**Лабораторная работа 1**

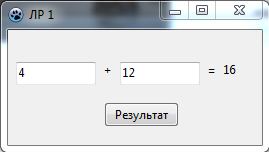
**Введение. Знакомство с интегрированной средой Lazarus**

**Задача:**

Создать простейший пользовательский интерфейс, для программы сложения двух чисел (a + b = c), содержащий объекты Button, TextBox - для ввода значений переменных a и b и вывода значения с, Label - для поясняющих надписей.

Написать процедуру обработки события - щелчок по кнопке - сложение двух чисел. Для этого необходимо создать процедуру обработки события (двойным щелчком по объекту кнопка), объявить переменные, ввести значение переменных a и b, вычислить значение переменной с и вывести значение с на экран.

**Скриншот:**



**Код:**

unit Unit1;

{$mode objfpc}{$H+}

interface

uses

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

type

{ TForm1 }

TForm1 = class(TForm)

Button1: TButton;

Edit1: TEdit;

Edit2: TEdit;

Label1: TLabel;

Label2: TLabel;

Label3: TLabel;

procedure Button1Click(Sender: TObject);

private

public

end;

var

Form1: TForm1;

implementation

{$R \*.lfm}

{ TForm1 }

procedure TForm1.Button1Click(Sender: TObject);

begin

Label1.Caption:=FloatToStr(StrToFloat(Edit1.Text)+StrToFloat(Edit2.Text));

end;

end.

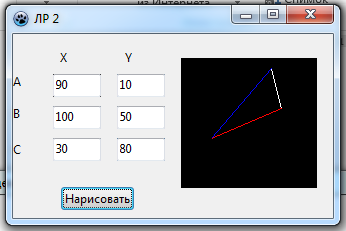
**Лабораторная работа 2**

**Работа с графикой**

**Задача:**

Нарисовать в объекте TImage треугольник, одна сторона которого черная, другая красная и третья - синяя. Создать программу рисующую треугольник, координаты вершин которого можно изменять через пользовательский интерфейс.

**Скриншот:**



**Код процедуры:**

procedure TForm1.Button1Click(Sender: TObject);

var x1, y1, x2, y2, x3, y3: integer;

begin

x1:=StrToInt(Edit1.Text);

y1:=StrToInt(Edit2.Text);

x2:=StrToInt(Edit3.Text);

y2:=StrToInt(Edit4.Text);

x3:=StrToInt(Edit5.Text);

y3:=StrToInt(Edit6.Text);

Image1.Canvas.pen.Color := clWhite;

Image1.Canvas.Line(x1,y1,x2,y2);

Image1.Canvas.pen.Color := clRed;

Image1.Canvas.Line(x2,y2,x3,y3);

Image1.Canvas.pen.Color := clBlue;

Image1.Canvas.Line(x1,y1,x3,y3);

end;

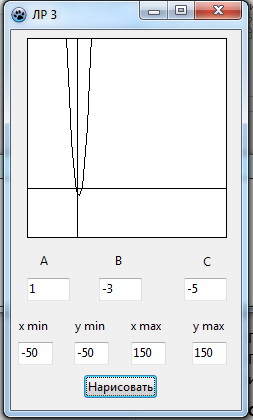
**Лабораторная работа 3**

**Построение графика заданной функции в указанном диапазоне**

**Задача:**

Построить график функции y = Ax^2 + Bx + C в диапазоне ( x min , y min ) – ( x max , y max ). Постоянные А, В, С, x min , y min , x max , y max вводятся через пользовательский интерфейс.

**Скриншот:**



**Код процедуры:**

procedure TForm1.Button1Click(Sender: TObject);

var

xmin, ymin, xmax, ymax, hg, w, n, xsc, ysc : integer;

a, b, c, h, x, y, kx, ky : real;

begin

hg:=image1.Height;

w:=image1.Width;

xmin:=StrtoInt(Edit4.Text);

ymin:=StrtoInt(Edit5.Text);

xmax:=StrtoInt(Edit6.Text);

ymax:=StrtoInt(Edit7.Text);

a:=StrtoInt(Edit1.Text);

b:=StrtoInt(Edit2.Text);

c:=StrtoInt(Edit3.Text);

n:=77;

h:=w/n;

kx:=w/(xmax-xmin);

ky:=hg/(ymax-ymin);

image1.Canvas.Rectangle(0,0,w,hg);

image1.Canvas.MoveTo(0,round(ymax\*ky));

image1.Canvas.LineTo(w,round(ymax\*ky));

image1.Canvas.MoveTo(round(-xmin\*kx),0);

image1.Canvas.LineTo(round(-xmin\*kx),hg);

x:=xmin;

image1.Canvas.MoveTo(xmin, round(a\*x\*x+b\*x+c));

while x<=xmax do

begin

y:=a\*x\*x+b\*x+c;

xsc:=round((x-xmin)\*kx);

ysc:=round(hg-(y-ymin)\*ky);

image1.Canvas.LineTo(xsc,ysc);

x:=x+h;

end;

end;

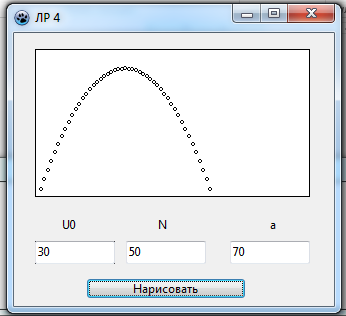
**Лабораторная работа 4**

**Движение тела брошенного под углом к горизонту**

**Задача:**

Изобразить местонахождение тела, брошенного с земли с начальной скорость U0 под углом к горизонту α, через равные интервалы времени ( N – число равных временных интервалов)

**Скриншот:**



**Код процедуры:**

procedure TForm1.Button1Click(Sender: TObject);

const

g=9.8;

var

u0, a, kx, ky,tmax, t, x, y, dt : real;

xsc, ysc, n : integer;

begin

u0:=StrtoInt(Edit1.Text);

n:=StrtoInt(Edit2.Text);

a:=StrtoInt(Edit3.Text);

Image1.Canvas.Rectangle(0,0,Image1.Width,Image1.Height);

Image1.Canvas.pen.Color := clblack;

a:=a\*pi/180;

kx:=Image1.Width/((u0\*u0)/g);

ky:=Image1.Height/((u0\*u0)/(2\*g));

tmax:=(2\*u0\*sin(a))/g;

t:=0;

dt:=tmax/n;

while t<=tmax do

begin

x:=u0\*cos(a)\*t;

y:=u0\*sin(a)\*t-g\*t\*t/2;

xsc:=round(x\*kx);

ysc:=round(Image1.Height-y\*ky);

Image1.Canvas.Ellipse(xsc,ysc,xsc+4,ysc+4);

t:=t+dt;

end;

end;

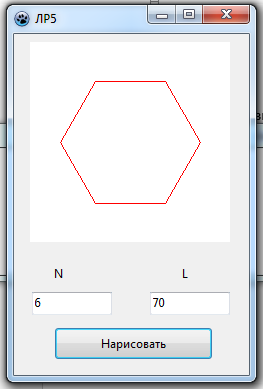
**Лабораторная работа 5**

**Построение правильного n-угольника.**

**Задача:**

Построить правильный n -угольник, количество углов которого задано и вводиться через пользовательский интерфейс.

**Скриншот:**



**Код процедуры:**

procedure TForm1.Button1Click(Sender: TObject);

var

l, n, xsc, ysc : integer;

x, y, a, b : real;

begin

n := StrtoInt(Edit1.Text);

l := StrtoInt(Edit2.Text);

Image1.Canvas.Pen.Color:=clWhite;

Image1.Canvas.Rectangle(0,0,Image1.Width,Image1.Height);

Image1.Canvas.Pen.Color:=clRed;

b:=0;

a:=((360/n)\*pi)/180;

x:=Round(Image1.Width/2);

y:=Round(Image1.Height/2);

xsc:=Round(x+l\*cos(b));

ysc:=Round(y+l\*sin(b));

Image1.Canvas.MoveTo(xsc,ysc);

while b<360 do

begin

xsc:=Round(x+l\*cos(b));

ysc:=Round(y+l\*sin(b));

Image1.Canvas.LineTo(xsc,ysc);

b:=b+a;

end;

end;

**Лабораторная работа 6**

**Преобразование объекта на плоскости.**

**Задача:**

Построить произвольный объект в первой четверти системы координат (например, флажок. кол-во вершин объекта должно быть на меньше 7) и организовать следующие преобразования объекта:

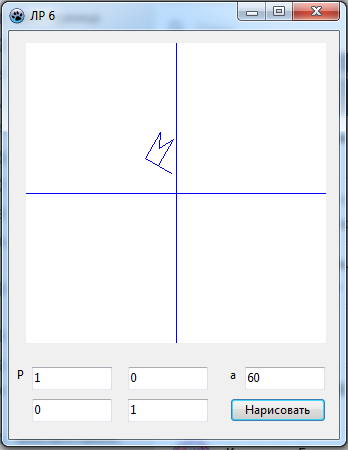
- общее преобразование (изменение масштаба, симметричное отражение относительно осей, сдвиг),

- вращение объекта относительно начала координат

- комбинированное преобразование (преобразование общего вида + вращение на произвольный угол относительно начала координат)

Матрица общего преобразования и угол поворота должны вводиться через пользовательский интерфейс, координаты вершин объекта можно указать явно в коде программы

**Скриншот:**



**Код процедуры:**

procedure TForm1.Button1Click(Sender: TObject);

const

n=6;

l=2;

M: array[1..n,1..l]of real=((10,10),(10,30),(30,30),(20,25),(30,20),(10,20));

var

p1, p2, p3 : array[1..l,1..l] of real;

p : array[1..n,1..l] of real;

i, j, t, xsc, ysc:integer;

kx, ky, xmax, ymin, ymax, xmin, a:real;

begin

xmax:=100;

ymin:=-100;

ymax:=100;

xmin:=-100;

Image1.Canvas.Rectangle(0, 0, Image1.Width, Image1.Height);

Image1.Canvas.pen.Color := clblue;

kx:=Image1.Width/(xmax-xmin);

ky:=Image1.Height/(ymax-ymin);

xsc:=Round((0-Xmin)\*kx);

ysc:=Round(Image1.Height-(0-ymin)\*ky);

Image1.Canvas.MoveTo(xsc,0);

Image1.Canvas.LineTo(xsc,Image1.Height);

Image1.Canvas.MoveTo(0,ysc);

Image1.Canvas.LineTo(Image1.Width,ysc);

Image1.Canvas.Pen.Color:=clblue;

Image1.canvas.moveto(Round((M[1,1]-Xmin)\*kx),Round(Image1.Height-(M[1,2]-ymin)\*ky));

for i:=2 to n do

Image1.canvas.lineto(Round((M[i,1]-Xmin)\*kx),Round(Image1.Height-(M[i,2]-ymin)\*ky));

a:=(StrtoInt(Edit1.Text))\*(pi/180);

p1[1,1]:=cos(a);

p1[1,2]:=sin(a);

p1[2,1]:=-sin(a);

p1[2,2]:=cos(a);

p2[1,1]:=StrtoInt(Edit2.Text);

p2[1,2]:=StrtoInt(Edit3.Text);

p2[2,1]:=StrtoInt(Edit4.Text);

p2[2,2]:=StrtoInt(Edit5.Text);

for i:=1 to l do

for j:=1 to l do

begin

p3[i,j]:=0;

for t:=1 to l do

p3[i,j]:=p3[i,j]+p1[i,t]\*p2[t,j];

end;

for i:=1 to n do

for j:=1 to l do

begin

p[i,j]:=0;

for t:=1 to l do

p[i,j]:=p[i,j]+M[i,t]\*p3[t,j];

end;

Image1.Canvas.pen.Color := clwhite;

Image1.Canvas.Rectangle(0, 0, Image1.Width, Image1.Height);

Image1.Canvas.pen.Color := clblue;

Image1.Canvas.MoveTo(xsc,0);

Image1.Canvas.LineTo(xsc,Image1.Height);

Image1.Canvas.MoveTo(0,ysc);

Image1.Canvas.LineTo(Image1.Width,ysc);

Image1.canvas.MoveTo(Round((p[1,1]-Xmin)\*kx),Round(Image1.Height-(p[1,2]-ymin)\*ky));

for i:=2 to n do

Image1.canvas.LineTo(Round((p[i,1]-Xmin)\*kx),Round(Image1.Height-(p[i,2]-ymin)\*ky));

end;