

Для работы этого компонента будем использовать сообщения Windows **CM_MOUSEENTER** и **CM_MOUSELEAVE**.

- Создаем объект, унаследованный от **TControl**.
- Определяем переменную **TCanvas** для того, чтобы рисовать на ней и переменную

Boolean

, которая будет иметь значение

True

, когда мышь будет находиться над управлением.

- Определяем процедуры для обработки сообщений **CM_MOUSEENTER**, **CM_MOUSELEAVE** и **WM_PAINT**.

Управление - белый прямоугольник. Когда мышь появляется над ним, он становится желтым.

```
THighLightControl = class(TControl) private FMouseOver: Boolean; FCanvas:
TCanvas; protected procedure CMMouseLeave( var msg : TMessage); message
CM_MOUSELEAVE; procedure CMMouseEnter( var msg : TMessage); message
CM_MOUSEENTER; procedure WMPaint( var msg: TMessage); message
WM_PAINT; public constructor Create(Owner: TComponent); override; destructor
Destroy; override; end; procedure Register; implementation constructor
THighLightControl.Create(Owner: TComponent); begin inherited Create(Owner); Width :=
100; Height := 100; FMouseOver := False; FCanvas := TControlCanvas.Create;
TControlCanvas(FCanvas).Control := Self; { Объект TControlCanvas унаследован от
TCanvas и будет использоваться для рисования на Control Surface. } end; destructor
THighLightControl.Destroy; begin FCanvas.Destroy; inherited Destroy; end; procedure
THighLightControl.CMMouseLeave( var msg : TMessage); begin FMouseOver := False;
Invalidate; { Установить значение FMouseOver в False, и перерисовать управление }
end; procedure THighLightControl.CMMouseEnter( var msg : TMessage); begin
FMouseOver := True; Invalidate; { Установить значение FMouseOver в True, и
перерисовать управление } end; procedure THighLightControl.WMPaint(var msg:
TMessage); begin { Процедура для рисования прямоугольника } if FMouseOver then
begin { Мышь переместилась над управлением } FCanvas.Brush.Color := clYellow;
end else begin FCanvas.Brush.Color := clWhite; end; FCanvas.Rectangle(0, 0, Width,
Height); end; procedure Register; begin RegisterComponents('SAMPLES',
[THighLightControl]); end; end. {jcomments on}
```