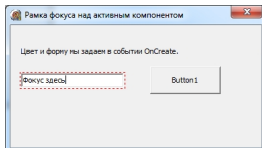


Для того, чтобы создать более дружелюбный интерфейс, можно рисовать рамку фокуса вокруг активного компонента. Для этой цели идеально подходит объект **TShape**, который представляет собой геометрическую фигуру, которая может быть помещена на форму. Каждый раз, когда компонент находится в фокусе, мы будем перемещать объект

TShape

в его местоположение.



Для начала нужно создать объект **TShape** в событии **OnCreate** формы и придать ему свойства:

```
uses { ... }, ExtCtrls; var Form1: TForm1; focusRect: TShape; { ... } procedure TForm1.FormCreate(Sender: TObject); begin focusRect := TShape.Create(self); focusRect.Shape := stRectangle; focusRect.Visible := false; focusRect.Brush.Style := bsClear; focusRect.Pen.Style := psDot; focusRect.Pen.Color := clRed; focusRect.Pen.Width := 1; Screen.OnActiveControlChange:= ActiveControlChange; end;
```

Примечание: переменная **focusRect** типа **TShape** объявляется в объявлении переменной формы. Теперь, когда фокус переходит от одного управления к другому, мы также передвигаем рамку фокуса.

Мы будем использовать процедуру **ActiveControlChange**, обрабатывая событие **OnActiveControlChange**

глобального объекта

Screen

```
private { Private declarations } procedure ActiveControlChange(Sender: TObject); { ... } procedure TForm1.ActiveControlChange(Sender: TObject); begin with focusRect do begin Parent := Screen.ActiveControl.Parent; Top := Screen.ActiveControl.Top - 2; Height := Screen.ActiveControl.Height + 4; Left := Screen.ActiveControl.Left - 2; Width := Screen.ActiveControl.Width + 4; Visible := true; end; end;
```

Размеры объекта **focusRect** подстраиваются под размеры активного управления, как мы и написали выше в коде.

Ну и наконец, нужно освободить память при закрытии приложения. Никогда не забывайте делать это!

```
procedure TForm1.FormDestroy(Sender: TObject); begin  
Screen.OnActiveControlChange:= nil; focusRect.Free; end;
```