



**«Московский государственный технический университет
имени Н.Э. Баумана»
(национальный исследовательский университет)**

ФАКУЛЬТЕТ
КАФЕДРА

ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ
КОМПЬЮТЕРНЫЕ СИСТЕМЫ И СЕТИ (ИУ6)

О т ч е т
по лабораторной работе №10
варианта №7

Название лабораторной работы:

Программирование в среде Delphi

Дисциплина:

Основы программирования

Студент гр. **ИУ6-12**

31/12/17 _____
(Подпись, дата)

Векшин Роман
(И.О. Фамилия)

Преподаватель

(Подпись, дата)

Черноусова Татьяна Геннадьевна
(И.О. Фамилия)

Задание 1

Разработать программу – калькулятор, выполняющий основные арифметические действия

Структура проекта Calc

- Главная программа
- Графическая библиотека классов форм Windows (VCL)
- Модуль главной формы

Пользовательский интерфейс приложения Calc

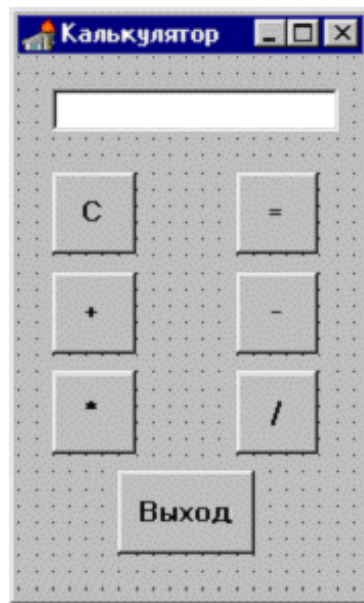


Рисунок 1-Пример интерфейса главной формы приложения Calc из задания

Схема алгоритма

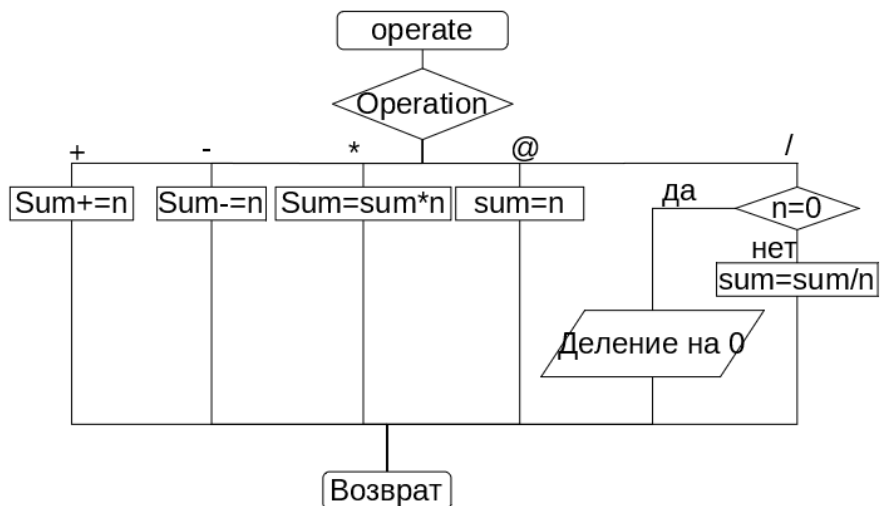


Рисунок 2— Схема алгоритма выполнения основных арифметических операций программы "Калькулятор"

Код программы

```
program project1;
uses
  Interfaces, Forms, Unit1;

begin
  Application.Initialize;
  Application.CreateForm(TForm1, Form1);
  Application.Run;
end.
```

Текст модуля

```
unit Unit1;
interface

uses
  Classes, Forms, StdCtrls;

type
  TForm1 = class(TForm)
    btnC: TButton;
    btnPrepare: TButton;
    btnPlus: TButton;
    btnMinus: TButton;
    btnMultiplecate: TButton;
    btnDivision: TButton;
    btnExit: TButton;
    eRes: TEdit;
    procedure btnCClick(Sender: TObject);
    procedure btnPrepareClick(Sender: TObject);
    procedure operators(Sender: TButton);
    procedure btnExitClick(Sender: TObject);
  end;
var
  Form1: TForm1;

implementation
{$R *.lfm}

var
  Sum: real;
  operation: string = '@';
//Совершение арифметических операций
procedure operate;
var
  s: string;
  code: integer;
  n: real;
begin
  s := Form1.eRes.Text;
  Form1.eRes.Clear;
  val(s, n, code);
  if code = 0 then
    case operation of
      '@': sum := n;
      '+': sum := sum + n;
      '-': sum := sum - n;
      'x': sum := sum * n;
      '/': if n = 0 then
        form1.Caption := 'Div by 0!'
      else
        sum := sum / n;
    end;
  end;
end;
//Обработка события нажатия кнопки очистки поля
procedure TForm1.btnCClick(Sender: TObject);
begin
  eRes.Clear;
  operation := '@';
  eRes.SetFocus;
end;
```

```
//Обработка события нажатия кнопки результата
procedure TForm1.btnPrepareClick(Sender: TObject);
var
  s: string;
begin
  operate;
  str(sum: 6: 3, s);
  eRes.Text := s;
  operation := '@';
  eRes.SetFocus;
end;
//Обработка события нажатия кнопок x, +, -, /
procedure TForm1.operators(Sender: TButton);
begin
  operate;
  operation := Sender.Caption;
  eRes.SetFocus;
end;
//Обработка события нажатия кнопки выхода
procedure TForm1.btnExitClick(Sender: TObject);
begin
  Close;
end;
end.
```

Тестирование

Таблица 1-Результаты тестирования программы «Калькулятор»

Исходные данные	Ожидаемый результат	Полученный результат
С	Пустое поле	(Пустое поле)
0.5+0.25	0.75	0.750
0.5-0.25	0.25	0.250
0.5/0.25	2	2.000
0.5*0.25	0.125	0.125
0.25=	0.25	0.250
=	(Последний полученный результат)	0.250
5/0	(Деление на 0)	Div by 0!
Выход	Заккрытие приложения	(Приложение закрыто)

Пример работы программы

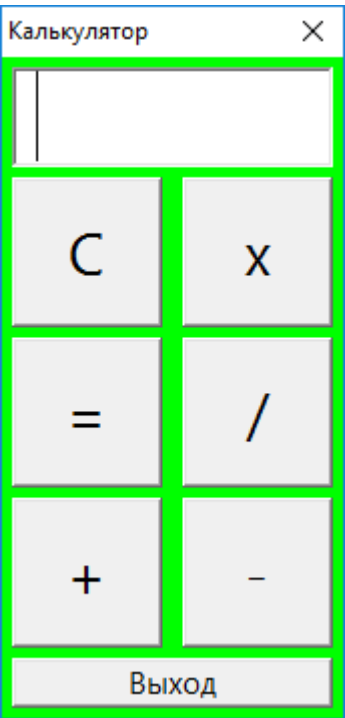


Рисунок 3-Пример работы программы 1

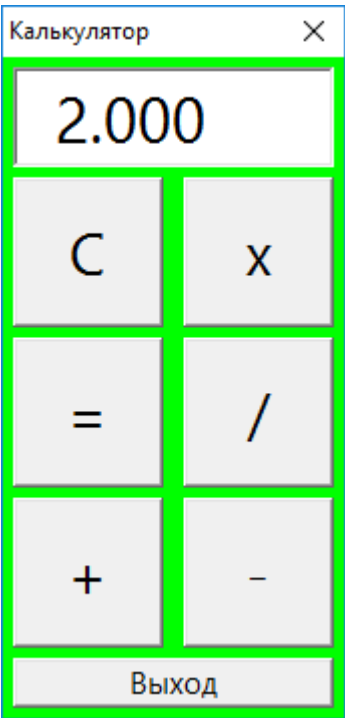


Рисунок 4-Пример работы программы 1

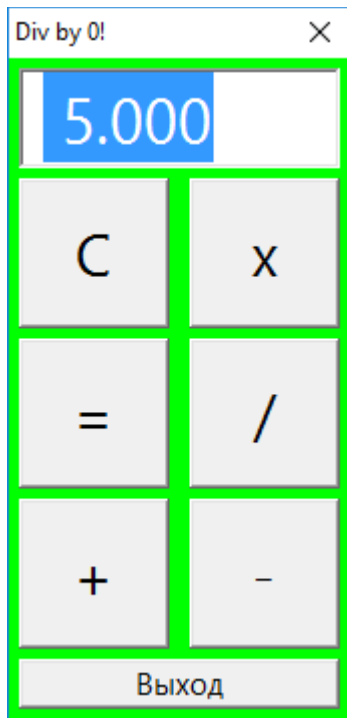


Рисунок 5-Пример работы программы 1

Вывод

- 1) Разработан алгоритм для оконного приложения «Калькулятор», и составлена его схема в среде LibreOffice Draw (см. рис. 2).
- 2) Создан код программы по схеме алгоритма в среде Lazarus.
- 3) Проведено тестирование (см. табл. 1).
- 4) Тестирование показало корректность работы программы (см. рис. 3-5).

Задание 2

Разработать приложение «Записная книжка».



Рисунок 6-Пример интерфейса главной формы приложения "Записная книжка" из задания

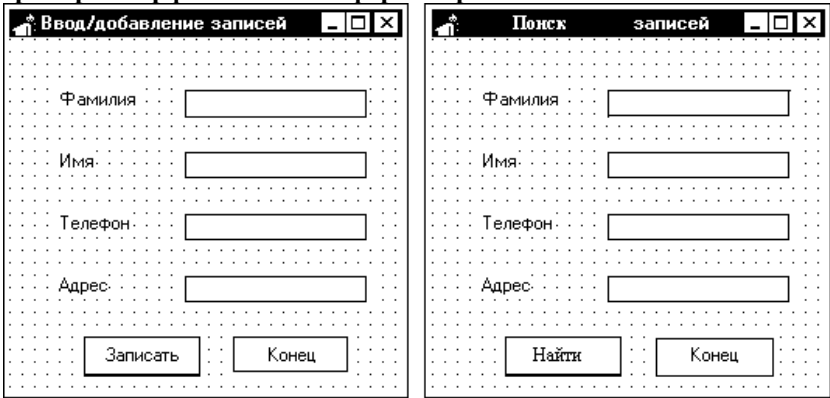


Рисунок 7-Пример интерфейса форм ввода и поиска информации приложения "Записная книжка" из задания

Схема алгоритма

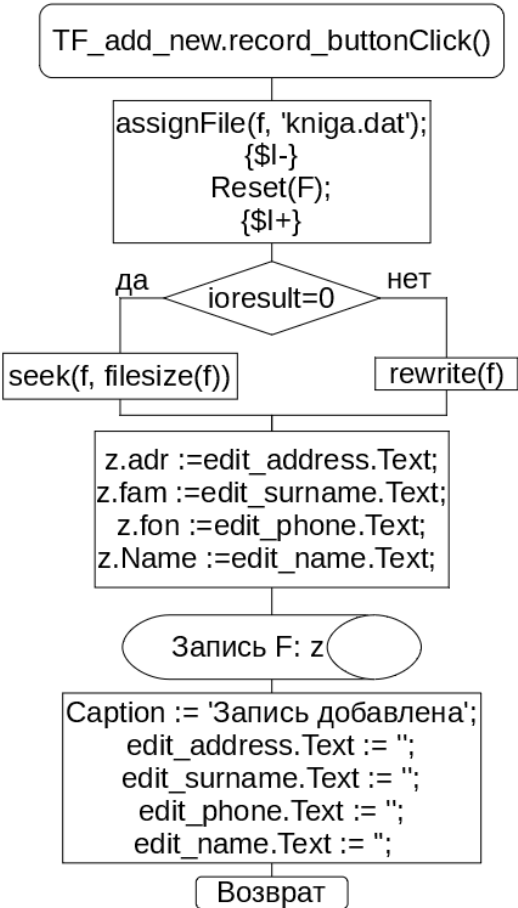


Рисунок 8-Схема алгоритма

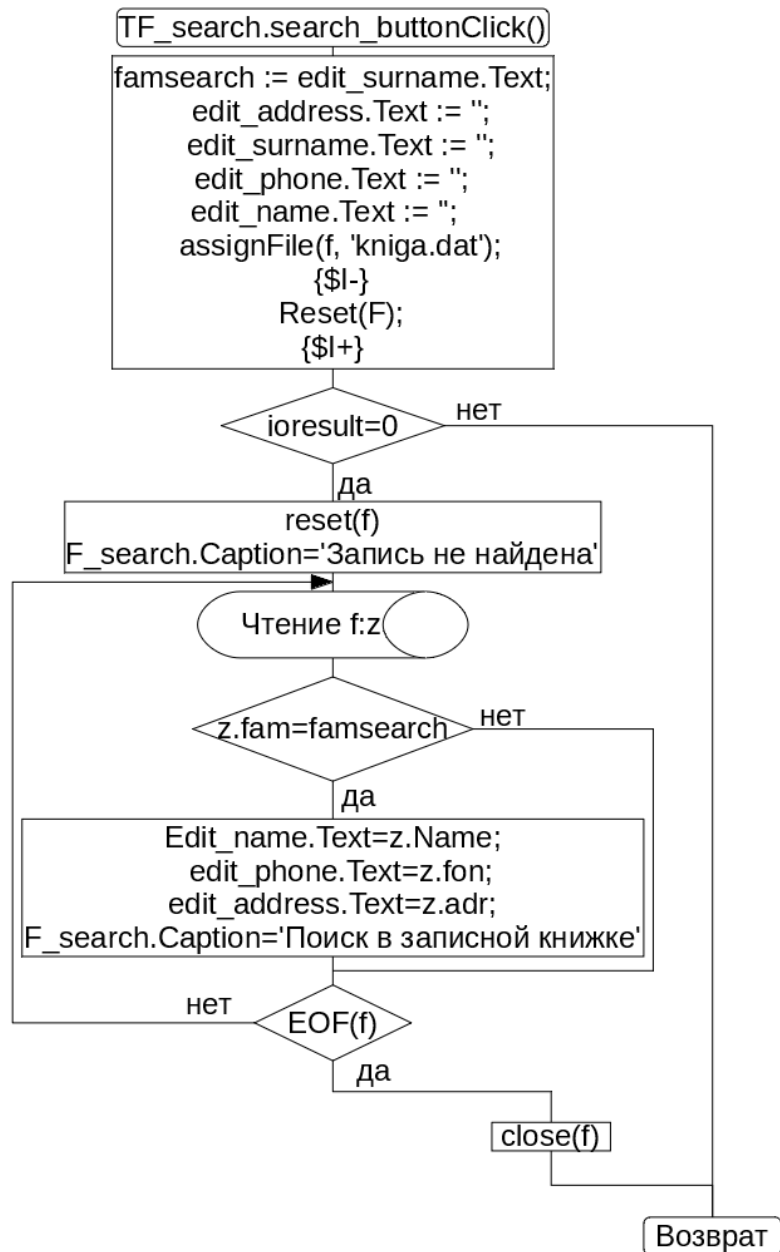


Рисунок 9-Схема алгоритма

Формат Записи

Таблица 2-Формат данных записи

RECORD
Surname: string
Name: string
Phone: string
Address: string

Код программы

```
program project1;
uses
  Interfaces, Forms, U_main, U_search, U_add;
begin
  RequireDerivedFormResource:=True;
  Application.Initialize;
  Application.CreateForm(TF_main, F_main);
  Application.CreateForm(TF_search, F_search);
  Application.CreateForm(TF_add_new, F_add_new);
  Application.Run;
end.
```

Текст модуля U_main

```
unit U_main;

interface
uses
  Classes, Forms, StdCtrls, U_search, U_add;

type
  TF_main = class(TForm)
    add_new_button: TButton;
    search_button: TButton;
    close_form_button: TButton;
    procedure open_to_add(Sender: TObject);
    procedure open_to_search(Sender: TObject);
    procedure close_form(Sender: TObject);
  end;

var
  F_main: TF_main;

implementation

{$R *.lfm}
//Обработка события нажатия кнопки на открытие формы для добавления записи
procedure TF_main.open_to_add(Sender: TObject);
begin
  F_add_new.Show;
end;
//Обработка события нажатия кнопки на поиск в файле
procedure TF_main.open_to_search(Sender: TObject);
begin
  F_search.Show;
end;
//Обработка события нажатия кнопки выхода
procedure TF_main.close_form(Sender: TObject);
begin
  Close;
end;
end.
```

Текст модуля U_search

```
unit U_search;

interface
uses
  Classes, Forms, StdCtrls;

type
  TF_search = class(TForm)
    search_button: TButton;
    close_form_button: TButton;
    edit_surname: TEdit;
    Edit_name: TEdit;
    edit_phone: TEdit;
    edit_address: TEdit;
    LSurname: TLabel;
    LName: TLabel;
```



```

    LPhone: TLabel;
    LAddress: TLabel;
    procedure search_buttonClick(Sender: TObject);
    procedure close_form_buttonClick(Sender: TObject);
end;
type
    zap = record
        fam: string[22];
        Name: string[22];
        fon: string[22];
        adr: string[22];
    end;
var
    F_search: TF_search;
implementation
{$R *.lfm}

    var
        f: file of zap;
        z: zap;
//Обработка события нажатия кнопки поиска по фамилии в файле
    procedure TF_search.search_buttonClick(Sender: TObject);
    var
        famsearch: string[22];
    begin
        famsearch := F_search.edit_surname.Text;
        F_search.edit_address.Text := '';
        F_search.edit_surname.Text := '';
        F_search.edit_phone.Text := '';
        F_search.edit_name.Text := '';
        assignFile(f, 'kniga.dat');
        {$I-}
        Reset(f);
    {$I+}
        if ioresult = 0 then
            begin
                reset(f);
                F_search.Caption := 'Запись не найдена';
                repeat
                    Read(f, z);
                    if z.fam = famsearch then
                        begin
                            F_search.Edit_name.Text := z.Name;
                            F_search.edit_phone.Text := z.fon;
                            F_search.edit_address.Text := z.adr;
                            F_search.Caption := 'Поиск в записной книжке';
                        end;
                until EOF(f);
                CloseFile(f);
            end;
        end;
//Обработка события нажатия кнопки выхода
    procedure TF_search.close_form_buttonClick(Sender: TObject);
    begin
        Close;
    end;

end.

```

Текст модуля U_add

```

unit U_add;

interface
uses
    Classes, Forms, StdCtrls, U_search;
type
    TF_add_new = class(TForm)
        record_button: TButton;
        close_form_button: TButton;
        edit_surname: TEdit;
    end;

```

```

edit_name: TEdit;
edit_phone: TEdit;
edit_address: TEdit;
LSurname: TLabel;
LName: TLabel;
LPhone: TLabel;
LAddress: TLabel;
procedure record_buttonClick(Sender: TObject);
procedure close_form_buttonClick(Sender: TObject);
end;

var
  F_add_new: TF_add_new;
implementation
{$R *.lfm}

var
  f: file of zap;
  z: zap;
//Обработка события нажатия кнопки добавления записи
procedure TF_add_new.record_buttonClick(Sender: TObject);
begin
  assignFile(f, 'kniga.dat');
  {$I-}
  Reset(f);
{$I+}
  if ioresult = 0 then
    seek(f, filesize(f))
  else
    rewrite(f);
  z.adr := F_add_new.edit_address.Text;
  z.fam := F_add_new.edit_surname.Text;
  z.fon := F_add_new.edit_phone.Text;
  z.Name := F_add_new.edit_name.Text;
  Write(f, z);
  CloseFile(f);
  F_add_new.Caption := 'Запись добавлена';
  F_add_new.edit_address.Text := '';
  F_add_new.edit_surname.Text := '';
  F_add_new.edit_phone.Text := '';
  F_add_new.edit_name.Text := '';
end;

//Обработка события нажатия кнопки выхода
procedure TF_add_new.close_form_buttonClick(Sender: TObject);
begin
  Close;
end;

end.

```

Пример работы программы

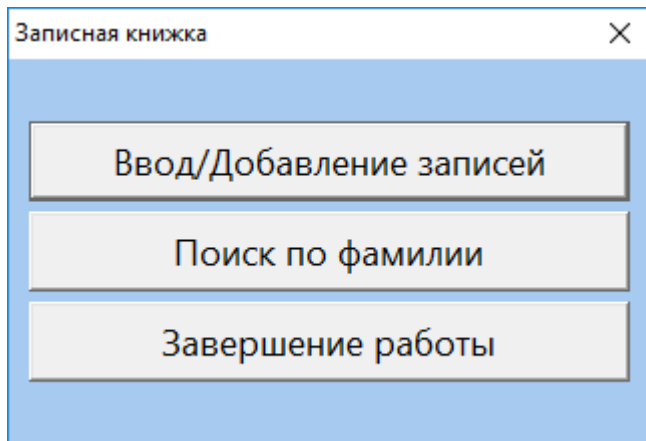


Рисунок 10-Пример работы программы

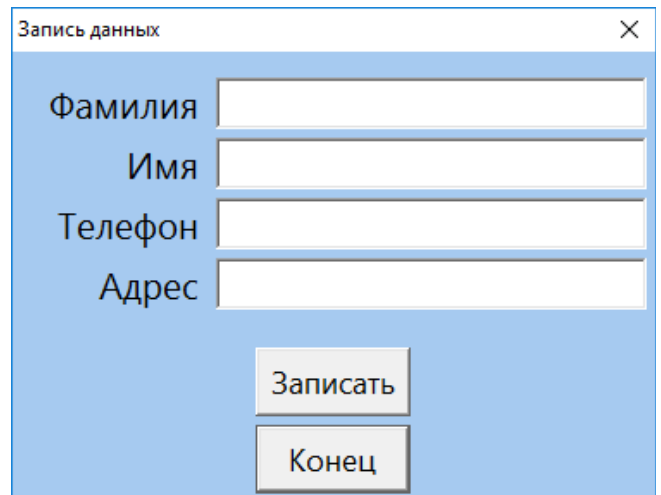


Рисунок 11-Пример работы программы

Поиск в записной книжке

Фамилия

Имя

Телефон

Адрес

Найти

Конец

Рисунок 12-Пример работы программы

Поиск в записной книжке

Фамилия

Имя

Телефон

Адрес

Найти

Конец

Рисунок 13-Пример работы программы

Вывод

- 1) Разработан алгоритм программы «Записная книжка», и составлена его схема в среде LibreOffice Draw(см. рис. 8-9).
- 2) Создан код программы по схеме алгоритма в среде Lazarus.
- 3) Проведено тестирование.
- 4) Тестирование показало корректность работы программы (см. рис. 10-13).