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 ТИПИЗИРОВАНЫМИ ФАЙЛАМИ»**

**ПО МДК 05.02 РАЗРАБОТКА КОДА ИНФОРМАЦИОННЫХ СИСТЕМ**

Выполнил: студент учебной группы

ИСПк-204-52-00

Карина Яна

Преподаватель:

Сергеева Елизавета Григорьевна

Киров

2023

**Цель работы**

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

**Скриншот задания**

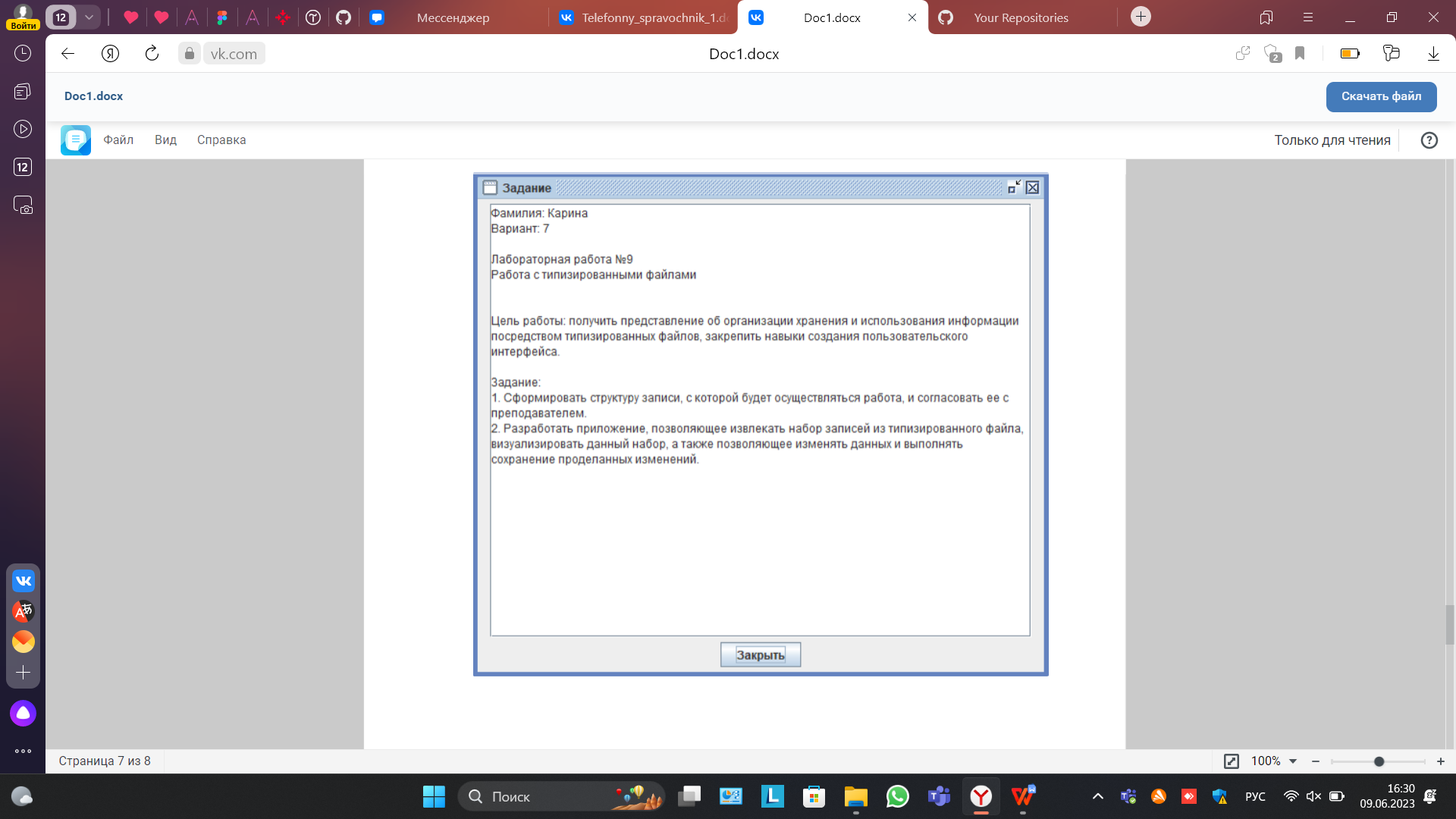


Рисунок 1 – Скриншот задания

**Описание алгоритма**

В программе присутствуют следующие возможности :

1. Возможность добавить новую запись
2. Возможность редактировать запись
3. Возможность удалить запись

**Код программы**

unit Unit1;

{$mode objfpc}{$H+}

interface

uses

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

Grids, edit;

type

{ TfMain }

TfMain = class(TForm)

Panel1: TPanel;

bAdd: TSpeedButton;

bEdit: TSpeedButton;

bDel: TSpeedButton;

bSort: TSpeedButton;

SG: TStringGrid;

procedure bAddClick(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;

type

Contacts = record

Name: string[100];

Telephon: string[20];

Note: string[20];

end; //record

var

fMain: TfMain;

adres: string; // адрес, откуда запущена программа

implementation

{$R \*.lfm}

{ TfMain }

procedure TfMain.bAddClick(Sender: TObject);

begin

fEdit.eName.Text:='';

fEdit.eTelephone.Text:='';

fEdit.ModalResult:=mrNone;

fEdit.ShowModal;

if (fEdit.eName.Text='') or (fEdit.eTelephone.Text='') then exit;

if fEdit.ModalResult <> mrOk then exit;

SG.RowCount:=SG.RowCount+1;

SG.Cells[0,SG.RowCount-1]:=fEdit.eName.Text;

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

SG.Cells[2,SG.RowCount-1]:=fEdit.Edit1.Text;

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

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

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

SG.Cells[6,SG.RowCount-1]:=fEdit.Job.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.Surname.Text:= SG.Cells[1,SG.Row];

fEdit.Edit1.Text:= SG.Cells[2,SG.Row];

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

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

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

fEdit.Job.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.Row]:= fEdit.eName.Text;

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

SG.Cells[3, SG.Row]:= fEdit.Edit1.Text;

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

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

SG.Cells[6, SG.Row]:= fEdit.eTelephone.Text;

SG.Cells[7, SG.Row]:= fEdit.Job.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 данныые текущей записи:

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

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

MyCont.Note:=SG.Cells[2, 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]:= 'Имя';

SG.Cells[1, 0]:= 'Фамилия';

SG.Cells[2, 0]:= 'Отчество';

SG.Cells[3, 0]:= 'Пол';

SG.Cells[4, 0]:= 'Дата рождения';

SG.Cells[5, 0]:= 'Номер телефона';

SG.Cells[6, 0]:= 'Должность';

SG.ColWidths[0]:=150;

SG.ColWidths[1]:=150;

SG.ColWidths[2]:=150;

SG.ColWidths[3]:=150;

SG.ColWidths[4]:=150;

SG.ColWidths[5]:=150;

SG.ColWidths[6]:=150;

//если файла данных нет, просто выходим:

if not FileExists(adres + 'telephones.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]:= MyCont.Name;

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

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

end;

finally

CloseFile(f);

end;

end;

end.

**Результат выполнения программы**

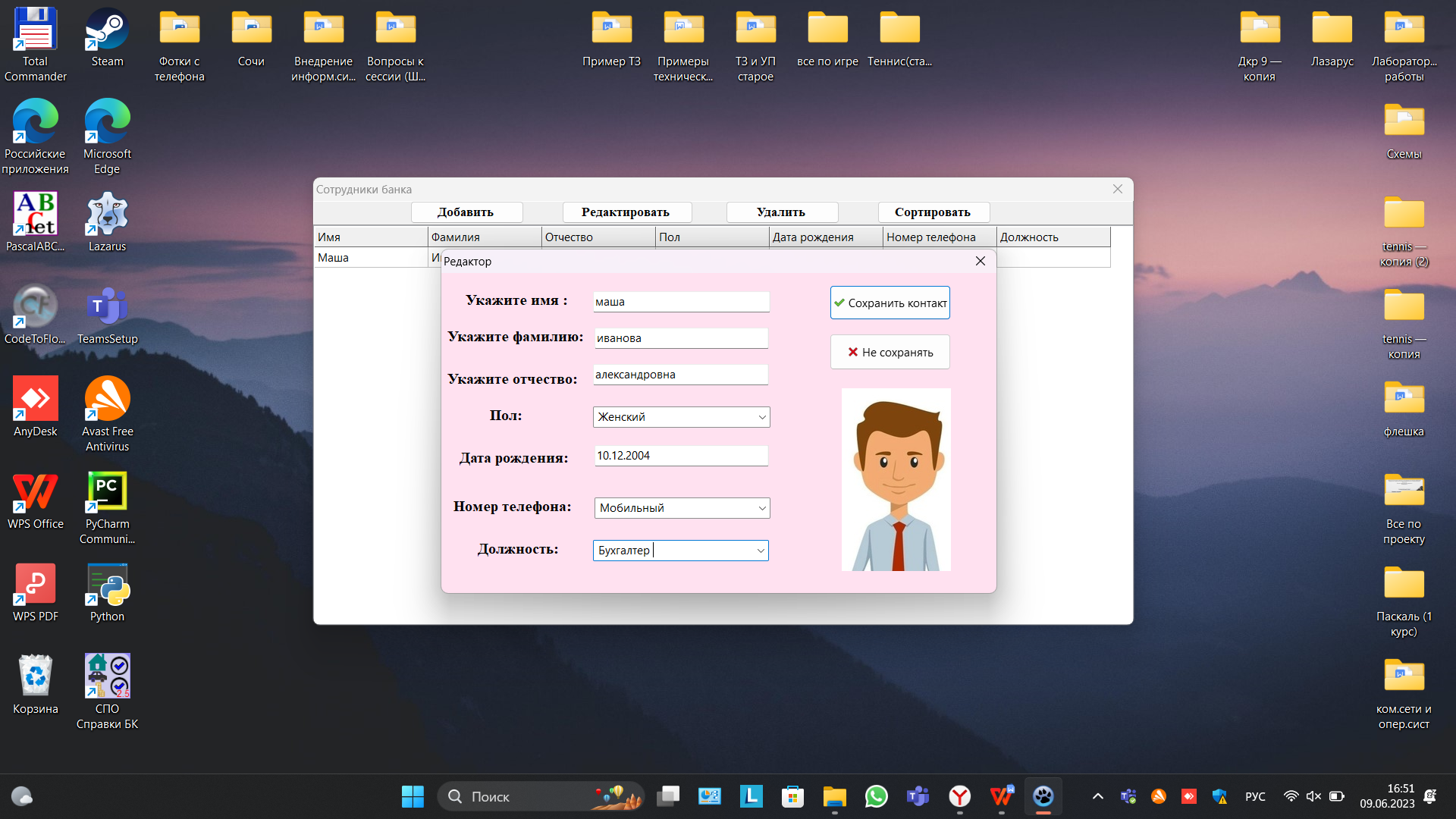


Рисунок 2 - Редактор

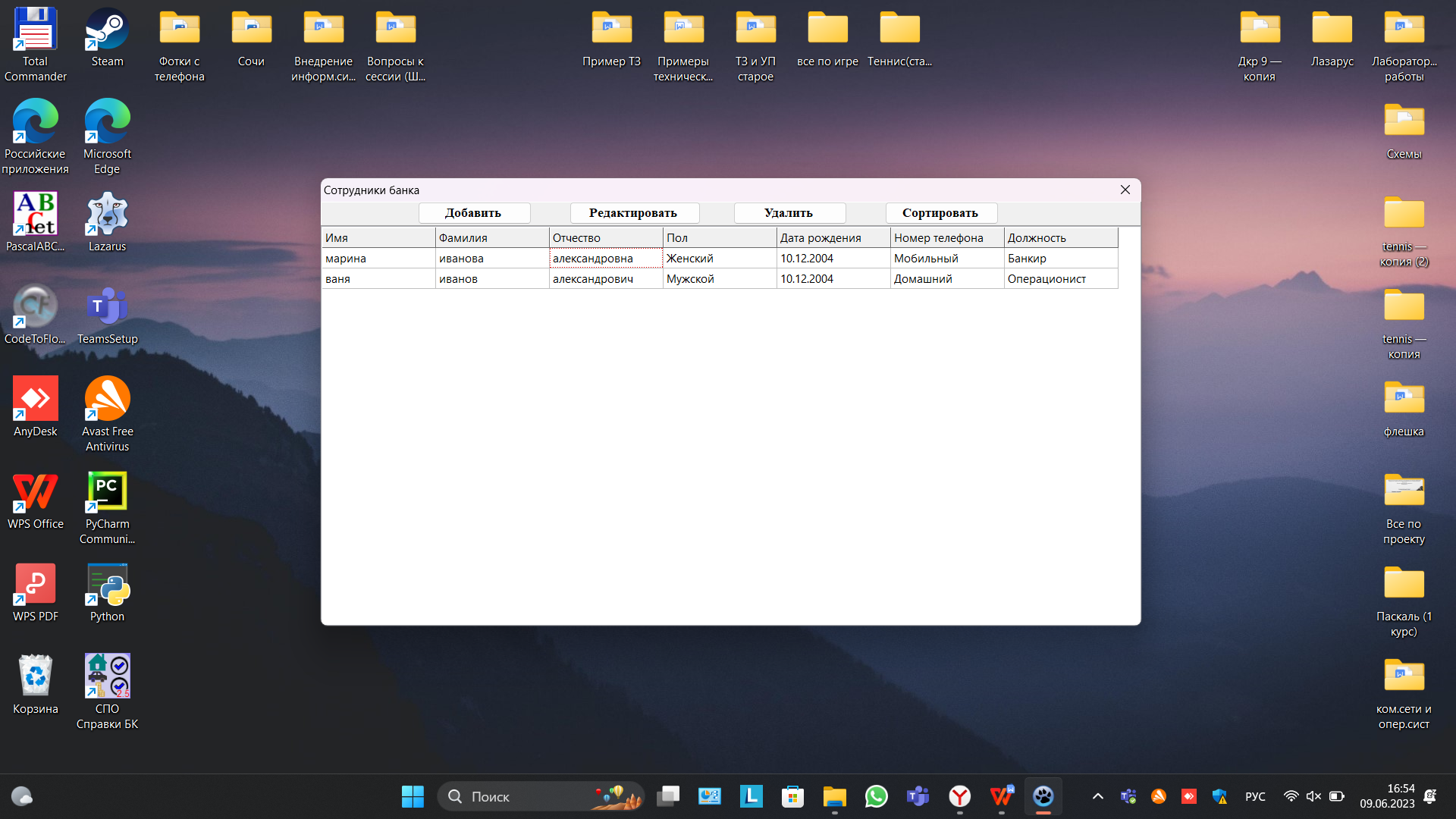


Рисунок 3 - Таблица сотрудников банка

**Вывод**

В процессе выполнения домашней контрольной работы №9 была реализована таблица сотрудников банка. При написании программы были использованы самые основные элементы, такие как TEdit, TComboBox, TLabel, DBGrid, TButton и другие. Программа имеет достаточно простой интерфейс, который доступен для всех пользователей. Можно сделать вывод, что данная практика является хорошим практическим опытом в дальнейшей самостоятельной деятельности.