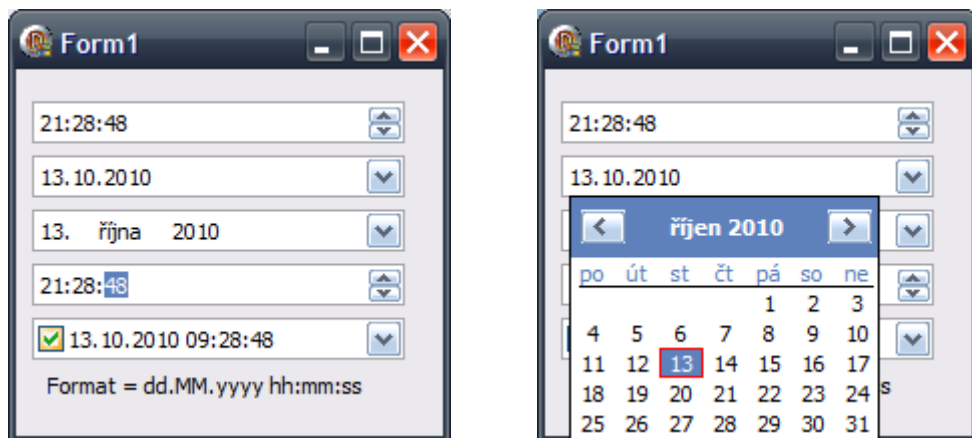


Delphi – lekce 9

TDateTimePicker

Tato komponenta slouží k výběru data nebo času (případně obojího). Na obrázku tato komponenta v několika módech.



Mód zobrazení je určován pomocí následujících property **DataFormat**, **Format**, **DateMode** a **Kind** a **ShowCheckbox** - nejlépe vyzkoušet přímo v IDE. Nejkomplexnější je nastavování pomocí **Format**, formátovací řetězec je shodný s formátováním data a času v Delphi.

Vlastní čas a datum se nastavuje pomocí property **Date** a **Time**.

TPaintBox a kreslení na TCanvas

V Delphi je kreslení implementováno pomocí třídy TCanvas, která představuje plátno a zároveň obsahuje metody pro kreslení (např. výpis textu, kreslení čar, kružnic nebo obdélníků).

Kde na TCanvas můžeme narazit? Hodně komponent obsahuje interně kreslení na TCanvas, můžeme přes canvas kreslit na bitmapu (a tu pak používat) a další možnosti.

Komponenta je TPaintBox je speciální tím, že sama nijak nevypadá, ale umožňuje jednoduše kreslit právě na TCanvas – například tak můžeme jednoduše provádět grafický výstup (nebo si zkusit práci s TCanvas). Na rozdíl od komponenty **TImage**, která zobrazuje bitmapu (na kterou ale můžeme kreslit přes její TCanvas), musíme u komponenty **TPaintBox** provádět kreslení vlastními silami přes obsluhu **OnPaint**.

Zkusíme malý (trochu zábavnější) program. Na formulář dejme komponentu **TPaintBox**, nastavíme mu zarovnání alClient a do její obsluhy **OnPaint** následující obsluhu:

```
procedure TForm1.pb1Paint(Sender: TObject);
var
  iX, iY: Integer;
  iAmp: Integer;
```

```

begin
  iAmp := pb1.Height div 2;
  with pb1.Canvas do
  begin
    Font.Size := 30; //velikost písma
    Font.Color := ClRed; // barva písma
    TextOut(0, iAmp, 'Delphi'); // nápis
    for iX := 0 to (Width - 1) div 2 do
    begin
      iY := Round(sin ((ix + fpos)/ (2*PI)) * iAmp);
      Pen.Color := RGB(0, 0, fPos + ix);
      MoveTo(iX*2, Height);
      LineTo(iX * 2, iAmp + iY);
      Pen.Color := RGB(0, 0, 0); // cerna
      MoveTo(iX*2 + 1, Height);
      LineTo(iX * 2 + 1, iAmp + iY);
    end;
  end;
end;

```

Pen.Color je barva čáry, **MoveTo** přesune pozici na kreslení a **LineTo** nakreslí čáru.

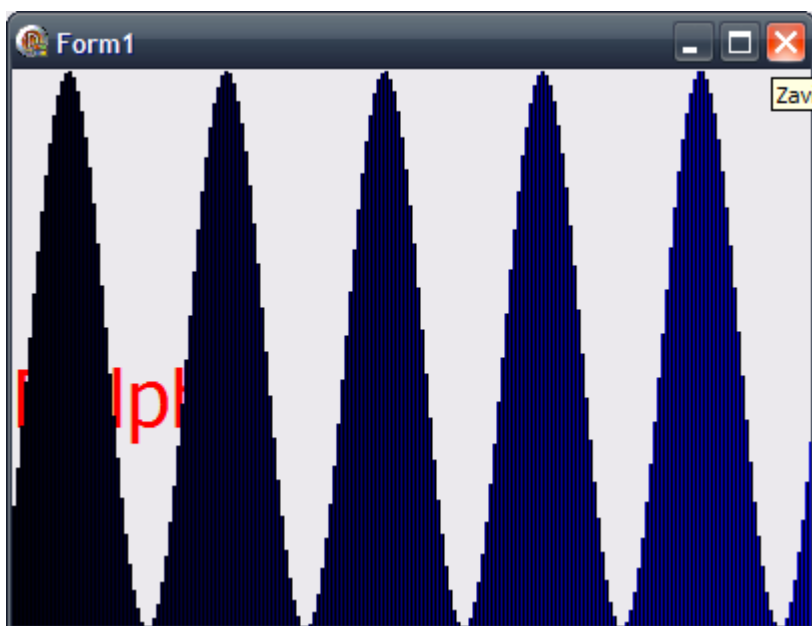
Proměnná **fpos:byte** je proměnnou formuláře a v OnCreate je inicializovaná na 0. Dejme na formulář **TTimer** a nastavíme mu `enabled = True` a interval na 20 (rychlost animace).

V obsluze časovače bude

```

procedure TForm1.tmr1Timer(Sender: TObject);
begin
  inc(fpos, 2);
  pb1.Invalidate; //
end;

```

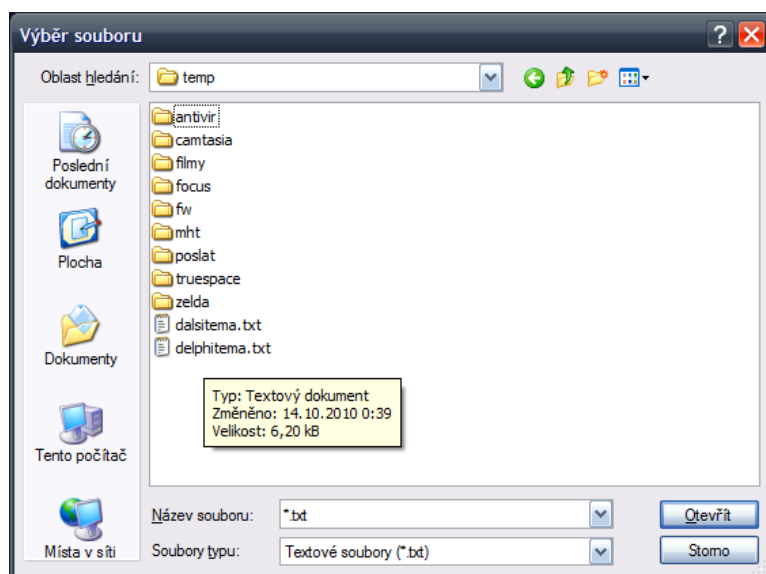


A to je vše. Náš program bude animovat zajímavě kreslenou sinusovku a na pozadí bude text. Dále můžeme vykreslovat obrázky (např. z imagelistu) atd. Program je v příloze.

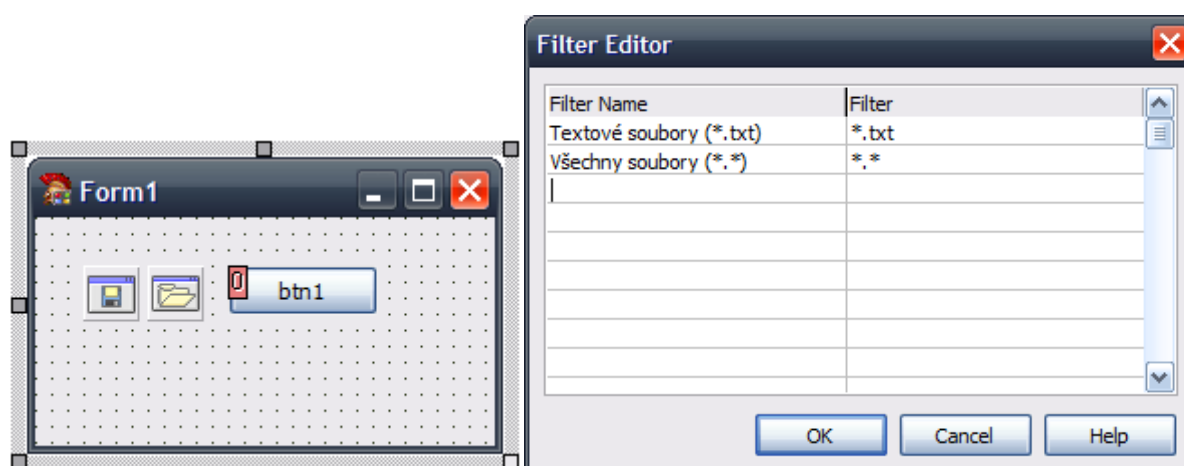
Dialogy pro otevření a uložení souboru

Tyto dvě nevizuální komponenty tj. **TSaveDialog** a **TOpenDialog** jsou velmi důležitou součástí každého programu a slouží k výběru souboru pro otevření nebo uložení.

Obě komponenty jsou velmi podobné a pomocí property **Options** a **OptionsEx** se dají konfigurovat možnosti vzhledu (např. zda soubor musí existovat, zda se dá dialog rozšiřovat atd.).



Obě komponenty můžete buďto vložit na formulář a tam přímo specifikovat jejich vlastnosti nebo je vytvořit až za běhu.



Důležité property jsou **FileName** (vybraný nebo předvybraný soubor), **Filter** (řetězec s filtry pro soubory – lze využít integrovaný editor viz. obrázek a pak se podívat jak filtr vypadá), **FilterIndex** (pořadové číslo předvybraného filtru), **InitialDir** (počáteční adresář) a **Title** (titulek okna).

Pokud máme dialog na formuláři tak můžeme jednoduše napsat

```
if dlgOpen1.Execute then  
    ShowMessage('Vybrán ' + dlgOpen1.FileName);
```

Dialog se zobrazí přes metodu Execute a vrátí boolean hodnotu určující zda se výběr povedl. Vybraný soubor je uložen v property FileName (včetně cesty). Před zavoláním můžeme naplnit Filename počátečním souborem.

Pokud nechceme dávat komponenty na formulář, můžeme vytvářet dialog dynamicky (jako já – komponenta se pak vytváří až je třeba, což je vhodné pro málo používané místa v kódu) např. takto:

```
with TOpenDialog.Create(nil) do  
begin  
    try  
        Options := [ofFileMustExist, ofEnableSizing];  
        if Execute then  
            ShowMessage('Vybrán ' + FileName);  
        finally  
            Free;  
        end;  
    end;  
end;
```

Domácí úkol: Vytvořte jednoduchý textový editor s možností načítání a ukládání souborů.