**

**end;**

**end;**

**end.**

Результаты выполнения задания.

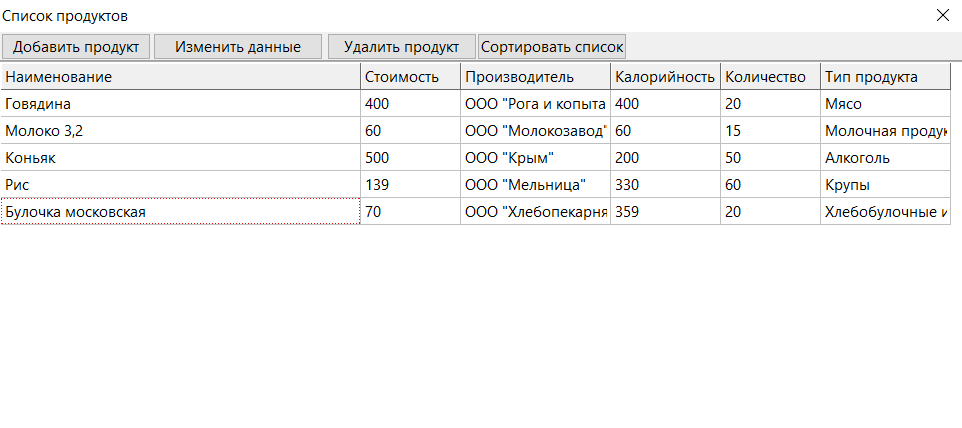


Рисунок 1 – результат выполнения задания

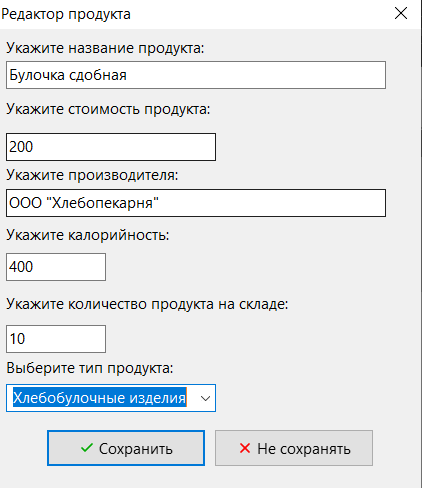
****

Рисунок 2 – результат выполнения задания

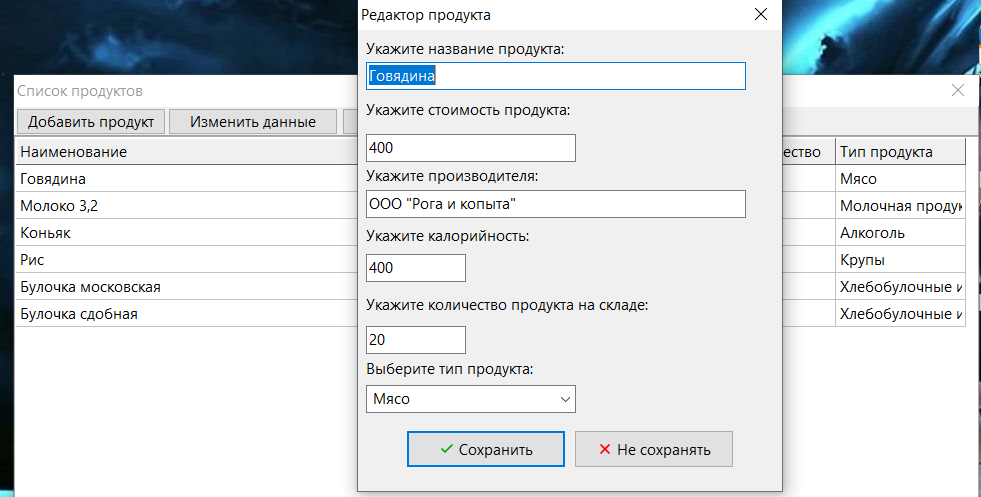
****

Рисунок 3 – результат выполнения задания

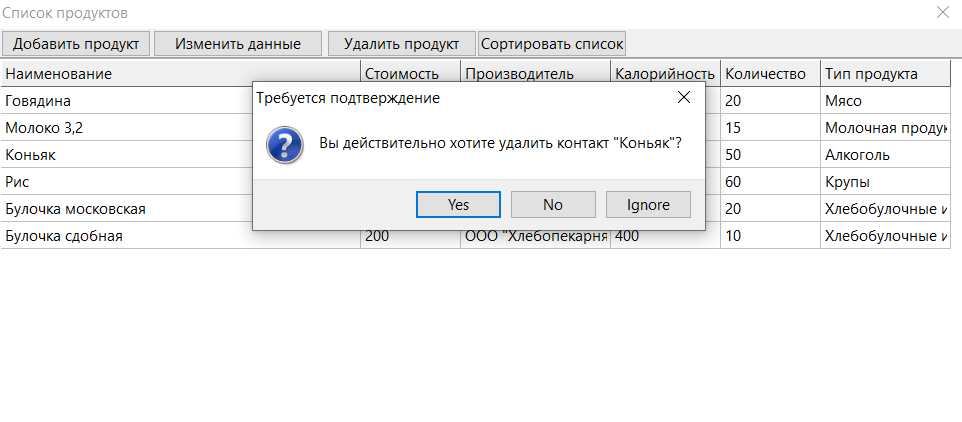
****

Рисунок 4 – результат выполнения задания

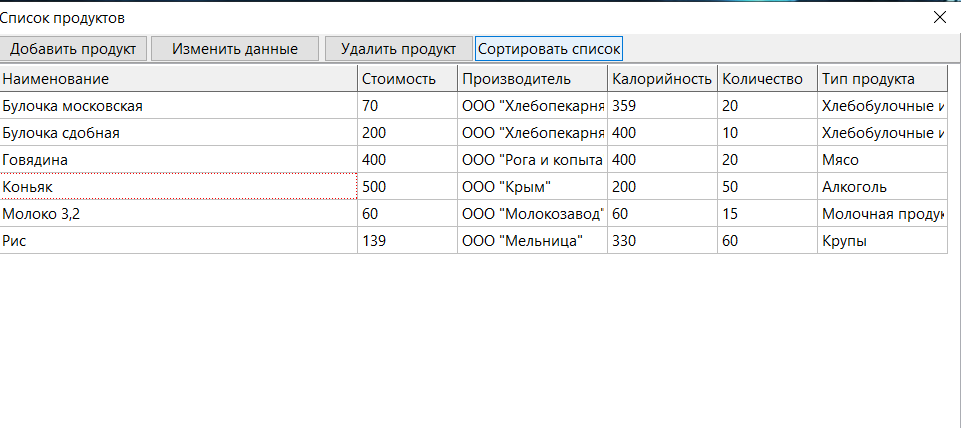
****

Рисунок 5 – результат выполнения задания

**Вывод:** в процессе выполнения задания проблем не возникло, была реализована программа для записи данных о продуктах с возможностью изменения и удаления данных. Были получены навыки реализации приложений с хранением данных в типизированных файлах. Закреплены навыки создания пользовательского интерфейса.