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-203-52-00

Обухов Илья Николаевич

Преподаватель:

Сергеева Елизавета Григорьевна

Киров

2024

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

**Задание: (Танки)**

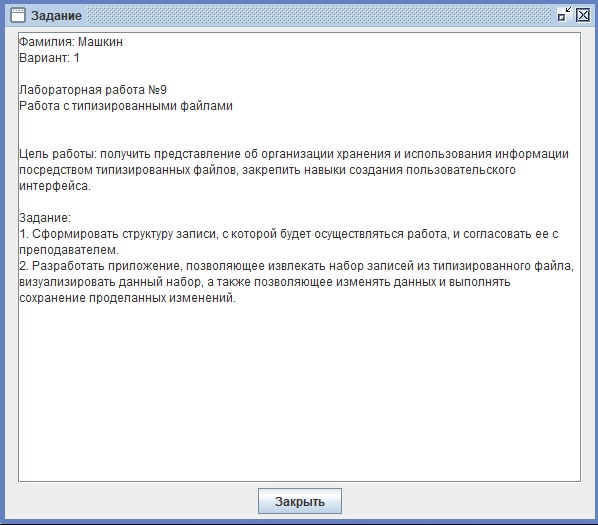


Рисунок 1 – Задание

**Код 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];**

**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:= '';**

**//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;**

**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',**

**'Удалить запись "' +**

**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];**

**//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;**

**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];**

**//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]:= 'МОДЕЛЬ';**

**SG.Cells[1, 0]:= 'ДАТА ВЫПУСКА';**

**SG.Cells[2, 0]:= 'СТРАНА';**

**SG.Cells[3, 0]:= 'ОБЪЕМ ДВИГАТЕЛЬ';**

**SG.Cells[4, 0]:= 'МАКСИМАЛЬНАЯ СКОРОСТЬ';**

**SG.ColWidths[0]:= 200;**

**SG.ColWidths[1]:= 130;**

**SG.ColWidths[2]:= 100;**

**SG.ColWidths[3]:= 160;**

**SG.ColWidths[4]:= 220;**

**//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;**

**end;**

**finally**

**CloseFile(f);**

**end;**

**end;**

**end.**

**Результаты выполнения программы:**

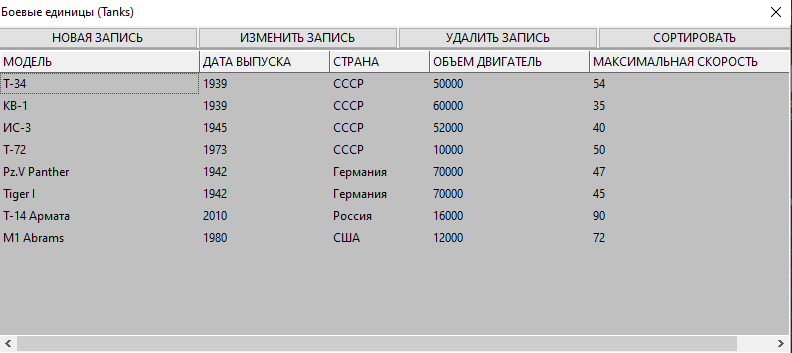
****

Рисунок 2 – Список танкоу

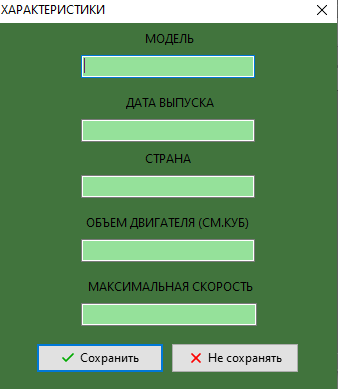


Рис. 3 – Добавление новых танкоу

**Вывод:**

В ходе выполнения домашней контрольной работы номер 9 мы столкнулись с различными рода задачами. Но к счастью, все поставленные цели были достигнуты и задачи выполнены. Проблем при выполнении домашней контрольной работы номер 9 не возникало, если не брать в расчёт то, что, когда мы открыли файл с заданием мы очень удивились размеру домашней контрольной работы номер 9. Это нагоняло какого-то страха и смятения, но как говорит Павел Техник: «глаза боятся - руки делают». Опираясь на это, и не оставляя без внимания всё остальное домашнюю контрольную работу номер 9 можно считать полностью выполненной.