

16.3 Button (gb.qt4)

Diese Klasse implementiert einen Button – eine Taster-Schaltfläche. Ein Button kann einen Text, ein Bild oder beides anzeigen:



Abbildung 16.3.1: Drei Button im Projekt GeoMap → Kapitel 17.15.1

16.3.1 Eigenschaften

- Verwendet man in der Beschriftung (Eigenschaft Text oder synonym Caption) ein &-Zeichen, dann fügt man einen so genannten ShortCut ein. Die Tastenkombination *ALT+Buchstabe*, der dem &-Zeichen folgt, löst die Aktion aus, die auch bei einem Klick auf den Button ausgelöst worden wäre.
- Ein Button in einem Programm-Fenster kann über die Eigenschaft *Button.Default = True* als Default-Button definiert werden. Der Default-Button wird automatisch aktiviert, wenn die RETURN-Taste gedrückt wird.
- Wenn für einen Button die Eigenschaft *Button.Cancel* auf True gesetzt wird, dann wird das Drücken der ESC-Taste diesen Button automatisch aktivieren.

Button	Datentyp	Default	Beschreibung
.AutoSize	Boolean	False	Gibt an, ob sich die Größe des Buttons automatisch an den Text anpasst
.Caption	String	Null	Synonym für .Text
.Text	String	Null	Setzt den anzuzeigenden Text auf dem Button oder gibt diesen zurück
.Value	Boolean	False	Wird der Wert auf True gesetzt, dann wird der Button aktiviert – das Klick-Ereignis wird simuliert.
.Picture	Picture	Null	Setzt das anzuzeigende Bild auf dem Button oder gibt dieses Bild zurück
.Tag	Variant	Null	Setzt die Eigenschaft oder liest den Wert aus. Die Tag-Eigenschaft kann verwendet werden, um eine nutzerspezifische Information in einem Objekt zu speichern.
.ToolTip	String	Null	Setzt den anzuzeigenden ToolTip-Text des Buttons oder gibt diesen zurück

Tabelle 16.3.1.1: Ausgewählte Button-Eigenschaften

16.3.2 Ereignisse – Events

Für einen Button ist das Klick-Ereignis das dominante Ereignis. Es wird ausgelöst, wenn der Benutzer mit der Maus auf den Button klickt.

```
' Diese Funktion liefert die Namen der Klassen mit der gesuchten Eigenschaft in einer Matrix
Private Function SearchClasses(sProperty As String) As String[]
    Dim hClass As Class
    Dim aNamesMatrix As New String[]

    For Each hClass In Classes
        If hClass.Symbols.Exist(sProperty, gb.IgnoreCase) Then aNamesMatrix.Add(hClass.Name)
    Next
    Return aNamesMatrix.Sort() ' Rückgabe des sortierten Arrays der Namen
End ' Function SearchClasses(..)

Public Sub btnAnzeigeLast_Click()
    Dim sElement As String

    TextArea.Clear
    For Each sElement In SearchClasses("PopupMenu")
        TextArea.Insert(sElement & ", ")
    Next
End ' btnAnzeigeLast_Click()
```

Erweiterungen existieren, wenn die Eigenschaften *Button.Default*, *Button.Cancel* oder *Button.Value* auf den Wert *True* gesetzt worden sind → Eigenschaften.

16.3.3 Beispiel 1

Beispiel 1 zeigt einen Button, mit dem drei unterschiedliche Aktionen (Starten, Stoppen und Reset) ausgelöst werden können. Dreh- und Angelpunkt ist hier die Verwendung der Tag-Eigenschaft. Ein Vorgang wird gestartet und die Zeit-Dauer in Sekunden hexadezimal angezeigt. Nach dem Stoppen kann die Zeit abgelesen werden und abschließend mit Reset der Startzustand gesetzt werden:



Abbildung 16.3.3.1: Kurzzeit-Messung mit hexadezimaler Anzeige

Der Quelltext enthält nur die relevanten Abschnitte aus einem größeren Projekt:

```
[1] ...
[2]
[3] Private sSecond As Integer
[4]
[5] Public Sub Form_Open()
[6]     ...
[7]
[8]     btnStartStop.Tag = "start"
[9]     btnStartStop.Picture = Picture["icon:/16/play"]
[10]    btnStartStop.Caption = " Starten"
[11]    Timer1.Delay = 1000
[12]    LCDNumber1.Mode = 2 ' Hexadecimal
[13]
[14]    ...
[15] End ' Form_Open()
[16]
[17] Public Sub btnStartStop_Click()
[18]
[19]     Select Case btnStartStop.Tag
[20]     Case "start"
[21]         btnStartStop.Picture = Picture["icon:/16/stop"]
[22]         btnStartStop.Tag = "stop"
[23]         btnStartStop.Caption = " Stoppen"
[24]         Timer1.Start ' Aktion starten
[25]     Case "stop"
[26]         btnStartStop.Picture = Picture["icon:/16/redo"]
[27]         btnStartStop.Tag = "reset"
[28]         btnStartStop.Caption = " Reset"
[29]         Timer1.Stop ' Aktion stoppen
[30]     Case "reset"
[31]         btnStartStop.Picture = Picture["icon:/16/play"]
[32]         btnStartStop.Tag = "start"
[33]         btnStartStop.Caption = " Starten"
[34]         sSecond = 0
[35]         LCDNumber1.Value = 0
[36]     End Select ' btnStartStop.Tag
[37]
[38] End ' btnStartStop_Click()
[39]
[40] Public Sub Timer1_Timer()
[41]
[42]     Inc sSecond
[43]     LCDNumber1.Value = sSecond
[44]
[45] End ' Timer1_Timer()
```

16.3.4 Beispiel 2

Das Beispiel 2 zeigt das Projekt zur Demonstration der Komponente *gb.map* zur Anzeige von Karten von *openstreetmap.org*. Es steht aber nicht die Anzeige der Karten im Fokus, sondern der Einsatz von Eigenschaften, Methoden und des Klick-Ereignisses für die drei Button auf dem Formular → Abbildung 16.3.4.2.

Vom Quelltext (komplett) sind nur die Zeilen 1 bis 33 relevant für das Thema Button. Der Quelltext wird anschließend kommentiert:

```
[1] ' Gambas class file
[2]
[3] Public Sub Form_Open()
[4]   FMain.Center
[5]   FMain.Caption = "GeoMap - Daten von OpenStreetMap ©"
[6]
[7]   btnShowMap.Picture = Picture["icon:/16/internet"]
[8]   btnShowMap.Text = "  Anzeige GeoMap"
[9]   btnShowMap.Default = True
[10]  btnShowMap.SetFocus
[11]
[12]  btnClose.Caption = "&Beenden"
[13]
[14]  btnShowHelp.Text = Null
[15]  btnShowHelp.Tooltip = "Hinweise zur Nutzung von GeoMap"
[16]  btnShowHelp.Picture = Picture["icon:/16/help"]
[17]
[18]End ' Form_Open()
[19]
[20]Public Sub btnShowMap_Click()
[21]  If btnShowHelp.Tag = Null Then
[22]    btnShowHelp.Value = True
[23]    btnShowHelp.Tag = 1
[24]  Endif ' btnHelp.Tag = Null ?
[25]  ShowMap()
[26]End ' btnShowMap_Click()
[27]
[28]Public Sub btnShowHelp_Click()
[29]  HelpMessage()
[30]End ' btnShowHelp_Click()
[31]
[32]Public Sub btnClose_Click()
[33]  FMain.Close
[34]End ' btnClose_Click()
[35]
[36]*****
[37]Private Sub ShowMap()
[38]  Dim iZoom As Integer
[39]  Dim fLatitude, fLongitude As Float
[40]  Dim sCacheName, sTileName, sTilePattern As String
[41]  Dim cArguments As New Collection
[42]
[43]  iZoom = 13
[44]  fLatitude = 52.78631 ' °Breite Osterburg
[45]  fLongitude = 11.73872 ' °Länge Osterburg
[46]  sTileName = "OpenStreetMap"
[47]  sTilePattern = "85.30.190.241/{z}/{x}/{y}.png"
[48]  cArguments = Null
[49]  sCacheName = Null
[50]  MapView1.Map.AddTile(sTileName, sTilePattern, cArguments, sCacheName)
[51]  MapView1.AllowEffect = False
[52]  MapView1.Map[sTileName].Visible = True
[53]  MapView1.Map.Center = MapPoint(fLatitude, fLongitude)
[54]  MapView1.Map.Zoom = iZoom
[55]
[56]End ' ShowMap
[57]
[58]Private Sub HelpMessage()
[59]  Dim sMessage As String
[60]
[61]  sMessage = "<hr><b>Hinweise zum Einsatz der Komponente Map</b><hr>"
[62]  sMessage &= "» Verschieben der Karte mit der linken Maustaste"
```

```
[63] sMessage &= "<br>» Zoomen mit dem Maus-Rad>"
[64] sMessage &= "<br>» Original-Zoom mit 'Anzeige GeoMap'<br>"
[65] sMessage &= "» Enter-Taste setzt Standard-Anzeige (Original-Zoom)"
[66] sMessage &= "<br>» Tastenkombination ALT+b beendet das Programm<br>"
[67] sMessage &= "■ Entwickler der Komponente gb.map ist Fabien Bodard"
[68] sMessage &= "<br>■ EMail: gambix@users.sourceforge.net<br>"
[69] Message.Info(sMessage)
[70]End ' HelpMessage()
```

Kommentare:

- Nach dem in den Zeilen 7 und 8 dem Button *btnShowMap* ein Icon aus dem Bestand von Gambas zugewiesen und die Beschriftung festgelegt wurde, wird in der Zeile 9 der Button *btnShowMap* zum Default-Button erklärt und zur Programmlaufzeit automatisch aktiviert, wenn Sie die RETURN-Taste drücken würden. Ob Sie die Return-Taste drücken oder auf den Button *btnShowMap* drücken zeigt die gleiche Wirkung – es wird eine geografischen Karte angezeigt.
- In der Zeile 12 wird ein ShortCut definiert, so dass auch die Tastenkombination *ALT+b* das Programm beendet – genauso wie ein Klick auf den Button *btnClose*.
- Die Tag-Eigenschaft des Buttons *btnShowMap* wird in den Zeilen 21 bis 24 genutzt, um dem Benutzer bereits vor der ersten Anzeige einer Karte die 'Hinweise zum Einsatz der Komponente Map' einmalig zu präsentieren.
- Die (automatische) Anzeige des Hilfetextes wird in der Zeile 22 realisiert, weil die Eigenschaft Value durch die Zuweisung *btnShowMap.Value = True* den Wert True erhält. Das Klick-Ereignis wird simuliert. Alternativ hätten Sie in der Zeile 22 auch *btnShowMap_Click()* eintragen können.
- Nach der ersten Anzeige des Hilfe-Textes ändert sich der Wert der Tag-Eigenschaft des Button *btnShowMap* in der Zeile 23 vom Wert Null auf 1. Damit werden weitere Anzeigen des Hilfe-Textes bei einem Klick auf den Button *btnShowMap* sicher unterbunden, weil die Abfrage in der Zeile 21 dann stets *False* liefert.



Abbildung 16.3.4.1: Programm-Hilfe

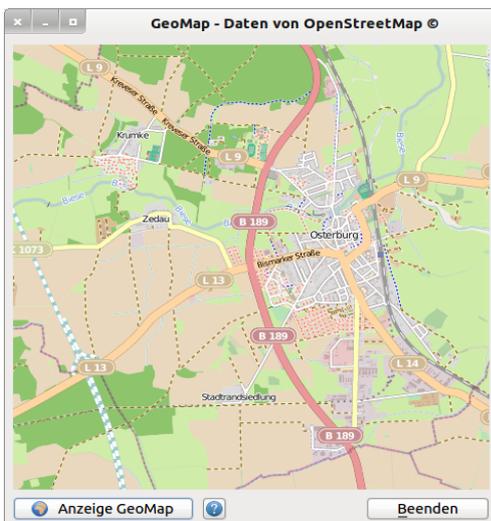


Abbildung 16.3.4.2: Projekt GeoMap

Hinweis:

Alle Message-Boxen als (Unter-)Programmfenster können bis zu drei Button besitzen. Dabei gilt:

- Die erste Taste ist immer der Default-Button → Return-Taste.
- Der letzte Button ist immer der Cancel-Button → Escape-Taste.

Wenn die Funktion nur mit einem Argument aufgerufen, dann gibt es nur den "OK"-Button.