

23.4.4 Klasse PictureBox

Die Klasse PictureBox (gb.qt4) implementiert ein Steuerelement zur Anzeige eines Pictures → Kapitel 23.4.3 Klasse Picture.

23.4.4.1 Eine neue PictureBox erzeugen

```
Dim hPictureBox As PictureBox
hPictureBox = New PictureBox ( Parent As Container ) As "event name"
```

Es wird eine neue PictureBox erzeugt. Der Parameter *Parent* gibt den Container an, in den die PictureBox eingefügt wird.

23.4.4.2 Eigenschaften

Interessant für den Einsatz einer PictureBox sind die folgenden, ausgewählten Eigenschaften:

Eigenschaft	Datentyp	Beschreibung
Alignment	Integer	Gibt den Wert der Ausrichtung eines Pictures in der PictureBox zurück oder setzt diesen Wert. Die Standard-Ausrichtung ist 'TopLeft'. Eine Liste der Konstanten zur Festlegung der Ausrichtung finden Sie im → Kapitel 20.7.1 Klasse Align.
Border	Integer	Gibt den spezifischen Rand um eine PictureBox zurück oder setzt diesen Wert. Im Kapitel 23.2.1 Klasse Border finden Sie eine Liste der Rand-Konstanten. Der Standard-Wert ist 'Border.None'.
Padding	Integer	Gibt den Padding-Wert zurück oder setzt den inneren Abstand (Pixel) um das Picture. Der Standard-Wert ist 0 Pixel.
Picture	Picture	Gibt das Picture zurück, das in der PictureBox angezeigt wird oder setzt das anzuzeigende Picture.
Stretch	Boolean	Gibt an, ob das anzuzeigende Picture an die Breite und Höhe der PictureBox angepasst werden soll oder setzt diese Eigenschaft.

Tabelle 23.4.4.2.1: Ausgewählte Eigenschaften der Klasse PictureBox

23.4.4.3 Projekt

Im Projekt wird ein Bild in einer PictureBox angezeigt. Die Pfade ausgewählter Bilder werden in einem Dialog (MulitSelect) von einem Datenträger geladen und in einem Array gespeichert.



Abbildung 23.4.4.3.1: Anzeige eines geladenen Bildes in einer PictureBox

Quelltext-Ausschnitt:

```
[1] ' Gambas class file
[2]
[3] Private sImagePaths As String[]
[4] Private iPictureIndex As Integer
```

```

[5]
[6] Public Sub Form_Open()
[7]     FMain.Center
[8]     FMain.Utility = True ' Minimal-Fenster
[9]     FMain.Text = "Dialog.OpenFile(True) mit Multi-Select"
[10]
[11]     PictureBoxD.W = 512 ' Startwert nur für Bildformat 4:3
[12]     PictureBoxD.H = 384 ' Startwert nur für Bildformat 4:3
[13]     PictureBoxD.Border = Border.Solid
[14]     PictureBoxD.Alignment = Align.Center
[15]     PictureBoxD.Stretch = True
[16]     PictureBoxD.Padding = 6
[17]     PictureBoxD.Background = &FFFFFF
[18]     PictureBoxD.Picture = Picture["Symbols/intro.jpg"]
[19]
[20]     SetEnabled()
[21] End ' Form_Open()
[22]
[23] Public Sub btnOpenFileImage_Click()
[24]     OpenFileImage()
[25] End ' btnOpenFileImage_Click()
[26]
[27] Public Sub btnNext_Click()
[28]     GoNext()
[29] End ' btnNext_Click()
[30]
[31] Public Sub btnPrevious_Click()
[32]     GoPrevious()
[33] End ' btnPrevious_Click()
[34]
[35] Public Sub btnClose_Click()
[36]     PictureBoxD.Picture = Null
[37]     Wait 0.5 ' Nur zur Sicht-Kontrolle: Leeren der PictureBox
[38]     FMain.Close
[39] End ' btnClose_Click()
[40]
[41] '-----
[42]
[43] Private Sub OpenFileImage()
[44]     Dim sMessage1, sMessage2 As String
[45]
[46]     Dialog.Title = "Wählen Sie Bild-Dateien aus (Bilder-Liste)..."
[47]     Dialog.Filter = [ "*.png;*.jpg;*.gif;*.jpeg;*.svg;*.ico", " Bild-Dateien ", " * ", " Alle Dateien " ]
[48]     Dialog.ShowHidden = False
[49]     Dialog.Path = Application.Path & "/ Images"
[50]
[51] ' Bilder auf einem Datenträger auswählen (True -> Multiselect aktiviert)
[52] If Dialog.Openfile(True) Then
[53]     Return
[54] Endif
[55]
[56] SetEnabled()
[57] sImagePath = New String[] ' Neue Bild-Liste
[58] sImagePaths = Dialog.Paths ' Bild-Liste sichern (Navigation)
[59] iPictureIndex = 0
[60] PictureBoxD.Picture = Picture.Load(Dialog.Paths[0]) ' Anzeige des ersten Bildes
[61]
[62] sMessage1 = "Die Bild-Liste enthält genau " & Dialog.Paths.Count & " Bild"
[63] sMessage2 = "Die Bild-Liste enthält " & Dialog.Paths.Count & " Bilder"
[64] FMain.Text = IIf(Dialog.Paths.Count = 1, sMessage1, sMessage2)
[65]
[66] ' Einschalten der Bild-Navigation, wenn die Bild-Liste mehr als ein Bild enthält
[67] If Dialog.Paths.Count > 1 Then
[68]     btnNext.Enabled = True
[69] Else
[70]     btnNext.Enabled = False
[71] Endif
[72]
[73] Catch
[74]     Message.Info(Error.Text)
[75] End ' OpenFileImage()
[76]
[77] Private Sub GoNext()
[78]     If iPictureIndex < sImagePaths.Count
[79]         Inc iPictureIndex
[80]         If iPictureIndex = 1 Then btnPrevious.Enabled = True
[81]         If iPictureIndex = sImagePaths.Max Then
[82]             PictureBoxD.Picture = Picture.Load(sImagePaths[iPictureIndex])
[83]             btnNext.Enabled = False
[84]             Return
[85]         Endif
[86]         PictureBoxD.Picture = Picture.Load(sImagePaths[iPictureIndex])
[87]     Endif
[88] End ' GoNext()

```

```

[89]
[90] Private Sub GoPrevious()
[91]     If iPictureIndex < sImagePaths.Count
[92]         Dec iPictureIndex
[93]         If iPictureIndex = sImagePaths.Max - 1 Then btnNext.Enabled = True
[94]         If iPictureIndex = 0 Then
[95]             PictureBoxD.Picture = Picture.Load(sImagePaths[iPictureIndex])
[96]             btnPrevious.Enabled = False
[97]             Return
[98]         Endif
[99]         PictureBoxD.Picture = Picture.Load(sImagePaths[iPictureIndex])
[100]     Endif
[101] End ' GoPrevious()
[102]
[103] Private Sub SetEnabled()
[104]     btnPrevious.Enabled = False
[105]     btnNext.Enabled = False
[106] End ' SetEnabled()

```

Kommentar:

- In der Zeile 8 wird die *Utility-Eigenschaft* von Form auf True gesetzt. Damit wird gesichert, dass das Fenster keine kleineren Maße hat als die Form in der Gambas-IDE.
- Erprobte Eigenschaften für die PictureBoxD werden in den Zeilen 11 bis 18 festgelegt.
- Mit *PictureBoxD.Alignment = Align.Center* und *PictureBoxD.Stretch = True* wird das anzuzeigende Bild vom Zentrum aus an die Weite und Höhe der PictureBoxD angepasst und angezeigt. Achtung: Die Streckung erfolgt nicht maßstabsgetreu!
- Mit den Eigenschaften *PictureBoxD.Padding = 6* und *PictureBoxD.Background = &FFFFFF* wird ein schmaler weißer Rand um das anzuzeigende Picture gelegt.
- Formate für die Bilder (PNG, JPEG, JPG, GIF, ICO, SVG, BMP und XPM), die sicher in ein Picture konvertiert werden können, erkennen Sie auch im Dialog-Filter in der Zeile 46. Animierte GIF-Bilder zum Beispiel werden nicht animiert dargestellt. Das gelingt nur in einer MovieBox → Kapitel 23.4.5 MovieBox.
- Vor der Anzeige wird das aktuelle Bild vom Datenträger geladen und der Eigenschaft *PictureBoxD.Picture* zugewiesen → Zeilen 60, 82, 86, 95 und 99.
- Damit keine weiße Fläche in der PictureBoxD angezeigt wird beim Programmstart, wird ein dezentes Startbild eingefügt → Zeile 18:



Abbildung 23.4.4.3.2: Startbild (intro.jpg)

23.4.4.4 PictureBox-Inhalt löschen

Die Eigenschaft *Picture* von PictureBox gibt nur eine Kopie des internen Pictures zurück. Das heißt, dass

```
PictureBox.Picture
```

ein anderes Picture-Objekt ist als jenes, das in der PictureBox dargestellt wird – aber mit gleichem Inhalt. Wenn Sie die Kopie mit ihrer *Clear()*-Methode leeren, so hat das keine Auswirkungen auf das Picture *in* der PictureBox. Das bedeutet, dass die Anweisung

```
PictureBox.Picture.Clear()
```

nicht ausreicht, um die PictureBox zu leeren. Das erledigen Sie mit:

```
PictureBox.Picture = Null .
```