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

Вальковская Арина Константиновна

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: 4)**

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**

В разработанном интерфейсе использовались данные компоненты:

* TLabel: Для отображения текстовых надписей, пояснений и заголовков.
* TEdit: Для ввода текстовой информации пользователем.
* TCombobox: Для предоставления пользователю выбора из заранее определенного списка значений.
* TBitBtn: Для создания кнопок с текстом и/или изображениями, запускающих определенные действия.
* TPanel: Для структурирования формы и группировки элементов управления.
* TSpeedButton: Для создания кнопок с изображениями, которые часто используются для быстрых действий.
* TStringGrid: Для организации и редактирования данных в виде таблицы с строками и столбцами. Данные, хранящиеся в таблице, могут быть добавлены, удаленя, отредактированы и отсортированы.

Также были использованы следующие типы данных:

* Строковый тип (string): Для хранения текстовой информации, такой как названия книг, имен авторов и изданий.
* Целочисленный тип (integer): Для хранения ISBN, так как это целочисленное значение.
* Вещественный тип (real): Для хранения количества страниц в книге.

**4. Код программы:**

unit Edit;

{$mode objfpc}{$H+}

interface

uses

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

Grids, Unit2;

type

{ TfMain }

TfMain = class(TForm)

Panel1: TPanel;

bAdd: TSpeedButton;

dEdit: TSpeedButton;

bDel: TSpeedButton;

bSort: TSpeedButton;

SG: TStringGrid;

procedure bAddClick(Sender: TObject);

procedure bDelClick(Sender: TObject);

procedure bSortClick(Sender: TObject);

procedure dEditClick(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];

Avtor: string[40];

Note: string[40];

ISBN: integer;

Izdat: string[100];

Stran: real;

end; //record

var

fMain: TfMain;

adres: string; //адрес, откуда запущена программа

implementation

{$R \*.lfm}

{ TfMain }

procedure TfMain.Panel1Click(Sender: TObject);

begin

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 книги';

SG.Cells[1, 0]:= 'Автор';

SG.Cells[2, 0]:= 'Жанр';

SG.Cells[3, 0]:= 'ISBN';

SG.Cells[4, 0]:= 'Издательство';

SG.Cells[5, 0]:= 'Количество страниц';

SG.ColWidths[0]:= 300;

SG.ColWidths[1]:= 230;

SG.ColWidths[2]:= 250;

SG.ColWidths[3]:= 150;

SG.ColWidths[4]:= 230;

SG.ColWidths[5]:= 200;

//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.Name;

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

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

SG.Cells[3, SG.RowCount-1]:= IntToStr(MyCont.ISBN);

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

SG.Cells[5, SG.RowCount-1]:= FloatToStr(MyCont.Stran);

end;

finally

CloseFile(f);

end;

end;

procedure TfMain.bAddClick(Sender: TObject);

begin

//очищаем поля, если там что-то есть:

fEdit.eName.Text:= '';

fEdit.eAvtor.Text:= '';

fEdit.eISBN.Text:= '';

fEdit.eIzdat.Text:= '';

fEdit.eStran.Text:= '';

//устанавливаем ModalResult редактора в mrNone:

fEdit.ModalResult:= mrNone;

//теперь выводим форму:

fEdit.ShowModal;

//если пользователь ничего не ввел - выходим:

if (fEdit.eName.Text= '') or (fEdit.eAvtor.Text= '') or (fEdit.eISBN.Text= '') or (fEdit.eIzdat.Text= '') or (fEdit.eStran.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.eName.Text;

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

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

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

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

SG.Cells[5, SG.RowCount-1]:= fEdit.eStran.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 данные "' +

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

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

SG.DeleteRow(SG.Row);

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.dEditClick(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.eAvtor.Text:= SG.Cells[1, SG.Row];

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

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

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

fEdit.eStran.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.eName.Text;

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

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

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

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

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

end;

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

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

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

MyCont.ISBN:= StrToInt(SG.Cells[3, i]);

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

MyCont.Stran:= StrToFloat(SG.Cells[5, i]);

//записываем их:

Write(f, MyCont);

end;

finally

CloseFile(f);

end;

end;

end.

**5. Результат выполнения программы**

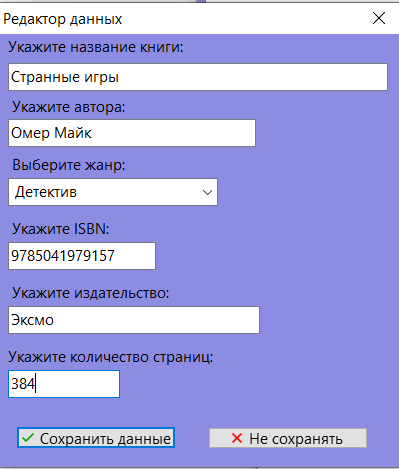
****

Рисунок 1 – Добавление данных

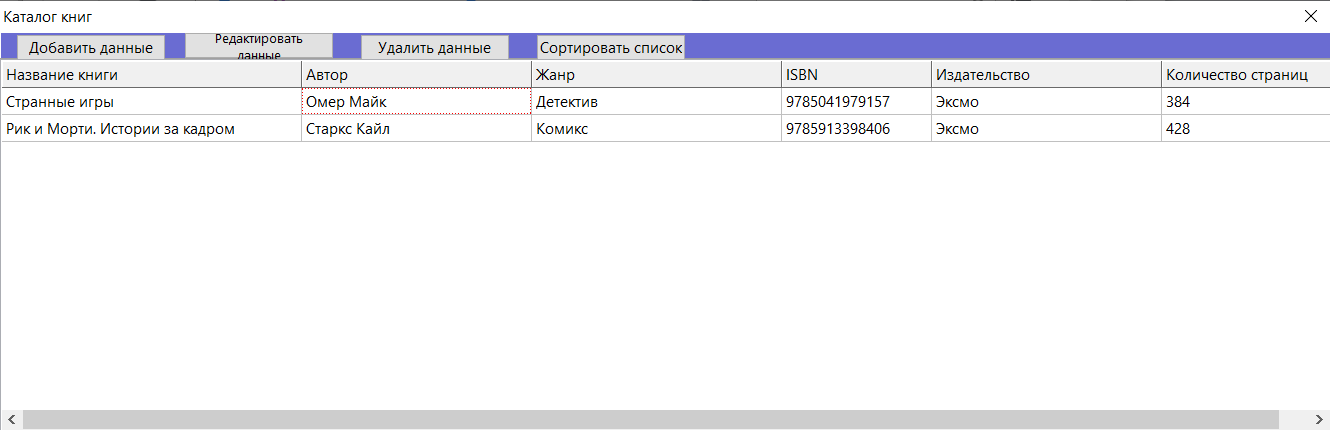


Рисунок 2 – Добавленные данные в таблицу

**6. Вывод**

Контрольная работа №9 была успешно выполнена. В рамках задания было разработано приложение, способное извлекать данные из типизированного файла, отображать их в удобной форме, а также позволять вносить изменения и сохранять их. В процессе выполнения работы не возникло сложностей, и все поставленные задачи были выполнены в полном объеме.