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»**

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

Выполнила: студентка учебной группы

ИСПк-203-52-00

Бочкарёва Виктория Дмитриевна

Преподаватель:

Сергеева Елизавета Григорьевна

Киров

2024

**Цель работы**

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

**Формулировка задания**

Задание:

1. Сформировать структуру записи, с которой будет осуществляться работа, и согласовать её 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.

**Описание алгоритма**

Рассмотрим основные части кода и их алгоритмы:

1. Процедура FormCreate:
   * Получает путь к директории, где находится приложение
   * Настраивает заголовки и ширину столбцов в StringGrid
   * Проверяет наличие файла данных Barbie\_Molds.dat
   * Если файл существует, то открывает его для чтения
   * В цикле читает записи из файла и добавляет их в StringGrid
2. Процедура FormClose:
   * Проверяет, есть ли данные в StringGrid
   * Если есть данные, то открывает файл Barbie\_Molds.dat для записи
   * В цикле читает данные из StringGrid и записывает их в файл
3. Процедура bAddClick:
   * Очищает поля в форме редактирования
   * Показывает форму редактирования в модальном режиме
   * Если пользователь ввел данные и нажал "Сохранить", добавляет новую строку в StringGrid с введенными данными
4. Процедура bDelClick:
   * Проверяет, есть ли данные в StringGrid
   * Если есть данные, то показывает диалог подтверждения удаления
   * Если пользователь подтвердил удаление, удаляет выбранную строку из StringGrid
5. Процедура bEditClick:
   * Проверяет, есть ли данные в StringGrid
   * Если есть данные, то заполняет поля в форме редактирования данными из выбранной строки StringGrid
   * Показывает форму редактирования в модальном режиме
   * Если пользователь нажал "Сохранить", обновляет данные в выбранной строке StringGrid
6. Процедура bSortClick:
   * Проверяет, есть ли данные в StringGrid
   * Определяет индекс выбранного столбца для сортировки из ComboBox
   * Сортирует данные в StringGrid по выбранному столбцу
7. Функция BoolToStrYesNo:
   * Принимает логическое значение
   * Возвращает строку "да" или "нет" в зависимости от логического значения

**Код программы**

Модуль Main:

unit Main;

{$mode objfpc}{$H+}

interface

uses

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

Grids, StdCtrls, AnchorDockPanel, edit;

type

Molds = record

Mold: string[50];

Year: integer;

Limited: boolean;

end; //record

{ TfMain }

TfMain = class(TForm)

bAdd: TBitBtn;

bEdit: TBitBtn;

bDel: TBitBtn;

bSort: TBitBtn;

cbSort: TComboBox;

Image1: TImage;

Panel1: TPanel;

SG: TStringGrid;

procedure bDelClick(Sender: TObject);

procedure bEditClick(Sender: TObject);

procedure bSortClick(Sender: TObject);

function BoolToStrYesNo(Value: Boolean): string;

procedure bAddClick(Sender: TObject);

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

procedure FormCreate(Sender: TObject);

private

public

end;

var

fMain: TfMain;

adres: string; //адрес, откуда запущена программa

implementation

{$R \*.lfm}

{ TfMain }

function TfMain.BoolToStrYesNo(Value: Boolean): string;

begin

if Value then

Result := 'да'

else

Result := 'нет';

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

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

if SG.Cells[2, SG.RowCount-1] = 'да'

then fEdit.ChB.Checked := True

else

fEdit.ChB.Checked := False;

//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.eMold.Text;

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

if fEdit.ChB.Checked then

SG.Cells[2, SG.Row] := 'да'

else

SG.Cells[2, SG.Row] := 'нет';

end;

end;

procedure TfMain.bSortClick(Sender: TObject);

var

SelectedColumn: Integer;

begin

// Проверяем, есть ли данные для сортировки

if SG.RowCount <= 1 then Exit;

// Определяем индекс выбранного столбца по названию

SelectedColumn := cbSort.Items.IndexOf(cbSort.Text);

// Сортируем данные в StringGrid по выбранному столбцу

SG.SortColRow(True, SelectedColumn);

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 запись "' +

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

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

SG.DeleteRow(SG.Row);

end;

procedure TfMain.bAddClick(Sender: TObject);

begin

//очищаем поля, если там что-то есть:

fEdit.eMold.Text:= '';

fEdit.eYear.Text:= '';

//устанавливаем ModalResult редактора в mrNone:

fEdit.ModalResult:= mrNone;

//теперь выводим форму:

fEdit.ShowModal;

//если пользователь ничего не ввел - выходим:

if (fEdit.eMold.Text= '') or (fEdit.eYear.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.eMold.Text;

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

if fEdit.ChB.Checked then

SG.Cells[2, SG.RowCount-1] := 'да'

else

SG.Cells[2, SG.RowCount-1] := 'нет';

end;

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

var

MyCont: Molds; //для очередной записи

f: file of Molds; //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 + 'Barbie\_Molds.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.Mold:= SG.Cells[0, i];

MyCont.Year:= strtoint(SG.Cells[1, i]);

MyCont.Limited:= SG.Cells[2, i] = 'да';

//записываем их:

Write(f, MyCont);

end;

finally

CloseFile(f);

end;

end;

procedure TfMain.FormCreate(Sender: TObject);

var

MyCont: Molds; //для очередной записи

f: file of Molds; //файл данных

i: integer; //счетчик цикла

begin

//сначала получим адрес программы:

adres:= ExtractFilePath(ParamStr(0));

//настроим сетку:

SG.Cells[0, 0]:= 'Название Молда';

SG.Cells[1, 0]:= 'Год выпуска';

SG.Cells[2, 0]:= 'Эксклюзив';

SG.ColWidths[0]:= 295;

SG.ColWidths[1]:= 150;

SG.ColWidths[2]:= 100;

//если файла данных нет, просто выходим:

if not FileExists(adres + 'Barbie\_Molds.dat') then exit;

//иначе файл есть, открываем его для чтения и

//считываем данные в сетку:

try

AssignFile(f, adres + 'Barbie\_Molds.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.Mold;

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

SG.Cells[2, SG.RowCount-1]:= booltostryesno(MyCont.Limited);

end;

finally

CloseFile(f);

end;

end;

end.

Модуль Edit:

unit edit;

{$mode ObjFPC}{$H+}

interface

uses

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

MaskEdit;

type

Barbie = record

Mold: string[50];

Year: integer;

Limited: boolean;

end;

{ TfEdit }

TfEdit = class(TForm)

BitBtn1: TBitBtn;

BitBtn2: TBitBtn;

ChB: TCheckBox;

eMold: TEdit;

Label1: TLabel;

Label2: TLabel;

Label3: TLabel;

eYear: TMaskEdit;

procedure FormShow(Sender: TObject);

private

public

end;

var

fEdit: TfEdit;

implementation

{$R \*.lfm}

{ TfEdit }

procedure TfEdit.FormShow(Sender: TObject);

begin

eMold.SetFocus;

end;

end.

**Результат выполнения программы**

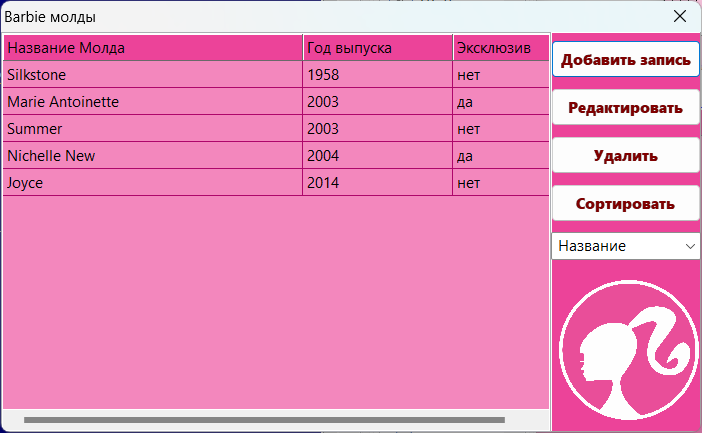


Рисунок 1 – Результат выполнения программы

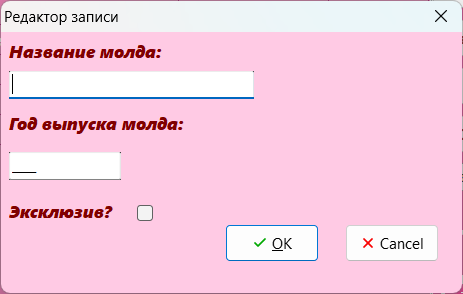


Рисунок 2 – Результат выполнения программы

**Вывод**

В результате выполнения домашней контрольной работы поставленная цель была достигнута. Были получены представления об организации хранения и использования информации посредством типизированных файлов, закреплены навыки создания пользовательского интерфейса.

В процессе работы была написана программа, которая дает пользователю вносить, изменять, удалять и просматривать записи из набора записей. Для осуществления графического интерфейса пользователя используются кнопки, метки, поле для ввода, изображение, сетка.

Таким образом, решив все поставленные задачи, а также получив новые знания, были достигнута поставленная цель, получены удовлетворяющие результаты.