12.4.4.1 Eigene Dialoge – Projekt 1

Im ersten Beispiel wird ein Dialog entwickelt, der vor dem Haupt-Programm aufgerufen wird und mit dem zwei Daten – ein frei wählbarer Name und ein Passwort – erfasst werden. Bei der Planung eines eigenen Dialogs sind u.a. folgende Fragen zu beantworten, wobei die Reihenfolge keine Rangfolge impliziert:

(F1) In welchem Modus wird der Dialog aufgerufen?
(F2) Wird der Dialog als Vorschalt-Fenster des (Haupt-)Programms oder zur Laufzeit des (Haupt-)Programms angezeigt?
(F3) Soll im Dialog ein Titel (Fensterzeile) angezeigt werden?
(F4) Müssen Daten an den Dialog zur Änderung übergeben werden? Mit welchem Datentyp werden diese Daten an den Dialog übergeben (Typ, Array, Collection)?
(F5) Werden Daten aus dem Dialog ausgelesen? Sind es mehrere Daten? Welchen Daten-Typ haben diese Daten?
(F6) Wie soll die grafische Benutzeroberfläche für das Dialog-Fenster aussehen (Steuerelemente, Anordnung)?

In Bezug auf das 1. Beispiel hier die Antworten:

(A1) Der Dialog wird *modal* aufgerufen → *_call(..)-Methode*.
(A2) Vorschalt-Fenster des (Haupt-)Programms.
(A3) Ja → Dialog-Titel-Text: 'Geben Sie die Daten ein!'
(A4) Nein.
(A5) Ja – genau zwei. Beide Daten sind vom gleichen Typ 'String' und werden als Schlüssel-Wert-Paar im Dialog in einer Collection gespeichert. Der Rückgabewert der Methode, mit dem der Dialog aufgerufen wird, hat deshalb den selben Datentyp.
(A6) Die folgende → Abbildung 12.4.4.1.1 zeigt zwei Label, zwei Textboxen und zwei Schaltflächen:

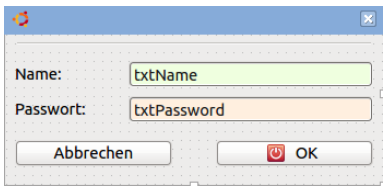

Abbildung 12.4.4.1.1: Dialog-Box (IDE)

Mit diesen Vorleistungen kann die Entwicklung eines eigenen Dialogs in Angriff genommen werden.

Im Kern definiert man ein Dialog-Fenster → Abbildung 12.4.4.1.1 und ruft es auf mit Hilfe der speziellen *_call(..)-Methode* im Haupt-Programm auf. Informationen zu dieser Methode finden Sie in der Gambas-Dokumentation unter → http://gambaswiki.org/wiki/lang/special/call. In den optionalen Argumenten der _call(..)-Methode können Sie Daten an den Dialog übergeben und im Rückgabewert der Methode (Funktionswert) Daten aus dem Dialog lesen. Mit dem Aufruf der _call(..)-Methode kann die Dialog-Box sich selbst *modal* anzeigen! Nun können Sie die Daten im Dialog bearbeiten. Wird die Dialog-Box geschlossen, so kehrt der ShowModal()-Aufruf – das passierte *automatisch* – in den _call()-Aufruf zurück und Sie können Daten des Dialogs auslesen.

Das alles findet im ersten Beispiel im Open-Event des Hauptformulars statt – bevor dieses Formular angezeigt wird! Die _call()-Methode erlaubt es, ein *Objekt* syntaktisch wie eine *Funktion* zu benutzen. Nutzen Sie die _call(..)-Methode immer dann, wenn ein Formular (oder eine Klasse generell) eine ganz bestimmte Aufgabe hat, die sie selbst erledigen kann, ohne zum Beispiel Event-Handler in anderen Klassen zu benötigen.

Die folgenden beiden Quelltexte für das Beispiel 1 zeigen die Implementierung eines eigenen Dialogs in Gambas:

Quelltext (Haupt-)Programm (FMain.class):

```
[1]  ' Gambas class file
[2]
[3]  Public Sub Form_Open()
```

```
[4]    Dim cData As Collection
[5]    Dim sDialogTitel As String
[6]    Dim frmLogin As New FLogin
[7]
[8]    FMain.Center
[9]    FMain.Resizable = False
[10]   sDialogTitel = "Geben Sie die Daten ein!"
[11]
[12] ' Bevor das Hauptfenster angezeigt wird, wird das Login-Fenster automatisch 'modal' geöffnet
[13]   cData = frmLogin(sDialogTitel)
[14]   If Not cData Then ' Abbruch?
[15]      Me.Close() ' Ohne Kommentar das Haupt-Programm beenden.
[16]      Return
[17]   Endif
[18]
[19]   If cData["password"] <> "mgA+" Then ' Fehlerhafter Login-Versuch? Mit Kommentar beenden.
[20]      Message.Title = "Fehler in der Dialog-Box"
[21]      Message.Error("Login fehlgeschlagen!")
[22]      Me.Close()
[23]      Return
[24]   Else
[25]    ' Sonst Haupt-Programm starten ...
[26]     lblGreet.Text = Subst$("Sie sind also &1. Gut zu wissen!", cData["name"])
[27]   Endif
[28]
[29] End ' Form_Open()
```

Kommentar:

- Um den Rückgabewert der _call(..)-Methode zu sichern, wird in der Zeile 4 die Variable *cData* vom Daten-Typ Collection erzeugt. Er entspricht dem Typ des Rückgabewerts.
- In der Zeile 6 wird ein neues Fenster vom Typ *FLogin* – das ist der Dialog – erzeugt.
- Die komplette Funktionalität des Dialogs wird in einer einzigen _call(..)-Methode gekapselt. In der Zeile 13 wird die _call(..)-Methode aufgerufen. Als Argument wird der Dialog-Titel gesetzt.
- Wenn die Dialog-Box geschlossen wird, dann wird der Rückgabewert der _call(..)-Methode in der Variable *cData* gespeichert.
- Ist die Variable cData leer, weil der Dialog abgebrochen wurde, wird das Haupt-Programm beendet (Zeilen 14 bis 17).
- Wurde ein falsches Passwort eingegeben (Zeile 19), dann wird ein Fehler angezeigt und das Haupt-Programm beendet (Zeilen 22 und 23).
- Mit der Eingabe des richtigen Passworts wird das Haupt-Programm angezeigt und eine Meldung ausgegeben, die den gespeicherten Namen (Zeile 26) aus dem Dialog verwendet.

Quelltext Dialog (FLogin.class):

```
[1]  ' Gambas class file
[2]
[3]  Public Sub _call(sTitel As String) As Collection
[4]    Me.Text = sTitel
[5]
[6]    If Me.ShowModal() = 0 Then
[7]       Return Null
[8]    Else
[9]       Return ["name": txtName.Text, "password": txtPassword.Text]
[10]   Endif
[11]
[12] End ' _call(sTitel As String)
[13]
[14] Public Sub btnCancel_Click()
[15]   Me.Close(0)
[16] End ' btnCancel_Click()
[17]
[18] Public Sub btnOK_Click()
[19]   Me.Close(1)
[20] End ' btnOK_Click()
[21]
[22] Public Sub txtName_Activate()
[23]   txtPassword.SetFocus()
[24] End ' txtName_Activate()
[25]
[26] Public Sub txtPassword_Activate()
[27]   btnOK_Click()
[28] End ' txtPassword_Activate()
```

Kommentar:

- Der _call(..)-Methode wird in der Zeile 3 der Dialog-Titel als Argument mitgegeben.
- Nach dem Schließen des Dialogs wird in den Zeilen 6 bis 10 ausgewertet, welchen Wert das Dialog-Fenster beim Schließen in *Me.ShowModal()* in Zeile 6 zurückgab. Beachten Sie: Dieser Rückgabewert hat nichts mit dem *Funktionswert* der _call(..)-Methode zu tun.
- Der Wert von *Me.ShowModal()* ist entweder 0, wenn der Dialog abgebrochen wurde (Zeile 15) oder 1 (Zeile19), wenn die bearbeiteten Daten übernommen wurden.
- Je nach Wert wird entweder eine leere Collection zurückgegeben oder die Collection mit dem eingegebenen Namen und dem Passwort.

Hinweise:

- Die Methode *Window.ShowModal()* hat einen (oft vergessenen) Rückgabewert. Hat man ein Fenster modal offen und schließt es mit ME.Close(iStatus), dann ist der Rückgabewert von ShowModal() gleich iStatus.
- Beachten Sie, dass Sie auch der Methode Window.Close() ein *optionales* Argument übergeben können, um es in geeigneter Weise auszuwerten.
- Wenn ein geöffnetes Fenster mit dem Schließen-Symbol [x] in der Fenster-Leiste geschlossen wird, so wird automatisch der Wert 0 zurückgegeben.

Die Ergebnisse im ersten Beispiel sind wenig spektakulär – aber genau jene, die zu erwarten waren. Der Dialog wird als *Vorschalt*-Programm angezeigt und Sie werden zur Eingabe der erforderlichen Daten aufgefordert:

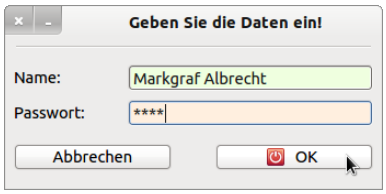

Abbildung 12.4.4.1.2: Dialog-Box – Daten-Eingabe

Bei einem fehlerhaften Passwort – gespeichert im Rückgabewert des Dialogs cData["passwort"] – sehen Sie diese Meldung und dann nichts mehr … .

Abbildung 12.4.4.1.3: Fehlermeldung

War das Einloggen erfolgreich, wird das Hauptprogramm angezeigt. Im Kommentar wird der im Dialog eingegebene Name (cData["name"]) verwendet:

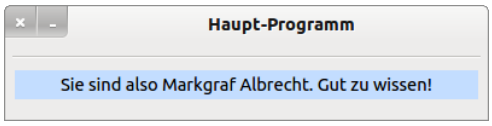

Abbildung 12.4.4.1.4: Das Haupt-Programm wird angezeigt ...