

Создайте новое приложение Delphi. На главной форме поместите *Menu1* (**TMenu**), **TStatusBar**

и

TApplicationEvents

. Добавьте несколько пунктов в меню. Назначьте нескольким пунктам меню свойство *Hint*

.

Вот полный исходный код модуля *Формы* вместе с выполнением класса **TMenuItemHint**.

```
unit Unit1; interface uses Windows, Messages, SysUtils, Variants, Classes, Graphics,
  Controls, Forms, Dialogs, Menus, AppEvnts, StdCtrls, ExtCtrls, ComCtrls; type
TMenuItemHint = class(THintWindow) private activeMenuItem : TMenuItem;
showTimer : TTimer; hideTimer : TTimer; procedure HideTime(Sender : TObject) ;
procedure ShowTime(Sender : TObject) ; public constructor Create(AOwner :
TComponent) ; override; procedure DoActivateHint(menuitem : TMenuItem) ; destructor
Destroy; override; end; TForm1 = class(TForm) { ... } procedure
FormCreate(Sender: TObject) ; procedure ApplicationEvents1Hint(Sender: TObject) ;
private miHint : TMenuItemHint; procedure WMMenuSelect(var Msg: TWMMenuSelect)
; message WM_MENUSELECT; end; var Form1: TForm1; implementation {$R *.dfm}
procedure TForm1.FormCreate(Sender: TObject) ; begin miHint :=
TMenuItemHint.Create(self) ; end; { *FormCreate* } procedure
TForm1.ApplicationEvents1Hint(Sender: TObject) ; begin StatusBar1.SimpleText :=
'App.OnHint : ' + Application.Hint; end; { *Application.OnHint* } procedure
TForm1.WMMenuSelect(var Msg: TWMMenuSelect) ; var menuItem : TMenuItem;
hSubMenu : HMENU; begin inherited; // от TCustomForm (гарантирует, что
Application.Hint применен) menuItem := nil; if (Msg.MenuFlag $FFFF) or (Msg.IDItem
0) then begin if Msg.MenuFlag and MF_POPUP = MF_POPUP then begin
hSubMenu := GetSubMenu(Msg.Menu, Msg.IDItem) ; menuItem :=
Self.Menu.FindItem(hSubMenu, fkHandle) ; end else begin menuItem :=
Self.Menu.FindItem(Msg.IDItem, fkCommand) ; end; end;
miHint.DoActivateHint(menuitem) ; end; { *WMMenuSelect* } { TMenuItemHint } constructor
TMenuItemHint.Create(AOwner: TComponent) ; begin inherited; showTimer :=
TTimer.Create(self) ; showTimer.Interval := Application.HintPause; hideTimer :=
TTimer.Create(self) ; hideTimer.Interval := Application.HintHidePause; end; (*Create*)
destructor TMenuItemHint.Destroy; begin hideTimer.OnTimer := nil; showTimer.OnTimer
:= nil; self.ReleaseHandle; inherited; end; (*Destroy*) procedure
TMenuItemHint.DoActivateHint(menuitem: TMenuItem) ; begin // силой удаляет "старое"
окно подсказки hideTime(self) ; if (menuItem = nil) or (menuItem.Hint = "") then begin
activeMenuItem := nil; Exit; end; activeMenuItem := menuItem;
showTimer.OnTimer := ShowTime; hideTimer.OnTimer := HideTime; end; (*DoActivateHint*)
procedure TMenuItemHint.ShowTime(Sender: TObject) ; var r : TRect; width : integer;
```

```
hght : integer; begin  if activeMenuItem nil then  begin  // позиция и размеры  width
:= Canvas.TextWidth(activeMenuItem.Hint) ;  hght :=
Canvas.TextHeight(activeMenuItem.Hint) ;  r.Left := Mouse.CursorPos.X + 16;  r.Top :=
Mouse.CursorPos.Y + 16;  r.Right := r.Left + width + 6;  r.Bottom := r.Top + hght + 4;
ActivateHint(r,activeMenuItem.Hint) ;  end;  showTimer.OnTimer := nil; end; {*ShowTime*}
  procedure TMenuItemHint.HideTime(Sender: TObject) ; begin  // скрывает (разрушает)
окно подсказки  self.ReleaseHandle;  hideTimer.OnTimer := nil; end; {*HideTime*}  end.
```

Продолжение следует ...