

Brh 9 W2-Programmieren Grafik-Canvas

Grafik mit Delphi Das Canvas-Objekt

Grafikoperationen sind unter Delphi auf die Zeichenfläche (Canvas) eines Objektes bezogen. Mögliche Ausgabeobjekte sind:

- **Form** (Standard-Register)
- **Image** (Zusätzlich-Register)
- **Paintbox** (System-Register)

Unterobjekte des Canvas-Objektes sind:

Pen:

Mit dem „Stift“ wird die Form der Linien bzw. Begrenzungen bei Rechtecken, Kreisen etc. festgelegt.

Canvas.pen.width

Stiftbreite in Pixeln

Canvas.pen.color

RGB-Farbangabe (TrueColor 16 Mio. Farben), entweder per Integerwert oder mit den vordefinierten Windowsstandardfarben (siehe Tcolor) wie Clred, Clyellow etc.

Canvas.pen.style

Neben der normalen Linie (psolid = Standard) sind auch gestrichelte (psdash) oder gepunktete (psdot) Linien möglich.

Brush

Der „Pinsel“ legt die Füllungen geometrischer Figuren fest.

Canvas.brush.color

wie pen.color

Canvas.brush.style

Möglich sind die 100%Füllung (bssolid), keine Füllung (bsclear) und verschiedene Strukturen (bsCross, bsHorizontal, bsBdiagonal u.a.)

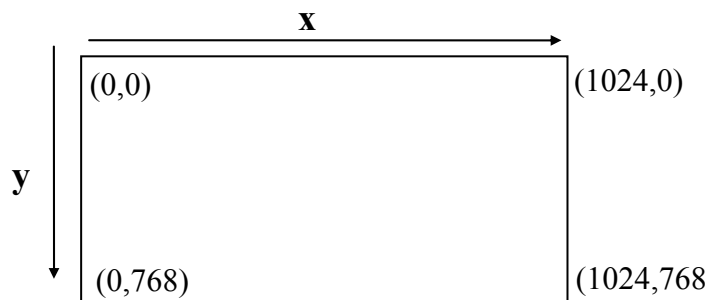
Brh 9 W2-Programmieren Grafik-Canvas

Font

Über Font wird die Schrift in der bekannten Form über

Font.name	'Arial', 'Times New Roman' etc.
Color	clred, clgreen etc.
style	fsbold etc.
size	Textgrösse in Punkten: 10,12,14 etc.

Das Koordinatensystem bei Bildschirmauflösung 1024 x 768



Die Position von Bildschirmpunkten wird in der Form (x,y) angegeben, wobei x der Abstand vom linken Rand und y der Abstand vom oberen Rand der Zeichenfläche ist.

Elementare Zeichenfunktionen

Punkte

Einen Punkt kannst du z.B. mit

Canvas.pixels[200,100] := cblack

setzen.

Brh 9 W2-Programmieren Grafik-Canvas

Linien

Bevor du eine Linie zeichnest, legst du einen Startpunkt mit `MoveTo` fest:

Canvas.MoveTo(100,100)

Von diesem Punkt aus kannst du mit **Canvas.LineTo(200,100)** eine gerade Linie zeichnen. Der Endpunkt (200,100) ist nun der Anfangspunkt der nächsten Linie, falls du die Koordinaten nicht mit `MoveTo` veränderst. Die Linieneigenschaften kannst du vorher mit `Pen.width` etc. festlegen oder verändern.

Beispiel1:

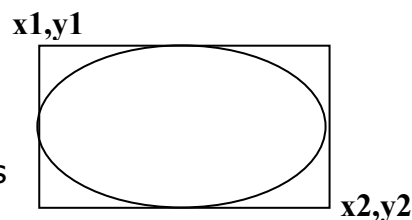
```
Form1.Canvas.MoveTo(100,100); Form1.Canvas.Pen.Color := clred;  
Form1.Canvas.Pen.Width := 5; Form1.Canvas.LineTo(300,150);  
Form1.Canvas.Pen.Color := cllime; Form1.Canvas.LineTo(200,200);  
Form1.Canvas.Pen.Color := cllime; Form1.Canvas.LineTo(100,100);  
Mit
```

Canvas.Rectangle(x1,y1,x2,y2)

erhalten Sie ein Rechteck. Falls es nicht mit `Brush.color` gefüllt sein soll, muss `Brush.Style = bsclear` sein.

Ellipse und Kreis

Der Kreis ist Sonderfall der Ellipse. Angegeben werden die Koordinaten des umgebenden Rechtecks:



Canvas.Ellipse(x1,y1,x2,y2)

Textausgabe

Nachdem mit `Canvas.Font` die Schrifteigenschaften festgelegt worden sind, kannst du mit

Canvas.Textout(100,100, 'Mein Text') oder
Canvas.Textout(100,100, Edit1.text)

Text positionieren.

Brh 9 W2-Programmieren Grafik-Canvas

Aufgaben:

1. Haus vom Nikolaus: Schreibe ein Programm, das mit acht Buttons das Haus vom Nikolaus zeichnet. (Das-ist-das-Haus-vom-Ni-ko-laus)
2. Lege diese Prozedur in einen Button:

```
var
i: integer;
Begin
  With Form1.Canvas do begin
    For i := 1 to 10 do begin
      MoveTo (i * 10, 0);
      LineTo (screen.width - i * 10, screen.height);
    end;
  end;
end;
```

Erweitere nun das Programm, indem du

- am Anfang als Stiftfarbe clblue zuweist
 - die Linienstärke jeweils um ein Pixel erhöhst
 - die Anzahl der Linien auf 50 erhöhst
 - einen Farbverlauf erstellen, indem du der Wert von Color in jedem Schleifendurchgang erhöhst.
3. Erstelle einen Bildschirmschoner, der in einem Timer zufällig Kreise und/oder Rechtecke ausgibt.