GRAFIK

In Delphi gibt es grundsätzlich drei Varianten um Grafik anzuzeigen:

- Anzeigen von fertigen Bildern mit Image oder PaintBox
- Zeichnen von Figuren mit Shape
- Einsatz von Grafikmethoden (d. h. Programmierung) auf dem Canvas (engl. Leinwand)

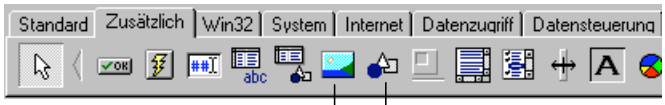


Image Shape

PaintBox

SHAPE

Mit Shape können Sie Formen wie Rechteck, Ellipsen, Kreise auf die gleiche Art wie z. B. ein Button oder ein Label erzeugen.

Im Objektinspektor können Sie die Figur gestalten:

- Shape Form auslesen: Rechteck und Quadrat (auch mit abgerundeten Ecken)
  Ellipse und Kreis

- Brush Füllmuster und Füllfarbe;
  Damit diese Optionen zum Vorschein kommen, müssen sie das + vor Brush doppelklicken.
  Unter Color finden Sie die Farbe, unter Style diverse Füllmuster.

- Pen Gestaltung der Randlinie;
  + doppelklicken für die Optionen Color, Style (Strichmuster), Width (Strichbreite in Punkt).
  Achtung! nur ausgezogene Linien (psSolid) können breiter als 1 gezeichnet werden.

Muster dazu auf den nächsten Seiten!

Auf den folgenden Seiten geht es nun vor allem darum, Grafiken mittels Programmierung herzustellen.

TCanvas dient als Zeichenfläche für Objekte, die sich selbst zeichen. Die unter Windows üblichen Steuerelemente wie z.B. Eingabe- oder Listenfelder benötigen keine Zeichenfläche, da sie von Windows gezeichnet werden.
TCanvas stellt Eigenschaften, Ereignisse und Methoden zur Verfügung, die beim Erzeugen von Bildern hilfreich sind, indem sie

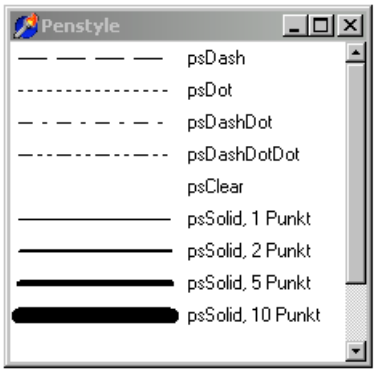
- die Art des verwendeten Pinsels und Stiftes sowie die Schriftart festlegen.
- eine Vielzahl von Fomen und Linien zeichnen und füllen.
- Text ausgeben.
- Grafiken zeichnen.
- eine Reaktion auf Änderungen am aktuellen Bild ermöglichen.

Muster zu Pen

Der Befehl heisst:

```
Canvas.Pen.Style:=psDash;
```

Der gewählte Style gilt für Linien und Ränder von Figuren; er bleibt eingestellt, bis er durch einen neuen Style-Befehl ersetzt wird.

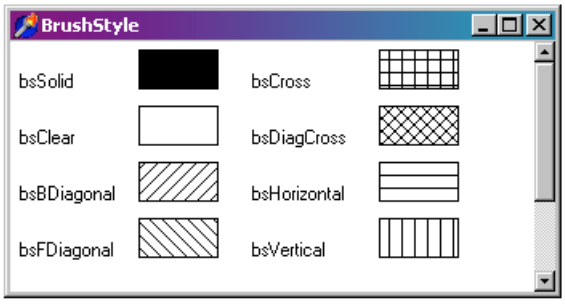


Muster zu Brush

Der Befehl heisst:

```
Canvas.Brush.Style:=psSolid;
```

Die Musterlinien werden in der aktuellen Musterfarbe auf dem bestehenden Hintergrund gezeichnet. Der Rand kann mit Pen.Color oder Pen.Style unabhängig davon gezeichnet werden. Der gewählte Style bleibt eingestellt, bis er durch einen neuen Style-Befehl ersetzt wird.



Farben

Systemfarben	clAqua	Aqua (Blaugrün)	clMaroon	Maroon (Kastanienbraun)
	clBlack	Schwarz	clNavy	Marineblau
	clBlue	Blau	clOlive	Olivgrün
	clDkGray	Dunkelgrün	clPurple	Purpur
	clFuchsia	Fuchsia (rosa)	clRed	Rot
	clGray	Grau	clSilver	Silber
	clGreen	Grün	clTeal	Teal (dunkles Blaugrau)
	clLime	Lime (mittleres Grün)	clWhite	Weiß
	clLtGray	Hellgrün	clYellow	Gelb

Systemfarben werden folgendermassen zugewiesen:

```
Canvas.Pen.Color:=clSilver;
Canvas.Brush.Color:= clBlue;
```

Pen- und Brush-Farben können unabhängig voneinander gewählt werden.
Sie bleiben eingestellt, bis sie durch einen neuen Farbbefehl aufgehoben werden.

RGB-Farben

Die Farben werden durch Zahlen dargestellt, die die Anteile an Rot, Grün, Blau wiedergeben und zwar üblicherweise in Hexadezimalzahlen.

z. B.
```
Canvas.Brush.Color:=$000000FF     (rot)
Canvas.Brush.Color:=$0000FF00     (grün)
Canvas.Brush.Color:=$00FF0000     (blau)
Canvas.Brush.Color:=$00000000     (schwarz)
Canvas.Brush.Color:=$00FFFFFF     (weiss)
```

FF ist die grösste zweistellige Zahl im Hexadezimalsystem, das die 16 Ziffern 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F aufweist und entspricht im Dezimalsystem der Zahl 255.

00	BF	7A	CD
	Blau	Grün	Rot

BF: $11 \cdot 16 + 15 = 191$ Teile Blau
7A: $7 \cdot 16 + 10 = 122$ Teile Grün
CD: $12 \cdot 16 + 13 = 205$ Teile Rot

Ohne Hexadezimalzahlen lässt sich die Farbe $00BF7ACD zuweisen als:

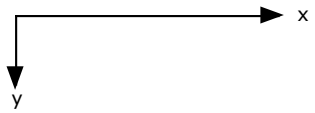
```
Canvas.Pen.Color:=RGB(191,122,205)
```

Mit den Extremwerten:

```
Canvas.Pen.Color:=RGB(0,0,0)          (Schwarz)
Canvas.Pen.Color:=RGB(255,255,255)    (Weiss)
```

KOORDINATEN

Das Koordinatensystem beruht auf der Einheit Punkt oder Pixel. Der Nullpunkt ist in der linken oberen Ecke des Formulars.



Sie sehen im Objektinspektor wie breit (Width) und wie hoch (Hight) das Formular momentan ist.
Falls Sie die Grösse des Formulars noch verändern, arbeiten Sie besser mit den Zahlen CustomWidth und CustomHight.

(CustomWidth-20, CustomHight-10) sind die Koordinaten eines Punktes, der vom rechten Rand 20 Pixel und vom untern Rand 10 Pixel Abstand hat.

Der Punkt mit den Koordinaten (CustomWidth **div** 2, CustomHight **div** 2) liegt etwa in der Mitte des Formulars. Beachten Sie das **div**! Koordinaten müssen ganzzahlig sein!

Tipp: Wenn Sie sich die Koordinaten nicht ganz vorstelllen können:
zeichnen Sie ein Label und lesen Sie die Zahlen im Objektinspektor ab.
(Left,Top) ergibt die Ecke oben links, (Left+Width,Top+Hight) die Ecke unten rechts.

FIGUREN

Punkt

Auf einen Punkt greifen Sie mit der Eigenschaft Pixels zu:

z. B. den Punkt (50,20) schwarz färben: Canvas.Pixels[50,20]:=clBlack

Da Pixels eine Eigenschaft ist, können Sie damit auch die Farbe eines Punktes ablesen:

```
var
  farbe: Tcolor;
farbe:=Canvas.Pixels[50,20];
```

Um abzulesen, wo sich der Stift im Moment befindet, benützen Sie die Eigenschaft PenPos:

```
var
  Ecke: Tpoint;
  xwert: LongInt;
Ecke:=Canvas.Pos;
xwert:=Canvas.Pos.X;
```

Geraden

Um Geraden zu zeichnen benötigen Sie zwei Befehle;

```
Canvas.MoveTo(x1,y1);
Canvas.LineTo(x2,y2);
```

Zahlenbeispiel:

```
Canvas.MoveTo(70,50);
Canvas.LineTo(410,120);
```

Es ginge auch mit der Methode PolyLine:

```
Canvas.PolyLine([Point(x1,y1),Point(x2,y2)]);
```

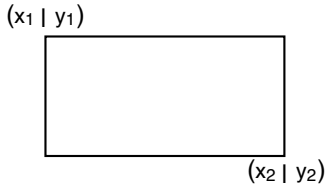
Rechteck und Quadrat

Linke obere und rechte untere Grenze angeben:

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

Zahlenbeispiel:

```
Canvas.Rectangle(70,50,170,100);
```



Ein Quadrat hat gleiche Seitenlängen:

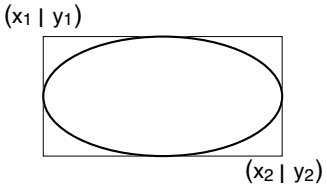
```
Canvas.Rectangle(70,50,100,80);
```

$$\begin{pmatrix} 70+30=100 \\ 50+30=100 \end{pmatrix}$$

Ellipse und Kreis

Linke obere und rechte untere Grenze des Rechtecks angeben:

Canvas.Ellipse(x_1,y_1,x_2,y_2);



Zahlenbeispiel:

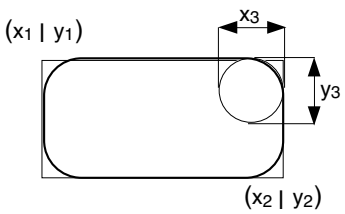
Canvas.Ellipse(70,50,170,100);

Ein Kreis mit Mittelpunkt (u,v) und Radius r:

Canvas.Ellipse(u-r,v-r,u+r,v+r);

Abgerundetes Rechteck

Canvas.RoundRect(x_1,y_1,x_2,y_2,x_3,y_3);



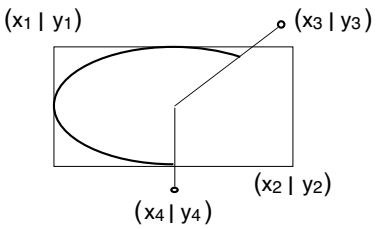
Zahlenbeispiel:

Canvas. RoundRect(70,50, 170,100,20,20);

Bogenstück

Das Bogenstück kann nicht gefüllt werden.

Canvas.Arc(x_1,y_1,x_2,y_2,x_3,y_3,x_4, y_4);

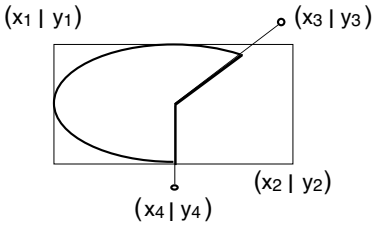


Zahlenbeispiel:

Canvas. Arc(70,50, 170,100, 150,50, 120,100);

Tortenstück

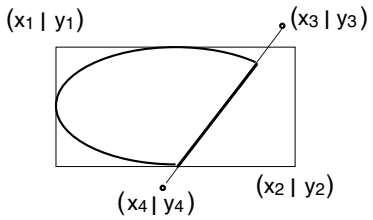
Canvas.Pie(x_1,y_1,x_2,y_2,x_3,y_3,x_4, y_4);



Zahlenbeispiel:

Canvas. Pie(70,50, 170,100, 150,50, 120,100);

Kreissegment



Canvas.Cord(x1,y1,x2,y2,x3,y3,x4, y4);

Zahlenbeispiel:

Canvas. Cord(70,50, 170,100, 150,50, 120,100);

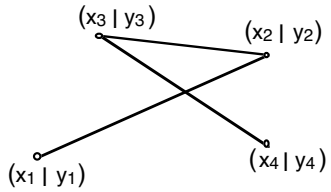
Streckenzug

Der Streckenzug kann nicht gefüllt werden.

Canvas.PolyLine([Point(x1,y1),Point(x2,y2), Point(x3,y3), Point(x4, y4), . . .]);

Zahlenbeispiel:

Canvas.PolyLine([Point(5,22), Point(37,9), Point(14,4), Point(37,19), . . .]);



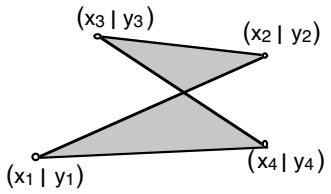
Vieleck

Das Polygon schliesst sich selbst und kann gefüllt werden.

Canvas.Polygon([Point(x1,y1),Point(x2,y2), Point(x3,y3), Point(x4, y4), . . .]);

Zahlenbeispiel:

Canvas.Polygon([Point(5,22), Point(37,9), Point(14,4), Point(37,19), . . .]);



TEXTAUSGABE

Einfachste Methode um Text auf den Bildschirm zu bringen.

Als Schriftart wird der aktuelle Wert von Font verwendet.
Zur Bezeichnung der Lage geben Sie die Koordinaten der Ecke oben links an.

Beispiel:

```
Canvas.Font.Height:=20;
Canvas.TextOut(20,10,'HALLO!');
```