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

Кононоа С.А.

Преподаватель:

Сергеева Елизавета Григорьевна

Киров

2024

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

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.

**Описание алгоритма:**

Файл сохраняется в массив строк, затем строка разделяется на компоненты по запятым в строке, и они выводятся в текстовые поля, при переключении на следующую строку берется следующая строки из массива и выполняются аналогичные действия, при переключении изменения сохраняются в массив строк, при закрытии массив записывается в файл.

**Код программы:**

unit Unit1;

{$mode objfpc}{$H+}

interface

uses

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

Buttons;

type

{ TFmain }

TFmain = class(TForm)

CheckIsExists: TCheckBox;

EditPointTo: TEdit;

LabelIsValue: TLabel;

LabelIsExists: TLabel;

LabelDateExport: TLabel;

LabelPointTo: TLabel;

EditCountry: TEdit;

Nextt: TButton;

LabelValue: TLabel;

Previouss: TButton;

EditTown: TEdit;

EditValue: TEdit;

EditDateExport: TEdit;

CheckValue: TCheckBox;

LabelCountry: TLabel;

TownLabel: TLabel;

procedure CheckValueChange(Sender: TObject);

procedure DiagonalClick(Sender: TObject);

procedure EditPointToChange(Sender: TObject);

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

procedure FormCreate(Sender: TObject);

procedure FormShow(Sender: TObject);

procedure LabelIsValueClick(Sender: TObject);

procedure LabelIsExistsClick(Sender: TObject);

procedure LabelDateExportClick(Sender: TObject);

procedure NexttClick(Sender: TObject);

procedure PrevioussClick(Sender: TObject);

procedure LabelCountryClick(Sender: TObject);

procedure EditCountryChange(Sender: TObject);

private

procedure GetRow(i: integer);

procedure editdata(i:integer);

public

end;

var

Fmain: TFmain;

Data: TStringList;

cur\_row: integer;

implementation

{$R \*.lfm}

{ TFmain }

procedure TFmain.FormShow(Sender: TObject);

Begin

Data := TStringList.create;

cur\_row := 0;

if FileExists('data.txt') then begin

Data.LoadFromFile('data.txt');

getRow(cur\_row);

end;

end;

procedure TFmain.LabelIsValueClick(Sender: TObject);

begin

end;

procedure TFmain.LabelIsExistsClick(Sender: TObject);

begin

end;

procedure TFmain.LabelDateExportClick(Sender: TObject);

begin

end;

procedure TFmain.NexttClick(Sender: TObject);

begin

editdata(cur\_row);

if cur\_row < data.count-1 then

begin

cur\_row := cur\_row + 1;

getRow(cur\_row);

end

else

begin

data.add('');

cur\_row += 1;

EditCountry.clear;

EditTown.clear;

CheckValue.checked := False;

CheckIsExists.checked := False;

EditPointTo.clear();

EditValue.clear;

EditDateExport.clear;

end;

end;

procedure TFmain.PrevioussClick(Sender: TObject);

begin

editdata(cur\_row);

if cur\_row > 0 then

begin

cur\_row := cur\_row - 1;

getRow(cur\_row);

end;

end;

procedure TFmain.LabelCountryClick(Sender: TObject);

begin

end;

procedure TFmain.EditCountryChange(Sender: TObject);

begin

end;

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

begin

editdata(cur\_row);

data.saveToFile('data.txt')

end;

procedure TFmain.EditPointToChange(Sender: TObject);

begin

end;

procedure TFmain.CheckValueChange(Sender: TObject);

begin

if CheckValue.checked = False then begin

EditValue.visible := False;

LabelValue.visible := False;

end

else begin

EditValue.visible := True;

LabelValue.visible := True;

end;

end;

procedure TFmain.DiagonalClick(Sender: TObject);

begin

end;

procedure TFmain.FormCreate(Sender: TObject);

begin

end;

procedure TFmain.GetRow(i: integer);

var str: string;

begin

str := Data.Strings[i];

EditCountry.text:=copy(str, 1, pos(',',str)-1);

delete(str,1,pos(',',str));

EditTown.text:=copy(str, 1, pos(',',str)-1);

delete(str,1,pos(',',str));

if copy(str, 1, pos(',',str)-1) = 'True' then

CheckValue.checked := True

else

CheckValue.checked := False;

delete(str,1,pos(',',str));

if copy(str, 1, pos(',',str)-1) = 'True' then

CheckIsExists.checked := True

else

CheckIsExists.checked := False;

delete(str,1,pos(',',str));

EditPointTo.text:=copy(str, 1, pos(',',str)-1);

delete(str,1,pos(',',str));

EditValue.text:=copy(str, 1, pos(',',str)-1);

delete(str,1,pos(',',str));

EditDateExport.text:=str;

end;

procedure TFmain.editdata(i:integer);

var check1, check2:string;

begin

if CheckValue.checked = True then

check1:= 'True'

else check1:='False';

if CheckIsExists.checked = True then

check2:= 'True'

else check2:='False';

data.strings[i]:=

EditCountry.text +

',' +

EditTown.text +

',' +

check1 +

',' +

check2 +

',' +

EditPointTo.text +

',' +

EditValue.text +

',' +

EditDateExport.Text;

end;

end.

**Результат выполнения программы:**

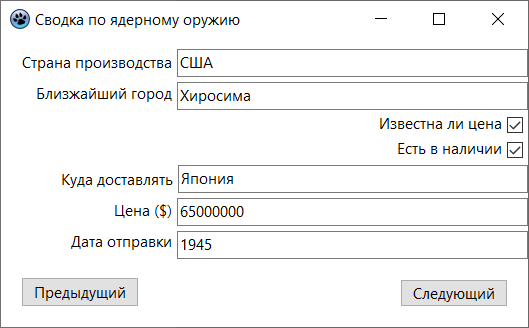


Рисунок 1 - Открытие программы

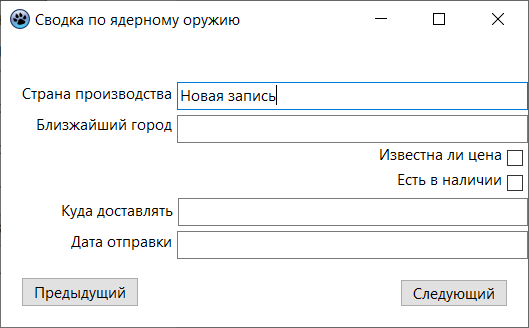


Рисунок 2 - Создание пустой записи

**Вывод:**

В ходе выполнения лабораторной работы были получено представление об организации хранения и использования информации посредством типизированных файлов, закреплены навыки создания пользовательского интерфейса