

20.5 Observer

Die Klasse *Observer* (gb) implementiert ein Objekt, das ein anderes Objekt beobachten kann, in dem es dessen Ereignisse abfängt.

20.5.1 Eigenschaften

Die Klasse *Observer* besitzt diese beiden Eigenschaften – *Object* und *Tag*.

- Die markante Eigenschaft *Observer.Object* vom Typ *Object* gibt das beobachtete Objekt zurück.
- Die *Tag*-Eigenschaft kann ausgelesen oder gesetzt werden. Über die Verwendung dieser Eigenschaft kann der Programmierer frei verfügen.

20.5.2 Erzeugung eines Observer-Objekts

Die folgende Anweisung generiert einen neuen *Observer* für das angegebene Objekt:

```
Dim hObserver As Observer
...
hObserver = New Observer ( ObjectA As Object [ , After As Boolean ] ) As EventName
```

- ObjectA* ist das zu beobachtende Objekt.
- Normalerweise bekommt ein *Observer* das Event eines Objekts *vor* dessen Default-*Observer*. Wenn Sie den optionalen Parameter *After* auf *False* setzen (Standard), dann ist es möglich, alle Ereignisse von *ObjectA* abzufangen, bevor sie tatsächlich ausgelöst werden. Das eröffnet die Möglichkeit, dass der *Observer* optional auch ein Ereignis abbrechen kann, um zu vermeiden, dass ein Objekt ein Ereignis überhaupt auslöst.
- Setzen Sie jedoch den optionalen Parameter *After* auf *True*, so wird der *Observer* Ereignisse des beobachteten Objekts abfangen, nachdem sie verarbeitet wurden. In diesem Fall sind Sie zum Beispiel nicht mehr in der Lage, Ereignisse des zu beobachtenden Objekts auszuwerten oder abzubrechen.
- Für jedes abgefangene Ereignis wird das *Observer*-Objekt ein Ereignis mit dem Präfix *EventName_* mit dem gleichen Namen und den gleichen Argumenten auslösen → Projekt 1, 2 und 3.
- Das erzeugte *Observer*-Objekt *hObserver* ist an das zu beobachtende Objekt gebunden und wird nur dann freigegeben, wenn das beobachtete Objekt freigegeben wird.

20.5.3 Projekt 1

Im Projekt 1 wird ein ausgewähltes Ereignis von einem Button zeitweise beobachtet. Ausgaben in der Konsole der Gambas-IDE unterstützen das Verständnis für diese Beobachtung:



Abbildung 20.5.3.1: Temporäre Beobachtung von Button1



Abbildung 20.5.3.2: Beobachtung von Button1 wurde abgeschaltet

- Der Quelltext wurde so geschrieben, dass das Click-Ereignis von *Button1* (Button-Text 'Beobachtungsstatus ermitteln') zeitweise beobachtet wird.
- Nach dem 3. Klick auf den *Button1* wird die Beobachtung abgeschaltet, weil dann das ausgelöste Ereignis nicht mehr abgebrochen wird.
- Daher können Sie dann die Ausgabe im *Button1_Click-Ereignis* in der Konsole der IDE sehen.

```
' Gambas class file

Public hObserver1 As Observer
Public iCount As Integer = 1

Public Sub _new()
    hObserver1 = New Observer(Button1, False) As "ObservedButton1"
End

Public Sub Form_Open()
    FMain.Center
    FMain.Resizable = False
    MovieBox1.Border = Border.None
    MovieBox1.Alignment = Align.Center
    MovieBox1.Playing = True
End ' Form_Open()

Public Sub ObservedButton1_Click()
    If iCount <= 3 Then
        Print "Der Button wurde zum " & iCount & ". Mal angeklickt. Das Click-Ereignis wird jedoch verworfen!"
        Stop Event
    Endif
    If iCount = 4 Then
        hObserver1 = Null
        MovieBox1.Playing = False
        FMain.Text = "Ende der Beobachtung ..."
    Endif
    Inc iCount
End ' Observer1_Click()

Public Sub Button1_Click() ' Original
    Print "Der Observer für Button1 wurde nach 3 Beobachtungszyklen abgeschaltet."
End ' Button1_Click()

Public Sub Form_Close()
    If hObserver1 Then hObserver1 = Null
End ' Form_Close()

Public Sub Button2_Click()
    FMain.Close
End ' btnClose_Click()
```

Diese Ausgaben ergeben sich beim Testen von Projekt 1:

```
Der Button wurde zum 1. Mal angeklickt. Das Click-Ereignis wird jedoch verworfen!
Der Button wurde zum 2. Mal angeklickt. Das Click-Ereignis wird jedoch verworfen!
Der Button wurde zum 3. Mal angeklickt. Das Click-Ereignis wird jedoch verworfen!
Der Observer für Button1 wurde nach 3 Beobachtungszyklen abgeschaltet.
```

20.5.4 Projekt 2

Im Gegensatz zum Projekt 1 wird im Projekt 2 die Beobachtung eines Objekts so angelegt, dass der Observer das *Button1_Click-Ereignis* abfängt, nachdem es verarbeitet wurde. Im generierten Ereignis *ObservedButton1_Click()* wird eine Zufallszahl ausgegeben, die im *Button1_Click-Ereignis* erzeugt und in der Eigenschaft *hObserver1.Tag* gespeichert wurde.

Quelltext:

```
' Gambas class file

Public hObserver1 As Observer

Public Sub _new()
    hObserver1 = New Observer(Button1, True) As "ObservedButton1"
End ' _new()

Public Sub Form_Open()
    FMain.Center
    FMain.Resizable = False
    MovieBox1.Alignment = Align.Center
    MovieBox1.Border = Border.None
    MovieBox1.Playing = True

End ' Form_Open()

Public Sub ObservedButton1_Click()
    Dim sMessage As String

    sMessage = "Vor 0.2 Sekunden wurde die folgende Zahl im beobachteten Button1_Click-Ereignis erzeugt: "
    Print sMessage; hObserver1.Tag
```

```

End ' Observer1_Click()

Public Sub Button1_Click()
    Dim fNumber As Float

    Randomize
    fNumber = Rnd(-2.0, 2.01)
    hObserver1.Tag = Round(fNumber, -2)
    Wait 0.2
End ' Button1_Click()

Public Sub btnClose_Click()

    If hObserver1 Then hObserver1 = Null
    FMain.Close
End ' btnClose_Click()

```

Diese Ausgaben zeigen in der Konsole der Gambas-IDE:

```

Vor 0.2 Sekunden wurde die folgende Zahl im beobachteten Button1_Click-Ereignis erzeugt: -0,87
Vor 0.2 Sekunden wurde die folgende Zahl im beobachteten Button1_Click-Ereignis erzeugt: 1,18
Vor 0.2 Sekunden wurde die folgende Zahl im beobachteten Button1_Click-Ereignis erzeugt: 0,33

```

20.5.5 Projekt 3

Folgende Besonderheiten kennzeichnen das 3. Projekt:

- Es werden *alle* Steuerelemente – auch die nicht sichtbaren – im Programmfenster beobachtet.
- Es wird nur das *Enter*-Ereignis beobachtet.
- Für jedes Steuerelement wird ein *eigener* Observer erzeugt.
- Für alle Observer wird ein *gemeinsamer* Event-Gruppenname (ObservedAll) vergeben.
- Um die Steuerelemente derselben Gruppe im Eventhandler zu unterscheiden, besitzt jedes Steuerelement einen speziellen Wert für die *Tag-Eigenschaft*.

Das vollständige Projekt finden Sie im Download-Bereich. Es wird hier nur der relevante Quelltext vorgestellt, in dem die o.a. Besonderheiten gut ablesbar sind:

```

...
Public Sub _new()
    SetAllObservers(ME)
End ' _new()

Private Sub SetAllObservers(hContainer As Container)
    Dim hObject As Object
    Dim hObserver As Observer

    For Each hObject In hContainer.Children
        hObserver = New Observer(hObject, False) As "ObservedAll"
        ' Rekursiver Abstieg
        If hObject Is Container Then SetAllObservers(hObject)
    Next
End ' SetAllObservers(..)

Public Sub ObservedAll_Enter()
    Select Last.Tag
        Case "PB"
            Print Object.Type(Last); " : "; Last.Name; " ->> Höhe = "; Last.H; "px"
        Case "HBOX"
            Print Object.Type(Last); " : "; Last.Name; " ->> Spacing = "; Last.Spacing
        Case "OI"
            Print Object.Type(Last); " : "; Last.Name; " ->> Text = ' "; Last.Text; " '"
        Case "PRE"
            Print Object.Type(Last); " : "; Last.Name; " ->> Aktiv? = "; Last.Enabled
        Case "NEXT"
            Print Object.Type(Last); " : "; Last.Name; " ->> Tag-Eigenschaft = "; Last.Tag
        Case "PS"
            Print Object.Type(Last); " : "; Last.Name; " ->> Expand? = "; Last.Expand
        Case "CLOSE"
            Print Object.Type(Last); " : "; Last.Name; " ->> Font = "; Last.Font.ToString()
        Case "S"
            Print Object.Type(Last); " : "; Last.Name; " ->> X = "; Last.X; "px"
    End Select
End ' ObservedAll_Enter()

```

Diese Ausgaben zeigen sich zum Beispiel in der Konsole der Gambas-IDE, wenn Sie sich mit der Maus über den Steuerelementen bewegen:

```
PictureBox : PictureBoxD ->> Höhe = 360px
HBox : HBox1 ->> Spacing = True
Button : btnOpenFileImage ->> Text = ' Bilder auswählen '
Button : btnPrevious ->> Aktiv? = False
Button : btnNext ->> Tag-Eigenschaft = NEXT
Panel : panSpace ->> Expand? = True
Button : btnClose ->> Font = Ubuntu,11
Separator : Separator1 ->> X = 8px
```



Abbildung 20.5.5.1: Projekt 3 – GUI