Delphi – lekce 2

Formulář

Obecný formulář (`TForm`) představuje okno a je nepřímým následníkem `TComponent`. Pokud vytvoříme vlastní formulář – je to následník právě `TForm`.

Celý popis formuláře je složen z dat komponent a informací o tom, kdo kterou komponentu vlastní. Pokud máme např. na formuláři tlačítko, ve skutečnosti to znamená, že formulář má mezi podřízenými komponentami jedno tlačítko.

Pokud tedy například pohneme tlačítkem doleva, tak se u dané komponenty změní property (vlastnost) `Left`. Při ukládání komponenty (do souboru *.dfm) je hodnota `Left` zapsaná (tedy pokud není aktuální hodnota hodnotou defaultní). Během fáze kompilace (resp. linkování) jsou dfm soubory vloženy do výsledné binárky (tj. do EXE).

Soubor *.dfm tedy obsahuje textový popis komponent na formuláři a je generovaný IDE. Zobrazení popisu formuláře, tj. obsahu souboru dfm, lze provést v IDE při zobrazeném formuláři např. ALT+F12 (a zase zpět).

Při kompilaci se textový popis převede na binární a přilinkuje se k aplikaci. Při vytváření formuláře za běhu, je vytvořena instance formuláře, a pak postupně všech dalších komponent, které na formulář patří. Při jejich vytváření jsou ze spustitelného souboru nataženy uložené vlastnosti (každá komponenta svoje, tedy tlačítko i to svoje `Left`). No a jelikož je Left property dojde (skrze zápisovou metodu) k posunutí komponenty na správné místo určené při návrhu.

Ukážeme si popis formuláře s jednou komponentou `TMemo`. Formulář má jméno frmMain a komponenta Editor. Formulář je vlastníkem Editoru. Popis formuláře v souboru fMain.dfm:

```
object frmMain: TfrmMain
  Left = 200
  Top = 157
  Width = 783
  Height = 540
  ActiveControl = Editor
  Caption = 'Hlavní okno'
  Color = clBackground
  PixelsPerInch = 75
  TextHeight = 16
  TextWidth = 7
  object Editor: TMemo
    Left = 0
    Top = 0
    Width = 783
    Height = 540
    Align = alClient
    TabOrder = 0
  end
end
```

a třída formuláře v fMain.pas:

```
 1unit fMain;
 2interface
 3
 4type
 5  TfrmMain = class(TForm)
 6    Editor: TMemo;
 7  end;
 8var
 9  frmMain:TfrmMain;
10
11implementation
12
13 {$R *.dfm}  // přilinkujeme výše uvedený dfm soubor
14end.
```

Teď se určitě ptáte, kdo vytváří formuláře. Při pohledu do souboru dpr (menu Project/View Source) uvidíme následující:

```
 1program Test1;
 2
 3uses
 4  Forms,
 5  fMain in 'fMain.pas' {frmMain};
 6
 7begin
 8  Application.Initialize;   // inicializace objektu Application
 9  Application.MainFormOnTaskbar := True;
10  Application.CreateForm(TfrmMain, frmMain); // vytvoření formuláře
11  Application.Run;         // jdeme na to
12end.
```

V případě více formulářů se řádek s `CreateForm` opakuje (samozřejmě s jinými parametry).

Pokud nechceme při startu aplikace vytvářet všechny formuláře (a to určitě nechceme, protože to u složitějších formulářů chvilku trvá a zbytečně to zabírá paměť), můžeme tyto řádky smazat. Druhou možností je říct IDE, že si to nepřejeme (menu Project/Options/Forms). V tom případě musíme vytvářet formuláře sami v okamžiku potřeby. První vytvořený formulář se stane hlavním formulářem aplikace. Pokud se tedy po startu zobrazuje jiné okno než je Vaše ctěná libost - zkontrolujte pořadí vytváření formulářů.

Některé důležité vlastnosti formulářů

Jelikož formulář (okno) je jednou z nejčastějších věcí co bude zákazník sledovat (tedy pokud píšeme GUI aplikaci) je vhodné věnovat některým detailům pozornost. Pozn. GUI = Graphic User Interface tedy grafické uživatelské rozhraní, v podstatě cokoliv co jde vidět na obrazovce a slouží pro komunikaci s uživatelem.

Důležitou vlastností je **BorderStyle**, který určuje styl okraje formuláře:

-bsSizeable – klasické okno, změna velikosti

-bsDialog – standardní dialogové okno, nelze měnit velikost

-bsNone – okno nemá okraje

-bsSingle – nelze měnit velikost

-bsSizeToolWin – menší titulek, lze měnit velikost

-bsToolWindows – menší titulek, nelze měnit velikost

Dále je důležité **Caption** – což je titulek okna, který jde vidět.

AlphBlend a **AlphaBlendValue –** určuje zda formulář je částečně průhledný, a jak moc.

BorderIcons nastavují viditelnost systémových ikon pro maximalizaci, minimalizaci případně systémového menu.

FormStyle určuje styl formuláře: normální, fsMDIxxx (formulář je součástí MDI aplikace – někdy příště), fsStayOnTop (vždy nahoře)

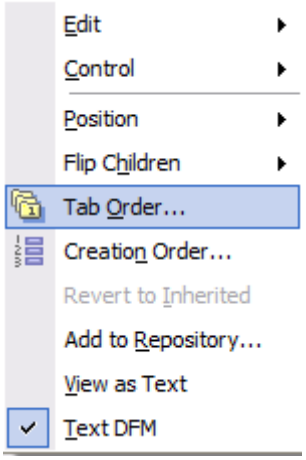
Icon je ikona formuláře, pokud není je použita ikona aplikace.

KeyPreview je velmi zajímavá vlastnost. Pokud je nastavena tak všechny stisknuté klávesy kdekoliv ve formuláři se objevují v příslušné obsluze na formuláři a ne jen u komponenty, ve které se klávesa stiskla. To je v určitých případech velmi výhodné.

Position určuje kde se formulář zobrazí (střed monitoru, střed pracovní plochy, …).

WindowsState určuje jak se formulář zobrazí (minimalizovaně, maximalizovaně nebo normálně)

Jedním z detailů, který ale dokáže pěkně uživatele nakrknout je pořadí prvků v okně, tzv. TabOrder. Jde o to, že uživatel očekává, že focus (zaměření) prvků v okně bude mít nějaké logické pořadí. Toto se hlavně projeví, když uživatel při opakovaném stisku klávesy TAB očekává postupné přecházení mezi prvky. TabOrder pro celý formulář se nastavuje přes popmenu nad formulářem.



Pokud je naistalováno rozšíření CnPack, lze použít pro nastavování některou z jeho funkcí, které jsou podle mne lepší.

Show a ShowModal

Jedná se o dvě metody formuláře, které slouží k jeho zobrazení. Rozdíl je v tom, že Show formulář zobrazí a hned se vrátí, kdežto ShowModal zobrazí formulář a do volacího formuláře se vrátí až po uzavření zobrazeného formuláře. Častěji je používán ShowModal, který se používá pro výběr z dialogového okna – něco jako např. výběr jména souboru.

Příklad

Mějme projekt, ve kterém jsou dva formuláře Form1 a Form2. Form1 je v projektu hned po vytvoření nové aplikace, druhý formulář přidáme přes menu New – Form.

Do sekce uses v implementační části Form1 napíšeme **uses Unit2;** (pozn. Form2 je standardně uložen v souboru Unit2.pas).

Na Form1 vložíme tlačítko a do jeho obsluhy napíšeme Form2.Show; a vložíme druhé tlačítko do jehož obsluhy OnClick napíšeme Form2.ShowModal.

Tím máme nejjednodušší příklad zobrazování oken.

Nyní dáme na Form2 tlačítko s popiskou “OK” a druhé s popiskou “Cancel” a chceme se dozvědět zda Form2 byl zavřen pomocí OK nebo Cancel. K tomu slouží vlastnost **ModalResult** u tlačítka. Osobně to moc nepoužívám a raději používám **ModalResult** přímo z kódu, ale pro ukázku to stačí.

U tlačítka s popiskou OK nastavíme ModalResult na mrOK a u tlačítka Cancel na mrCancel. Navíc u tlačítka s OK nastavíme vlastnost **Default** (čímž řekneme, že toto tlačítko se bude mačkat při stisku Enter) a u tlačítka s Cancel vlastnost **Cancel** (čímž řekneme, že toto tlačítko je stejné jako stisk klávesy ESC).

Poslední věc co je třeba udělat je v prvním formuláři upravit kód u tlačítka s ShowModal na tento kód:

```
case Form2.ShowModal of
  mrOK:
    ShowMessage('OK');
  mrCancel:
    ShowMessage('Cancel');
end;
```

Program je v příloze.

Domácí úkol

Bylo by vhodné upravit předchozí příklad tak, aby se druhé okno nedalo zvětšovat (a ani maximalizovat nebo minimalizovat) a objevovalo se ve středu okna. Dále na Form2 a Form1 přidat po TEditBoxu a upravit volání kódu se ShowModal tak, aby se v případě stisku tlačítka OK překopírovala hodnota z EditBoxu ve Form2 do EditBoxu ve Form1. V případě Cancel se nic neděje. Pozor na TabOrder!

Jen pro jistotu kopírování hodnoty: Edit1.Text := Form2.Edit1.Text;