

### 23.3.2 Methoden

Die Klasse `Paint` (gb.qt4) besitzt nur (statische) Methoden. Die vielen Methoden sind in den u.a. Tabellen gruppiert notiert, um so Zusammenhänge besser sichtbar werden zu lassen. Nach den Tabellen finden Sie Hinweise zur Verwendung ausgewählter Methoden und Beschreibungen der Parameter.

Das Projekt *Painting* aus den Gambas-Beispielen sollten Sie sich unbedingt ansehen, weil es nicht nur das Zeichnen mit der Klasse `Paint` demonstriert, sondern auch den verwendeten (Paint-)Quelltext für die 22 Linien und Flächen anzeigt:

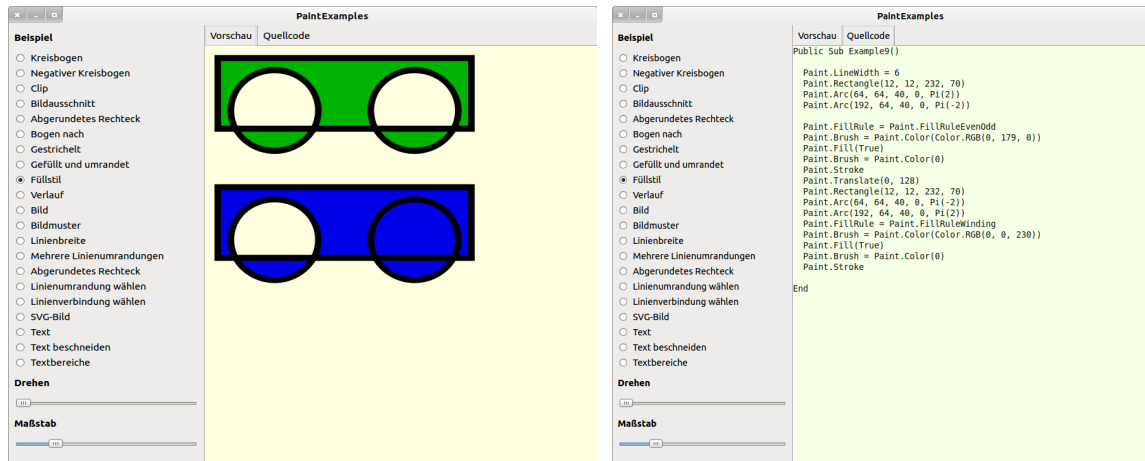


Abbildung 23.3.2.1: Projekt Painting – Beispiel 9: Füllstil

#### 23.3.2.1 Methoden

Die Klasse `Paint` verfügt über diese Methoden, bei denen optionale Argumente farblich hervorgehoben sind:

Methoden	Beschreibung
<code>Begin ( Device As Object [ Area As RectF ] )</code>	Startet das Zeichnen auf dem angegebenen Device (Zeichenfläche: <code>DrawingArea</code> , <code>Picture</code> , <code>Image</code> , <code>Printer</code> , <code>SVGImage</code> ). Der Aufruf kann verschachtelt werden.
<code>End ( )</code>	Beendet das Zeichnen. Sie müssen diese Methode genauso oft aufrufen, wie die <code>Paint.Begin(..)</code> -Methode.
<code>NewPath ( )</code>	Löscht den aktuellen Zeichen-Pfad. Nach dem Aufruf existiert weder ein Pfad und noch ein aktueller (Start-)Punkt.
<code>ClosePath ( )</code>	Fügt ein Linien-Segment zum aktuellen Pfad hinzu und zwar vom aktuellen Punkt bis zum Startpunkt des aktuellen Teil-Pfades (das ist der Punkt, der zuletzt an <code>Paint.MoveTo()</code> übergeben wurde). So wird der aktuelle Pfad geschlossen und der aktuelle Punkt wird zum gemeinsamen Anfangs- und Endpunkt des Pfades.
<code>Scale ( SX As Float, SY As Float )</code>	Ändert die <i>aktuelle Transformationsmatrix</i> (CTM) durch Skalierung der x- und y-User-Space-Achsen mit den Faktoren SX und SY. SX: Skalierungsfaktor für die x-Richtung SY: Skalierungsfaktor für die y-Richtung
<code>Translate ( TX As Float, TY As Float )</code>	Ändert die CTM durch Verschieben des Koordinatenursprungs um TX und TY. TX: Verschiebungsweite in x-Richtung TY: Verschiebungsweite in y-Richtung
<code>Rotate ( Angle As Float )</code>	Ändert die CTM durch Drehen der Koordinaten-Achsen bei gegebenem Drehwinkel Angle (Bogenmaß). Die Drehrichtung für positive Drehwinkel entspricht einer Drehung von der positiven x-Achse in Richtung der positiven y-Achse (mathematisch positiv).

Methode	Beschreibung
Clip ( [ Preserve As Boolean ] )	Erstellt eine neue Clip-Region, indem die aktuelle Clip-Region mit dem aktuellen Pfad geschnitten wird – unter Beachtung der aktuellen FillRule. Nach der Anwendung von Clip() wird der aktuelle Pfad gelöscht, außer das Preserve-Argument ist TRUE. Die aktuelle Clip-Region beeinflusst alle Zeichenoperationen: alle Änderungen außerhalb dieser Region werden ignoriert. Ein Aufruf von Clip() kann die Clip-Region nur kleiner machen, niemals größer, aber der aktuelle Clip ist eine Eigenschaft des Paint-Zustands. Eine temporäre Clip-Einschränkung können Sie erreichen, indem Sie innerhalb eines Paint.Save()-/Paint.Restore()-Paares clippen. Nur Paint.ResetClip() vergrößert die Clip-Region wieder.
ResetClip ( )	Setzt die aktuelle Clip-Region auf ihren ursprünglichen, uneingeschränkten Zustand zurück.
Reset()	Setzt die CTM auf die Identitätsmatrix (Ausgangsmatrix).
Save ( )	Erstellt eine Kopie des aktuellen Paint-Zustands und speichert diese in einem internen Stapel der gespeicherten Zustände. Wenn Wiederherstellen aufgerufen wird, so wird der Zeichnung-Zustand aus dem gespeicherten Zustand wieder hergestellt. Mehrere Aufrufe zum Speichern und Wiederherstellen können verschachtelt werden.
Restore ( )	Stellt den Paint-Zustand wieder her, indem der gespeicherte Zustand aufgerufen wird. Anschließend wird der gespeicherte Zustand aus dem Stapel der gespeicherten Zustände gelöscht.

23.3.2.2 Methoden – Farbe und Muster

Methode	Beschreibung
Color ( Color As Integer ) As PaintBrush	Erstellt einen neuen Pinsel entsprechend einer opaken oder durchscheinenden Farbe. Die Farbe 'Color' wird definiert wie für alle GUI-Komponenten → Kapitel 25.3.5 Arbeit mit Farben
LinearGradient ( X0 As Float, Y0 As Float, X1 As Float, Y1 As Float, Colors As Integer[], Positions As Float[] [ , Extend As Integer ] ) As PaintBrush	Erstellt einen neuen linearen Farbverlauf-Pinsel entlang der durch (X0 Y0) und (X1 Y1) definierten Linie und definiert Farbstellen aus den Farb- und Positionsargumenten.
RadialGradient ( CX As Float, CY As Float, Radius As Float, FX As Float, FY As Float, Colors As Integer[], Positions As Float[] [ , Extend As Integer ] ) As PaintBrush	Erstellt einen neuen radialen Farbverlauf-Pinsel, wobei die Farben zwischen einem Brennpunkt (FX FY) und dem Endpunkt auf einer von (CX0 CY0, Radius0) definierten Kreisfläche interpoliert werden und definiert Farbstellen aus den Farb- und Positionsargumenten.

23.3.2.3 Methoden – Linien und Flächen

Methode	Beschreibung
MoveTo ( X As Float, Y As Float )	Beginnt einen neuen (Unter-)Pfad. Nach diesem Aufruf hat der aktuelle Punkt die Koordinaten P(x y).
RelMoveTo(DX, DY)	Beginnt einen neuen (Unter-)Pfad. Nach diesem Aufruf besitzt der aktuelle Punkt P' ein Offset von DX und DY gegenüber dem Ausgangspunkt P. Bei einem Start-Punkt P(x y) hat der aktuelle Punkt nach Anwendung von RelMoveTo(DX,DY) die Koordinaten P'(x+DX   y+DY).
LineTo ( X As Float, Y As Float )	Fügt dem Pfad eine Strecke (Geradenabschnitt) vom existierenden Start-Punkt P(x0 y0) zum Punkt P'(X Y) in User-Space-Koordinaten hinzu.
RelLineTo ( DX As Float, DY As Float )	Fügt dem Pfad eine Strecke (Geradenabschnitt) vom Start-Punkt P(x0 y0) zum aktuellen Punkt P'(x0+DX   y0+DY) in User-Space-Koordinaten hinzu.

Methoden	Beschreibung
Rectangle ( X As Float, Y As Float, Width As Float, Height As Float [ , Radius As Float ] )	Fügt für das Rechteck der vorgegebenen Größe einen geschlossenen Unter-Pfad zum bestehenden Pfad in User-Space-Koordinaten hinzu.
FillRect ( X As Float, Y As Float, Width As Float, Height As Float, Color As Integer )	Zeichnet das durch Startpunkt sowie Länge und Höhe definierte Rechteck und füllt es mit der angegebenen Farbe.
Polygon ( Points As Float[] )	Fügt für das Polygon mit den vorgegebenen Punkten einen geschlossenen Unter-Pfad zum bestehenden Pfad hinzu. <i>Points</i> ist ein Float-Array, wobei jedes Paar von Werten einen Punkt des Polygons beschreibt. Folglich muss die Anzahl der Elemente im Float-Array gerade sein. Die Mindestanzahl ist vier. Das Polygon wird automatisch geschlossen.
Ellipse ( X As Float, Y As Float, Width As Float, Height As Float [ , Angle As Float, Length As Float, Pie As Boolean ] )	Fügt dem Pfad einen elliptischen Bogen hinzu. Die Ellipse wird durch ihren Begrenzungsrahmen (X, Y, Breite, Höhe) definiert. Der Ellipsenbogen beginnt mit dem Start-Winkel 'Angle' und verläuft gegen den Uhrzeigersinn mit dem Winkel 'Length'. Die Bezugsachse für 0 Grad ist die positive x-Achse. Alle Winkelangaben müssen im Bogenmaß erfolgen. Setzen Sie die Rad-Funktion ein, um ein Gradmaß in das Bogenmaß zu konvertieren.
Arc ( XM As Float, YM As Float, Radius As Float [ , Angle As Float, Length As Float, Pie As Boolean ] )	Fügt dem Pfad einen Kreisbogen mit dem gegebenen Radius hinzu. Der Bogen wird bei (XM, YM) zentriert und beginnt mit dem (Start-) Winkel Angle und dreht den Zentriwinkel gegen den Uhrzeigersinn mit dem Winkel Length. Die Bezugsachse für 0 Grad ist die positive x-Achse. Alle Winkelangaben müssen im Bogenmaß erfolgen. Setzen Sie die Rad-Funktion ein, um ein Gradmaß in das Bogenmaß zu konvertieren.
CurveTo ( X1 As Float, Y1 As Float, X2 As Float, Y2 As Float, X3 As Float, Y3 As Float )	Fügt eine kubische Bezier-Kurve (Spline) von der aktuellen Position bis zur Position (X3 Y3) in User-Space-Koordinaten dem Pfad hinzu, wobei (X1 Y1), (X2 Y2) und (X3 Y3) Stütz-Punkte sind. Nach diesem Aufruf wird der aktuelle Punkt (X3 Y3) sein.
RelCurveTo ( X1 As Float, Y1 As Float, X2 As Float, Y2 As Float, X3 As Float, Y3 As Float )	Fügt, genau wie CurveTo(), eine kubische Bezier-Kurve hinzu. Hier werden aber alle Argumente als Offsets relativ zum aktuellen Punkt aufgefasst.
Stroke ( [ Preserve As Boolean ] )	Der (Linien-)Pfad wird mit der aktuellen Linien-Definition (Strichstärke, Strich-Form und Linien-Endform) nachgezeichnet. Der Pfad wird nach Paint.Stroke() gelöscht – es sei denn, das Preserve-Argument ist auf den Wert TRUE gesetzt.
Fill ( [ Preserve As Boolean ] )	Füllt die durch den aktuellen Pfad begrenzte Fläche anhand der aktuellen Flächen-Definition (FillRule). Jeder Teil-Pfad wird vorher automatisch geschlossen. Nach Paint.Fill() wird der Pfad gelöscht – aber nur dann, wenn das optionale Preserve-Argument nicht auf TRUE gesetzt wurde.

### 23.3.2.4 Methoden – Text und Bilder

Methoden	Beschreibung
Text ( Text As String [ , X As Float, Y As Float, Width As Float, Height As Float, Alignment As Integer ] )	Fügt den angegebenen Text zum Pfad hinzu. Es wird der aktuelle Font benutzt, der über die Font-Eigenschaft gesetzt ist.
TextSize ( Text As String ) As RectF	Gibt den Begrenzungsrahmen (Datentyp RectF (mit Koordinaten vom Typ Float)) zurück, der durch die Textzeichenfolge benötigt wird. Im Gegensatz zu TextExtents wird die Font-Schriftgröße verwendet – nicht die Höhe des tatsächlich gezeichneten Textes.
TextExtents ( Text As String ) As PaintExtents	Ermittelt die Ausmaße (Datentyp PaintExtents) für die Textzeichenfolge. Die Ausmaße beschreiben ein User-Space-Rechteck, das den tatsächlich gezeichneten Text umschließt, wie es von Paint.Text und Paint.Fill ohne Transformationsmatrix erstellt worden wäre.
TrimText ( Text As String, W As Float [ , H ] )	Gibt einen eingekürzten Text zurück, der in das angegebene Recht-

Methode	Beschreibung
<code>As Float ] ) As String</code>	eck passt. Wenn der Text von vornherein in das Rechteck passt, wird er unverändert zurückgegeben. Sonst wird er gekürzt und mit einem Auslassungszeichen ergänzt, so dass er passt.
<code>RichText ( Text As String [ , X As Float, Y As Float, Width As Float, Height As Float, Alignment As Integer ] )</code>	Fügt den angegebenen RichText zum Pfad hinzu. Es wird der aktuelle Font benutzt, der über die Font-Eigenschaft gesetzt ist.
<code>RichTextSize ( Text As String [ , Width As Float ] ) As RectF</code>	Gibt den Begrenzungsrahmen (Datentyp RechteckF (mit Koordinaten vom Typ Float)) zurück, der durch die RichText-Zeichenfolge benötigt wird. Im Gegensatz zu RichTextExtents wird die Font-Schriftgröße verwendet - nicht der tatsächlich gezeichnete Text.
<code>RichTextExtents ( Text As String [ , Width As Float ] ) As PaintExtents</code>	Ermittelt die Ausmaße (Datentyp PaintExtents) für die RichText-Zeichenfolge. Die Ausmaße beschreiben ein User-Space-Rechteck, das den tatsächlich gezeichneten Text umschließt, wie es von Paint.RichText und Paint.Fill ohne Transformationsmatrix erstellt worden wäre. Da nur ein Pfad erstellt wird, werden durch die <font>-Auszeichnung definierte Schrift-Farben <u>nicht</u> berücksichtigt!
<code>TrimRichText ( RichText As String, W As Float [ , H As Float ] ) As String</code>	Gibt eine gekürzte Version des gegebenen RichTexts zurück, die in das gegebene Rechteck passt. Wenn der RichText hinein passt, wird er unverändert zurückgegeben. Sonst wird er eingekürzt und um ein Auslassungszeichen ergänzt, so dass er passt.
<code>DrawImage ( Image As Image, X As Float, Y As Float [ , Width As Float, Height As Float, Opacity As Float, Source As Rect ] )</code>	Zeichnet ein Bild oder einen Teil eines Bildes. Dabei geben die X-, Y-, Width- und Height-Argumente die Position und maximale Ausdehnung des Bildes auf der Zeichenfläche an. Im Source-Argument kann das gewünschte Teilbild aus dem übergebenen Image als Rechteck spezifiziert werden.
<code>DrawPicture ( Picture As Picture, X As Float, Y As Float [ , Width As Float, Height As Float, Source As Rect ] )</code>	Zeichnet ein Bild oder einen Teil eines Bildes. Dabei geben die X-, Y-, Width- und Height-Argumente die Position und maximale Ausdehnung des Bildes auf der Zeichenfläche an. Im Source-Argument kann das gewünschte Teilbild aus dem übergebenen Picture als Rechteck spezifiziert werden.
<code>DrawText ( Text As String [ , X As Float, Y As Float, Width As Float, Height As Float, Alignment As Integer ] )</code>	Zeichnet den angegebenen Text. Wenn Sie die optionalen Parameter angeben, so wird der Text durch das angegebene Rechteck begrenzt und nach dem Alignment-Parameter ausgerichtet. Die Methode ist schneller, als den Text mit Paint.Text und anschließendem Paint.Fill zu zeichnen.
<code>DrawTextShadow ( Text As String, X As Float, Y As Float, W As Float, H As Float [ , Alignment As Integer, Radius As Integer, Opacity As Float ] )</code>	Zeichnet den Schatten eines Textes.
<code>DrawRichText ( Text As String [ , X As Float, Y As Float, Width As Float, Height As Float, Alignment As Integer ] )</code>	Zeichnet den angegebenen Rich-Text. Wenn Sie die optionalen Parameter angeben, so wird der Text durch das angegebene Rechteck begrenzt und nach dem Alignment-Parameter ausgerichtet. Die Methode ist schneller, als den Text mit Paint.RichText und anschließendem Paint.Fill zu zeichnen.
<code>DrawRichTextShadow ( Text As String, X As Float, Y As Float, W As Float, H As Float [ , Alignment As Integer, Radius As Integer, Opacity As Float ] )</code>	Zeichnet den Schatten eines RichTextes.
<code>Image ( Image As Image [ , X As Float, Y As Float ] ) As PaintBrush</code>	Erstellt einen neuen Pinsel aus einem Bild. Image ist das Bildobjekt und X sowie Y sind die (optionalen) Werte (Datentyp Float) aus der Brush-Matrix und geben die initiale Translation des Bildes an.

Tabellen 23.3.2.4.1 : Methoden der Klasse Paint

Die folgenden Hinweise ergänzen die Inhalte der Tabellen. Es werden Argumente näher beschrieben und auf Besonderheiten hingewiesen. Beispiele und vollständige Projekte dagegen finden Sie in den folgenden beiden → Kapiteln 23.3.3 *Projekte 1* und 23.3.4 *Projekte 2*.

- ClosePath  
ClosePath() ist verschieden von einem LineTo() mit den geeigneten Koordinaten. Wenn ein (mit der ersten Methode) geschlossener Pfad mit Stroke() gezeichnet wird, gibt es keine Kappen (Paint.LineCap) an den Linien-Enden, sondern die Linien werden zusammengeführt (Paint.LineJoin). Gibt es keinen aktuellen Punkt wenn ClosePath() aufgerufen wird, bewirkt diese Methode nichts.

- Kreis und Kreisbogen

Die Hinweise für einen Kreisbogen sind einerseits sehr umfangreich und andererseits zum Teil nur in bestimmten *Kombinationen* gültig:

- XM : x-Koordinate des Zentrums des Bogens
- YM : y-Koordinate des Zentrums des Bogens
- Radius : Radius des Kreisbogens
- Angle : Startwinkel - gemessen gegen die positive x-Achse
- Length : Maß für den Zentriwinkel
- Pie : Für True ergibt sich ein geschlossener Sektor ('Tortenstück'). Standard ist ein offener Bogen.
- Wenn Length *negativ* ist, so wird der Bogen im Uhrzeigersinn gezeichnet.
- Werden Angle und Length nicht angegeben, dann wird ein Vollkreis gezeichnet.
- Wenn der Wert Angle festgelegt ist, aber nicht Length, dann wird nichts gezeichnet. Die aktuelle Position wird auf den (End-)Punkt auf dem Kreisbogen eingestellt.
- Um einen elliptischen Bogen statt eines Kreisbogens zu zeichnen, müssen Sie die Ellipse-Methode mit geeigneten Argumenten verwenden.

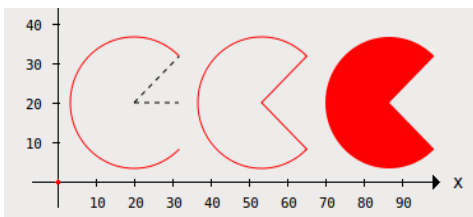


Abbildung 23.3.2.4.1: Kreisbogen - Kreissektoren

Quelltext:

```
Paint.AntiAlias = True
Paint.LineWidth = 1
Paint.Brush = Paint.Color(Color.Red)
Paint.Arc(60, 60, 50, Rad(45), 3 * Pi / 2, False)
Paint.Stroke
Paint.Arc(160, 60, 50, Pi / 4, Rad(270), True)
Paint.Stroke
Paint.Arc(260, 60, 50, Pi / 4, 3 * Pi / 2, True)
Paint.Fill
Paint.AntiAlias = False
```

- Abgerundete Rechtecke  
Wenn das optionale Radius-Argument definiert ist, so wird das Rechteck abgerundete 'Ecken' besitzen. Das Argument *Radius* bestimmt den Radius der Rundung. Der Radius darf nicht größer als die Hälfte der *kürzesten* Seite des Rechtecks sein.

- Polygon – Dreieck  
Mit diesem Quelltext wird ein hellblaues Dreieck ABC gezeichnet:

```
Paint.LineWidth = 1
Paint.Brush = Paint.Color(&C3DDFF) ' hellblau
Paint.Polygon([20, 20, 30, 160, 200, 70]) ' A(20|20), B(30|160) und C(200|70)
Paint.Fill
```

- Ellipsen

```
Ellipse ( X As Float, Y As Float, Width As Float, Height As Float [ , Angle As Float, Length As Float, Pie As Boolean ] )
```

- X, Y, Width, Height : Begrenzungsrahmen der Ellipse
- Angle : Startwinkel
- Length : Überstrichener Winkel
- Pie : Für True ergibt sich ein geschlossener Sektor. Standard ist ein offener Bogen.
- Wenn Length negativ ist, so wird der Ellipsen-Bogen im Uhrzeigersinn gezeichnet.
- Werden Angle und Length nicht angegeben, so wird eine vollständige Ellipse gezeichnet.
- Wenn der Wert Angle festgelegt ist – aber nicht Length – dann wird nichts gezeichnet. Die aktuelle Position wird auf den (End-)Punkt auf dem Ellipsen-Bogen eingestellt.

■ Kubische Bezier-Kurve

```
CurveTo ( X1 As Float, Y1 As Float, X2 As Float, Y2 As Float, X3 As Float, Y3 As Float )
```

- X1 : x-Koordinate des ersten Stütz-Punktes (Start-Punkt)
- Y1 : y-Koordinate des ersten Stütz-Punktes
- X2 : x-Koordinate des zweiten Stütz-Punktes
- Y2 : y-Koordinate des zweiten Stütz-Punktes
- X3 : x-Koordinate des dritten Stütz-Punktes (Endpunkt)
- Y3 : y-Koordinate des dritten Stütz-Punktes

Gab es keinen aktuellen Punkt vor dem Aufruf von Paint.CurveTo(..), so wird sich die Funktion so verhalten, als ob dem Aufruf die Methode Paint.MoveTo(X1|Y1) vorausging, um einen definierten Anfang zu sichern.

■ Kubische Bezier-Kurve mit relativen Koordinaten – Version von CurveTo(..)

```
RelCurveTo ( X1 As Float, Y1 As Float, X2 As Float, Y2 As Float, X3 As Float, Y3 As Float )
```

- DX1 : x-Offset des ersten Stütz-Punktes (Start-Punkt)
- DY1 : y-Offset des ersten Stütz-Punktes
- DX2 : x-Offset des zweiten Stütz-Punktes
- DY2 : y-Offset des zweiten Stütz-Punktes
- DX3 : x-Offset des dritten Stütz-Punktes (Endpunkt)
- DY3 : y-Offset des dritten Stütz-Punktes

Paint.CurveTo(X+DX1, Y+DY1, X+DX2, Y+DY2, X+DX3, Y+DY3) ist logisch äquivalent zum Aufruf Paint.RelCurveTo (DX1, DY1, DX2, DY2, DX3, DY3) bei einem aktuellen Punkt (X|Y).

■ Text

```
Text(Text As String [, X As Float, Y As Float, Width As Float, Height As Float, Alignment As Integer ] )
```

- X, Y, Width, Height : ein Rechteck, in dem der Text gezeichnet wird
- Alignment : Ausrichtung des Textes im Rechteck
- Ist nur der Text angegeben, so wird der Text am aktuellen Punkt gezeichnet. Die y-Koordinate des aktuellen Punktes bestimmt die BaseLine für den Text.
- Der Begriff 'BaseLine' lässt sich sehr einfach am Wort *Ärger* beschreiben. Ziehen Sie gedanklich eine Linie vom Fußpunkt des Buchstabens A zum Fußpunkt des kleinen r – das ist die BaseLine.
- Sind neben dem Text nur die Argumente X und Y angegeben, dann wird der Text am Punkt P(X|Y) gezeichnet. Die y-Koordinate von P bestimmt die BaseLine für den Text.
- Andernfalls wird der Text innerhalb des angegebenen Rechtecks gezeichnet. Beachten sie, dass das Rechteck den Text nicht abschneidet!

■ RichText

```
RichText ( Text As String [ , X As Float, Y As Float, Width As Float, Height As Float, Alignment As Integer ] )
```

- Für die optionalen Argumente gelten die gleichen Hinweise wie für die Text()-Methode.
- Da nur ein Pfad erstellt wird, werden durch die *<font>*-Auszeichnung definierte Schrift-Farben nicht berücksichtigt!

## TrimText – TrimRichText

```
TrimText ( Text As String, W As Float [ , H As Float ] ) As String
```

```
TrimRichText ( RichText As String, W As Float [ , H As Float ] ) As String
```

- Text : Text, der anzupassen ist
- W : Länge des (umschließenden) Rechtecks
- H : optionale Höhe des (umschließenden) Rechtecks. Wenn sie nicht angegeben ist, so wird von der Schriftgröße ausgegangen. Mit anderen Worten: Der Text soll in eine einzige Zeile passen.

## DrawImage

```
DrawImage ( Image As Image, X As Float, Y As Float [ , Width As Float, Height As Float, Opacity As Float, Source As Rect ] )
```

- Image : Bild, das gezeichnet werden soll
- X, Y : Koordinaten des Anfangspunktes (oben links)
- Width, Height : Wenn angegeben, so wird das Bild auf Breite und Höhe skaliert
- Opacity : Die Deckkraft, die für die Erstellung des Bildes verwendet wird, liegt zwischen 0 (völlig transparent) und 1,0 (völlig undurchsichtig). Standardmäßig liegt der Wert bei 1.
- Source : Wenn angegeben (Datentyp Rect), so wird nur der durch das Rechteck beschriebene Bild-Teil berücksichtigt.

## DrawPicture

```
DrawPicture ( Picture As Picture, X As Float, Y As Float [ , Width As Float, Height As Float, Source As Rect ] )
```

- Picture : Bild, das gezeichnet werden soll
- X, Y : Koordinaten des Anfangspunktes
- Width, Height : Wenn angegeben, so wird das Bild auf Breite und Höhe skaliert
- Source : Wenn angegeben (Datentyp Rect), so wird nur der durch das Rechteck beschriebene Bild-Teil berücksichtigt.

## DrawTextShadow

```
DrawTextShadow ( Text As String, X As Float, Y As Float, W As Float, H As Float [ , Alignment As Integer, Radius As Integer, Opacity As Float ] )
```

```
DrawRichTextShadow ( Text As String, X As Float, Y As Float, W As Float, H As Float [ , Alignment As Integer, Radius As Integer, Opacity As Float ] )
```

- X, Y, W, H : Definieren das Rechteck, in dem gezeichnet wird
- Alignment : Bestimmt die Ausrichtung des Textes, Standard ist Align.Left
- Radius : Überblendradius in Pixeln. Standardmäßig ist der Wert 1/8 der Schrifthöhe
- Opacity : Die Deckkraft liegt zwischen 0,0 und 1,0. Standardmäßig beträgt der Wert 0,5 (50% transparent)
- Sie müssen Paint.DrawText() oder Paint.RichText() aufrufen, um den eigentlichen Text zum Schatten zu zeichnen.

## LinearGradient

```
LinearGradient ( X0 As Float, Y0 As Float, X1 As Float, Y1 As Float, Colors As Integer[], Positions As Float[] [ , Extend As Integer ] ) As PaintBrush
```

- X0 : x-Koordinate Startpunkt
- Y0 : y-Koordinate Startpunkt
- X1 : x-Koordinate Endpunkt
- Y1 : y-Koordinate Endpunkt
- Durch die Punkte (X0, Y0) und (X1, Y1) wird eine Strecke der Länge L definiert, die beide Punkte verbindet.
- Colors : Array von Farbwerten
- Positions : Array von Farb-Positionen; 0 für Gradient-Start und 1 für Gradient-Ende. Eine

Position  $q$  (Float) zwischen 0 und 1 entspricht dem Punkt  $Q$  auf der definierten Strecke, der  $q \cdot L$  weit vom Startpunkt und  $(1-q) \cdot L$  weit vom Endpunkt entfernt ist. Eine Position im Positions-Array wird auch Stopper genannt. Zu einem Stopper gehört die Farbe mit gleichem Index aus dem Colors-Array. Am Stopper hat diese Farbe die größte Intensität.

- Die folgende Abbildung zeigt einen LinearGradient von (0|0) nach (200|70) über die Farben Schwarz, Gelb und Blau. Die Stopper sind bei 0 und 0,3 sowie 1,0 als rote Punkte eingezeichnet.

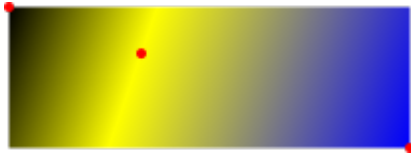


Abbildung 23.3.2.4.2: LinearGradient

### RadialGradient

```
RadialGradient ( CX As Float, CY As Float, Radius As Float, FX As Float, FY As Float, Colors As Integer[], Positions As Float[] [ , Extend As Integer ] ) As PaintBrush
```

- CX : x-Koordinate Kreis-Mittelpunkt
- CY : y-Koordinate Kreis-Mittelpunkt
- Radius : Radius des Kreises
- FX : x-Koordinate für den Gradienten-Mittelpunkt
- FY : y-Koordinate für den Gradienten-Mittelpunkt
- Colors : Array von Farbwerten (Stop)
- Positions : Array von Farb-Positionen; 0 für Gradient-Start und 1 für Gradient-Ende