

## Лабораторная работа № 2

### Построение правильного n-угольника

#### Задание 1.

1. **Задание:** построить правильный n-угольник, количество вершин которого вводится через пользовательский интерфейс. Также через пользовательский интерфейс вводится длина одной стороны многоугольника.

#### 2. Математическая модель:

$$x_i = x_0 + R \cos(\varphi_0 + \frac{2\pi i}{n})$$

$$y_i = y_0 + R \sin(\varphi_0 + \frac{2\pi i}{n})$$

где  $x_0$  и  $y_0$  - координаты центра,  $R$  - радиус описанной вокруг правильного многоугольника окружности,  $\varphi_0$  - угловая координата первой вершины

#### 3. Список идентификаторов в программе

Имя переменной	Описание переменной	Тип данных
fi	Угловая координата первой вершины	real
x	Координата x точки стороны многоугольника для рисования линии	real
x0	Координаты центра ширины Image1	real
y	Координата y точки стороны многоугольника для рисования линии	real
y0	Координаты центра высоты Image1	real
d	Длина одной стороны многоугольника, вводится с клавиатуры	real
R	Радиус описанной вокруг правильного многоугольника окружности	real
i	Переменная цикла для рисования сторон многоугольника	integer
n	Количество углов многоугольника, вводится 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);
Button1.Enabled := False;
Button2.Enabled := False;
end;

```

```

procedure TForm1.Button1Click(Sender: TObject);
var

```

```

    fi, x, x0, y, y0, d, R: real;
    i, n: integer;

```

```

begin

```

```

    x0 := Image1.Width div 2;
    y0 := Image1.Height div 2;
    n := Round(StrToInt(Edit1.Text));
    d := StrToInt(Edit2.Text);
    R := d / (2*Sin(180/n));

```

```

    fi := 0;

```

```

    i := 0;

```

```

    x := x0 + R*Cos(fi + (2*PI*i/n));

```

```

    y := y0 + R*Sin(fi + (2*PI*i/n));

```

```

    Image1.Canvas.Pen.Color := clPurple;

```

```

    Image1.Canvas.Pen.width := 3;

```

```

    Image1.Canvas.MoveTo(Round(x), Round(y));

```

```

    for i := 1 to n do

```

```

    begin

```

```

        x := x0 + R*Cos(fi + (2*PI*i/n));

```

```

        y := y0 + R*Sin(fi + (2*PI*i/n));

```

```

        Image1.Canvas.LineTo(Round(x), Round(y));

```

```

    end;

```

```

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);

```

```

end;

```

```

procedure TForm1.Edit1Change(Sender: TObject);

```

```

begin

```

```

    if ((Edit1.Text = "") or

```

```

        (Edit2.Text = "")) then Button1.Enabled := False

```

```

    else Button1.Enabled := True;

```

```

    if ((Edit1.Text = "") or

```

```

        (Edit2.Text = "")) then Button2.Enabled := False

```

```

    else Button2.Enabled := True;

```

```

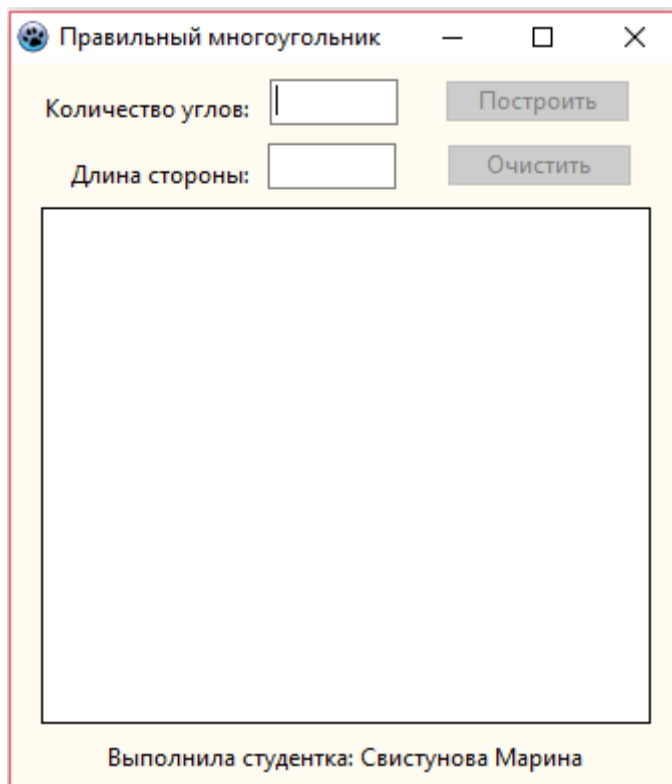
end;

```

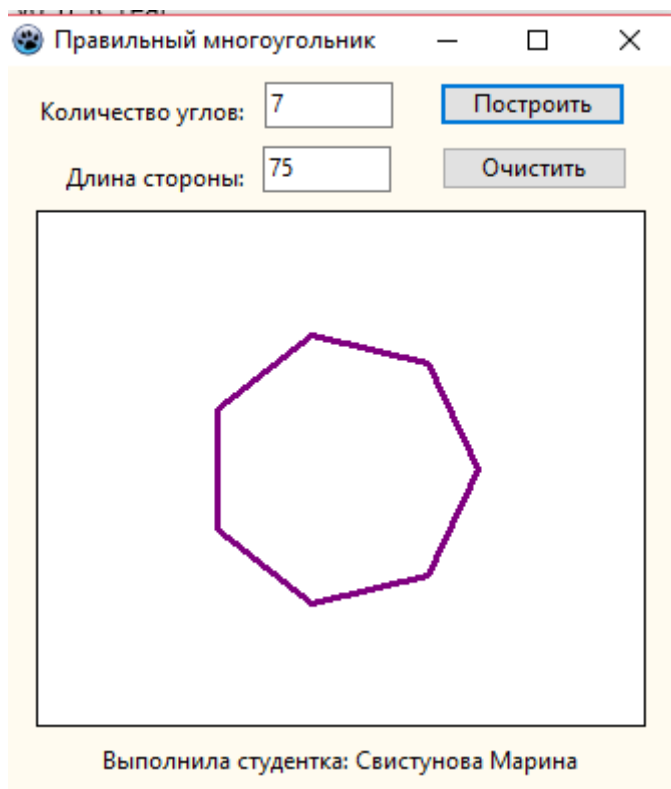
```
procedure TForm1.Edit1KeyPress(Sender: TObject; var Key: char);
begin
  case Key of
    '0'..'9',#8: ;
  else Key:=chr(0);
  end;
end;
```

##### 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. Название формы «Правильный многоугольник».



Кнопки «Построить» и «Очистить» заблокированы до тех пор, пока пользователь не введет значения в поля «Количество углов» и «Длина сторон». В данные поля можно вводить только цифры.



Кнопка «Очистить» очищает поле.