

## Region

### Цели:

- Получить навыки рисования на форме
- Закрепить навыки создания обработчиков событий.

## Команды

Команда для:

Создание квадрата: `F2=CreateRectRgn(x1,y1,x2,y2);`

Создание эллипса: `R3=CreateEllipticRgn(x1,y1,x2,y2);`

Указание точки: `P{0}=Point(x, y);`

Создание полигона по точкам: `CreatePolygonRgn(P, 3,WINDING);`

Объединение: `CombineRgn(R1,R2,R3,RGN_ "x");`

RGN\_ "x"-название команды: «RGN\_OR» - Объединение,  
«RGN\_DIFF» - вычитание из R2 R3, «RGN\_AND» -  
нахождение общих областей R2 и R1;

Назначение полученный регион форме:

`SetWindowRgn(Handle,R1,True).`

## Создание формы

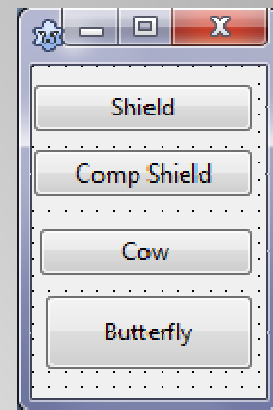
Создадим две формы, одну исключим из автосоздания. На первой форме разместить четыре кнопки. Каждой кнопки задать процедуру Click.

Для Shield:

```
procedure TForm1.ShieldClick(Sender:  
TObject);  
begin  
    Form2:=TShield.Create;  
    Form2.Show;  
end;
```

Для CompShield:

```
procedure TForm1.CompShieldClick(Sender: TObject);  
begin  
    Form2:=TCompShield.Create;  
    Form2.Show;  
end;
```



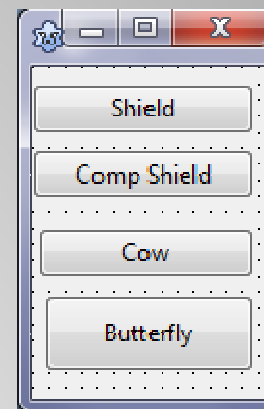
## Создание формы

Для Cow:

```
procedure TForm1.CowBTNClick(Sender:TObject);  
begin  
    Fomr2:=TCow.Create;  
    Fomr2.Show;  
end;
```

Для ButterFly:

```
procedure TForm1.Button1Click(Sender: TObject);  
begin  
    Fomr2 :=TButterfly.Create;  
    Fomr2.Show;  
end;
```



## Создание формы

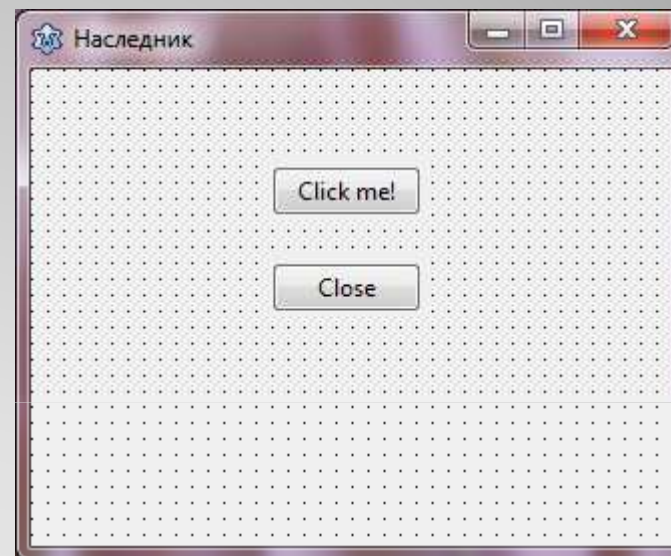
На второй форме разместим две кнопки.

Для Click me!:

```
procedure TForm2.ClickmeClick  
(Sender: TObject);  
begin  
  ShowMessage(IntToStr(Width));  
end;
```

Для Close:

```
procedure TForm2.CloseClick(Sender: TObject);  
begin  
  Close;  
end;
```



## Создание формы

Создадим классы объектов.

Для Shield:

```
TShield=class (TForm2)
  private
  public
    constructor Create;
end;
```

Для CompShield:

```
TCompShield=class (TForm2)
  private
  public
    constructor Create;
end;
```

## Создание формы

Для Cow:

```
TCow=class (TForm2)
  private
  public
    constructor Create;
end;
```

Для Butterfly:

```
TButterfly=class (TForm2)
  private
  public
    constructor Create;
end;
```

## Создание формы

Сделаем что бы кнопки находились на середине формы по горизонтали при её создании.

```
Clickme.Left:=(ClientWidth-Clickme.Width) div 2;
```

```
Close.Left:=(ClientWidth-Clickme.Width) div 2;
```



## Создание Regions

А теперь напишем код для обрезки форм.

Форма виде щита:

```
constructor TShield.Create;  
Var R1,R2: HRgn;  
Begin  
  inherited Create(Owner);  
  Width:=353; Height:=350;           //Размер формы  
  Caption:='It's a SHIELD!!!';       //Название формы  
  Brush.Color:=clGreen;              //Цвет формы  
  R1:=CreateEllipticRgn(-150,-300,360,400); //Правый вырез  
  R2:=CreateEllipticRgn(360,-300,5,400);    //Левый вырез  
  CombineRgn(R1,R1,R2,RGN_AND); //Комбинирование двух  
  вырезов
```

## Создание Regions

```
//Верхний левый вырез  
R2:=CreateEllipticRgn(-40,-30,180,20);  
//Комбинирование с предыдущими вырезами  
CombineRgn(R1,R1,R2,RGN_DIFF);  
//Верхний правый вырез  
R2:=CreateEllipticRgn(180,-30,400,20);  
//Комбинирование с предыдущими вырезами  
CombineRgn(R1,R1,R2,RGN_DIFF);  
//Назначаем полученный регион форме  
SetWindowRgn(Handle,R1,True);  
end;
```

## Создание Regions

Форма виде комп.щита:

```
constructor TCompShield.Create;  
var  
  R1,R2 : HRgn;  
  P : array [0..2] of TPoint;  
begin  
  inherited Create(Owner);  
  Width:=263; Height:=350;  
  Caption:='It's a COMPLICATED SHIELD!!!!';  
  Brush.Color:=clBlue;  
  R1:=CreateRectRgn(0,0,270,233); //Вырез прям-ник  
  R2:=CreateEllipticRgn(-20,-20,135+5,20); //Верхний левый  
  CombineRgn(R1,R1,R2,RGN_DIFF); //Комбинирование
```

## Создание Regions

//Верхний правый

```
R2:=CreateEllipticRgn(135-5,-20,270+20,20);
```

```
CombineRgn(R1,R1,R2,RGN_DIFF);
```

//Левый вырез

```
R2:=CreateEllipticRgn(-15,10,8,233);
```

```
CombineRgn(R1,R1,R2,RGN_DIFF);
```

//Правый вырез

```
R2:=CreateEllipticRgn(270-7,10,270+16,233);
```

```
CombineRgn(R1,R1,R2,RGN_DIFF);
```

//Нижний вырез

```
R2:=CreateEllipticRgn(0,150,270,320);
```

```
CombineRgn(R1,R1,R2,RGN_OR);
```

## Создание Regions

```
//Треугольник внизу щита  
P[0]:=Point(135-40, 310);  
P[1]:=Point(135+40, 310);  
P[2]:=Point(135, 335);  
R2:=CreatePolygonRgn(P,3,WINDING);  
CombineRgn(R1,R1,R2,RGN_OR);  
// Назначаем полученный регион форме  
SetWindowRgn(Handle, R1, True);  
end;
```

## Создание Regions

Форма виде коровы:

```
constructor TCow.Create;  
Var R1,R2 : HRgn;  
P : array [0..2] of TPoint;  
begin  
  inherited Create(Owner);  
  Width:=280; Height:=350;  
  Caption:='I'am a COW!!!!';  
  Brush.Color:=clRed;
```

## Создание Regions

//Уши коровы

```
R1:=CreateEllipticRgn(135-110, 45, 135+40, 105);  
R2:=CreateEllipticRgn(135+40, 45, 135+110, 105);  
CombineRgn(R1,R1,R2,RGN_OR);  
P[0]:=Point(135-140, 40);  
P[1]:=Point(135+140, 40);  
P[2]:=Point(135, 140);  
R2:=CreatePolygonRgn(P, 3, WINDING);  
CombineRgn(R1,R1,R2,RGN_DIFF);
```

## Создание Regions

```
//Рога
R2:=CreateEllipticRgn(135-90, 0, 135+90, 100);
CombineRgn(R1,R1,R2,RGN_OR);
R2:=CreateEllipticRgn(135-70, -5, 135+70, 70);
CombineRgn(R1,R1,R2,RGN_DIFF);
//Морда
R2:=CreateEllipticRgn(135-70, 60, 135+70, 170);
CombineRgn(R1,R1,R2,RGN_OR);
//Нос
R2:=CreateEllipticRgn(135-40, 150, 135+40, 210);
CombineRgn(R1,R1,R2,RGN_OR);
//Назначаем полученный регион форме
SetWindowRgn(Handle, R1, True);
end;
```



## Создание Regions

Форма виде бабочки:

```
constructor TButterfly.Create;  
var  
  R1,R2:HRgn;  
  P:array [0..2] of TPoint;  
  X:Word;  
begin  
  inherited Create(Owner);  
  Width:=280; Height:=350;  
  Caption:='B U T T E R F L Y!!!!';  
  Brush.Color:=clYellow;
```

## Создание Regions

```
//Левое верхнее крыло  
R1:=CreateEllipticRgn(Round(-  
Width*0.4),0,Round(Width*0.49),Round(Height*1.1));  
//Правое верхнее крыло  
R2:=CreateEllipticRgn(Round(Width*0.51),0,Round(Width*1.4),  
Round(Height*1.1));  
CombineRgn(R2,R1,R2,RGN_OR);  
//Отсекаем лишнее от верхних крыльев,  
//Остаются линзы на пересечении эллипсов  
R1:=CreateEllipticRgn(0,Round(-Height*0.3),  
Width,Round(Height*0.71));  
CombineRgn(R1,R1,R2,RGN_AND);
```

## Создание Regions

//Эллипс - основа нижних крыльев

```
R2:=CreateEllipticRgn(Round(Width*0.1),  
Round(Height*0.65), Round(Width*0.9), Height);  
CombineRgn(R1,R1,R2,RGN_OR);
```

//Вырезаем эллипс - разрез между нижних крыльев

```
R2:=CreateEllipticRgn(Round(Width*0.3),  
Round(Height*0.7), Round(Width*0.7), Round(Height*1.5));  
CombineRgn(R1,R1,R2,RGN_DIFF);
```

//Вертикальный эллипс - туловище бабочки

```
R2:=CreateEllipticRgn(Round(Width*0.46),  
Round(Height*0.3), Round(Width*0.54),  
Round(Height*0.8));  
CombineRgn(R1,R1,R2,RGN_OR);
```

## Создание Regions

```
//Голова - круг; за основу берем меньшую  
//Из двух величин - высоты и ширины окна  
X:=Width;  
if Height < X then X:=Height;  
X:=Round(X/18);  
R2:=CreateEllipticRgn(Round(Width*0.5)-X,  
Round(Height*0.3)-X, Round(Width*0.5)+X,  
Round(Height*0.3)+X);  
CombineRgn(R1,R1,R2,RGN_OR);
```

## Создание Regions

//Левый усик

```
P[0]:=Point(Round(Width*0.5),Round(Height*0.3));  
P[1]:=Point(Round(Width*0.35),Round(Height*0.01));  
P[2]:=Point(Round(Width*0.355)+1, 0);  
R2:=CreatePolygonRgn(P, 3, WINDING);  
CombineRgn(R1,R1,R2,RGN_OR);
```

//Правый усик

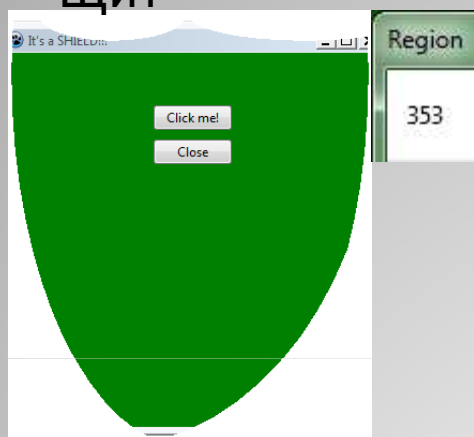
```
P[0]:=Point(Round(Width*0.5),Round(Height*0.3));  
P[1]:=Point(Round(Width*0.655+1),Round(Height*0.01));  
P[2]:=Point(Round(Width*0.65),0);  
R2:=CreatePolygonRgn(P, 3,WINDING);  
CombineRgn(R1,R1,R2,RGN_OR);
```

## Создание Regions

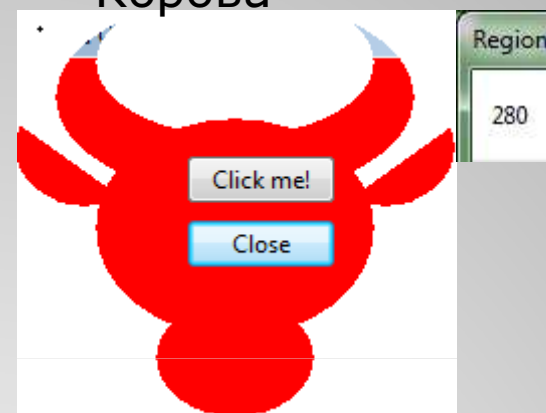
```
//Острие на крыле слева снизу
P[0]:=Point(Round(Width*0.15),Height);
P[1]:=Point(Round(Width*0.2),Round(Height*0.8));
P[2]:=Point(Round(Width*0.3),Round(Height*0.9));
R2:=CreatePolygonRgn(P, 3, WINDING);
CombineRgn(R1,R1,R2,RGN_OR);
//Острие на крыле справа снизу
P[0]:=Point(Round(Width*0.85), Height);
P[1]:=Point(Round(Width*0.8), Round(Height*0.8));
P[2]:=Point(Round(Width*0.7), Round(Height*0.9));
R2:=CreatePolygonRgn(P, 3, WINDING);
CombineRgn(R1,R1,R2,RGN_OR);
//Назначаем полученный регион форме
SetWindowRgn(Handle, R1, True);
end;
```

# Формы

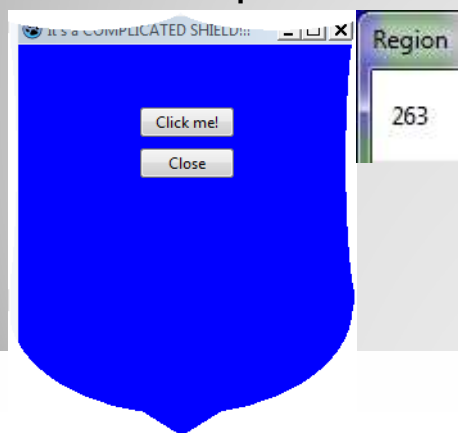
Щит



Корова



Комп. щит



Бабочка

