

Postup tvorby programu (projektu)

V této kapitole se budeme zabývat tím základním, co je nezbytné pro nejjednodušší příklady. Brzy se přesvědčíme, že Delphi je vlastně "komponentová stavebnice", kde navíc můžeme měnit vlastnosti komponent nebo vytvářet vlastní komponenty.

Vytvoření projektu (File new application)

Po spuštění Delphi nebo otevření nového projektu vidíme prázdný formulář s názvem Form1. Nyní spustíme kompilaci (překlad textového do strojového kódu) a po chvíli se znovu objeví formulář, ale také již běžící aplikace. Aplikace s oknem Form1 je prázdná a nic nedělá. Uložme projekt pod implicitním názvem do nějakého adresáře, kde máme přístup a spustíme kompilaci (ikona, <F9> nebo menu Run/Run), tím se vytvoří exe soubor se stejným názvem jako uložený projektový soubor. Pro lepší názornost opusťte na chvíli Delphi a průzkumníkem nebo obdobným správcem programu spustíte program s názvem *Project1 .exe.* a ověřte jeho chování. Pokud chcete otevřít existující projekt File Open Project.

Nastavování vlastností komponent

Zjednodušeně můžeme říci, že objektové programování je založeno na popisu tzv. objektů a jejich vlastností. U většiny komponent je umožněno nastavovat jejich vlastnosti *properties*

- v režimu návrhu. *design mode*
- při běhu programu - run-time

Při nastavování vlastností komponenty budeme postupovat tak, že nejdříve vybereme komponentu myší a poté nastavíme příslušnou vlastnost komponenty v objekt inspektoru. Název komponenty se po správném vybrání musí zobrazovat v záhlaví objekt inspektoru. Nejdříve si procvičíme nastavování vlastností formuláře v režimu návrhu, tedy ihned, po spuštění Delphi, kdy nemáme spuštěný žádný projekt. Vrátime se k prvnímu příkladu prázdného formuláře. Uložíme prázdný projekt. Vybereme komponentu Form1 a v inspektoru objektů vybereme její vlastnost Caption (titulek) a napíšeme Informatika je můj oblíbený obor. Spustíme program klávesou F9. Nastavte změnu dalších vlastností formuláře:

- Color (barva)
- ClientHeight (využitelná vnitřní výška)
- ClientWidth(využitelná vnitřní šířka)
- Height (výška)
- Width (šířka)

Ke každé vlastnosti můžeme vyvolat nápovědu stiskem klávesy F1. Vlastnosti můžeme nastavovat u všech komponent tedy u viditelných i neviditelných komponent při běhu programu. Neviditelných komponent při běhu programu je málo a poznáme je tak, že jejich rozměry ve formuláři nejdou změnit.

Vlastnosti komponenty často měníme programovým kódem, který se provede až při běhu programu. Např. červenou barvu formuláře, jehož jméno je *Form1* nastavíme v kódu `Form1.Color:=ClRed;` nebo modrou `Form1.Color:= ClBlue;`. V uvedených příkladech vidíme tzv. tečkovou konvenci. Nejdříve napíšeme název komponenty či objektu a dále název vlastnosti oddělený tečkou. Následuje speciální symbol přiřazení a hodnota vlastnosti. Zbývá ještě otázka, kam takový kód umístíme. Nejčastěji do událostní procedury. Příkaz přiřazení `:=` funguje tak že zjistí hodnotu výrazu na pravé straně a umístí ji do proměnné na levé straně.

KOMPONENTA.VLASTNOST:=HODNOTA

Je to část kódu, který reaguje na určitou událost, např. stisknutí tlačítka myši, klávesy apod.

Příklad

Změna vlastností formuláře při běhu programu:

- Spustíme nový prázdný projekt
- V Inspektoru objektů zvolíme záložku Events a uvidíme abecední seznam událostí, na které formulář reaguje. Poklepeme myši na řádek `OnCreate` a Delphi nám automaticky vygeneruje hlavičku i prázdné tělo událostní procedury. Pak dopíšeme řádek kódu, který zařídí červenou barvu formuláře.
- Spustíme program (F9)

```
procedure TForm1.FormCreate(Sender: TObject); //hlavička
```

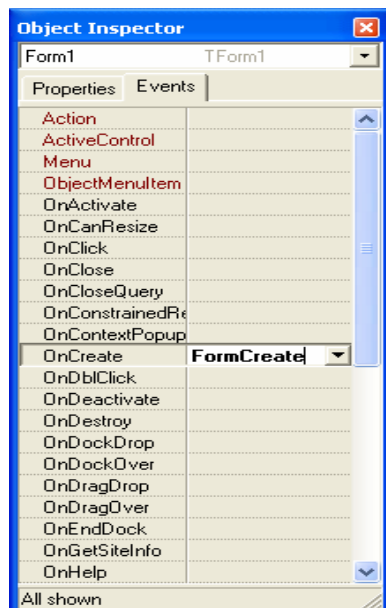
```
begin
```

```
  Form1.Color:= ClRed; {zde napíšememe kód}
```

```
end;
```

Při nastavování vlastností objektů používáme nadefinované konstanty. Pro barvu (color) se jedná o anglický název barvy s předponou `cl` (viz nápověda Color Property).

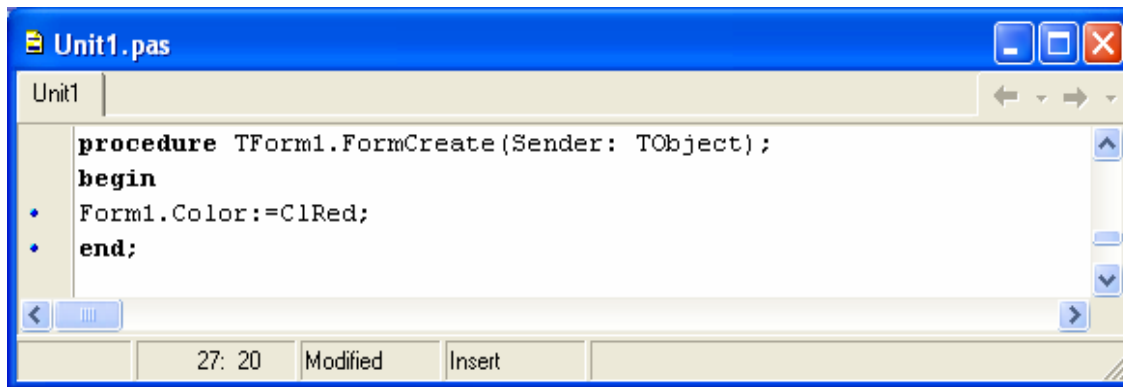
Reakce na události



Zjednodušeně si můžeme představit, že programy vytvořené v Delphi pracují ve Windows tak, že vlastně nic nedělají, pouze čekají na nějakou událost. Událost je vyvolána většinou uživatelem jako je třeba stisk klávesy, pohyb či stisk tlačítka myši nebo systémem např. po uplynutí časového intervalu, po příjmu signálu na vstupním portu a podobně. Při definici reakce na určitou událost postupujeme následujícím způsobem:

- Vybereme komponentu
- V objekt inspektoru vybereme záložku Events
- Po poklepání na příslušnou událost se automaticky vygeneruje událostní procedura
- Do událostní procedury dopíšeme zdrojový text programu, který reaguje na událost.

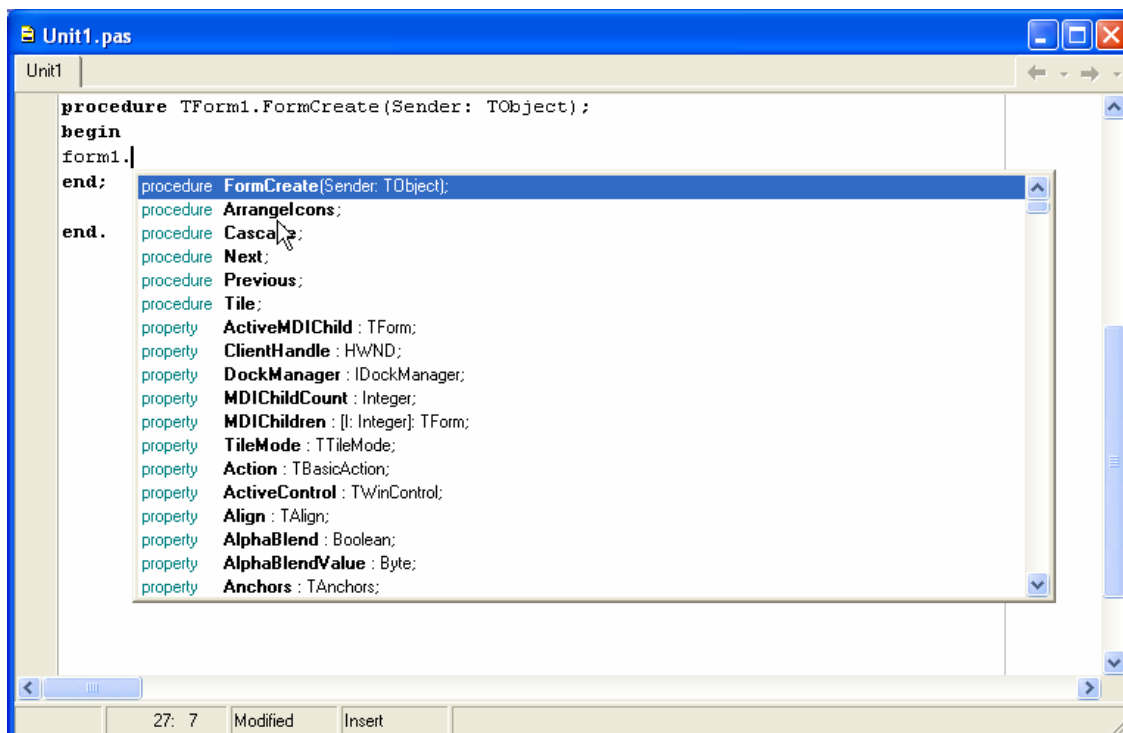
Např: Po vytvoření formuláře formulář zčervená



```
Unit1
procedure TForm1.FormCreate(Sender: TObject);
begin
  Form1.Color:=C1Red;
end;
```

27: 20 Modified Insert

Delphi disponují funkcí kompletace kódu, která výrazně usnadňuje psaní zdrojového kódu. Při zapisování kódu je zápis průběžně vyhodnocován, je automaticky hlídána syntaxe. Když napíšete jméno třídy nebo proměnné, tak za tečkou u proměnné se zobrazí vlastnosti, ze kterých lze vybírat. Např. Form1.Color:=CLBlue. Po zapsání názvu Form1. se zobrazí seznam vlastností, ze kterých lze vybírat.

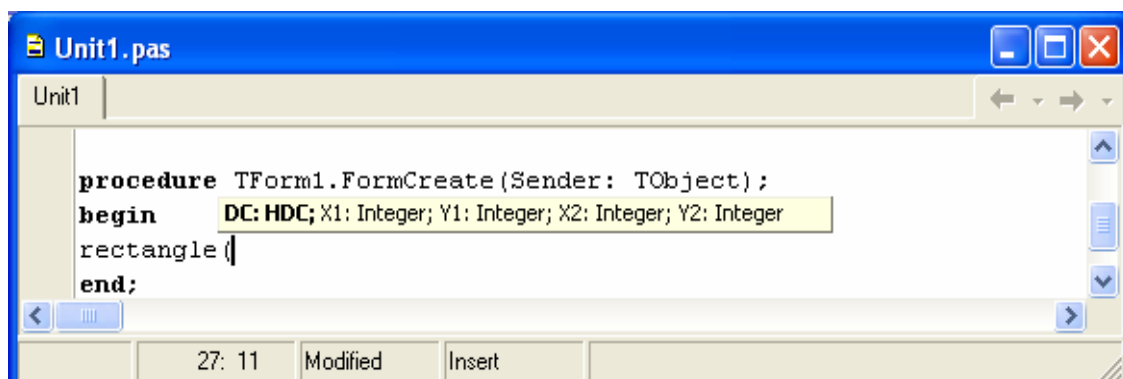


```
Unit1
procedure TForm1.FormCreate(Sender: TObject);
begin
  form1.
end;
```

- procedure FormCreate(Sender: TObject);
- procedure ArrangIcons;
- procedure Casca;
- procedure Next;
- procedure Previous;
- procedure Tile;
- property ActiveMDIChild : TForm;
- property ClientHandle : HWND;
- property DockManager : IDockManager;
- property MDIChildCount : Integer;
- property MDIChildren : [Integer]: TForm;
- property TileMode : TTileMode;
- property Action : TBasicAction;
- property ActiveControl : TWinControl;
- property Align : TAlign;
- property AlphaBlend : Boolean;
- property AlphaBlendValue : Byte;
- property Anchors : TAnchors;

27: 7 Modified Insert

Další funkcí jsou parametry kódu, což je plovoucí okénko, které se objevuje bezprostředně po zapsání otevírací závorky. Například po vytvoření formuláře Form1 se má narýsovat čtverec o souřadnicích X1,Y1,X2,Y2.



```
Unit1
procedure TForm1.FormCreate(Sender: TObject);
begin
  rectangle (
end;
```

27: 11 Modified Insert

Po návrhu uživatelského rozhraní ošetříme události uživatele. To znamená, že pro definované události (events) napíšeme příslušné procedury, které na události reagují. Většinou nám nestačí jen měnit vlastnosti komponent, ale musíme použít konstanty, proměnné a také je často potřeba napsat i vlastní (pomocné) procedury nebo funkce, které na události přímo nereagují. Zde se neobejdeme bez znalostí jazyka Pascal.