

## Самостоятельна работа № 1

### Знакомство с интегрированной средой Lazarus

#### Задание 1.

1. **Задание:** нарисовать в объекте TImage треугольник, одна сторона которого - черная, другая - красная и третья - синяя.

2. **Математическая модель:** перемещение пера в центр поля, затем рисование треугольника путем смещения координат.

#### 3. Список идентификаторов в программе

Имя переменной	Описание переменной	Тип данных
x	Координата середины поля Image по оси OX	integer
y	Координата середины поля Image по оси Oy	integer

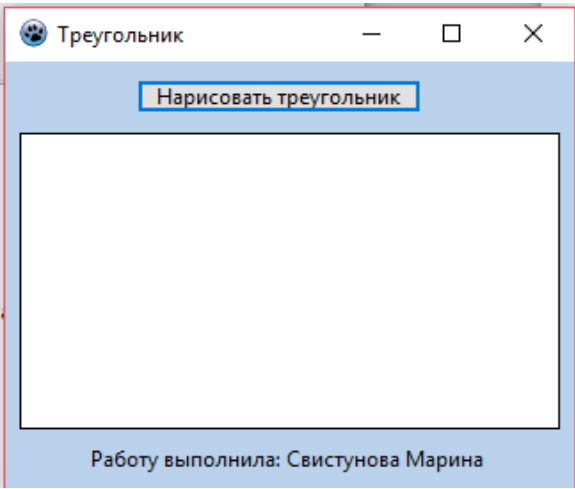
#### 4. Код программы (только процедуру обработки основного события - щелчок по кнопке):

```
procedure TForm1.Button1Click(Sender: TObject);
var
  x, y: integer;
begin
  x := Round(Image1.Width/2);
  y := Round(Image1.Height/2);
  Form1.Canvas.MoveTo(x, y);
  Form1.Canvas.Pen.Color := clRed;
  Form1.Canvas.LineTo(Canvas.PenPos.x + 70, Canvas.PenPos.y + 50);
  Form1.Canvas.Pen.Color := clBlue;
  Form1.Canvas.LineTo(Canvas.PenPos.x - 120, Canvas.PenPos.y + 10);
  Form1.Canvas.Pen.Color := clBlack;
  Form1.Canvas.LineTo(Canvas.PenPos.x + 50, Canvas.PenPos.y -60);
end;
```

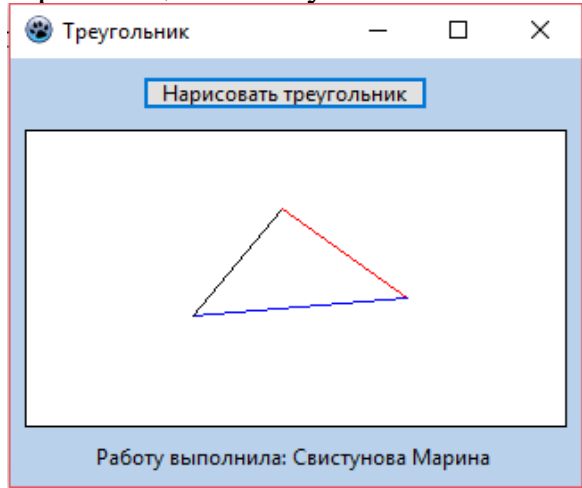
```
procedure TForm1.FormCreate(Sender: TObject);
begin
  Image1.Canvas.Pen.Color := clBlack; //окантовка
  Image1.Canvas.Brush.Color := clWhite; //заливка
  Image1.Canvas.Rectangle(0,0,Image1.Width,Image1.Height);
end;
```

#### 5. Протокол работы программы (отчет и скриншот пользовательского интерфейса с демонстрацией работы программы)

При открытии формы TImage заполняется белым цветом, окантовка черного цвета. Сама форма отлична по цвету от стандартной. Внизу указана мое имя.



При нажатии кнопки «Нарисовать треугольник» рисуется треугольник с разноцветными сторонами. Для рисования треугольника каждый раз позиция пера смещается на указанное количество пикселей и рисуется линия.



**Задание 2.**

- 1. **Задание:** создать программу, рисующую треугольник, координаты вершин которого можно изменять через пользовательский интерфейс.
- 2. **Математическая модель:** треугольник рисуется по координатам линиями от точки А до точки В, от В до С, от С до А.
- 3. **Список идентификаторов в программе**

Имя переменной	Описание переменной	Тип данных
ax, ay	Координаты вершины А, вводятся с клавиатуры	integer

bx, by	Координаты вершины В, вводятся с клавиатуры	integer
cx, cy	Координаты вершины С, вводятся с клавиатуры	integer

#### 4. Код программы (только процедуру обработки основного события - щелчок по кнопке):

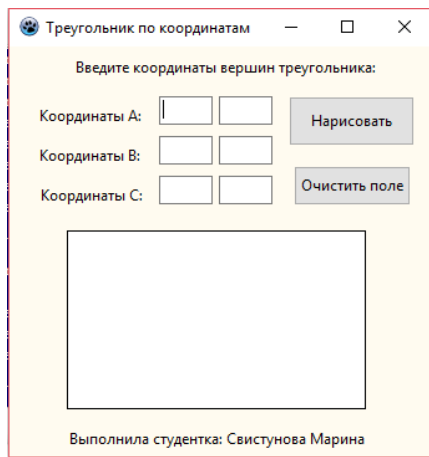
```
procedure TForm1.FormCreate(Sender: TObject);
begin
    Image1.Canvas.Pen.Color := clBlack; //окантовка
    Image1.Canvas.Brush.Color := clWhite; //заливка
    Image1.Canvas.Rectangle(0,0,Image1.Width,Image1.Height);
end;
```

```
procedure TForm1.Button1Click(Sender: TObject);
var
    ax, ay, bx, by, cx, cy: integer;
begin
    ax := StrToInt(Edit1.Text); ay := StrToInt(Edit4.Text);
    bx := StrToInt(Edit2.Text); by := StrToInt(Edit5.Text);
    cx := StrToInt(Edit3.Text); cy := StrToInt(Edit6.Text);
    Image1.Canvas.MoveTo(ax, ay);
    Image1.Canvas.Pen.Color := clBlack;
    Image1.Canvas.LineTo(bx, by);
    Image1.Canvas.LineTo(cx, cy);
    Image1.Canvas.LineTo(ax, ay);
end;
```

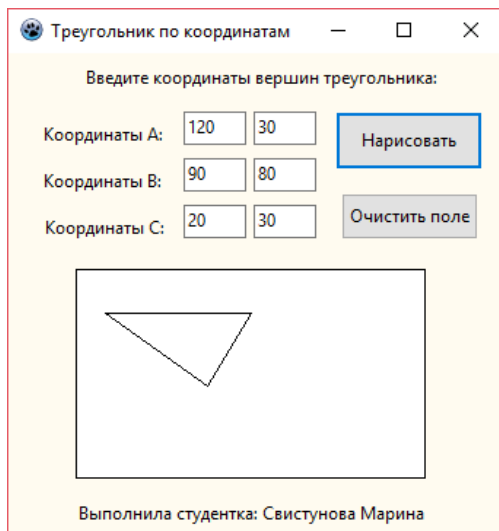
```
procedure TForm1.Button2Click(Sender: TObject);
begin
    Image1.Canvas.Pen.Color := clBlack; //окантовка
    Image1.Canvas.Brush.Color := clWhite; //заливка
    Image1.Canvas.Rectangle(0,0,Image1.Width,Image1.Height);
    Edit1.Clear; Edit2.Clear; Edit3.Clear;
    Edit4.Clear; Edit5.Clear; Edit6.Clear;
end;
```

#### 5. Протокол работы программы (отчет и скриншот пользовательского интерфейса с демонстрацией работы программы)

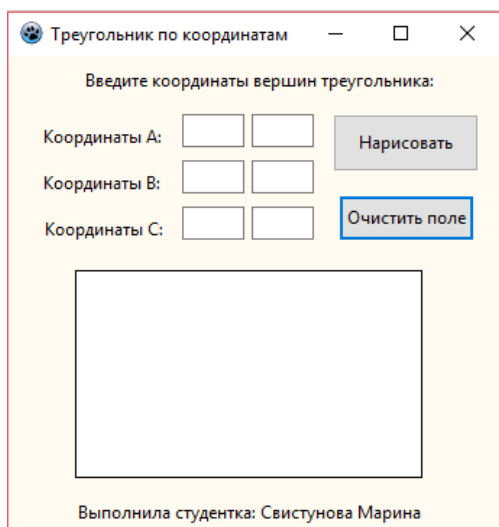
При открытии формы TImage заполняется белым цветом, окантовка черного цвета. Сама форма отлична по цвету от стандартной. Внизу указана мое имя.



При введении всех координат вершин треугольника и нажатии на кнопку «Нарисовать» рисуется треугольник.



При нажатии на кнопку «Очистить поле» очищаются как поля ввода вершин треугольника, так и само поле TImage.



### Задание 3.

1. **Задание:** создать программу, рисующую цилиндр и куб с прозрачными гранями.
2. **Математическая модель:** рисование цилиндра и куба по точкам со смещением.
3. **Список идентификаторов в программе**

Имя переменной	Описание переменной	Тип данных
x1	Координата оси ОХ верхнего левого угла прямоугольника, в который вписывается верхний эллипс для рисования цилиндра	integer
y1	Координата оси ОУ верхнего левого угла прямоугольника, в который вписывается верхний эллипс для рисования цилиндра	integer
dx	Длина верхней грани прямоугольника, в который вписан эллипс	integer
dy	Длина боковой грани прямоугольника, в который вписан эллипс	integer
s	Смещение нижнего прямоугольника с эллипсов по отношению к верхнему	integer
Cx1	Координата оси ОХ верхнего левого угла переднего квадрата, являющегося ближайшей гранью куба по отношению к смотрящему	integer
Cy1	Координата оси ОУ верхнего левого угла переднего квадрата, являющегося ближайшей гранью куба по отношению к смотрящему	integer
Cd	Длина грани куба	integer
Cs	Длина грани куба, которую видно, как диагональ	integer

### 4. Код программы (только процедуру обработки основного события - щелчок по кнопке):

```
procedure TForm1.FormCreate(Sender: TObject);
begin
    Image1.Canvas.Pen.Color := clBlack; //окантовка
    Image1.Canvas.Brush.Color := clWhite; //заливка
    Image1.Canvas.Rectangle(0,0,Image1.Width,Image1.Height);
end;

procedure TForm1.Button1Click(Sender: TObject);
var
    x1, y1, dx, dy, s: integer;
begin
    x1 := 30;
```

```

y1 := 30;
dx := 70;
dy := 20;
s := 50;
Image1.Canvas.MoveTo(x1, y1);
Image1.Canvas.Brush.Style := bsClear;
Image1.Canvas.Pen.Color := clBlack;
Image1.Canvas.Ellipse(x1, y1, x1 + dx, y1 + dy);
Image1.Canvas.Ellipse(x1, y1 + s, x1 + dx, y1 + dy + s);
Image1.Canvas.MoveTo(x1, y1 + Round(dy/2));
Image1.Canvas.LineTo(x1, y1 + s + Round(dy/2));
Image1.Canvas.MoveTo(x1 + dx, y1 + Round(dy/2));
Image1.Canvas.LineTo(x1 + dx, y1 + s + Round(dy/2));
end;

procedure TForm1.Button2Click(Sender: TObject);
var
  Cx1, Cy1, Cd, Cs: integer;
begin
  Cx1 := 150;
  Cy1 := 50;
  Cd := 60;
  Cs := Round(Cd/2);
  Image1.Canvas.MoveTo(Cx1, Cy1);
  Image1.Canvas.Brush.Style := bsClear;
  Image1.Canvas.Pen.Color := clBlack;
  Image1.Canvas.Rectangle(Cx1, Cy1, Cx1 + Cd, Cy1 + Cd);
  Image1.Canvas.Rectangle(Cx1 + Cs, Cy1 - Cs, Cx1 + Cd + Cs, Cy1 + Cd - Cs);
  Image1.Canvas.MoveTo(Cx1, Cy1);
  Image1.Canvas.LineTo(Cx1 + Cs, Cy1 - Cs);
  Image1.Canvas.MoveTo(Cx1 + Cd, Cy1);
  Image1.Canvas.LineTo(Cx1 + Cs + Cd, Cy1 - Cs);
  Image1.Canvas.MoveTo(Cx1 + 2, Cy1 + Cd - 2);
  Image1.Canvas.LineTo(Cx1 + Cs + 2, Cy1 - Cs + Cd - 2);
  Image1.Canvas.MoveTo(Cx1 + Cd, Cy1 + Cd - 2);
  Image1.Canvas.LineTo(Cx1 + Cd + Cs, Cy1 - Cs + Cd - 2);
end;

```

## 5. Протокол работы программы (отчет и скриншот пользовательского интерфейса с демонстрацией работы программы)

При открытии формы TImage заполняется белым цветом, окантовка черного цвета. Сама форма отлична по цвету от стандартной. Внизу указана мое имя.

При нажатии на кнопку «Цилиндр» рисуется цилиндр. При нажатии на кнопку «Куб» рисуется куб. В программе явно описаны координаты одного из углов каждого рисунка, при их смещении рисуются цилиндр и куб

