

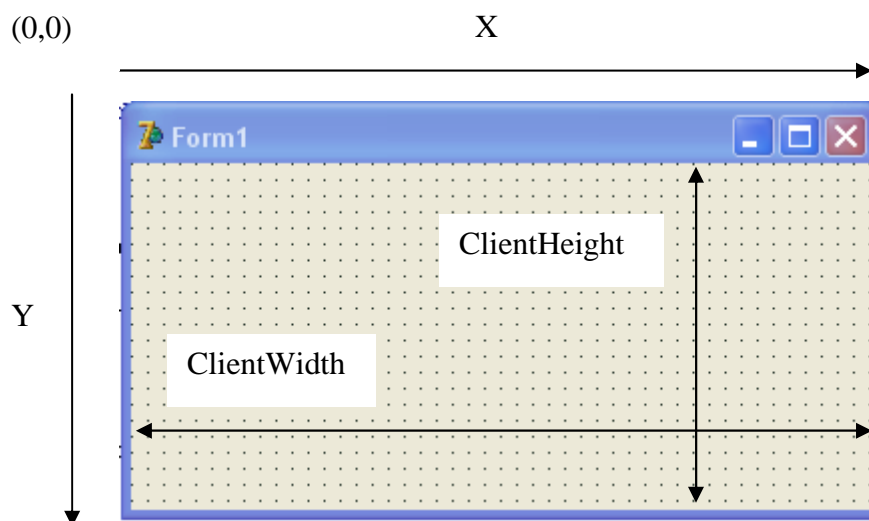
Grafika v Delphi

Kreslení, které nám umožňuje Delphi, je jednoduché. Máme k dispozici 3 možnosti vložení grafiky do aplikace:

- Vytvořit obrázek pomocí grafických nástrojů v režimu návrhu.
- Vykreslit obrázek až při běhu programu.
- Můžeme použít hotový obrázek v režimu návrhu.

Kreslení ve formuláři grafickými nástroji v režimu návrhu

Při vytváření grafického útvaru vlastně nekreslíme přímo ve formuláři, ale na jeho plátno Canvas.



Z obrázku je patrný počátek soustavy souřadnic (0,0) vlevo nahoře. X-ová souřadnice roste směrem doprava, Y-ová souřadnice roste směrem dolů. Počátek je volen vlevo nahoře, protože vykreslování obrázků se děje po řádcích zleva doprava a shora dolů. Zbývá určit souřadnice pravého dolního rohu. To závisí na velikosti okna a na rozlišení monitoru VGA. Např. při rozlišení 800 x 600 pixelů bude pravý dolní roh mít souřadnice (799,599). Abychom nemuseli okno v naší aplikaci přeměřovat, má každé okno vlastnost ClientWidth (využitelná šířka) a ClientHeight (využitelná výška).

Metodický postup kreslení grafickými nástroji v režimu návrhu:

Otevřeme novou aplikaci a uložíme ji pod názvem. Teď si musíme rozmyslet, k jaké události formuláře napíšeme výkonnou proceduru. V seznamu událostí formuláře v Inspektoru objektů (záložka Events) najdeme událost OnPaint (malovat). Je dobré si zapamatovat, že tato událost je vždy zavolána při obnovení formuláře (např. při vytvoření formuláře). Poklepeme myší na řádek, kde se nachází událost OnPaint a Delphi nám vygeneruje skelet procedury. V úvodu jsme říkali, že nekreslíme přímo ve formuláři, ale na plátně. Napíšeme tedy Canvas, tečku a chvíli počkáme na

nápovědu, kde se objeví abecedně seřazený seznam všech procedur, funkcí a vlastností plátna. Vybereme příslušnou proceduru a do kulatých závorek napíšeme parametry.

Kreslení úseček

K nakreslení úsečky line použijeme metodu LineTo (čára do) což vyjadřuje příkaz nakresli úsečku od aktuální pozice pera do souřadnic (x,y). LineTo nakreslí na plátně čáru od aktuální pozice pera (PenPos) do bodu určeného X,Y a nastaví aktuální pozici pera PenPos na X,Y. LineTo tedy změní hodnotu PenPos na X,Y.

Procedura pro kreslení úsečky:

```
LineTo(X, Y:Integer);
```

Příklad:

```
procedure TForm1.FormPaint(Sender: TObject);
```

```
begin
```

```
Canvas.LineTo(100,100); // Nakreslí úsečku od aktuální pozice do souřadnice (100,100)
```

```
end;
```

Vodorovná úsečka:

Vlastnosti: 10pixelů shora a 10 pixelů od obou okrajů formuláře

Nejdříve přesuneme pero do počátečního bodu. To nám zařídí procedura MoveTo(10,10), která přesune pero do daného bodu. Pak nakreslíme úsečku do koncového bodu, jehož x-ovou souřadnici vypočteme jako využitelnou šířku formuláře. Na vlastnost formuláře OnPaint napíšeme příslušnou reakci:

```
procedure TForm1.FormPaint(Sender: TObject);
```

```
begin
```

```
Canvas.MoveTo(10,10);//Přesune aktuální pozici pera do bodu o souřadnicích(10,10)
```

```
Canvas.LineTo(ClientWidth -10,10);//Nakreslí vodorovnou úsečku
```

```
end;
```

Vlastnosti pera

Perem (Pen) můžeme ovládat kreslení čar na plátně. Samotné pero má 4 vlastnosti, které můžeme měnit:

1. Color
2. Width
3. Style
4. Mode

Vlastnosti Pera jsou po spuštění formuláře nějak nastaveny. Platí zásada, že toto nastavení platí tak dlouho dokud je v programu nezměníme. Mód copy překresluje vše co se nachází na plátně.

Nakreslete ve formuláři obdélník ze 4 úseček.

```
procedure TForm1.FormPaint(Sender: TObject);
begin
  canvas.Pen.Color:=CIBlue;//Modrá barva pera
  Canvas.MoveTo(20,20);//Přesune ukazatel do pozice (120,20)
  Canvas.LineTo(ClientWidth-20,20);// Kreslí vpravo
  Canvas.Pen.Color:=Clred;//Kreslí červenou
  Canvas.LineTo(ClientWidth-20,ClientHeight-20);//Kreslí dolů
  Canvas.Pen.Color:=CIYellow;//Kreslí žlutou
  Canvas.LineTo(20,ClientHeight-20);//Kreslí vlevo
  Canvas.Pen.Color:=CIGreen; //Kreslí zelenou
  Canvas.Pen.Width:=5;//Nastaví tloušťku na 5 pixelů
  Canvas.LineTo(20,20);//Kreslí nahoru
end;
```

Kreslení obdélníku

Metoda Rectangle má 4 parametry. 2 souřadnice levého horního rohu a 2 souřadnice pravého dolního rohu. Procedura Rectangle nakreslí obrysy obdélníka, ale vyplní jej bílou barvou. Je to dáno tím, že barva štětce (Brush), který právě maluje plochy je nastaven na bílou.

Canvas.Brush.Color := Clgreen nastaví barvu pozadí na zelenou
Cancas.Brush.Style nastaví typ výplně na křížovou.

```
procedure TForm1.FormPaint(Sender: TObject);
begin
  Canvas.Pen.Color:=CIBlue;//Nastaví barvu pera na modrou
  Canvas.Brush.Style:=BsdiagCross;//Nastaví šrafování výplně
  Canvas.Brush.Color:=Clgreen;//Nastaví barvu výplně na zelenou
  Canvas.Rectangle(30,30,ClientWidth-30,ClientHeight-30);//Nastaví rozměry
  obdélníka
end;
```

Elipsa a eliptický oblouk

```
procedure TForm1.FormPaint(Sender: TObject);
begin
  //plná elipsa
  canvas.Font.Size:=12;//Nastaví Velokost fontu
  canvas.Font.Color:= clred;//Nastaví Barvu fontu
  canvas.TextOut(200,50,'Elipsa');//Zobrazí text na pozici X,Y
  canvas.Pen.Color:=clred;//Nastaví barvu pera
  canvas.Pen.Width:=3;//Nastaví tloušťku pera
  Canvas.Brush.Color:=clblue;//Nastaví barvu výplně
  canvas.Ellipse(100,100,400,300);//Nakreslí elipsu
  //eliptický oblouk
  canvas.Font.Size:=12;
  canvas.Font.Color:= clred;
  canvas.TextOut(200,350,'Eliptický oblouk');
```

```
canvas.Pen.Color:=clred;
canvas.Pen.Width:=3;
Canvas.Brush.Color:=clblue;
canvas.Pie(100,400,400,600,250,300,300,500); //Nakreslí eliptický oblouk
end;
```

Události myši

Událostí myši máme na mysli:

1. Reakci na stisk tlačítka myši
2. Reakci na uvolnění tlačítka myši
3. Reakci na posun myši

V prázdném formuláři přepneme na záložku Inspektoru objektů událost Events a tam uvidíme 3 události myši:

1. OnMouseDown stisknutí tlačítka myši
2. OnMouseUp uvolnění tlačítka myši
3. OnMouseMove posun tlačítka myši

Parametry události myši:

Sender – objekt detekující akci myši

Button – Určuje příslušné tlačítko myši: MbLeft, MbMiddle, MbRight

Shift – Indikuje stav kláves Alt, Ctrl, Shiftv okamžiku akce myši,

X,Y – souřadnice bodu, kde událost nastala

Logická položka kreslení nám vypomůže s rozlišením, kdy se má kreslit a kdy se už kreslit nemá. Kreslení nastavíme na hodnotu True při stisknutí tlačítka myši. Kreslení nastavíme na False při uvolnění tlačítka myši.

Nakreslete čáru za pomoci tlačítek myši:

var

Form1: TForm1;

Kresleni:boolean;

implementation

{ \$R *.dfm }

//Tato procedura se provede po stisku tlačítka myši

procedure TForm1.FormMouseDown(Sender: TObject; Button: TMouseButton;

Shift: TShiftState; X, Y: Integer);

begin

Kresleni:=True; //Zapne Kresleni

Canvas.Pen.Color:=Clblue; //Nastaví barvu pera na modrou

Canvas.MoveTo(X,Y); {přesun pera do pozice X,Y}

end;

//tato procedura se provede po uvolnění tlačítka myši

procedure TForm1.FormMouseUp(Sender: TObject; Button: TMouseButton;

Shift: TShiftState; X, Y: Integer);

begin

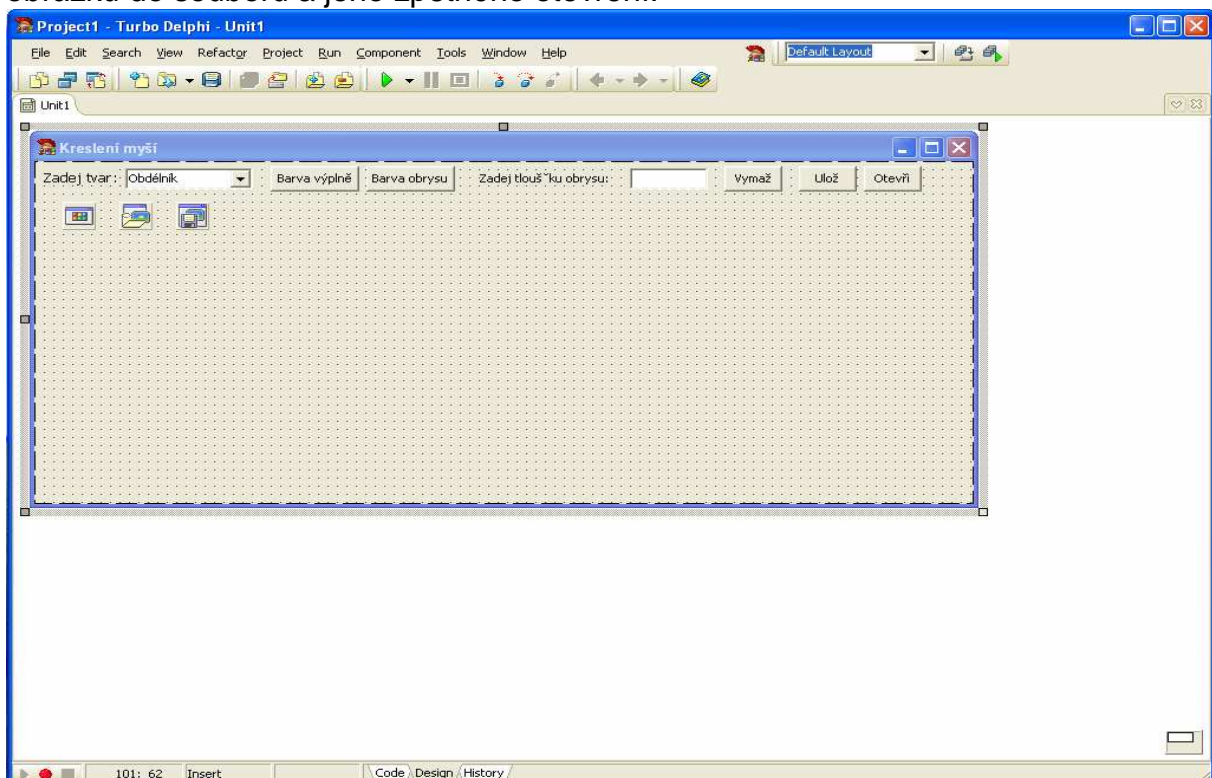
```
Canvas.LineTo(X,Y);//Kreslí čáru do pozice X,Y  
Kresleni:=False; //Vypne Kreslení  
end;
```

```
//Tato procedura se provede během tažení maši po podložce při stisknutém tlačítku  
procedure TForm1.FormMouseMove(Sender: TObject; Shift: TShiftState; X,  
  Y: Integer);  
begin  
If kresleni=true then  
Canvas.LineTo(X,Y);//kreslí čáru do pozice X,Y  
end;
```

Pro kreslení úsečky myši procedurou LineTo(X,Y) je třeba si zapamatovat počáteční bod v okamžiku stlačení tlačítka myši, do kterého pak přesuneme pero. Zavedeme si 2 položky celočíselného typu POcX a PocY ve veřejné části formuláře. V okamžiku uvolnění tlačítka jsou souřadnice koncového bodu známy, takže jenom přesuneme pero do souřadnic počátečního bodu a pak zavoláme metodu LineTo a aktuálními souřadnicemi. Dosáhli jsme kreslení úsečky s nežádoucí stopou.

Odstranění stopy: vtip spočívá ve smazání každé úsečky před vykreslením další úsečky. Musíme proto zaznamenat polohu předchozí úsečky. To je cílem položek posunX a PosunY. Nastavíme PosunX a PosunY na koncový bod každé bezprostředně nakreslené úsečky a pak použijeme posunutý a počáteční bod ke smazání úsečky před tím, než bude nakreslena nová úsečka. Změnou Pera na mód PmNotXor dojde k překreslení úsečky barvou pozadí. Po nastavení na původní hodnotu módu PnCopy bude opět kreslená čára aktuálně nastavenou barvou pera. To co jsme si řekli pro kreslení úsečky také platí pro kreslení jakéhokoliv geometrického obrazce např. čtverce, elipsy, kružnice atd.

Následující příklad ukazuje vykreslování zadaného tvaru, zadanou barvou výplně a obrysu a zadanou tloušťkou pera na komponentu image včetně možnosti uložení obrázku do souboru a jeho zpětného otevření.



```

unit Unit1;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, ExtCtrls, ExtDlgs;

type
  TForm1 = class(TForm)
    Label1: TLabel;
    ComboBox1: TComboBox;
    Button1: TButton;
    Button2: TButton;
    ColorDialog1: TColorDialog;
    Button3: TButton;
    Label2: TLabel;
    Edit1: TEdit;
    Image1: TImage;
    OpenPictureDialog1: TOpenPictureDialog;
    SavePictureDialog1: TSavePictureDialog;
    Button4: TButton;
    Button5: TButton;
    procedure Button1Click(Sender: TObject);
    procedure Button2Click(Sender: TObject);
    procedure Edit1Change(Sender: TObject);
    procedure Image1MouseDown(Sender: TObject; Button: TMouseButton;
      Shift: TShiftState; X, Y: Integer);
    procedure Image1MouseUp(Sender: TObject; Button: TMouseButton;
      Shift: TShiftState; X, Y: Integer);
    procedure Button3Click(Sender: TObject);
    procedure Button4Click(Sender: TObject);
    procedure Button5Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form1: TForm1;
  x1,y1:integer;
implementation

{$R *.dfm}
procedure TForm1.Button1Click(Sender: TObject);
begin
  if colordialog1.Execute then
    image1.canvas.Brush.Color:=colordialog1.color;
end;

```

```

procedure TForm1.Button2Click(Sender: TObject);
begin
if colordialog1.Execute then
image1.canvas.pen.Color:=colordialog1.color;
end;

procedure TForm1.Button3Click(Sender: TObject);
begin
image1.Canvas.Brush.Color:=clwhite;
image1.Canvas.pen.Color:=clwhite;
image1.Canvas.Rectangle(0,0,image1.Width,image1.Height);
end;

procedure TForm1.Button4Click(Sender: TObject);
begin
if savepicturedialog1.execute then
image1.Picture.SaveToFile(savepicturedialog1.fileName);
end;

procedure TForm1.Button5Click(Sender: TObject);
begin
if openpicturedialog1.execute then
image1.Picture.loadfromFile(openpicturedialog1.fileName);
end;

procedure TForm1.Edit1Change(Sender: TObject);
begin
image1.canvas.Pen.Width:=strtoint(edit1.Text);
end;

procedure TForm1.Image1MouseDown(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
begin
x1:=x; y1:=y;
end;

procedure TForm1.Image1MouseUp(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
begin
if combobox1.ItemIndex=0 then image1.canvas.rectangle(x1,y1,x,y);
if combobox1.ItemIndex=1 then image1.canvas.ellipse(x1,y1,x,y);
if combobox1.ItemIndex=2 then
begin
image1.canvas.moveto(x1,y1);
image1.canvas.lineto(x,y);
end;
end;

end.

```