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**

**«РАБОТЫ С ТИПИЗИРОВАННЫМИ ФАЙЛАМИ»**

**ПО**

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

Выполнила: студентка учебной группы

ИСПк-202-52-00

Широнина Анна Андреевна

Преподаватель:

Сергеева Елизавета Григорьевна

Киров

2025

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

1. **Формулировка заданий. Вариант 0**

Задание:

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.

1. **Описание приложения**

Был создан справочник для записи различных позиций одежды

Для отображения самых форм используется компонент TfMain и TfEdit.

Для кнопок «Добавить», «Редактировать», «Удалить», «Сохранить», «Открыть» был использован компонент TSpeedButton, а располагаются эти кнопки на панели TPanel.

Для формирования самой таблицы в справочнике используется TStringGrid и TGridColumns.

Для сохранения файла используется диалоговый компонент TSaveDialog, а для открытия файла – TOpenDialog.

Также для фона был использован компонент для изображения TImage.

Для подписей к полям ввода используется компонент TLabel.

Для самих полей вывода используются элементы: TEdit, TDateTimePicker, TComboBox.

Для кнопок «Сохранить» и «Не сохранять» запись используются кнопки TBitBit.

1. **Код программы (файл Main.pas)**

unit Main;

{$mode objfpc}{$H+}

interface

uses

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

Grids,edit;

type

Item = record

Name: string[100];

Date1: string[20];

Price: string[20];

Country:string[40];

tip: string[40];

bir: string[40];

sostav:string[40];

end; //record

{ TfMain }

TfMain = class(TForm)

bAdd1: TSpeedButton;

Image1: TImage;

OpenDialog1: TOpenDialog;

Panel1: TPanel;

bEdit: TSpeedButton;

bDel: TSpeedButton;

bSave: TSpeedButton;

SaveDialog1: TSaveDialog;

bOpen: TSpeedButton;

SG:TStringGrid;

procedure bAdd1Click(Sender: TObject);

procedure bDelClick(Sender: TObject);

procedure bEditClick(Sender: TObject);

procedure bOpenClick(Sender: TObject);

procedure bSaveClick(Sender: TObject);

procedure FormCreate(Sender: TObject);

procedure SaveItemDataToFile(const FileName: string);

procedure OpenItemDataFromFile(const FileName: string);

private

public

end;

var

fMain: TfMain;

adres: string;

implementation

{$R \*.lfm}

{ TfMain }

procedure TfMain.bAdd1Click(Sender: TObject);

begin

//очищаем поля, если там что-то есть:

fEdit.eName.Text:= '';

fEdit.eSrok.Text:= '';

fEdit.eCountry.Text:='';

fEdit.Sostav.Text:='';

fEdit.CBNote.Text:='';

fEdit.Bir.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.eSrok.Text= '') or (fEdit.eCountry.Text= '') or (fEdit.Sostav.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]:= DatetoStr(fEdit.Date1.Date);

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

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

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

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

SG.Cells[6, SG.RowCount-1]:= fEdit.Sostav.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 контакт "' +

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.Date1.Date:= StrtoDate(SG.Cells[1, SG.Row]);

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

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

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

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

fEdit.Sostav.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.RowCount-1]:= fEdit.eName.Text;

SG.Cells[1, SG.RowCount-1]:= DatetoStr(fEdit.Date1.Date);

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

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

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

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

SG.Cells[6, SG.RowCount-1]:= fEdit.Sostav.Text;

end;

end;

procedure TfMain.bOpenClick(Sender: TObject);

begin

// Настраиваем диалог сохранения

OpenDialog1.Title := 'Открыть данные о вещи';

OpenDialog1.DefaultExt := 'dat';

OpenDialog1.Filter := 'Файлы данных (\*.dat)|\*.dat|Все файлы (\*.\*)|\*.\*';

OpenDialog1.FileName := 'items.dat';

if OpenDialog1.Execute then

begin

OpenItemDataFromFile(OpenDialog1.FileName);

ShowMessage('Данные успешно сохранены в файл: ' + OpenDialog1.FileName);

end;

end;

procedure TfMain.bSaveClick(Sender: TObject);

begin

// Настраиваем диалог сохранения

SaveDialog1.Title := 'Сохранить данные о вещи';

SaveDialog1.DefaultExt := 'dat';

SaveDialog1.Filter := 'Файлы данных (\*.dat)|\*.dat|Все файлы (\*.\*)|\*.\*';

SaveDialog1.FileName := 'items.dat';

// Показываем диалог и сохраняем данные

if SaveDialog1.Execute then

begin

SaveItemDataToFile(SaveDialog1.FileName);

ShowMessage('Данные успешно сохранены в файл: ' + SaveDialog1.FileName);

end;

end;

procedure TfMain.FormCreate(Sender: TObject);

begin

SG.Cells[0, 0]:= 'Название';

SG.Cells[1, 0]:= 'Дата изготовления';

SG.Cells[2, 0]:= 'Цена в рублях';

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

SG.Cells[4, 0]:= 'Тип';

SG.Cells[5, 0]:= 'Присутствие бирки';

SG.Cells[6, 0]:= 'Состав';

SG.ColWidths[0]:= 200;

SG.ColWidths[1]:= 150;

SG.ColWidths[2]:= 150;

SG.ColWidths[3]:= 150;

SG.ColWidths[4]:= 200;

SG.ColWidths[5]:= 150;

end;

procedure TfMain.SaveItemDataToFile(const FileName: string);

var

f: file of Item;

i: integer;

MyItem: Item;

begin

if SG.RowCount = 1 then exit;

try

assignfile(f, FileName);

rewrite(f);

for i:=1 to SG.RowCount-1 do

begin

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

MyItem.Date1 := SG.Cells[1, i];

MyItem.Price := SG.Cells[2, i];

MyItem.Country := SG.Cells[3, i];

MyItem.tip := SG.Cells[4, i];

MyItem.bir := SG.Cells[5, i];

MyItem.sostav := SG.Cells[6, i];

Write(f, MyItem);

end;

finally

CloseFile(f);

end;

end;

procedure TfMain.OpenItemDataFromFile(const FileName: string);

var

f: file of Item;

MyItem: Item;

begin

if not FileExists(FileName) then exit;

try

AssignFile(f, FileName);

Reset(f);

SG.RowCount := 1;

while not Eof(f) do

begin

Read(f, MyItem);

SG.RowCount := SG.RowCount + 1;

SG.Cells[0, SG.RowCount-1] := MyItem.Name;

SG.Cells[1, SG.RowCount-1] := MyItem.Date1;

SG.Cells[2, SG.RowCount-1] := MyItem.Price;

SG.Cells[3, SG.RowCount-1] := MyItem.Country;

SG.Cells[4, SG.RowCount-1] := MyItem.tip;

SG.Cells[5, SG.RowCount-1] := MyItem.bir;

SG.Cells[6, SG.RowCount-1] := MyItem.sostav;

end;

finally

CloseFile(f);

end;

end;

end.

1. **Результат выполнения программы**

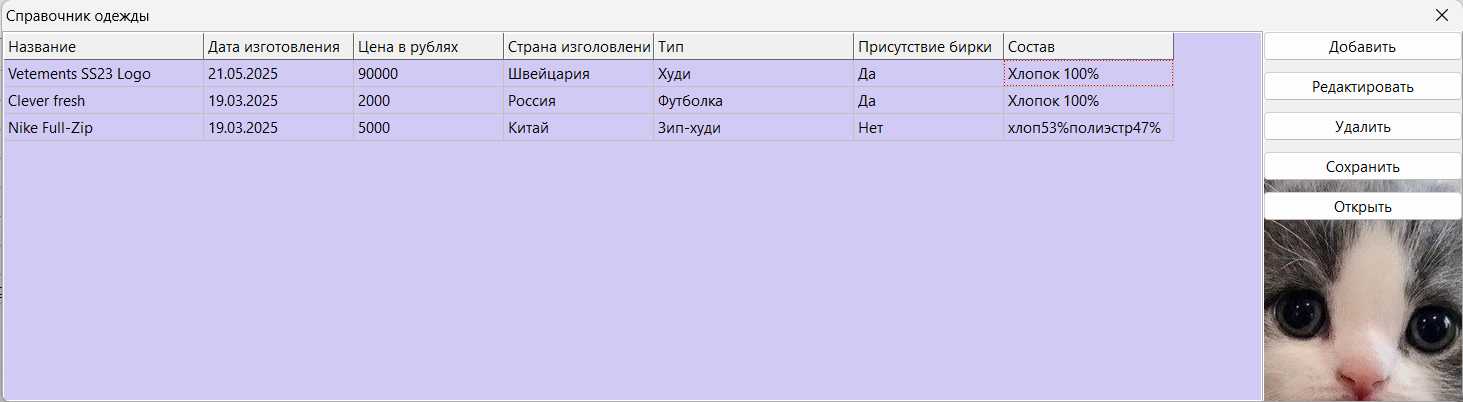


Рисунок 1 – Результат выполнения программы главный экран

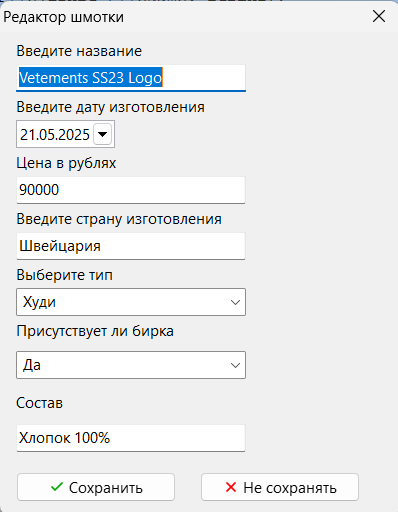


Рисунок 2 – Экран редактирования записей

1. **Вывод**

В ходе выполнения домашней контрольной работы было разработано приложение для работы с типизированными файлами, представляющее собой справочник позиций одежды. Цель работы была успешно достигнута.

В процессе работы были изучены и применены основные компоненты Lazarus, такие как TStringGrid, TSpeedButton, TPanel, TEdit, TComboBox, TDateTimePicker, TSaveDialog и TOpenDialog. Эти компоненты позволили создать удобный и функциональный интерфейс, обеспечивающий выполнение следующих операций: добавление, редактирование и удаление записей; сохранение данных в типизированный файл и их загрузка из файла; визуализация данных в табличном виде.

Приложение корректно обрабатывает ввод данных, проверяет их на полноту и сохраняет в структурированном формате. Для удобства пользователя реализованы диалоговые окна для открытия и сохранения файлов, а также подтверждение критических действий, таких как удаление записей.

Таким образом, выполнение данной работы позволило не только освоить работу с типизированными файлами, но и получить практический опыт разработки приложений с графическим интерфейсом. Разработанный справочник может быть расширен дополнительными функциями, такими как поиск, сортировка или фильтрация записей, что делает его хорошей основой для более сложных проектов в будущем.