**О Т Ч Е Т**

**ПО ЛАБОРАТОРНЫМ РАБОТАМ**

**по курсу:**

***" РАЗРАБОТКА ПРИЛОЖЕНИЙ В ВИЗУАЛЬНЫХ СРЕДАХ "***

|  |  |
| --- | --- |
| Выполнил: | Студент: Воробей И.А.  Гр. 10701118 |
| Проверил, доцент: | Гурский Н.Н. |

Минск – 2019**ЛИТЕРАТУРА**

**Основная литература**

1. Фаронов В.В. DELPHI. Программирование на языке высокого уровня: Учебник для вузов. – СПб.: Питер, 2009. – 640 с.
2. Сухарев М. DELPHI. Полное руководство. – СПб.: Питер, 2008. – 1040 с.
3. Архангельский А.Я. Программирование в DELPHI-6. – М.: ЗАО «Издательство БИНОМ», 2002. – 1120 с.
4. Елмакова Н., Трепалин С., Тенцер А. DELPHI 6 и технология COM. – СПб.: Питер, 2002. – 640 с.
5. Понамарев В. COM и ActiveX в DELPHI. – СПб.: БХВ-Петербург, 2001. – 320 с.

**Дополнительная литература**

1. Фаронов В.В. Программирование баз данных в DELPHI 7. Учебный курс. – СПб.: Питер, 2005. – 459 с.
2. Подольский С.В., Скиба С.А., Кожедуб О.А. Разработка интернет-приложений в DELPHI. – СПб.: БХВ-Петербург. 2002. – 432 с.
3. Краснов М.В. OpenGL. Графика в проектах Delphi. – СПб.: БХВ-Санкт-Петербург, 2000. – 352 с.
4. Краснов М.В. DirectX. Графика в проектах Delphi. – СПб.: БХВ-Петербург, 2003. – 416 с.
5. Марко Кэнту. DELPHI 7: Для профессионалов. – СПб.: Питер, 2004. – 1101 с.

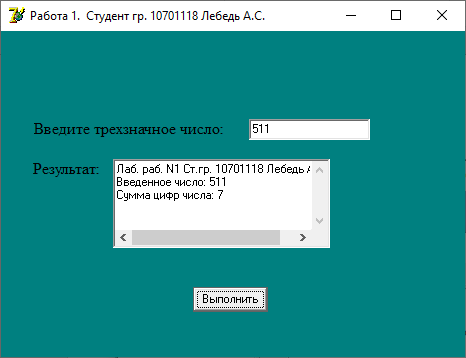
**Работа 1.**

**РАЗРАБОТКА ПРОСТЕЙШЕЙ ПРОГРАММЫ В СРЕДЕ DELPHI**

**Цель работы:**изучить основы среды DELPHI и составить простейшую программу для линейного алгоритма.

**Задание:** Целой переменной S присвоить сумму цифр трехзначного целого числа .

**Скриншоты результатов:**



**Код программы:**

unit Unit1;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,

Dialogs, StdCtrls, ExtCtrls, Math;

type

TForm1 = class(TForm)

Edit1: TEdit;

Label1: TLabel;

Button1: TButton;

Label2: TLabel;

Memo1: TMemo;

procedure Button1Click(Sender: TObject);

procedure FormCreate(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

Form1: TForm1;

implementation

{$R \*.dfm}

procedure TForm1.Button1Click(Sender: TObject);

var Text:string;

sum:integer;

begin

Text:=Edit1.Text;

Memo1.Clear();

Memo1.Lines.Add(‘Лаб. Раб. N1 Ст.гр.10701118 Лебедь А.С.’);

Memo1.Lines.Add(‘Введенное число: ‘ + text);

sum:=

StrToInt(Text[1]) +

StrToInt(Text[2]) +

StrToInt(Text[3]);

Memo1.Lines.add(‘Сумма цифр числа: ‘ + IntToStr(sum));

end;

procedure TForm1.FormCreate(Sender: TObject);

begin

end;

end.

**Вывод:** изучил основы среды DELPHI и составил простейшую программу для линейного алгоритма.

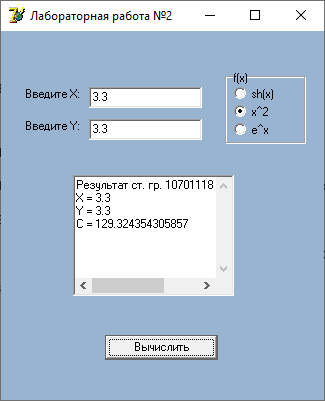
**Работа 2.**

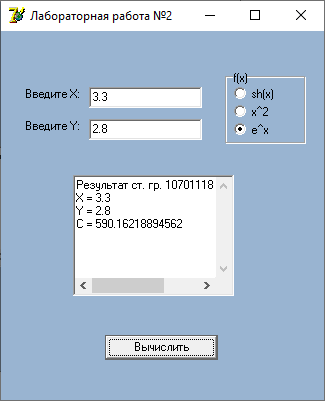
# **КОНСТРУИРОВАНИЕ ПРОГРАММ ДЛЯ РАЗВЕТВЛЯЮЩИХСЯ ВЫЧИСЛИТЕЛЬНЫХ ПРОЦЕССОВ**

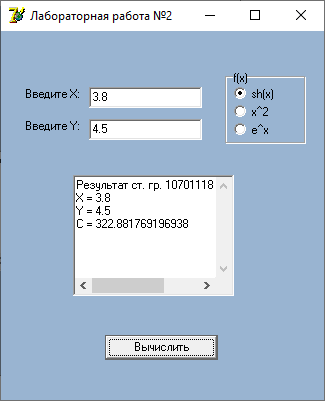
**Цель работы:**научиться пользоваться стандартными компонентами организации переключений (TCheckBox, TradioGroup и др.). Используя компонентами организации переключений разработать интерфейс и программу для заданного разветвляющегося алгоритма.

**Задание:** Целой переменной S присвоить сумму цифр трехзначного целого числа .

**Скриншоты результатов:**







**Код программы:**

unit Unit1;

**//VARABEI IVAN**

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,

Dialogs, StdCtrls, ExtCtrls, Math;

type

TForm1 = class(TForm)

Label1: TLabel;

Label2: TLabel;

Edit1: TEdit;

Edit2: TEdit;

Memo1: TMemo;

Button1: TButton;

RadioGroup1: TRadioGroup;

procedure Button1Click(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

Form1: TForm1;

implementation

{$R \*.dfm}

procedure TForm1.Button1Click(Sender: TObject);

var x, y, fx, c: extended;

begin

Memo1.Clear();

Memo1.Lines.Add('Результат ст. гр. 10701118 Лебедя А.С.');

Memo1.Lines.Add('X = ' + Edit1.Text);

Memo1.Lines.Add('Y = ' + Edit2.Text);

x:=StrToFloat(Edit1.Text);

y:=StrToFloat(Edit2.Text);

case RadioGroup1.ItemIndex of

0: fx := sinh(x);

1: fx := Power(x, 2);

2: fx := exp(x);

end;

if (x-y) = 0 then c:=Power(fx, 2) + Power(y, 2) + sin(y)

else if (x-y) > 0 then c:=Power((fx-y), 2) + cos(y)

else c:=Power((y-fx), 2) + tan(y);

Memo1.Lines.Add('C = ' + FloatToStr(c));

end;

end.

**Вывод:** научился пользоваться стандартными компонентами организации переключений (TCheckBox, TradioGroup и др.). Используя компонентами организации переключений разработал интерфейс и программу для заданного разветвляющегося алгоритма.

# **Работа 3.**

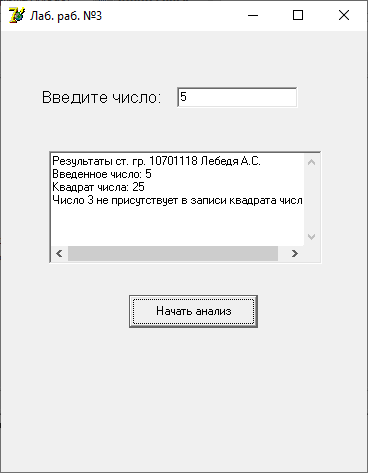
# **КОНСТРУИРОВАНИЕ ПРОГРАММ**

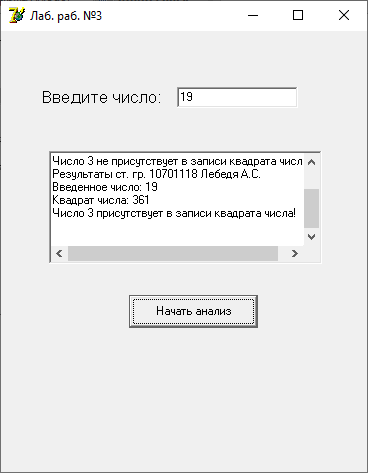
# **ДЛЯ ЦИКЛИЧЕСКИХ ВЫЧИСЛИТЕЛЬНЫХ ПРОЦЕССОВ**

**Цель работы:**изучить средства отладки программ в среде DELPHI. Составить и отладить программу для циклического вычислительного процесса.

**Задание:** Дано натуральное число n. Выяснить, входит ли цифра 3 в запись числа .

**Скриншоты результатов:**





**Код программы:**

program Project2;

**//VARABEI IVAN**

uses

Forms,

Unit2 in 'Unit2.pas' {Form2};

{$R \*.res}

begin

Application.Initialize;

Application.CreateForm(TForm2, Form2);

Application.Run;

end.

**Вывод:** изучил средства отладки программ в среде DELPHI. Составил и отладил программу для циклического вычислительного процесса.

# **Работа 4.**

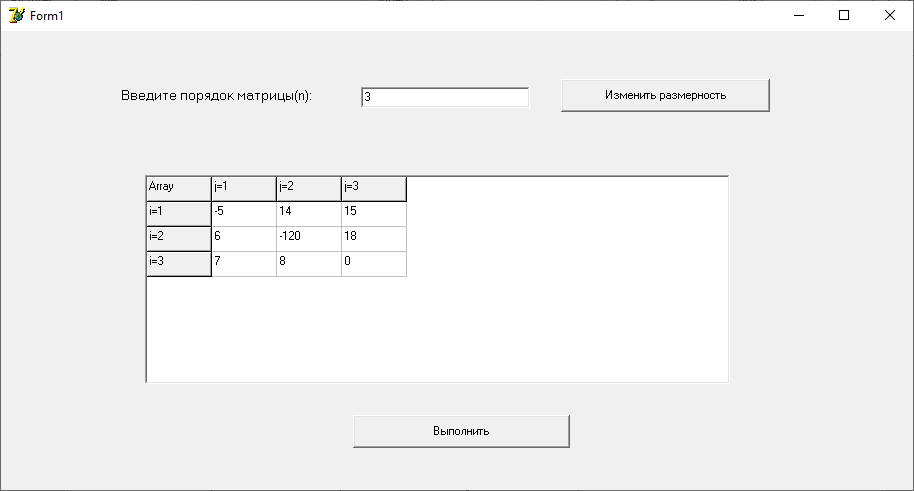
# **КОНСТРУИРОВАНИЕ ПРОГРАММ**

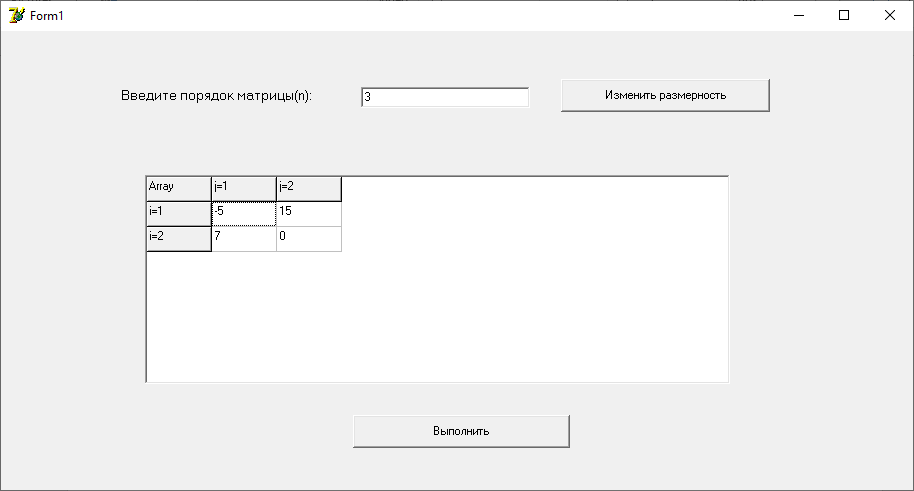
# **С ИСПОЛЬЗОВАНИЕМ МАССИВОВ**

**Цель работы:**изучить свойства компонента TStringGrid. Написать программу с использованием массивов

**Задание:** Из матрицы *п-*го порядка получить матрицу порядка *п-1* путем удаления из исходной матрицы строки и столбца, на пересечении которых расположен элемент с наибольшим по модулю значением.

**Скриншоты результатов:**





**Код программы:**

unit Unit1;

**//VARABEI IVAN**

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,

Dialogs, Grids, StdCtrls;

const

maxSize = 10;

type

A = array[1..maxSize, 1..maxSize] of integer;

TForm1 = class(TForm)

Label1: TLabel;

Edit1: TEdit;

StringGrid1: TStringGrid;

Button1: TButton;

Button2: TButton;

procedure FormCreate(Sender: TObject);

procedure Edit1Change(Sender: TObject);

procedure Button1Click(Sender: TObject);

procedure Button2Click(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

Form1: TForm1;

implementation

{$R \*.dfm}

procedure TForm1.FormCreate(Sender: TObject);

var n, i : integer;

begin

n:=StrToInt(Edit1.Text);

StringGrid1.RowCount := n + 1;

StringGrid1.ColCount := n + 1;

StringGrid1.Cells[0,0] := 'Array';

for i:=1 to n do begin

StringGrid1.Cells[0, i] := 'i=' + IntToStr(i);

StringGrid1.Cells[i, 0] := 'j=' + IntToStr(i);

end;

end;

procedure TForm1.Edit1Change(Sender: TObject);

begin

StringGrid1.RowCount := StrToInt(Edit1.Text) + 1;

end;

procedure TForm1.Button1Click(Sender: TObject);

var n, i: integer;

begin

n:=StrToInt(Edit1.Text);

StringGrid1.RowCount := n + 1;

StringGrid1.ColCount := n + 1;

for i:=1 to n do begin

StringGrid1.Cells[0, i] := 'i=' + IntToStr(i);

StringGrid1.Cells[i, 0] := 'j=' + IntToStr(i);

end;

end;

procedure TForm1.Button2Click(Sender: TObject);

var i, j, n, max, maxI, maxJ:integer;

arr : A;

begin

n:=StrToInt(Edit1.Text);

max:=StrToInt(StringGrid1.Cells[1,1]);

maxI:=1;

maxJ:=1;

for i:=1 to n do

for j:=1 to n do begin

arr[i, j] := StrToInt(StringGrid1.Cells[i, j]);

if abs(arr[i,j]) > max then begin

max := abs(arr[i, j]);

maxI := i;

maxJ := j;

end;

end;

for i:=maxI+1 to n do

for j:=1 to n do

StringGrid1.Cells[i-1, j] := StringGrid1.Cells[i, j];

for i:=1 to n do

for j:=maxJ+1 to n do

StringGrid1.Cells[i, j-1] := StringGrid1.Cells[i, j];

StringGrid1.RowCount:=n;

StringGrid1.ColCount:=n;

end;

end.

**Вывод:** изучил свойства компонента TStringGrid. Написал программу с использованием массивов

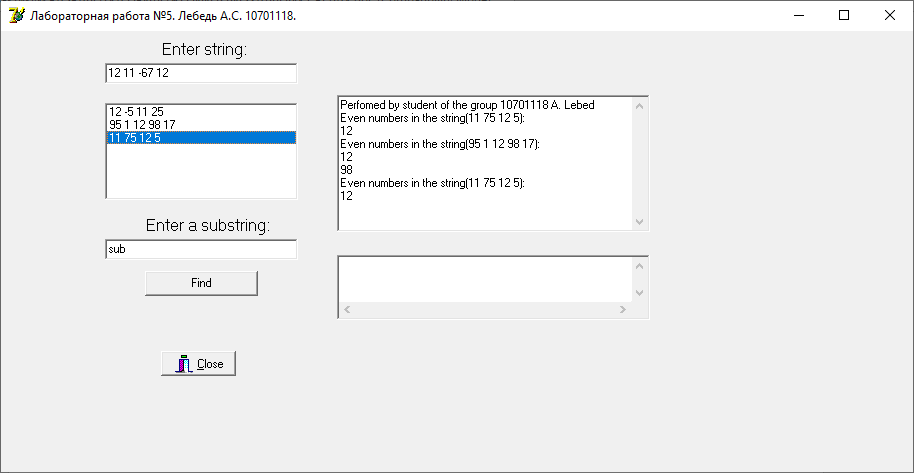
# **Работа 5.**

# **ПРОГРАММИРОВАНИЕ С ИСПОЛЬЗОВАНИЕМ СТРОК**

**Цель работы:**изучить методы программирования с использованием строк и правила работы с компонентами TListBox и TComboBox. Написать программу работы со строками.

**Задание:** Дана строка символов, состоящая из произвольных десятичных цифр, разделенных пробелами. Вывести четные числа этой строки.

**Скриншоты результатов:**



**Код программы:**

unit Unit1;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,

Dialogs, StdCtrls, Buttons;

type

TStringArray = array of string;

TForm1 = class(TForm)

ListBox1: TListBox;

Edit1: TEdit;

Label1: TLabel;

BitBtn1: TBitBtn;

Memo1: TMemo;

Label2: TLabel;

Edit2: TEdit;

Button1: TButton;

Memo2: TMemo;

procedure Edit1KeyPress(Sender: TObject; var Key: Char);

procedure FormActivate(Sender: TObject);

procedure ListBox1Click(Sender: TObject);

procedure FormCreate(Sender: TObject);

procedure Button1Click(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

Form1: TForm1;

implementation

{$R \*.dfm}

procedure TForm1.Edit1KeyPress(Sender: TObject; var Key: Char);

begin

if key = #13 then begin

if Edit1.Text <> '' then begin

ListBox1.Items.Add(Edit1.Text);

Edit1.Text := '';

end;

end;

end;

procedure TForm1.FormActivate(Sender: TObject);

begin

Edit1.SetFocus;

end;

function IsNumber(str : string) : boolean;

var i,index : integer;

begin

IsNumber := True;

index:=1;

if str[1] = '-' then

inc(index);

for i:=index to length(str) do begin

if not (str[i] in ['0'..'9']) then

IsNumber := False;

end;

end;

function Split(str : string) : TStringArray;

var temp : string;

i, amount, tempIndex, wordsIndex : integer;

words : TStringArray;

begin

amount:=1;

for i:=1 to length(str) do begin

if str[i] = ' ' then

amount:= amount + 1;

end;

SetLength(words, amount);

tempIndex:=1;

wordsIndex:=0;

for i:=1 to length(str) do begin

if str[i] = ' ' then begin

words[wordsIndex]:=temp;

temp:='';

tempIndex:=1;

wordsIndex:=wordsIndex + 1;

end

**//VARABEI IVAN**

else begin

insert(str[i], temp, tempIndex);

tempIndex:=tempIndex + 1;

end;

end;

words[wordsIndex]:=temp;

Split := words;

end;

procedure TForm1.ListBox1Click(Sender: TObject);

var str : string;

i, number: integer;

words : TStringArray;

begin

str := ListBox1.Items[ListBox1.ItemIndex];

words := Split(str);

Memo1.Lines.Add('Even numbers in the string(' + str + '):');

for i:=0 to (length(words)-1) do begin

if IsNumber(words[i]) then begin

number:= strToInt(words[i]);

if (abs(number) mod 2) = 0 then

Memo1.Lines.Add(intToStr(number));

end;

end;

end;

procedure TForm1.FormCreate(Sender: TObject);

begin

Memo1.Clear();

Memo1.Lines.Add('Perfomed by student of the group 10701118 A. Lebed');

end;

procedure TForm1.Button1Click(Sender: TObject);

var str, substr, strCopy : string;

amount, index : integer;

begin

str:=Form1.ListBox1.Items[Form1.ListBox1.ItemIndex];

substr:=Edit2.Text;

strCopy:=str;

amount := 0;

repeat

index := pos(substr, strCopy);

if not(index = 0) then begin

delete(strCopy, index, length(substr));

inc(amount);

index := index + length(substr);

end else break;

until index >= length(strCopy);

if amount > 0 then

Memo2.Lines.Add('Подстрока(' + substr + ') встречается ' + inttostr(amount) + ' раз(а) в строке(' + str + ')')

else

Memo2.Lines.Add(Подсрока(' + substr + ') не встречается в строке (' + str + ')');

end;

end.

**Вывод:** изучил методы программирования с использованием строк и правила работы с компонентами TListBox и TComboBox. Написал программу работы со строками.

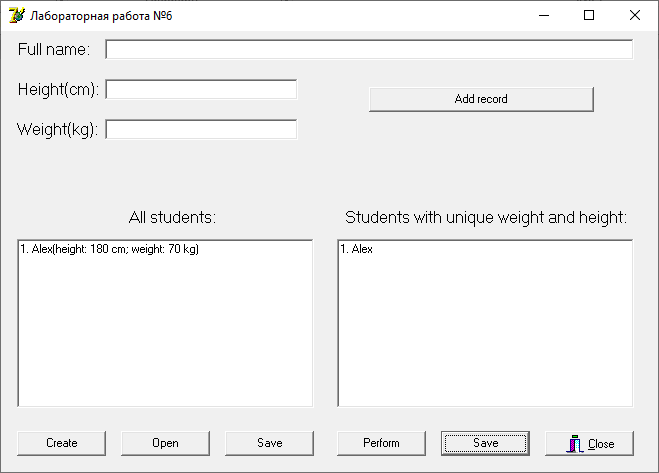
# **Работа 6.**

# **ПРОГРАММИРОВАНИЕ С ИСПОЛЬЗОВАНИЕМ ЗАПИСЕЙ И ФАЙЛОВ**

**Цель работы:**изучить правила работы с компонентами TopenDialog и TSaveDialog. Написать программу с использованием файлов и данных типов запись.

**Задание:** Список группы студентов содержит следующую информацию: Ф.И.О., рост и вес. Вывести Ф.И.О. студентов, рост и вес которых чаще всего встречаются в списке.

**Скриншоты результатов:**



**Код программы:**

unit Unit6;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,

Dialogs, StdCtrls, Buttons;

type

TStudent = record

fullName : string[255];

height : integer;

weight : extended;

end;

TForm1 = class(TForm)

Button1: TButton;

Button2: TButton;

BitBtn1: TBitBtn;

Memo1: TMemo;

Button3: TButton;

Edit1: TEdit;

Label1: TLabel;

Label2: TLabel;

Label3: TLabel;

Edit2: TEdit;

Edit3: TEdit;

Button4: TButton;

OpenDialog1: TOpenDialog;

SaveDialog1: TSaveDialog;

Label4: TLabel;

Memo3: TMemo;

Label5: TLabel;

Button5: TButton;

Button6: TButton;

procedure FormCreate(Sender: TObject);

procedure Button1Click(Sender: TObject);

procedure Button4Click(Sender: TObject);

procedure Button2Click(Sender: TObject);

procedure Button5Click(Sender: TObject);

procedure Button6Click(Sender: TObject);

procedure Button3Click(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

Form1: TForm1;

datFile : file of TStudent;

datFileName, textFileName : string;

txtFile : TextFile;

students : array of TStudent;

recordsQuantity : integer;

implementation

{$R \*.dfm}

procedure TForm1.FormCreate(Sender: TObject);

begin

Button4.Hide;

recordsQuantity := 0;

end;

//file creation

procedure TForm1.Button1Click(Sender: TObject);

begin

OpenDialog1.Title := 'Ñreate a new file';

if OpenDialog1.Execute then

begin

datFileName := OpenDialog1.FileName;

AssignFile(datFile, datFileName);

Rewrite(datFile);

Button4.Show;

end;

end;

//adding a record

procedure TForm1.Button4Click(Sender: TObject);

begin

inc(recordsQuantity);

SetLength(students, recordsQuantity);

with students[recordsQuantity-1] do begin

fullname := Edit1.Text;

height := strToInt(Edit2.Text);

weight := strToFloat(Edit3.Text);

Memo1.Lines.Add(IntToStr(recordsQuantity) + '. ' + fullname + '(height: ' +

IntToStr(height) + ' cm; weight: ' + FloatToStr(weight) + ' kg)');

end;

Edit1.Text := '';

Edit2.Text := '';

Edit3.Text := '';

end;

//opening file

procedure TForm1.Button2Click(Sender: TObject);

begin

OpenDialog1.Title := 'Open a file';

if OpenDialog1.Execute then

begin

datFileName := OpenDialog1.FileName;

AssignFile(datFile, datFileName);

Reset(datFile);

end;

recordsQuantity := 0;

Memo1.Clear;

while not eof(datFile) do begin

inc(recordsQuantity);

SetLength(students, recordsQuantity);

read(datfile, students[recordsQuantity-1]);

with students[recordsQuantity-1] do begin

Memo1.Lines.Add(IntToStr(recordsQuantity) + '. ' + fullname + '(height: ' +

IntToStr(height) + ' cm; weight: ' + FloatToStr(weight) + ' kg)');

end;

end;

Button4.Show;

closeFile(datFile);

end;

//saving to dat

procedure TForm1.Button5Click(Sender: TObject);

var

i : integer;

begin

if not(Memo1.Lines.Count = 0) then begin

SaveDialog1.Title := 'Save records';

if SaveDialog1.Execute then

begin

datFileName := SaveDialog1.FileName;

AssignFile(datFile, datFileName);

rewrite(datFile);

for i:=0 to recordsQuantity-1 do begin

write(datFile, students[i]);

end;

end;

closeFile(datFile);

end;

end;

//saving to txt

procedure TForm1.Button6Click(Sender: TObject);

begin

if not(Memo3.Lines.Count = 0) then begin

SaveDialog1.Title := 'Save records';

if SaveDialog1.Execute then

begin

textFileName := SaveDialog1.FileName;

Memo3.Lines.SaveToFile(textFileName);

end;

end;

end;

**//VARABEI IVAN**

procedure TForm1.Button3Click(Sender: TObject);

var

i, j, index : integer;

flag : boolean;

begin

Memo3.Clear;

index := 1;

for i:=0 to recordsQuantity-1 do begin

flag := True;

for j:=0 to recordsQuantity-1 do begin

if j = i then continue;

if (students[i].height = students[j].height)

or (students[i].weight = students[j].weight) then begin

flag := False;

end;

end;

if flag then begin

Memo3.Lines.Add(IntToStr(index) + '. ' + students[i].fullName);

inc(index);

end;

end;

end;

end.

**Вывод:** изучил правила работы с компонентами TopenDialog и TSaveDialog. Написал программу с использованием файлов и данных типов запись.

# **Работа 7.**

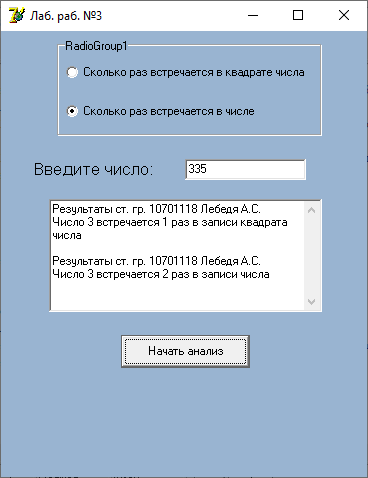
# **ПРОГРАММИРОВАНИЕ**

# **С ИСПОЛЬЗОВАНИЕМ ПОДПРОГРАММ И МОДУЛЕЙ**

**Цель работы:**изучить возможности DELPHI для написания подпрограмм и создания модулей. Составить и отладить программу, использующую внешний модуль Unit с подпрограммой.

**Задание:** По указанию преподавателя выберите вариант задачи из заданий, приведенных в работе 3. Предусмотрите возможность выбора функции, для которой будет рассчитываться таблица. Функции поместите в отдельный модуль. Вызывать выбранную функцию должна процедура, использующая в качестве входного параметра имя соответствующей функции.

**Скриншоты результатов:**



**Код программы:**

unit Unit2;

**//VARABEI IVAN**

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,

Dialogs, StdCtrls, Func, ExtCtrls;

type

TForm2 = class(TForm)

Label1: TLabel;

Edit1: TEdit;

Memo1: TMemo;

Button1: TButton;

RadioGroup1: TRadioGroup;

procedure Button1Click(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

Form2: TForm2;

implementation

{$R \*.dfm}

procedure useFunc(f : func1; number:integer; str : string);

begin

amount := f(number);

Form2.Memo1.Lines.Add('Ðåçóëüòàòû ñò. ãð. 10701118 Ëåáåäÿ À.Ñ.');

if amount > 0 then

Form2.Memo1.Lines.Add('×èñëî 3 âñòðå÷àåòñÿ ' + IntToStr(amount) + ' ðàç â çàïèñè ' + str)

else

Form2.Memo1.Lines.Add('×èñëî 3 íå ïðèñóòñòâóåò â çàïèñè ' + str);

Form2.Memo1.Lines.Add('');

end;

procedure TForm2.Button1Click(Sender: TObject);

var

n : integer;

begin

n:=strToInt(Edit1.Text);

case RadioGroup1.ItemIndex of

0: useFunc(amountOfDigitThreeInNumberSquare, n, 'êâàäðàòà ÷èñëà');

1: useFunc(amountOfDigitThreeInNumber, n, '÷èñëà');

end;

end;

end.

**//VARABEI IVAN**

unit Func;

interface

uses SysUtils;

function amountOfDigitThreeInNumberSquare(number : integer) : integer;

function amountOfDigitThreeInNumber(number : integer) : integer;

type

func1 = function(number:integer):integer;

var

amount : integer;

implementation

function amountOfDigitThreeInNumberSquare(number:integer):integer;

var

square, i, amount : integer;

squareStr : string;

begin

square:=number\*number;

squareStr:=IntToStr(square);

amount:=0;

for i:=1 to length(squareStr) do

begin

if squareStr[i] = '3' then

inc(amount);

end;

amountOfDigitThreeInNumberSquare := amount;

end;

function amountOfDigitThreeInNumber(number:integer):integer;

var

i, amount : integer;

numberStr : string;

begin

numberStr:=IntToStr(number);

amount:=0;

for i:=1 to length(numberStr) do

begin

if numberStr[i] = '3' then

inc(amount);

end;

amountOfDigitThreeInNumber := amount;

end;

end.

**Вывод:** изучил возможности DELPHI для написания подпрограмм и создания модулей. Составил и отладил программу, использующую внешний модуль Unit с подпрограммой.

# **Работа 8.**

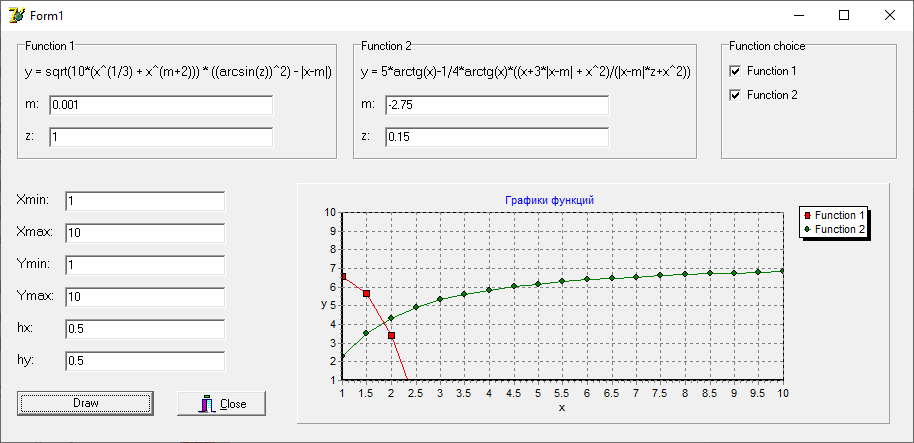
# **ПРОГРАММИРОВАНИЕ С ИСПОЛЬЗОВАНИЕМ СРЕДСТВ**

# **ДЛЯ ОТОБРАЖЕНИЯ ГРАФИЧЕСКОЙ ИНФОРМАЦИИ**

**Цель работы:**изучить возможности построения графиков с помощью компонента отображения графической информации TChart. Написать и отладить программу построения на экране графика заданной функции.

**Задание:** Постройте графики функций для соответствующих вариантов из работы №1. Таблицу данных получить путем изменения параметра X с шагом h. Вывод исходных данных организовать через окна TЕdit. Самостоятельно выбрать удобные параметры настройки.

**Скриншоты результатов:**



**Код программы:**

unit Unit8;

interface

**//VARABEI IVAN**

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,

Dialogs, TeEngine, Series, ExtCtrls, TeeProcs, Chart, Buttons, StdCtrls, Math;

type

TForm1 = class(TForm)

GroupBox1: TGroupBox;

Label1: TLabel;

Label2: TLabel;

Edit1: TEdit;

Label3: TLabel;

Edit2: TEdit;

GroupBox2: TGroupBox;

Label4: TLabel;

Label5: TLabel;

Label6: TLabel;

Edit3: TEdit;

Edit4: TEdit;

GroupBox3: TGroupBox;

CheckBox1: TCheckBox;

CheckBox2: TCheckBox;

Label7: TLabel;

Label8: TLabel;

Label9: TLabel;

Label10: TLabel;

Label11: TLabel;

Label13: TLabel;

Edit5: TEdit;

Edit6: TEdit;

Edit7: TEdit;

Edit8: TEdit;

Edit9: TEdit;

Edit10: TEdit;

Button1: TButton;

BitBtn1: TBitBtn;

Chart1: TChart;

Series1: TLineSeries;

Series2: TLineSeries;

procedure Button1Click(Sender: TObject);

procedure FormCreate(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

Form1: TForm1;

implementation

{$R \*.dfm}

function Function1(x, m, z : extended) : extended;

begin

Function1 := Sqrt(10\*(Power(x, 1.0/3) + Power(x,m+2))) \* (Power(arcsin(z), 2) - abs(x-m));

end;

function Function2(x, m, z : extended) : extended;

begin

Function2 := 5\*arctan(x)-1.0/4\*arctan(x)\*((x+3\*abs(x-m)+x\*x)/(abs(x-m)\*z+x\*x));

end;

procedure TForm1.Button1Click(Sender: TObject);

var xMin, xMax, yMin, yMax, hx, hy, x, y1, y2, m1,z1,m2,z2 : extended;

begin

xMin := strToFloat(Edit5.Text);

xMax := strToFloat(Edit6.Text);

yMin := strToFloat(Edit7.Text);

yMax := strToFloat(Edit8.Text);

hx := strToFloat(Edit9.Text);

hy := strToFloat(Edit10.Text);

Chart1.BottomAxis.SetMinMax(xMin, xMax);

Chart1.BottomAxis.Increment := hx;

Chart1.LeftAxis.SetMinMax(yMin, yMax);

Chart1.LeftAxis.Increment := hy;

Series1.Clear;

Series2.Clear;

x := xMin;

if CheckBox1.Checked then begin

m1 := strToFloat(Edit1.Text);

z1 := strToFloat(Edit2.Text);

end;

if CheckBox2.Checked then begin

m2 := strToFloat(Edit3.Text);

z2 := strToFloat(Edit4.Text);

end;

Repeat

if CheckBox1.Checked then begin

y1 := Function1(x, m1, z1);

Series1.AddXY(x, y1, '');

end;

if CheckBox2.Checked then begin

y2 := Function2(x, m2, z2);

Series2.AddXY(x, y2, '');

end;

x := x + hx;

Until (x > Xmax);

end;

procedure TForm1.FormCreate(Sender: TObject);

begin

Chart1.Title.Text[0] := 'Ãðàôèêè ôóíêöèé';

end;

end.

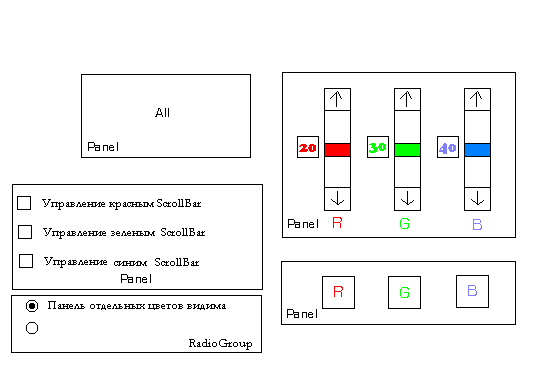
**Вывод:** изучил возможности построения графиков с помощью компонента отображения графической информации TChart. Написал и отладил программу построения на экране графика заданной функции.

# **Работа 9.**

# **СОЗДАНИЕ НЕСКОЛЬКИХ ФОРМ И СРЕДСТВА УПРАВЛЕНИЯ ИМИ**

**Цель работы:**Изучить основные свойства и методы, связанные с созданием и активизацией форм

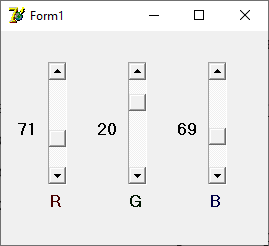
**Задание:** Взять за основу задание к лаб. раб. №7. Добавить к проекту дополнительную форму, на которой реализовать смешивание цветов. При этом должен быть организован диалог, как показано ниже.

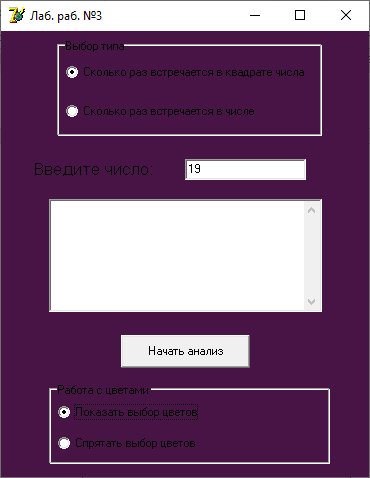


Выбранный цвет необходимо применить к одной из компонент главной формы.

Замечание. Вызов формы смешивания цветов реализовать в модальном и немодальном режимах

**Скриншоты результатов:**





**Код программы:**

unit Unit2;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,

Dialogs, StdCtrls, Func, ExtCtrls;

type

TForm2 = class(TForm)

Label1: TLabel;

Edit1: TEdit;

Memo1: TMemo;

Button1: TButton;

RadioGroup1: TRadioGroup;

RadioGroup2: TRadioGroup;

procedure Button1Click(Sender: TObject);

procedure RadioGroup2Click(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

**//VARABEI IVAN**

var

Form2: TForm2;

implementation

uses Unit1;

{$R \*.dfm}

procedure useFunc(f : func1; number:integer; str : string);

begin

amount := f(number);

Form2.Memo1.Lines.Add('Ðåçóëüòàòû ñò. ãð. 10701118 Ëåáåäÿ À.Ñ.');

if amount > 0 then

Form2.Memo1.Lines.Add('×èñëî 3 âñòðå÷àåòñÿ ' + IntToStr(amount) + ' ðàç â çàïèñè ' + str)

else

Form2.Memo1.Lines.Add('×èñëî 3 íå ïðèñóòñòâóåò â çàïèñè ' + str);

Form2.Memo1.Lines.Add('');

end;

procedure TForm2.Button1Click(Sender: TObject);

var

n : integer;

begin

n:=strToInt(Edit1.Text);

case RadioGroup1.ItemIndex of

0: useFunc(amountOfDigitThreeInNumberSquare, n, 'êâàäðàòà ÷èñëà');

1: useFunc(amountOfDigitThreeInNumber, n, '÷èñëà');

end;

end;

procedure TForm2.RadioGroup2Click(Sender: TObject);

begin

if RadioGroup2.ItemIndex = 0 then

begin

Form1.Show;

end else begin

Form1.Hide;

end;

end;

end.

unit Unit1;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,

Dialogs, StdCtrls;

type

TForm1 = class(TForm)

ScrollBar1: TScrollBar;

ScrollBar2: TScrollBar;

ScrollBar3: TScrollBar;

Label1: TLabel;

Label2: TLabel;

Label3: TLabel;

Label4: TLabel;

Label5: TLabel;

Label6: TLabel;

procedure ScrollBar2Change(Sender: TObject);

procedure ScrollBar1Change(Sender: TObject);

procedure ScrollBar3Change(Sender: TObject);

procedure FormCreate(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

Form1: TForm1;

implementation

uses Unit2;

{$R \*.dfm}

procedure changeInitialFormColor();

begin

Form2.Color := rgb(

Form1.ScrollBar1.Position,

Form1.Scrollbar2.Position,

Form1.Scrollbar3.Position

);

end;

procedure TForm1.ScrollBar2Change(Sender: TObject);

begin

label2.Font.Color := rgb(0, ScrollBar2.Position, 0);

changeInitialFormColor();

label5.Caption := IntToStr(Scrollbar2.Position);

end;

procedure TForm1.ScrollBar1Change(Sender: TObject);

begin

label1.Font.Color := rgb(ScrollBar1.Position, 0, 0);

changeInitialFormColor();

label4.Caption := IntToStr(Scrollbar1.Position);

end;

procedure TForm1.ScrollBar3Change(Sender: TObject);

begin

label3.Font.Color := rgb(0, 0, ScrollBar3.Position);

changeInitialFormColor();

label6.Caption := IntToStr(Scrollbar3.Position);

end;

procedure TForm1.FormCreate(Sender: TObject);

begin

label4.Caption := IntToStr(Scrollbar1.Position);

label5.Caption := IntToStr(Scrollbar2.Position);

label6.Caption := IntToStr(Scrollbar3.Position);

end;

end.

unit Func;

interface

uses SysUtils;

function amountOfDigitThreeInNumberSquare(number : integer) : integer;

function amountOfDigitThreeInNumber(number : integer) : integer;

type

func1 = function(number:integer):integer;

var

amount : integer;

implementation

function amountOfDigitThreeInNumberSquare(number:integer):integer;

var

square, i, amount : integer;

squareStr : string;

begin

square:=number\*number;

squareStr:=IntToStr(square);

amount:=0;

for i:=1 to length(squareStr) do

begin

if squareStr[i] = '3' then

inc(amount);

end;

amountOfDigitThreeInNumberSquare := amount;

end;

function amountOfDigitThreeInNumber(number:integer):integer;

var

i, amount : integer;

numberStr : string;

begin

numberStr:=IntToStr(number);

amount:=0;

for i:=1 to length(numberStr) do

begin

if numberStr[i] = '3' then

inc(amount);

end;

amountOfDigitThreeInNumber := amount;

end;

end.

**Вывод:** изучил основные свойства и методы, связанные с созданием и активизацией форм.

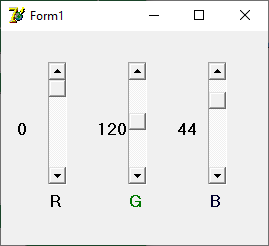
# **Работа 10.**

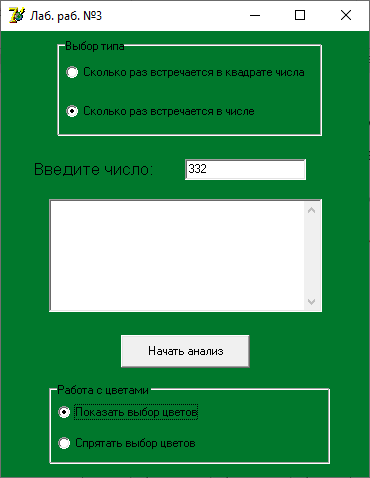
# **СОЗДАНИЕ НЕСКОЛЬКИХ ФОРМ И СРЕДСТВА УПРАВЛЕНИЯ ИМИ**

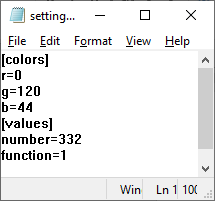
**Цель работы:**Изучить возможности автоматического сохранения параметров и установок, принятых в программе

**Задание:** Реализовать чтение и сохранение параметров при запуске и завершении программы. В качестве задания использовать л.р. №5.

**Скриншоты результатов:**







**Код программы:**

**//VARABEI IVAN**

unit Unit2;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,

Dialogs, StdCtrls, Func, ExtCtrls, IniFiles;

procedure changeInitialFormColor();

type

TForm2 = class(TForm)

Label1: TLabel;

Edit1: TEdit;

Memo1: TMemo;

Button1: TButton;

RadioGroup1: TRadioGroup;

RadioGroup2: TRadioGroup;

procedure Button1Click(Sender: TObject);

procedure RadioGroup2Click(Sender: TObject);

procedure FormCreate(Sender: TObject);

procedure FormDestroy(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

Form2: TForm2;

r, g, b : integer;

iniFile : TIniFile;

implementation

uses Unit1;

{$R \*.dfm}

procedure useFunc(f : func1; number:integer; str : string);

begin

amount := f(number);

Form2.Memo1.Lines.Add('Ðåçóëüòàòû ñò. ãð. 10701118 Ëåáåäÿ À.Ñ.');

if amount > 0 then

Form2.Memo1.Lines.Add('×èñëî 3 âñòðå÷àåòñÿ ' + IntToStr(amount) + ' ðàç â çàïèñè ' + str)

else

Form2.Memo1.Lines.Add('×èñëî 3 íå ïðèñóòñòâóåò â çàïèñè ' + str);

Form2.Memo1.Lines.Add('');

end;

procedure TForm2.Button1Click(Sender: TObject);

var

n : integer;

begin

n:=strToInt(Edit1.Text);

case RadioGroup1.ItemIndex of

0: useFunc(amountOfDigitThreeInNumberSquare, n, 'êâàäðàòà ÷èñëà');

1: useFunc(amountOfDigitThreeInNumber, n, '÷èñëà');

end;

end;

procedure TForm2.RadioGroup2Click(Sender: TObject);

begin

if RadioGroup2.ItemIndex = 0 then

begin

Form1.ShowModal;

end else begin

Form1.Hide;

end;

end;

procedure changeInitialFormColor();

begin

Form2.Color := rgb(r,g,b);

end;

procedure TForm2.FormCreate(Sender: TObject);

begin

iniFile := TiniFile.Create(extractfilepath(paramstr(0))+'settings.ini');

RadioGroup1.ItemIndex := inifile.ReadInteger('values', 'function', 0);

r := iniFile.ReadInteger('colors', 'r', 255);

g := iniFile.ReadInteger('colors', 'g', 255);

b := iniFile.ReadInteger('colors', 'b', 255);

Edit1.Text := iniFile.ReadString('values', 'number', '0');

changeInitialFormColor();

end;

procedure TForm2.FormDestroy(Sender: TObject);

begin

inifile.WriteInteger('colors', 'r', r);

inifile.WriteInteger('colors', 'g', g);

inifile.WriteInteger('colors', 'b', b);

inifile.WriteString('values', 'number', Edit1.Text);

inifile.WriteInteger('values', 'function', RadioGroup1.ItemIndex);

end;

end.

unit Unit1;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,

Dialogs, StdCtrls, Unit2;

type

TForm1 = class(TForm)

ScrollBar1: TScrollBar;

ScrollBar2: TScrollBar;

ScrollBar3: TScrollBar;

Label1: TLabel;

Label2: TLabel;

Label3: TLabel;

Label4: TLabel;

Label5: TLabel;

Label6: TLabel;

procedure ScrollBar2Change(Sender: TObject);

procedure ScrollBar1Change(Sender: TObject);

procedure ScrollBar3Change(Sender: TObject);

procedure FormCreate(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

Form1: TForm1;

implementation

{$R \*.dfm}

procedure TForm1.ScrollBar2Change(Sender: TObject);

begin

g := ScrollBar2.Position;

label2.Font.Color := rgb(0, ScrollBar2.Position, 0);

changeInitialFormColor();

label5.Caption := IntToStr(Scrollbar2.Position);

end;

procedure TForm1.ScrollBar1Change(Sender: TObject);

begin

r := ScrollBar1.Position;

label1.Font.Color := rgb(ScrollBar1.Position, 0, 0);

changeInitialFormColor();

label4.Caption := IntToStr(Scrollbar1.Position);

end;

procedure TForm1.ScrollBar3Change(Sender: TObject);

begin

b := ScrollBar3.Position;

label3.Font.Color := rgb(0, 0, ScrollBar3.Position);

changeInitialFormColor();

label6.Caption := IntToStr(Scrollbar3.Position);

end;

procedure TForm1.FormCreate(Sender: TObject);

begin

ScrollBar1.Position := r;

ScrollBar2.Position := g;

ScrollBar3.Position := b;

label4.Caption := IntToStr(r);

label5.Caption := IntToStr(g);

label6.Caption := IntToStr(b);

end;

**//VARABEI IVAN**

end.

**Вывод:** изучил возможности автоматического сохранения параметров и установок, принятых в программе.

# **Работа 11.**

# **ГРАФИЧЕСКИЕ КОМПОНЕНТЫ**

**Цель работы:**Изучить основные графические компоненты, их свойства и методы.

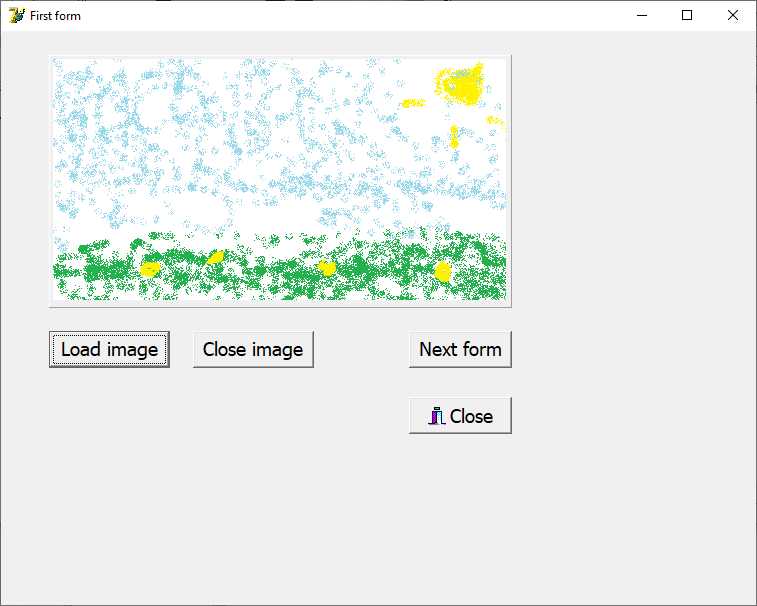
**Задание:** Разработать программу, содержащую три формы – три способа представления графической информации.

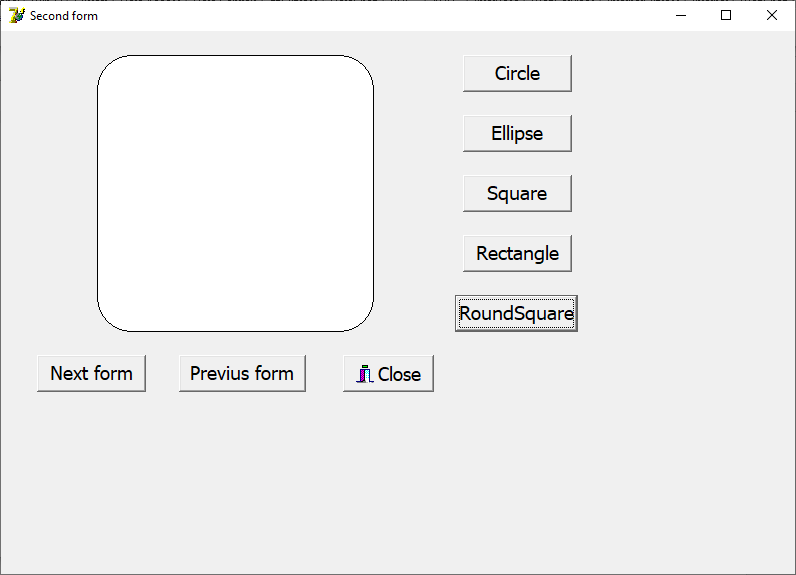
На 1-й форме продемонстрировать отображение графических картинок, созданных в других графических редакторах.

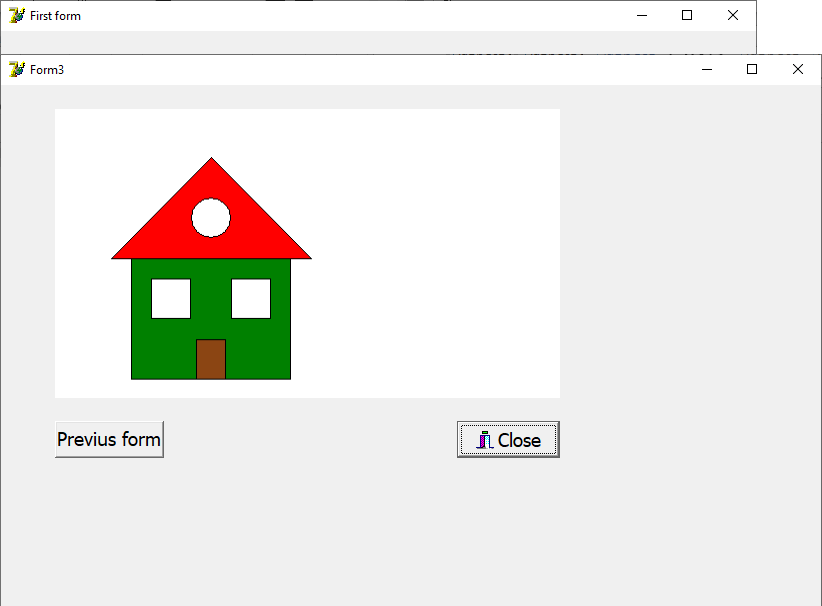
На 2-й форме с помощью кнопочного меню рисовать различные графические фигуры посредством компоненты класса TShape.

На 3-й форме реализовать рисование простейшими примитивами, типа линия, прямоугольник, эллипс и т.д.

**Скриншоты результатов:**







**Код программы:**

unit first;

**//VARABEI IVAN**

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,

Dialogs, ExtCtrls, ExtDlgs, StdCtrls, Buttons, JPEG;

type

TForm1 = class(TForm)

dlgOpenPic1: TOpenPictureDialog;

pnl1: TPanel;

btn1: TBitBtn;

btn2: TButton;

btn3: TButton;

btn4: TButton;

img1: TImage;

procedure FormCreate(Sender: TObject);

procedure btn2Click(Sender: TObject);

procedure btn3Click(Sender: TObject);

procedure btn4Click(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

Form1: TForm1;

implementation

uses second;

{$R \*.dfm}

procedure TForm1.FormCreate(Sender: TObject);

begin

Form1.Caption := 'First form';

pnl1.Caption := '';

btn1.Caption := 'Close';

btn1.Font.Size := 14;

btn2.Caption := 'Load image';

btn2.Font.Size := 14;

btn3.Caption := 'Close image';

btn3.Font.Size := 14;

btn4.Caption := 'Next form';

btn4.Font.Size := 14;

end;

procedure TForm1.btn2Click(Sender: TObject);

begin

pnl1.Visible := True;

dlgOpenPic1.Filter := GraphicFilter(TGraphic);

if dlgOpenPic1.Execute then

img1.Picture.LoadFromFile(dlgOpenPic1.FileName);

end;

procedure TForm1.btn3Click(Sender: TObject);

begin

img1.Picture := nil;

end;

procedure TForm1.btn4Click(Sender: TObject);

begin

Form1.Visible := False;

Form2.Visible := True;

Form2.SetFocus;

end;

end.

unit second;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,

Dialogs, StdCtrls, Buttons, ExtCtrls;

type

TForm2 = class(TForm)

shp1: TShape;

btn1: TBitBtn;

btn2: TButton;

btn3: TButton;

btn4: TButton;

btn5: TButton;

btn6: TButton;

btn7: TButton;

btn8: TButton;

procedure FormCreate(Sender: TObject);

procedure btn8Click(Sender: TObject);

procedure btn1Click(Sender: TObject);

procedure btn2Click(Sender: TObject);

procedure btn3Click(Sender: TObject);

procedure btn4Click(Sender: TObject);

procedure btn5Click(Sender: TObject);

procedure btn6Click(Sender: TObject);

procedure btn7Click(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

Form2: TForm2;

implementation

uses first, third;

{$R \*.dfm}

procedure TForm2.FormCreate(Sender: TObject);

begin

Form2.Caption := 'Second form';

btn1.Caption := 'Close';

btn1.Font.Size := 14;

btn2.Caption := 'Circle';

btn2.Font.Size := 14;

btn3.Caption := 'Ellipse';

btn3.Font.Size := 14;

btn4.Caption := 'Square';

btn4.Font.Size := 14;

btn5.Caption := 'Rectangle';

btn5.Font.Size := 14;

btn6.Caption := 'RoundSquare';

btn6.Font.Size := 14;

btn7.Caption := 'Next form';

btn7.Font.Size := 14;

btn8.Caption := 'Previus form';

btn8.Font.Size := 14;

end;

procedure TForm2.btn8Click(Sender: TObject);

begin

Form2.Visible := False;

Form1.Visible := True;

Form1.SetFocus;

end;

procedure TForm2.btn1Click(Sender: TObject);

begin

Form1.Close();

Form2.Close();

Form3.Close();

end;

procedure TForm2.btn2Click(Sender: TObject);

begin

shp1.Shape:=stCircle;

end;

procedure TForm2.btn3Click(Sender: TObject);

begin

shp1.Shape:=stEllipse;

end;

procedure TForm2.btn4Click(Sender: TObject);

begin

shp1.Shape := stSquare;

end;

procedure TForm2.btn5Click(Sender: TObject);

begin

shp1.Shape := stRectangle;

end;

procedure TForm2.btn6Click(Sender: TObject);

begin

shp1.Shape := stRoundSquare;

end;

procedure TForm2.btn7Click(Sender: TObject);

begin

Form2.Visible := False;

Form3.Visible := True;

Form3.SetFocus;

end;

end.

unit third;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,

Dialogs, StdCtrls, Buttons, ExtCtrls;

type

TForm3 = class(TForm)

img1: TImage;

btn1: TBitBtn;

btn2: TButton;

procedure FormCreate(Sender: TObject);

procedure btn1Click(Sender: TObject);

procedure btn2Click(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

Form3: TForm3;

implementation

uses first, second;

{$R \*.dfm}

procedure TForm3.FormCreate(Sender: TObject);

begin

btn1.Caption := 'Close';

btn1.Font.Size := 14;

btn2.Caption := 'Previus form';

btn2.Font.Size := 14;

Form3.img1.Canvas.MoveTo(50, 200);

Form3.img1.Canvas.LineTo(100, 150);

Form3.img1.Canvas.LineTo(150, 200);

Form3.img1.Canvas.LineTo(50, 200);

Form3.img1.Canvas.LineTo(50, 275);

Form3.img1.Canvas.LineTo(150, 275);

Form3.img1.Canvas.LineTo(150, 200);

Form3.img1.Canvas.Rectangle(100, 100, 200, 200);

Form3.img1.Canvas.Rectangle(50, 301, 207, 130);

Form3.img1.Canvas.Rectangle(70, 540, 607, 100);

Form3.img1.Canvas.Rectangle(110, 10, 11, 250);

Form3.img1.Canvas.Ellipse(100, 110, 200, 210);

Form3.img1.Canvas.MoveTo(120, 150);

Form3.img1.Canvas.LineTo(120, 170);

**//VARABEI IVAN**

end;

procedure TForm3.btn1Click(Sender: TObject);

begin

Form1.Close();

Form2.Close();

Form3.Close();

end;

procedure TForm3.btn2Click(Sender: TObject);

begin

Form3.Visible := False;

Form2.Visible := True;

Form2.SetFocus;

end;

end.

**Вывод:** изучить основные графические компоненты, их свойства и методы.

# **Работа 12.**

# **УПРАВЛЕНИЕ ПРОГРАММОЙ С ПОМОЩЬЮ ПАНЕЛИ ИНСТРУМЕНТОВ**

**Цель работы:** Научиться подключать инструментальную панель, изучить основные свойства и типы кнопок и использование их для управления вычислительным процессом

**Задание:** Разработать программу, в которой предусмотреть управление вычислительным процессом с помощью кнопок, расположенных на инструментальной панели. Задание использовать из л.р.№8.

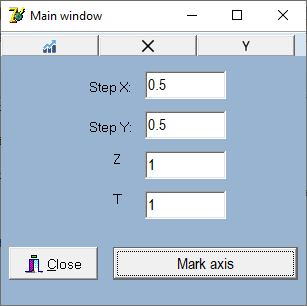
# **Работа 13.**

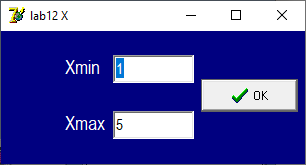
# **СОЗДАНИЕ МНОГОСТРАНИЧНОГО ДОКУМЕНТА**

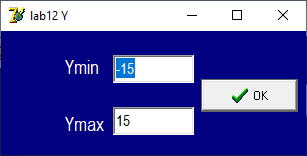
**Цель работы:**Изучить компоненты TPageControl, TTabSheet

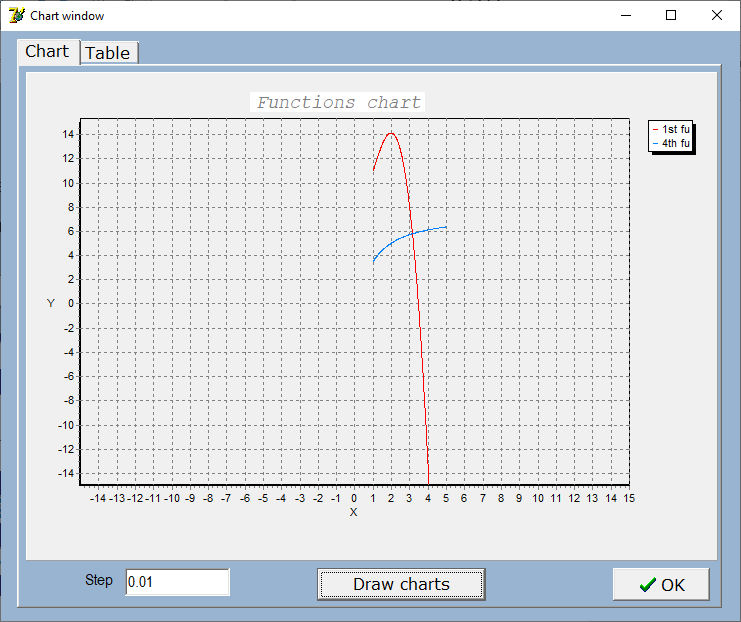
**Задание:** Разработать программу ввода данных, выбора метода расчета, расчета и представления результатов в табличной и графической формах на примере л.р. №8, реализуя отдельные вычислительные шаги на различных закладках многостраничного документа.

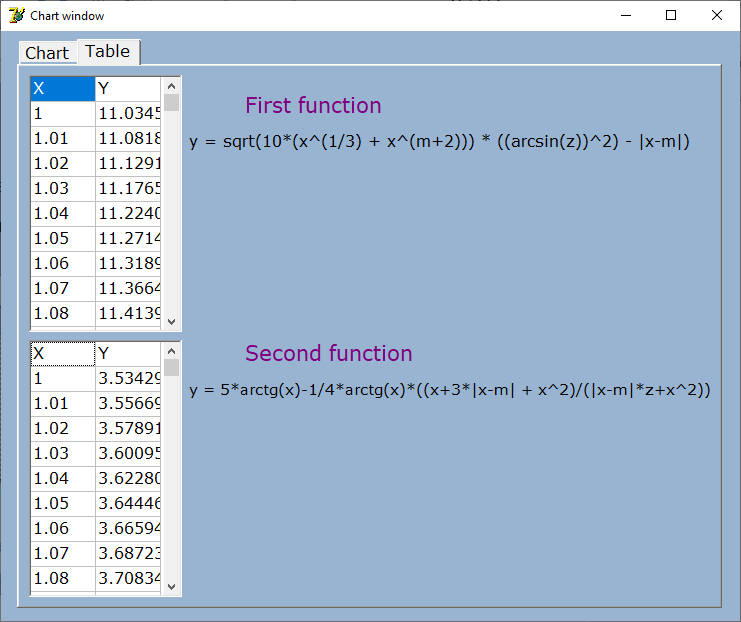
**Скриншоты результатов:**











**Код программы:**

**//VARABEI IVAN**

unit Unit1;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,

Dialogs, StdCtrls, Buttons, TeEngine, Series, ExtCtrls, TeeProcs, Chart,

ToolWin, ComCtrls, ImgList;

type

TForm1 = class(TForm)

Button2: TButton;

BitBtn1: TBitBtn;

Label5: TLabel;

Label6: TLabel;

Edit5: TEdit;

Edit6: TEdit;

Label8: TLabel;

Edit8: TEdit;

Label9: TLabel;

Edit9: TEdit;

ToolBar1: TToolBar;

ToolButton2: TToolButton;

ImageList1: TImageList;

ToolButton3: TToolButton;

ToolButton4: TToolButton;

procedure FormCreate(Sender: TObject);

procedure Button2Click(Sender: TObject);

procedure ToolButton1Click(Sender: TObject);

procedure ToolButton2Click(Sender: TObject);

procedure ToolButton3Click(Sender: TObject);

procedure ToolButton4Click(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

Form1: TForm1;

Xmin, Xmax, Ymin, Ymax, Hx, Hy, h, Z, T : extended;

implementation

uses Unit2, Unit3, Unit4;

{$R \*.dfm}

procedure TForm1.FormCreate(Sender: TObject);

begin

ToolBar1.Images := ImageList1;

Hx := 0.5;

Hy := 0.5;

Z := 1;

T := 1;

Edit5.Text := FloatToStr(Hx);

Edit6.Text := FloatToStr(Hy);

Edit8.Text := FloatToStr(Z);

Edit9.Text := FloatToStr(T);

end;

Procedure TForm1.Button2Click(Sender: TObject);

begin

Unit2.Form2.Chart2.BottomAxis.Increment := StrToFloat(Edit5.Text);

Unit2.Form2.Chart2.LeftAxis.Increment := StrToFloat(Edit6.Text);

end;

procedure TForm1.ToolButton1Click(Sender: TObject);

begin

Form2.Visible := False;

Form3.Visible := False;

Form4.Visible := False;

end;

procedure TForm1.ToolButton2Click(Sender: TObject);

begin

Form2.Visible := True;

end;

procedure TForm1.ToolButton3Click(Sender: TObject);

begin

Form3.Visible := True;

end;

procedure TForm1.ToolButton4Click(Sender: TObject);

begin

Form4.Visible := True;

end;

end.

unit Unit2;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,

Dialogs, StdCtrls, TeEngine, Series, ExtCtrls, TeeProcs, Chart,

ComCtrls, ToolWin, Buttons, Grids, Math;

type

TForm2 = class(TForm)

PageControl1: TPageControl;

TabSheet1: TTabSheet;

TabSheet2: TTabSheet;

Chart2: TChart;

LineSeries1: TLineSeries;

LineSeries2: TLineSeries;

Label1: TLabel;

Edit1: TEdit;

Button2: TButton;

BitBtn2: TBitBtn;

StringGrid2: TStringGrid;

Label2: TLabel;

**//VARABEI IVAN**

Label3: TLabel;

StringGrid1: TStringGrid;

Label4: TLabel;

Label6: TLabel;

procedure FormCreate(Sender: TObject);

procedure Button2Click(Sender: TObject);

procedure BitBtn2Click(Sender: TObject);

procedure BitBtn1Click(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

Form2: TForm2;

implementation

uses Unit1, Unit3, Unit4;

{$R \*.dfm}

procedure TForm2.FormCreate(Sender: TObject);

begin

h := 0.01;

Edit1.Text := FloatToStr(h);

StringGrid1.Cells[0,0]:='X';

StringGrid1.Cells[1,0]:='Y';

StringGrid2.RowCount := 1;

StringGrid2.Cells[0,0]:='X';

StringGrid2.Cells[1,0]:='Y';

end;

procedure TForm2.Button2Click(Sender: TObject);

var

prom1, prom2, prom3, prom4, z, m, x, y1, y2: extended;

index : integer;

begin

StringGrid1.RowCount := 1;

StringGrid2.RowCount := 1;

LineSeries1.Clear;

LineSeries2.Clear;

z := StrToFloat(Unit1.Form1.Edit9.Text);

m := StrToFloat(Unit1.Form1.Edit8.Text);

Xmin := StrToFloat(Unit3.Form3.Edit1.Text);

Xmax := StrToFloat(Unit3.Form3.Edit2.Text);

h := StrToFloat(Edit1.Text);

x := Xmin;

index := 1;

Repeat

StringGrid1.RowCount := StringGrid1.RowCount + 1;

StringGrid2.RowCount := StringGrid2.RowCount + 1;

//1st func

y1 := Sqrt(10\*(Power(x, 1.0/3) + Power(x,m+2))) \* (Power(arcsin(z), 2) - abs(x-m));

LineSeries1.AddXY(x, y1, '', clTeeColor );

StringGrid1.Cells[0,index] := FloatToStr(x);

StringGrid1.Cells[1,index] := FloatToStr(y1);

//4th func

y2 := 5\*arctan(x)-1.0/4\*arctan(x)\*((x+3\*abs(x-m)+x\*x)/(abs(x-m)\*z+x\*x));

LineSeries2.AddXY(x, y2, '', clTeeColor);

StringGrid2.Cells[0,index] := FloatToStr(x);

StringGrid2.Cells[1,index] := FloatToStr(y2);

x := x + h;

index := index+1;

Until (x > Xmax);

end;

procedure TForm2.BitBtn2Click(Sender: TObject);

begin

Form2.Visible := False;

end;

procedure TForm2.BitBtn1Click(Sender: TObject);

begin

Form2.Visible := False;

end;

end.

unit Unit3;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,

Dialogs, StdCtrls, Buttons;

type

TForm3 = class(TForm)

Label1: TLabel;

Label2: TLabel;

Edit1: TEdit;

Edit2: TEdit;

BitBtn1: TBitBtn;

procedure FormCreate(Sender: TObject);

procedure BitBtn1Click(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

Form3: TForm3;

implementation

{$R \*.dfm}

procedure TForm3.FormCreate(Sender: TObject);

var Xmin, Xmax : extended;

begin

Xmin := 1;

Xmax := 5;

Edit1.Text := FloatToStr(Xmin);

Edit2.Text := FloatToStr(Xmax);

end;

procedure TForm3.BitBtn1Click(Sender: TObject);

begin

Form3.Visible := False;

end;

end.

unit Unit4;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,

Dialogs, StdCtrls, Buttons;

type

TForm4 = class(TForm)

Edit3: TEdit;

Edit4: TEdit;

Label3: TLabel;

Label4: TLabel;

BitBtn1: TBitBtn;

procedure FormCreate(Sender: TObject);

procedure BitBtn1Click(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

Form4: TForm4;

implementation

{$R \*.dfm}

procedure TForm4.FormCreate(Sender: TObject);

var Ymin, Ymax : extended;

begin

Ymin := -15;

Ymax := 15;

Edit3.Text := FloatToStr(Ymin);

Edit4.Text := FloatToStr(Ymax);

end;

procedure TForm4.BitBtn1Click(Sender: TObject);

begin

Form4.Visible := False;

end;

end.

**Вывод:** научился подключать инструментальную панель, изучил основные свойства и типы кнопок и использование их для управления вычислительным процессом, а также изучил компоненты TPageControl, TTabSheet.

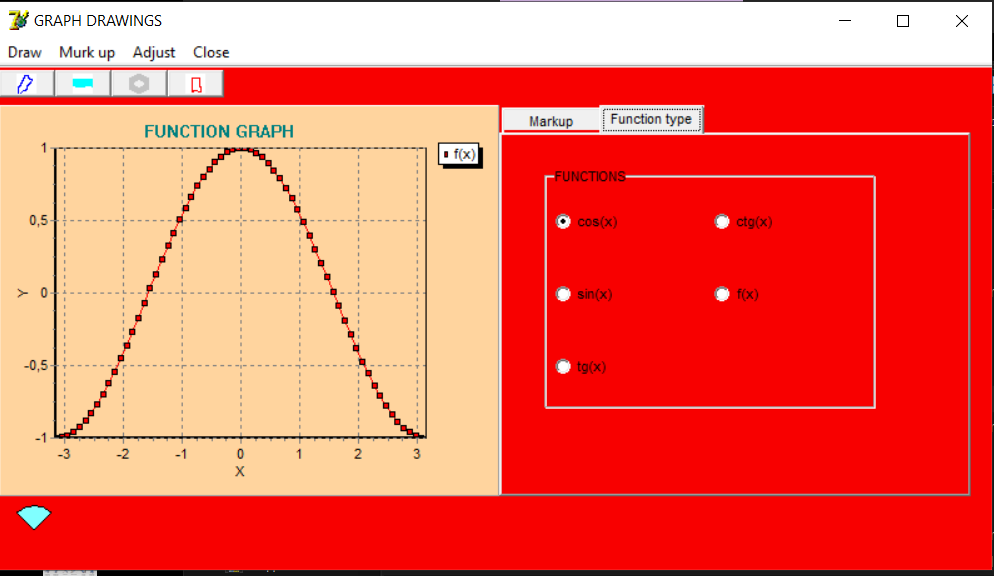
# **Работа 14.**

# **КОМПЛЕКСНАЯ ЛАБОРАТОРНАЯ РАБОТА**

**Цель работы:** Показать умение создания современного приложения на Delphi

**Задание:** Разработать программу, поддерживающую различные управления вычислительным процессом с помощьюTPageControl, TtabSheet, TСhar**,** Стандартные диалоговые компоненты, ТPopupMenu**,** TMainMenu, TtoolBar и др. на примере л.р. №8.

**Скриншоты результатов:**



**Код программы:**

unit lab8;

**//VARABEI IVAN**

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,

Dialogs, TeEngine, Series, Buttons, StdCtrls, ExtCtrls, TeeProcs, Chart, Math,

Menus, ComCtrls, ToolWin, ImgList, IniFiles;

type

TForm1 = class(TForm)

cht1: TChart;

sinxSeries1: TLineSeries;

il1: TImageList;

pgc1: TPageControl;

ts2: TTabSheet;

lbl1: TLabel;

lbl2: TLabel;

lbl3: TLabel;

lbl4: TLabel;

lbl6: TLabel;

lbl5: TLabel;

edt1: TEdit;

edt2: TEdit;

edt3: TEdit;

edt4: TEdit;

edt6: TEdit;

edt5: TEdit;

ts3: TTabSheet;

rg1: TRadioGroup;

mm1: TMainMenu;

N3: TMenuItem;

N4: TMenuItem;

N5: TMenuItem;

N1: TMenuItem;

N2: TMenuItem;

tlb1: TToolBar;

btn1: TToolButton;

btn2: TToolButton;

btn3: TToolButton;

btn4: TToolButton;

pb1: TPaintBox;

tmr1: TTimer;

N7: TMenuItem;

procedure N3Click(Sender: TObject);

procedure btn1Click(Sender: TObject);

procedure btn2Click(Sender: TObject);

procedure btn3Click(Sender: TObject);

procedure FormCreate(Sender: TObject);

procedure edt1Enter(Sender: TObject);

procedure edt2Enter(Sender: TObject);

procedure edt3Enter(Sender: TObject);

procedure edt4Enter(Sender: TObject);

procedure edt7Enter(Sender: TObject);

procedure edt6Enter(Sender: TObject);

procedure edt5Enter(Sender: TObject);

procedure rg1Click(Sender: TObject);

procedure N1Click(Sender: TObject);

procedure N2Click(Sender: TObject);

procedure FormDestroy(Sender: TObject);

procedure N5Click(Sender: TObject);

procedure N6Click(Sender: TObject);

procedure btn4Click(Sender: TObject);

procedure pb1Paint(Sender: TObject);

procedure tmr1Timer(Sender: TObject);

procedure N7Click(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

Form1: TForm1;

Xmin, Xmax, Ymin, Ymax, Hx, Hy: extended;

Ini : TIniFile;

choice : Integer;

colorofanimation : Integer;

implementation

{$R \*.dfm}

uses lab9;

procedure TForm1.N3Click(Sender: TObject);

begin

Close;

end;

procedure TForm1.btn1Click(Sender: TObject);

begin

N1.OnClick(Sender);

end;

procedure TForm1.btn2Click(Sender: TObject);

begin

N2.Click;

end;

procedure TForm1.btn3Click(Sender: TObject);

begin

Close;

end;

procedure TForm1.FormCreate(Sender: TObject);

begin

Ini:=TiniFile.Create(extractfilepath(paramstr(0)) + 'settings.ini');

Form1.Width:=Ini.ReadInteger('form','width', 300);

Form1.Height:=Ini.ReadInteger('form','height', 300);

Form1.Color := Ini.ReadInteger('form','color',0);

cht1.Color := Ini.ReadInteger('form','color\_of\_chart',0);

colorOfAnimation := Ini.ReadInteger('animation','color',clBlue);

tmr1.Interval := Ini.ReadInteger('animation','speed',500);

choice := Ini.ReadInteger('function', 'choice', 0);

rg1.ItemIndex := choice;

edt1.Text := inttostr(Ini.ReadInteger('function', 'x\_min', -5));

edt2.Text := inttostr(Ini.ReadInteger('function', 'x\_max', 5));

edt3.Text := inttostr(Ini.ReadInteger('function', 'y\_min', -5));

edt4.Text := inttostr(Ini.ReadInteger('function', 'y\_max', 5));

edt5.Text := inttostr(Ini.ReadInteger('function', 'step\_x', 1));

edt6.Text := inttostr(Ini.ReadInteger('function', 'step\_y', 2));

N2.Click;

N1.Click;

end;

procedure TForm1.edt1Enter(Sender: TObject);

begin

N2.Click;

N1.Click;

end;

procedure TForm1.edt2Enter(Sender: TObject);

begin

N2.Click;

N1.Click;

end;

procedure TForm1.edt3Enter(Sender: TObject);

begin

N2.Click;

N1.Click;

end;

procedure TForm1.edt4Enter(Sender: TObject);

begin

N2.Click;

N1.Click;

end;

procedure TForm1.edt7Enter(Sender: TObject);

begin

N2.Click;

N1.Click;

end;

procedure TForm1.edt6Enter(Sender: TObject);

begin

N2.Click;

N1.Click;

end;

procedure TForm1.edt5Enter(Sender: TObject);

begin

N2.Click;

N1.Click;

end;

procedure TForm1.rg1Click(Sender: TObject);

begin

N1.Click;

end;

procedure TForm1.N1Click(Sender: TObject);

Var

x, y2: extended;

begin

begin

x := Strtofloat(edt1.Text);

sinxSeries1.Clear;

Repeat

case rg1.ItemIndex of

0: y2 := exp(x);

1: y2 := cos(x);

2: y2 := sin(x);

3: y2 := tan(x);

4: y2 := x\*x;

end;

sinxSeries1.AddXY(x, y2, '');

x := x + strtoint(edt5.Text);

Until (x > Xmax);

end;

end;

procedure TForm1.N2Click(Sender: TObject);

begin

Xmin := StrToFloat(edt1.Text);

Xmax := StrToFloat(edt2.Text);

Ymin := StrToFloat(edt3.Text);

Ymax := StrToFloat(edt4.Text);

Hx := StrToFloat(edt5.Text);

Hy := StrToFloat(edt6.Text);

cht1.BottomAxis.Automatic := False;

cht1.BottomAxis.Minimum := Xmin;

cht1.BottomAxis.Maximum := Xmax;

cht1.LeftAxis.Automatic := False;

cht1.LeftAxis.Minimum := Ymin;

cht1.LeftAxis.Maximum := Ymax;

cht1.BottomAxis.Increment := Hx;

cht1.LeftAxis.Increment := Hy;

end;

procedure TForm1.FormDestroy(Sender: TObject);

begin

Ini.WriteInteger('form','width',Form1.Width);

Ini.WriteInteger('form','height',Form1.Height);

Ini.WriteInteger('form','color',Form1.Color);

Ini.WriteInteger('form','color\_of\_chart',cht1.Color);

Ini.WriteInteger('animation','speed',tmr1.Interval);

Ini.WriteInteger('animation','color',colorOfAnimation);

Ini.WriteInteger('function', 'choice', rg1.ItemIndex);

Ini.WriteInteger('function', 'x\_min', strToInt(edt1.Text));

Ini.WriteInteger('function', 'x\_max', strToInt(edt2.Text));

Ini.WriteInteger('function', 'y\_min', strToInt(edt3.Text));

Ini.WriteInteger('function', 'y\_max', strToInt(edt4.Text));

Ini.WriteInteger('function', 'step\_x', strToInt(edt5.Text));

Ini.WriteInteger('function', 'step\_y', strToInt(edt6.Text));

Ini.Free;

end;

procedure TForm1.N5Click(Sender: TObject);

begin

if (not Assigned(Form2)) then

Form2:=TForm2.Create(Self);

Form2.pgc1.TabIndex := 0;

Form2.ShowModal;

end;

procedure TForm1.N6Click(Sender: TObject);

begin

if (not Assigned(Form2)) then

Form2:=TForm2.Create(Self);

Form2.pgc1.TabIndex := 1;

Form2.ShowModal;

end;

procedure TForm1.btn4Click(Sender: TObject);

begin

if (not Assigned(Form2)) then

Form2:=TForm2.Create(Self);

Form2.ShowModal;

end;

procedure ClearPaintBox(Sender:TObject);

begin

with Form1.pb1.Canvas do

begin

Form1.pb1.Canvas.Brush.Color:=Form1.Color;

Form1.pb1.Canvas.FillRect(Rect(0,0,Form1.pb1.Width,Form1.pb1.Height));

end;

end;

procedure TForm1.pb1Paint(Sender: TObject);

var

x1,y1,x2,y2 : Integer;

begin

x1 := 0;

y1 := 0;

x2 := 40;

y2 := 40;

with Form1.pb1, canvas do

begin

Brush.Color := colorofanimation;

if choice = 0 then

begin

Arc(x1,y1,x2,y2,40,20,20,0);

end;

if choice = 1 then

begin

Arc(x1,y1,x2,y2,40,40,40,0);

end;

if choice = 2 then

begin

Arc(x1,y1,x2,y2,20,40,40,20);

end;

if choice = 3 then

begin

Arc(x1,y1,x2,y2,0,40,40,40);

end;

if choice = 4 then

begin

Arc(x1,y1,x2,y2,0,20,20,40);

end;

if choice = 5 then

begin

Arc(x1,y1,x2,y2,0,0,0,40);

end;

if choice = 6 then

begin;

Arc(x1,y1,x2,y2,20,0,0,20);

end;

if choice = 7 then

begin

Arc(x1,y1,x2,y2,40,0,0,0);

end;

end;

end;

procedure TForm1.tmr1Timer(Sender: TObject);

begin

ClearPaintBox(Sender);

pb1Paint(Sender);

choice := choice + 1;

if choice > 7 then

begin

choice := 0;

end;

end;

procedure TForm1.N7Click(Sender: TObject);

begin

if (not Assigned(Form2)) then

Form2:=TForm2.Create(Self);

Form2.pgc1.TabIndex := 2;

Form2.ShowModal;

end;

end.

unit lab9;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,

Dialogs, StdCtrls, ExtCtrls, ComCtrls;

type

TForm2 = class(TForm)

scrlbr1: TScrollBar;

scrlbr2: TScrollBar;

scrlbr3: TScrollBar;

rg1: TRadioGroup;

lbl1: TLabel;

lbl2: TLabel;

lbl3: TLabel;

lbl4: TLabel;

lbl5: TLabel;

lbl6: TLabel;

pgc1: TPageControl;

ts1: TTabSheet;

dlgFont1: TFontDialog;

ts3: TTabSheet;

lbl7: TLabel;

edt1: TEdit;

dlgColor1: TColorDialog;

btn3: TButton;

procedure scrlbr1Change(Sender: TObject);

procedure btn1Click(Sender: TObject);

procedure scrlbr2Change(Sender: TObject);

procedure scrlbr3Change(Sender: TObject);

procedure FormCreate(Sender: TObject);

procedure btn3Click(Sender: TObject);

procedure edt1Exit(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

Form2: TForm2;

implementation

uses lab8;

{$R \*.dfm}

procedure TForm2.scrlbr1Change(Sender: TObject);

begin

lbl4.Caption := IntToStr(scrlbr1.Position);

case rg1.ItemIndex of

0: Form1.color := RGB(scrlbr1.Position,scrlbr2.Position,scrlbr3.Position) ;

1: Form1.cht1.Color := RGB(scrlbr1.Position,scrlbr2.Position,scrlbr3.Position);

end;

lbl1.Font.Color := RGB(scrlbr1.Position,0,0) ;

end;

procedure TForm2.btn1Click(Sender: TObject);

begin

case rg1.ItemIndex of

0: Form1.color := RGB(scrlbr1.Position,scrlbr2.Position,scrlbr3.Position) ;

1: Form1.cht1.Color := RGB(scrlbr1.Position,scrlbr2.Position,scrlbr3.Position);

end;

end;

procedure TForm2.scrlbr2Change(Sender: TObject);

begin

lbl5.Caption := IntToStr(scrlbr2.Position);

case rg1.ItemIndex of

0: Form1.color := RGB(scrlbr1.Position,scrlbr2.Position,scrlbr3.Position) ;

1: Form1.cht1.Color := RGB(scrlbr1.Position,scrlbr2.Position,scrlbr3.Position);

end;

lbl2.Font.Color := RGB(0,scrlbr2.Position,0) ;

end;

procedure TForm2.scrlbr3Change(Sender: TObject);

begin

lbl6.Caption := IntToStr(scrlbr3.Position);

case Form2.rg1.ItemIndex of

0: Form1.color := RGB(scrlbr1.Position,scrlbr2.Position,scrlbr3.Position) ;

1: Form1.cht1.Color := RGB(scrlbr1.Position,scrlbr2.Position,scrlbr3.Position);

end;

lbl3.Font.Color := RGB(0,0,scrlbr3.Position) ;

end;

procedure TForm2.FormCreate(Sender: TObject);

begin

edt1.Text := '100';

lbl1.Font.Color := RGB(scrlbr1.Position,0,0) ;

lbl2.Font.Color := RGB(0,scrlbr2.Position,0);

lbl3.Font.Color := RGB(0,0,scrlbr3.Position);

end;

procedure TForm2.btn3Click(Sender: TObject);

begin

if dlgColor1.Execute then

begin

colorofanimation := dlgColor1.Color;

end;

end;

procedure TForm2.edt1Exit(Sender: TObject);

begin

Form1.tmr1.Interval := StrtoInt(edt1.Text);

end;

end.

**Вывод:** показал умение создания современного приложения на Delphi, используя все пройденные инструменты и материалы.