

12.4.4.2 Eigene Dialoge – Projekt 2

Im zweiten Projekt – einem Grafik-Programm – können Bildgrößen, Transparenz-Werte und Bild-Typen in 3 verschiedenen Combo-Boxen (CB) ausgewählt werden. Zur Laufzeit sollen die jeweils in der Eigenschaft `CB.List` gespeicherten Werte in drei Dialogen geändert werden. Beim allerersten Programm-Start werden die CB-Listen mit Standard-Werten gefüllt. Die aktuellen CB-Listen werden beim Programm-Ende in drei Dateien in einem gambas-spezifischen Format gespeichert. Bei jedem weiteren Programm-Start werden diese drei Dateien ausgelesen und der Inhalt den entsprechenden CB-Listen zugewiesen. Im Projekt 2 werden alle für die drei Dialoge nicht relevanten Steuerelemente im Haupt-Programm im Block `* GRAFIK-PROGRAMM*` zusammengefasst:

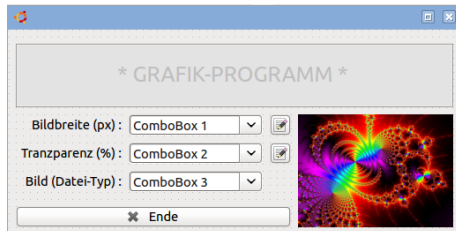


Abbildung 12.4.4.2.1: Grafik-Programm (IDE)

Auch im 2. Projekt gilt es diese Fragen zu beantworten:

- (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)?

Das sind die Antworten, die für alle drei Dialoge in gleicher Weise gelten, da sich für den Dialog nur die Argumente ändern:

- (A1) Der Dialog wird stets *modal* aufgerufen → `RunDialog(..)-Methode`.
- (A2) Anzeige zur Laufzeit des (Haupt-)Programms.
- (A3) Ja → Für jeden einzelnen Dialog gibt es einen eigenen Dialog-Titel-Text.
- (A4) Ja, es sind Listen vom Daten-Typ `String-Array`.
- (A5) Ja – jeweils eine Liste. Der Rückgabewert der `RunDialog(..)-Methode`, mit dem der Dialog aufgerufen wird, hat deshalb den selben Datentyp `String-Array`.
- (A6) Die Abbildung 12.4.4.2.1 zeigt 3 Label, drei Combo-Boxen und drei Schaltflächen.

Im ersten Projekt wurde für den Aufruf des Dialogs die spezielle `_call(..)-Methode` verwendet. Sie erlaubt es, ein *Objekt* syntaktisch wie eine *Funktion* zu benutzen. Was liegt also näher, als eine eigene Funktion zu schreiben, um mit deren Aufruf einen Dialog zu starten, Daten an den Dialog zu übergeben und bearbeitete Daten aus dem Dialog zu lesen. Dieser Funktion geben wir den passenden Namen `RunDialog(..)`. Diese Funktion wird im Projekt 2 verwendet; hier vorgestellt und anschließend kommentiert:

```
[1] Public Function RunDialog(aValue As String[], Optional sTitel As String) As String[]
[2]     Dim hForm As Form
[3]
[4]     hForm = New FEditList
[5]     If Not sTitel Then sTitel = "Bearbeiten Sie die Daten!"
[6]     Me.Text = sTitel
[7]     aTransfer = aValue
[8]
[9]     If Me.ShowModal() = 0 Then ' Bei einem Abbruch des Dialogs wird der Wert 0 zurückgegeben
[10]         Return aTransfer
[11]     Else
[12]         Return lboxList.List ' Die Daten werden im Dialog in einer ListBox bearbeitet
[13]     Endif
```

```
[14]
[15] End ' RunDialog(..)
```

Kommentar:

- Der Funktion *RunDialog(..)* in der Zeile 1 werden zwei Argumente übergeben; wobei das zweite optional ist. Über das erste Argument werden die zu bearbeitenden Daten (Daten-Typ String-Array) übergeben und mit dem optionalen Argument ein Titel, der in der Fensterzeile angezeigt wird. Der Funktionswert hat den gleichen Daten-Typ wie die zu bearbeitenden Daten – das ist jedoch nicht zwingend.
- Im Gegensatz zur *_call(..)*-Methode müssen Sie sich um das Erzeugen des Dialogs in der Zeilen 2 und 4 selbst kümmern.
- Über die Zeilen 5 und 6 realisieren Sie, dass in der Dialog-Fensterzeile immer ein passender Text angezeigt wird.
- In der Variable *aTransfer* (Daten-Typ String-Array) werden die dem Dialog übergebenen Daten gespeichert, damit u.a. die Originaldaten zur Verfügung stehen, wenn der Dialog abgebrochen wurde (Zeile 7).
- In den Zeilen 9 bis 13 werden entweder die Originaldaten zurückgegeben oder die bearbeiteten Daten aus der Listbox-Liste.

Neben den oberen Combo-Boxen befinden sich Schaltflächen, die jeweils einen Dialog öffnen. Zum Aufruf des Dialogs zur Bearbeitung der Liste der Bild-Typen wurde der entsprechenden Combo-Box ein Kontext-Menü spendiert.



Abbildung 12.4.4.2.2: Haupt-Programm

Wenn Sie den Dialog zur Bearbeitung der Bildweiten starten, dann können Sie die angezeigte Liste bearbeiten. Selbst das Löschen der kompletten Liste ist möglich.

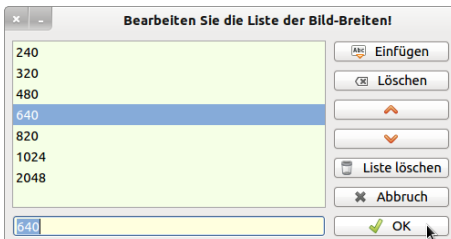


Abbildung 12.4.4.2.3: Dialog zur Bearbeitung spezieller Daten

Realisiert wird der o.a. Dialog durch diese Prozedur im Haupt-Programm:

```
Public Sub btnEditWidth_Click()
    Dim sTitel As String

    sTitel = "Bearbeiten Sie die Liste der Bild-Breiten!"
    cmbImageWidth.List = FEditList.RunDialog(cmbImageWidth.List, sTitel)
    If cmbImageWidth.Index < 0 Then ' Liste leer?
        cmbImageWidth.Add("640")
    Else
        cmbImageWidth.Text = cmbImageWidth.List[0]
    Endif
End ' btnEditWidth_Click()
```

Beachten Sie, dass im nächsten Abschnitt nur *ein* Dialog implementiert wird. Das ist hinreichend, da in allen drei Fällen nur String-Arrays bearbeitet werden, die jeweils als Argument in der Funktion *RunDialog(..)* übergebenen werden. Für den Fall, dass Sie Daten mit unterschiedlichem Daten-Typ bearbeiten wollen, müssten Sie verschiedene Dialoge erzeugen.

Die Quelltexte für das zweite Projekt werden für das Haupt-Programm und den Dialog vollständig vorgestellt:

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

```
[1] ' Gambas class file
[2]
[3] Public hFile As File
[4] Public sFilePath As String
[5] Public sProjectPath As String = Application.Path
[6]
[7] Public Sub Form_Open()
[8]     FMain.Center
[9]     FMain.Resizable = False
[10]    FMain.Text = "Grafik-Programm 'Kritzelt 2.0'"
[11]
[12] ' Initialisierungen:
[13] ' Bild-Weite
[14]    sFilePath = sProjectPath &/ cmbImageWidth.Name & ".list"
[15]    If Not Exist(sFilePath) Then
[16]        cmbImageWidth.List = ["240", "480", "640", "820", "1024"]
[17]    Else
[18]        cmbImageWidth.List = ImportData(sFilePath)
[19]    Endif ' Not Exist(sFilePath) ?
[20]    cmbImageWidth.Text = cmbImageWidth.List[2]
[21]    cmbImageWidth.ReadOnly = True
[22]
[23] ' Bild-Transparenz
[24]    sFilePath = sProjectPath &/ cmbImageTransparency.Name & ".list"
[25]    If Not Exist(sFilePath) Then
[26]        cmbImageTransparency.List = ["0", "10", "25", "50", "75", "85", "100"]
[27]    Else
[28]        cmbImageTransparency.List = ImportData(sFilePath)
[29]    Endif ' Not Exist(sFilePath) ?
[30]    cmbImageTransparency.Text = cmbImageTransparency.List[0]
[31]    cmbImageTransparency.ReadOnly = True
[32]
[33] ' Bild-Typ (Extension)
[34]    sFilePath = sProjectPath &/ cmbImageType.Name & ".list"
[35]    If Not Exist(sFilePath)
[36]        cmbImageType.List = ["png", "jpg", "gif", "ico", "svg"]
[37]    Else
[38]        cmbImageType.List = ImportData(sFilePath)
[39]    Endif ' Not Exist(sFilePath) ?
[40]    cmbImageType.Text = cmbImageType.List[0]
[41]    cmbImageType.ReadOnly = True
[42]
[43] End ' Form_Open()
[44]
[45] Public Sub btnEditWidth_Click()
[46]     Dim sTitel As String
[47]
[48]     sTitel = "Bearbeiten Sie die Liste der Bild-Breiten!"
[49]     cmbImageWidth.List = FEditList.RunDialog(cmbImageWidth.List, sTitel)
[50]     If cmbImageWidth.Index < 0 Then ' Liste leer?
[51]         cmbImageWidth.Add("640")
[52]     Else
[53]         cmbImageWidth.Text = cmbImageWidth.List[0]
[54]     Endif
[55]
[56] End ' btnEditWidth_Click()
[57]
[58] Public Sub btnEditTransparency_Click()
[59]     Dim sTitel As String
[60]
[61]     sTitel = "Bearbeiten Sie die Liste der Transparenz-Werte!"
[62]     cmbImageTransparency.List = FEditList.RunDialog(cmbImageTransparency.List)
[63]     If cmbImageTransparency.Index = -1 Then ' Alternative Abfrage zu Zeile 50: Liste leer?
[64]         cmbImageTransparency.Add("0")
[65]     Else
[66]         cmbImageTransparency.Text = cmbImageTransparency.List[0]
[67]     Endif
[68]
[69] End ' btnEditTransparenz_Click()
[70]
[71] Public Sub cmbImageType_Menu()
[72]     Dim mnuImageType As Menu
[73]     Dim mnuMenuItem As Menu
[74]
[75] ' Es wird ein neues Menü-Objekt für die Bildtyp-ComboBox erzeugt:
[76]     mnuImageType = New Menu(FMain, False)
[77] ' 1. Menü-Eintrag im Menü mnuImageType
```

```

[78] mnuMenuItem = New Menu(mnuImageType) As "mnuEditTyp"
[79] mnuMenuItem.Text = "Liste bearbeiten"
[80] mnuMenuItem.Picture = Stock["edit"]
[81] ' Das generierte Menü wird der Bildtyp-ComboBox als PopUp-Menü zugewiesen
[82] mnuImageType.Popup
[83]
[84] End ' cmbImageTyp_Menu()
[85]
[86] ' Aktion, wenn das Kontextmenü der 1. ComboBox ausgewählt wurde
[87] Public Sub mnuEditTyp_Click()
[88] Dim sTitel As String
[89]
[90] sTitel = "Bearbeiten Sie die Liste der Bild-Typen!"
[91] cmbImageType.List = FEditList.RunDialog(cmbImageType.List, sTitel)
[92] If cmbImageType.Count = 0 Then ' Liste leer?
[93]   cmbImageType.Add("png")
[94] Else
[95]   cmbImageType.Text = cmbImageType.List[0]
[96] Endif
[97]
[98] End ' mnuEditTyp_Click()
[99]
[100] Public Sub Form_Close()
[101] ' Daten-Export
[102]   sFilePath = sProjectPath & / cmbImageWidth.Name & ".list"
[103]   ExportData(sFilePath, cmbImageWidth.List)
[104]   sFilePath = sProjectPath & / cmbImageTransparency.Name & ".list"
[105]   ExportData(sFilePath, cmbImageTransparency.List)
[106]   sFilePath = sProjectPath & / cmbImageType.Name & ".list"
[107]   ExportData(sFilePath, cmbImageType.List)
[108] End ' Form_Close()
[109]
[110] Public Sub btnClose_Click()
[111]   FMain.Close
[112] End ' btnClose_Click()
[113]
[114] ' Diese Routine lädt Daten aus einer binären, gambas-spezifischen Datei in ein String-Array
[115] Private Function ImportData(sPath As String) As String[]
[116]   hFile = Open sPath For Read
[117]   Return Read #hFile As Array
[118]   Close #hFile
[119] Catch
[120]   Message.Error("Der Daten-Import war fehlerhaft!" & gb.NewLine & "Fehler: " & Error.Text)
[121] End ' ImportData(...)
[122]
[123] ' Diese Routine speichert Daten aus einem String-Array in einer binären, gambas-spezifischen Datei
[124] Private Sub ExportData(sPath As String, aData As String[])
[125]   If aData.Count = 0 Then
[126]     Return
[127]   Else
[128]     hFile = Open sPath For Write Create
[129]     Write #hFile, aData As Array
[130]     Close #hFile
[131]     Catch
[132]     Message.Error("Der Daten-Export war fehlerhaft!" & gb.NewLine & "Fehler: " & Error.Text)
[133]   Endif ' aData.Count = 0 ?
[134] End ' ExportData(...)

```

Quelltext Dialog (FLogin.class):

```

[1] ' Gambas class file
[2]
[3] Public aTransfer As String[]
[4]
[5] Public Function RunDialog(aValue As String[], Optional sTitel As String) As String[]
[6]   Dim hForm As Form
[7]
[8]   hForm = New FEditList
[9]   If Not sTitel Then sTitel = "Bearbeiten Sie die Daten!"
[10] Me.Text = sTitel
[11] aTransfer = aValue
[12]
[13] If Me.ShowModal() = 0 Then
[14]   Return aTransfer
[15] Else
[16]   Return lboxList.List
[17] Endif
[18]
[19] End ' RunDialog(..)
[20]
[21] Public Sub Form_Open()
[22]   FEditList.Center
[23]   FEditList.Resizable = False

```

```

[24]
[25]   listBox.List = aTransfer
[26]   If listBox.Count Then
[27]       listBox.Index = 0
[28]       txtElement.Text = listBox[0].Text
[29]       txtElement.SetFocus
[30]   Endif
[31]   Refresh
[32] End ' Form_Open()
[33]
[34] Public Sub btnCancel_Click()
[35]     Me.Close(0)
[36] End ' btnCancel_Click()
[37]
[38] Public Sub btnOK_Click()
[39]     Me.Close(1)
[40] End ' btnOK_Click()
[41]
[42] Public Sub btnInsertElement_Click()
[43]     listBox.Add("Neu_" & CStr(listBox.Count + 1))
[44]     Refresh
[45]     listBox.Index = listBox.Count - 1
[46]     txtElement.Text = listBox.Current.Text
[47]     txtElement.SetFocus
[48] End ' btnInsertSenderURL_Click()
[49]
[50] Public Sub btnDeleteElement_Click()
[51]     Dim iCurrentIndex As Integer
[52]
[53]     iCurrentIndex = listBox.Index
[54]     If iCurrentIndex < 0 Then Return
[55]
[56]     If listBox.Count > 1 Then
[57]         listBox.Remove(iCurrentIndex)
[58]     Else
[59]         If Message.Question("Letztes Element löschen?", "Löschen", "Abbruch") <> 1 Then Return
[60]         listBox.Remove(iCurrentIndex)
[61]     Endif
[62]
[63]     If iCurrentIndex >= listBox.Count Then Dec iCurrentIndex
[64]     listBox.Index = iCurrentIndex
[65]     txtElement.Text = listBox[iCurrentIndex].Text
[66]     Refresh
[67]
[68] End ' btnDeleteRadioURL_Click()
[69]
[70] Public Sub btnDown_Click()
[71]     Dim iCurrentIndex As Integer
[72]
[73]     iCurrentIndex = listBox.Index
[74]     If iCurrentIndex < (listBox.Count - 1) Then
[75]         Swap listBox[iCurrentIndex].Text, listBox[iCurrentIndex + 1].Text
[76]         listBox.Index = iCurrentIndex + 1
[77]     Endif ' iCurrentIndex < (lstValue.Count - 1)
[78]
[79] End ' btnDown_Click()
[80]
[81] Public Sub btnUp_Click()
[82]     Dim iCurrentIndex As Integer
[83]
[84]     iCurrentIndex = listBox.Index
[85]     If iCurrentIndex > 0 Then
[86]         Swap listBox[iCurrentIndex].Text, listBox[iCurrentIndex - 1].Text
[87]         listBox.Index = iCurrentIndex - 1
[88]     Endif ' iCurrentIndex > 0
[89]
[90] End ' btnUp_Click()
[91]
[92] Public Sub btnDeleteList_Click()
[93]     If Message.Question("Komplette Liste löschen?", "Löschen", "Abbruch") <> 1 Then Return
[94]     listBox.Clear
[95]     Refresh
[96] End ' btnDeleteList_Click()
[97]
[98] Public Sub listBox_Click()
[99]     If Not listBox.Current Then Return
[100]     txtElement.Text = listBox.Current.Text
[101]     txtElement.SetFocus
[102]     txtElement.SelectAll
[103] End ' listBox_Click()
[104]
[105] Public Sub txtElement_Change()
[106]     If listBox.Current Then
[107]         listBox.Current.Text = txtElement.Text

```

```
[108] Endif ' lsbCBLList.Current ?
[109] End ' txtElement_Change()
[110]
[111] Private Sub Refresh()
[112]     Dim bEnabled As Boolean
[113]
[114]     bEnabled = lboxList.Count
[115]
[116]     txtElement.Enabled = bEnabled
[117]     btnDeleteElement.Enabled = bEnabled
[118]     btnUp.Enabled = bEnabled
[119]     btnDown.Enabled = bEnabled
[120]     btnDeleteList.Enabled = bEnabled
[121]
[122]     If Not bEnabled Then txtElement.Clear
[123]
[124] End ' Refresh()
```