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-203-52-00

Леушина Анна Станиславовна

Преподаватель:

Сергеева Елизавета Григорьевна

Киров

2024

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

**2. Формулировка задания (Вариант: 15)**

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 данные и выполнять сохранение проделанных изменений.

**3**. **Описание алгоритма**

Для реализации интерфейса использовались данные компоненты:

* TLabel – компонент для отображения текста на форме.
* TEdit – компонент, который позволяет пользователю вводить текст.
* TCombobox – раскрывающийся список, из которого пользователь может выбирать один из предопределенных вариантов.
* TBitBtn – компонент кнопки с возможностью отображения изображения на ней.
* TPanel – область на форме, используемая для размещения других элементов управления;
* TSpeedButton – кнопка с изображением, обычно используемая для выполнения быстрых действий;
* TStringGrid представляет собой элемент управления сеткой, который позволяет отображать и редактировать данные в виде сетки строк и столбцов.

Записи хранятся в строковой таблице (StringGrid) и могут быть добавлены, удалены, отредактированы и отсортированы.

В работе были использованы данные типы данных:

* строковый тип (string[100] и string[50]) для текстовых данных, таких как имена, стили и техники, чтобы контролировать размер хранимых значений и использовать память эффективно;
* целочисленный тип (integer) для года, так как он представляет целое число;
* вещественный тип (real) для стоимости, поскольку она может содержать десятичные значения.

**4. Код программы:**

unit Main;

{$mode objfpc}{$H+}

interface

uses

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

Buttons, edit, Grids;

type

Contacts = record

nameK: string[100];

nameH: string[100];

Style: string[100];

Tehn: string[50];

Year: integer;

Place: string[20];

stoim: real;

end; //record

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);

private

public

end;

var

fMain: TfMain;

adres: string; //адрес, откуда запущена программа

implementation

{$R \*.lfm}

{ TfMain }

procedure TfMain.bAddClick(Sender: TObject);

var

MyCont: Contacts;

begin

fEdit.eName.Text:= '';

fEdit.eNameH.Text:= '';

fEdit.eYear.Text:= '';

fEdit.eStoim.Text:= '';

fEdit.ModalResult:= mrNone;

fEdit.ShowModal;

if (fEdit.eName.Text= '') or (fEdit.eNameH.Text= '') or

(fEdit.eYear.Text= '') or (fEdit.eStoim.Text= '') then exit;

if fEdit.ModalResult <> mrOk then exit;

MyCont.nameK := fEdit.eName.Text;

MyCont.nameH := fEdit.eNameH.Text;

MyCont.Style := fEdit.CBStyle.Text;

MyCont.Tehn := fEdit.CBTehn.Text;

MyCont.Year := StrToInt(fEdit.eYear.Text);

MyCont.Place := fEdit.CBPlace.Text;

MyCont.stoim := StrToFloat(fEdit.eStoim.Text);

SG.RowCount:= SG.RowCount + 1;

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

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

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

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

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

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

SG.Cells[6, SG.RowCount-1]:= FloatToStr(MyCont.stoim);

end;

procedure TfMain.bDelClick(Sender: TObject);

begin

if SG.RowCount = 1 then exit;

if MessageDlg('Требуется подтверждение',

'Вы действительно хотите удалить "' +

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

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

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

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

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

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

fEdit.eStoim.Text:= SG.Cells[6, 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.eName.Text;

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

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

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

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

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

SG.Cells[6, SG.Row]:= fEdit.eStoim.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 + 'katalog.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.nameK:= SG.Cells[0, i];

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

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

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

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

MyCont.Place:= SG.Cells[5, i];

MyCont.stoim:= StrToFloat(SG.Cells[6, 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]:= 'Название';

SG.Cells[1, 0]:= 'Художник';

SG.Cells[2, 0]:= 'Стиль';

SG.Cells[3, 0]:= 'Техника';

SG.Cells[4, 0]:= 'Год создания';

SG.Cells[5, 0]:= 'В музее';

SG.Cells[6, 0]:= 'Стоимость';

SG.ColWidths[0]:= 240;

SG.ColWidths[1]:= 240;

SG.ColWidths[2]:= 240;

SG.ColWidths[3]:= 240;

SG.ColWidths[4]:= 120;

SG.ColWidths[5]:= 100;

SG.ColWidths[6]:= 120;

//если файла данных нет, просто выходим:

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

//иначе файл есть, открываем его для чтения и

//считываем данные в сетку:

try

AssignFile(f, adres + 'katalog.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.nameK;

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

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

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

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

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

SG.Cells[6, SG.RowCount-1]:= FloatToStr(MyCont.stoim);

end;

finally

CloseFile(f);

end;

end;

end.

**5. Результат выполнения программы**

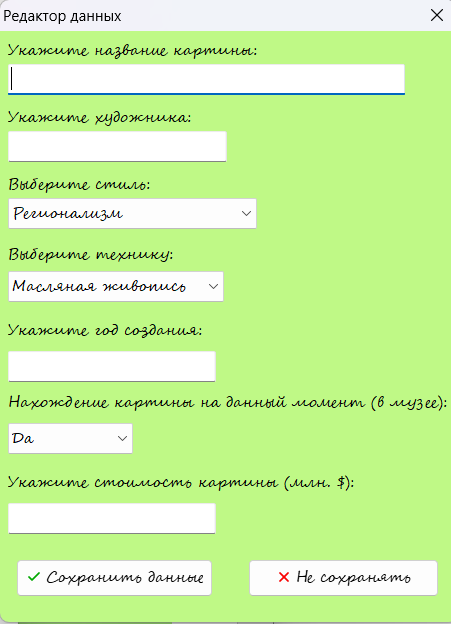


Рисунок 1 – Добавление данных в таблицу

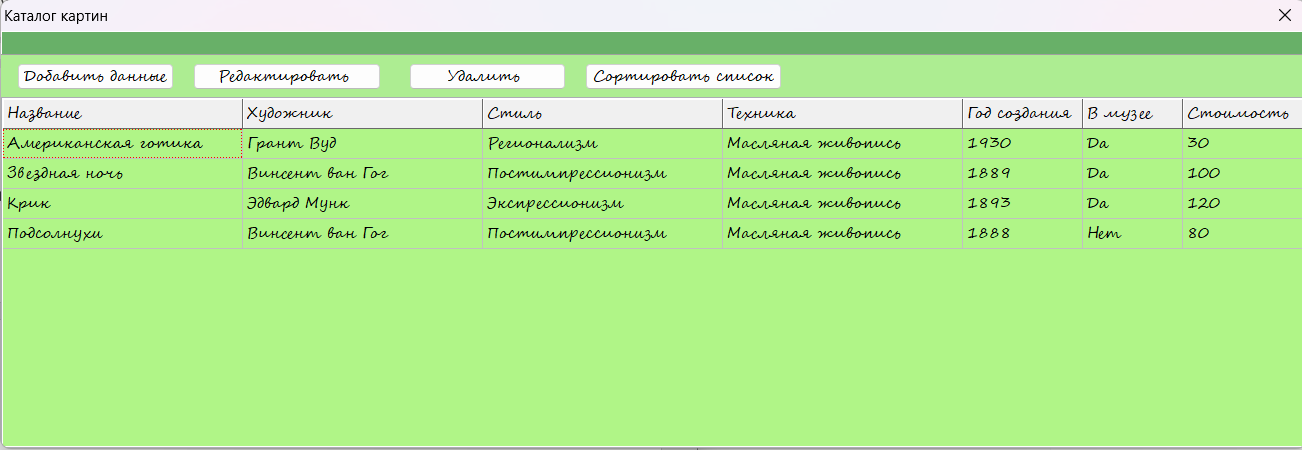


Рисунок 2 – Результат работы (каталог картин)

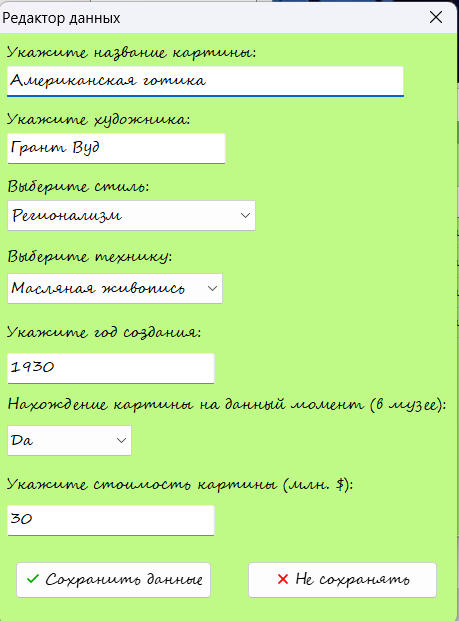


Рисунок 3 – Редактор для уже созданных записей

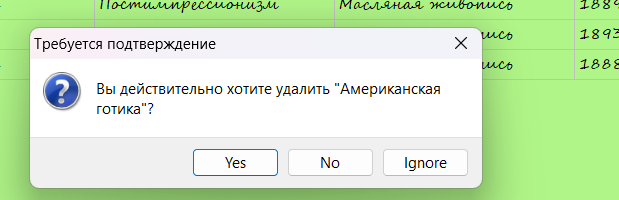


Рисунок 4 – Сообщение перед удалением записи

**6. Вывод**

В домашней контрольной работе №9 было реализовано приложение, позволяющее извлекать набор записей из типизированного файла, визуализировать данный набор, а также позволяющее изменять данные и выполнять сохранение проделанных изменений. В ходе выполнения работы затруднений не возникло, все поставленные задачи были выполнены в полном объеме.