Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Вятский государственный университет»

Колледж ВятГУ

**ОТЧЕТ**

**ПО ДОМАШНЕЙ КОНТРОЛЬНОЙ РАБОТЕ №9**

**«РАБОТА С ТИПИЗИРОВАННЫМИ ФАЙЛАМИ»**

**ПО «МДК 06.02»**

Выполнил: студент учебной группы

ИСПк 205-52-00 Касьянова Славяна Александровна

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

Формулировка задания (принтер)

Вариант 1

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 позволяющие изменять данных и выполнять

сохранение проделанных изменений.

1. Описание алгоритма

По средству использования процедур и функций происходит взаимодействие с пользовательским интерфейсом

1. Код программы

unit Main;

{$mode objfpc}{$H+}

interface

uses

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

Buttons, edit, Grids;

type

Contacts = record

Cost: integer;

Speed: integer;

Format: string[80];

Size: integer;

Color: string[80];

end; //record

type

{ TfMain }

TfMain = class(TForm)

Panel1: TPanel;

addPelmeni: TSpeedButton;

bEdit: TSpeedButton;

bDel: TSpeedButton;

bSort: TSpeedButton;

SG: TStringGrid;

procedure addPelmeniClick(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.addPelmeniClick(Sender: TObject);

begin

//очищаем поля, если там что-то есть:

fEdit.eSize.Text:= '';

fEdit.espeed.Text:= '';

fEdit.eColor.Text := '';

fEdit.eCost.Text := '';

//устанавливаем ModalResult редактора в mrNone:

fEdit.ModalResult:= mrNone;

//теперь выводим форму:

fEdit.ShowModal;

//если пользователь ничего не ввел - выходим:

if (fEdit.eSize.Text= '') or (fEdit.espeed.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.eCost.Text;

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

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

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

SG.Cells[4, SG.RowCount-1]:= fEdit.eColor.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 пельмени:( "' +

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

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

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

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

fEdit.eColor.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.eCost.Text;

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

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

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

SG.Cells[4, SG.Row]:= fEdit.eColor.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 + 'telephones.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.Cost:= StrToInt(SG.Cells[0, i]);

MyCont.Speed:= StrToInt(SG.Cells[1, i]);

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

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

MyCont.Color:= 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]:= 'цена';

SG.Cells[1, 0]:= 'скорость печати';

SG.Cells[2, 0]:= 'формат печати';

SG.Cells[3, 0]:= 'колво цветов';

SG.Cells[4, 0]:= 'размер';

SG.ColWidths[0]:= 150;

SG.ColWidths[1]:= 150;

SG.ColWidths[2]:= 400;

SG.ColWidths[3]:= 150;

SG.ColWidths[4]:= 100;

//если файла данных нет, просто выходим:

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

//иначе файл есть, открываем его для чтения и

//считываем данные в сетку:

try

AssignFile(f, adres + 'telephones.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]:= IntToStr(MyCont.Cost);

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

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

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

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

end;

finally

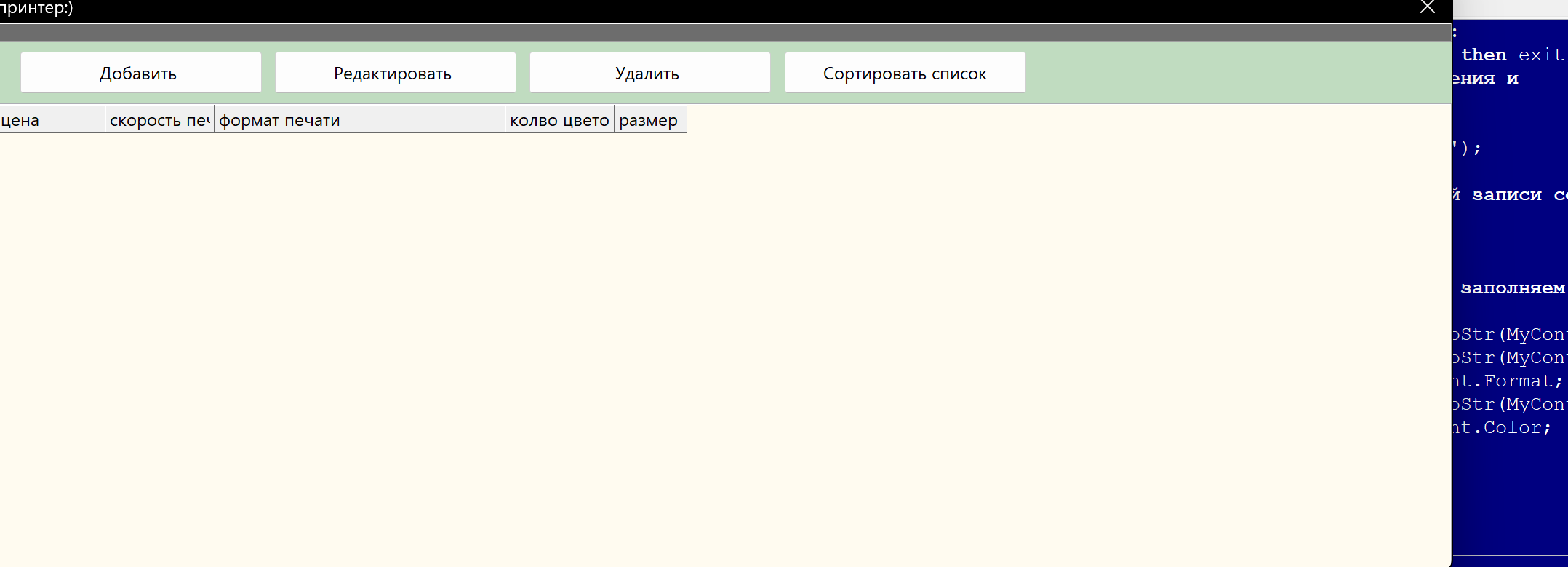
CloseFile(f);

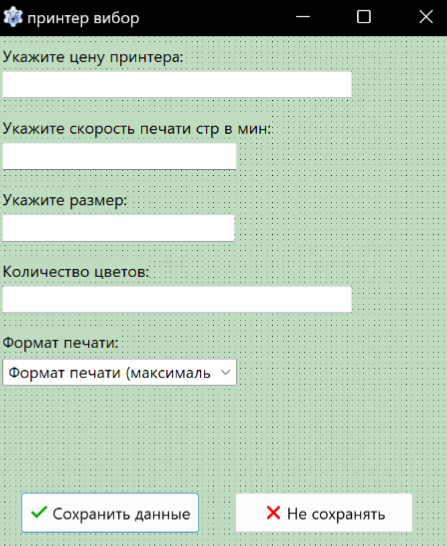
end;

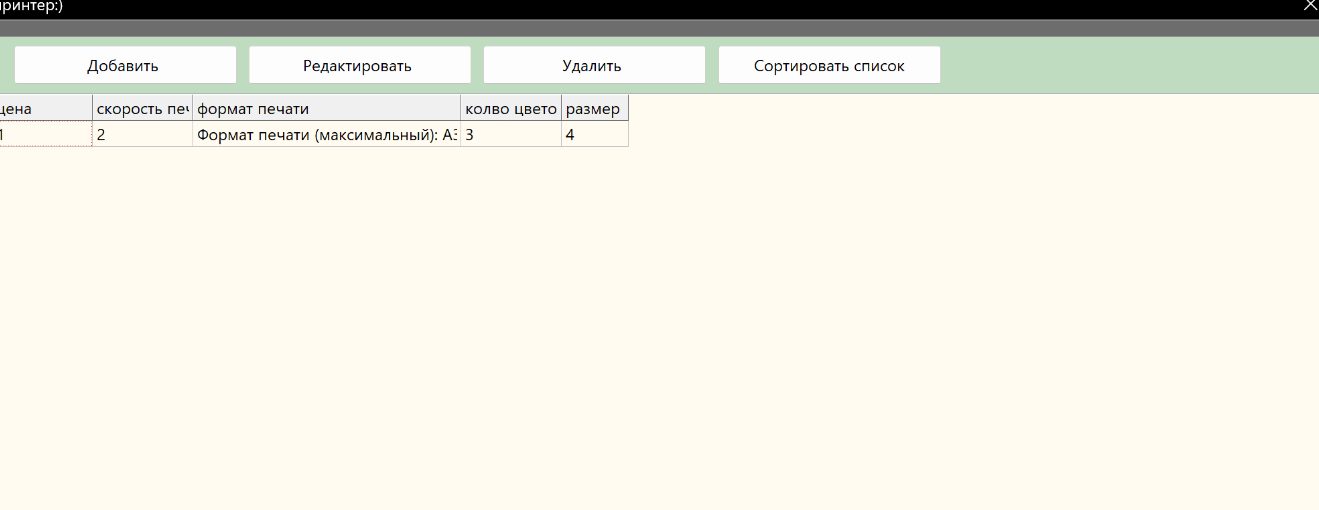
end;

end.

1. Результат при выводе







4. Вывод

Изучила базовую структуру организации программы и основные конструкции языка программирования lazarus

По итогу завершения работы, мы изучили базовую структуру организации программы и основы конструкции языка программирования Lazarus. В ходе выполнения контрольной работы, мы узнали о том как поменять фон приложения разобрались с их значениями. Так же столкнулись с некоторыми проблемами. Но решили их. Так же у нас возникли проблемы с выводом из-за небольших ошибок в написании программы, но мы их нашли, исправили и всё получилось.