

7.4.9.3 Sortierung abgeleiteter Arrays

Für die korrekte Umsetzung der Idee zur Sortierung von Daten in einer GridView wie im Beispiel im Kapitel 7.4.9.2 ist offensichtlich ein anderer Zugang zur Abbildung der Daten der GridView auf ein Array erforderlich, das korrekt sortiert werden kann.

Ansatz:

- Als Kernstück des neuen Ansatzes wird zuerst eine Klasse beschrieben, in der nur die für den Anwendungszweck erforderlichen öffentlichen Variablen mit dem notwendigen Daten-Typ deklariert sind. Die Klasse repräsentiert in diesem Ansatz einen Datensatz (Rekord) in einer Kursliste.
- Diese Klasse enthält neben der (reinen) Datenstruktur nur eine spezielle Methode.
- Die Methode `Object[].Sort(Mode As Integer)` sortiert Objekte, d.h. sie ruft die objekt-interne spezielle Methode `_compare(..)` auf, um die Objekte zu vergleichen. Daher kann ein Array sortiert werden, dessen Elemente Objekte sind – was ja für abgeleitete Arrays zutrifft.
- Der Vorteil des verfolgten Ansatzes besteht auch darin, dass man neben der Vorgabe der Sortier-Reihenfolge auch das Element im Objekt (statisch) vorgeben kann, nach dem das Objekt-Array schrittweise sortiert wird.

Hier werden Ihnen die vollständigen Quelltexte für die Klasse CDS (Klasse Datensatz) sowie für das Hauptprogramm angegeben und jeweils kurz kommentiert:

Klasse CDS.class

```
[1] ' Gambas class file
[2]
[3] Public Nachname As String
[4] Public Alter As Integer
[5]
[6] Public Function _compare(DS As CDS) As Integer
[7]   Dim iSelector As Integer
[8]
[9]   iSelector = 1
[10]  Select Case iSelector
[11]  Case 1
[12]     Return Comp(Nachname, DS.Nachname)
[13]  Case 2
[14]     Return Sgn(Alter - DS.Alter)
[15]  Default
[16]     Error.Raise(("Keine passende Sortier-Methode gefunden!"))
[17]  End Select
[18]
[19] End
```

Kommentare:

- Die Funktionseigenschaften der Funktion `_compare(DS As CDS)` müssen Sie aufgaben-angepasst festlegen.
- Als Argument der Funktion wird ein Objekt vom Klassen-Typ (\rightarrow CDS) verwendet.
- Je nach Selektor-Wert wird in der Select-Case-Kontroll-Struktur in den Zeilen 10 bis 17 entweder nach dem Nachnamen (1) oder nach dem Alter (2) sortiert.
- Es muss das Ergebnis des Vergleichs in den Zeilen 12 (Zeichenketten-Vergleich) und 14 (Ganzzahl-Vergleich) als ganze Zahl zurück gegeben werden:

```
+1 , wenn das Objekt größer ist als das übergebene Argument,
0  , wenn die Objekte gleich sind,
-1 , wenn das Objekt kleiner ist als das übergebene Argument.
```

- Wenn sich für die Objekte die `_compare(..)-Methode` nicht umsetzen lässt, dann werden die Objekte nach ihren Adressen im Speicher verglichen. Für diesen Fall ist es sinnvoll, mit einer entsprechenden Fehlermeldung (Zeilen 15 und 16) zu reagieren.

Quelltext für FMain.class

```
[1] ' Gambas class file
[2]
[3] Public $2DGridArray As CDS[]
```

```

[4] Public KL As New CDS[] ' Klassen-Liste
[5] Public aGArray As New CDS[]
[6]
[7] Public Sub Form_Open()
[8]     Dim hCDS As CDS
[9]
[10]    FMain.Center
[11]    FMain.Resizable = False
[12]
[13]    SetGridProperty()
[14]    FillGridView()
[15]    $2DGridArray = GridToArray()
[16]
[17]    ' Zur Kontrolle:
[18]    ' For Each hCDS In $2DGridArray
[19]    '     Print "---> ", hCDS.Nachname, hCDS.Alter
[20]    ' Next
[21]
[22] End ' Form_Open()
[23]
[24] Private Sub SetGridProperty()
[25]     GridView1.Header = GridView1.Both ' Kopfzeile wird angezeigt
[26]     GridView1.Mode = Select.Single ' Die selektierte Zeile wird farbig markiert
[27]     GridView1.Columns.Count = 2 ' Anzahl der Spalten
[28]     GridView1.Sorted = True
[29]
[30]     GridView1.Columns[0].Width = 140
[31]     GridView1.Columns[0].Alignment = Align.Center
[32]     GridView1.Columns[0].Title = "Nachname"
[33]     GridView1.Columns[1].Width = 60
[34]     GridView1.Columns[1].Alignment = Align.Center
[35]     GridView1.Columns[1].Title = "Alter"
[36] End ' SetGridProperty()
[37]
[38] Public Sub FillGridView()
[39]     Dim iCount As Integer
[40]     Dim aNames As String[] = ["Adler", "Bär", "Dachs", "Fuchs", "Meise", "Uhu", "Elch"] '
[41]
[42]     Randomize
[43]
[44]     For iCount = 0 To 4 ' 5 Datensätze mit Zufallsdaten
[45]         Inc GridView1.Rows.Count ' Anzeige der generierten Datensätze (Zeilen in der Gitteransicht)
[46]         GridView1.MoveTo(GridView1.Rows.Count - 1, 0) ' Optional
[47]         GridView1[iCount, 0].Text = aNames[Int(Rnd(0, aNames.Count))]
[48]         GridView1[iCount, 1].Text = Str(Int(Rnd(15, 20))) ' 15..19
[49]     Next ' iCount
[50]
[51] End ' FillGridView()
[52]
[53] Public Function GridToArray() As CDS[]
[54]     Dim i As Integer
[55]     Dim hCDS As CDS
[56]
[57]     For i = 0 To 4
[58]         hCDS = New CDS
[59]         hCDS.Nachname = GridView1[i, 0].Text
[60]         hCDS.Alter = GridView1[i, 1].Text
[61]         KL.Add(hCDS)
[62]     Next ' iCount
[63]
[64]     Return KL
[65]
[66] End ' Function GridToArray()
[67]
[68] Public Sub ArrayToGrid(aArray As CDS[])
[69]     Dim iRowCount As Integer = 0
[70]     Dim hCDS As CDS
[71]
[72]     GridView1.Clear
[73]     GridView1.Rows.Count = aArray.Count
[74]
[75]     For Each hCDS In aArray
[76]         GridView1[iRowCount, 0].Text = hCDS.Nachname
[77]         GridView1[iRowCount, 1].Text = hCDS.Alter
[78]         Inc iRowCount
[79]     Next ' hCDS
[80]
[81] End ' ArrayToGrid(..)
[82]
[83] Public Sub GridViewShow(Optional bSorted As Boolean)
[84]     Dim hCDS As CDS
[85]
[86]     aGArray = $2DGridArray.Copy()
[87]

```

```

[88] If bSorted = True Then
[89]     aGArray.Sort(gb.Ascent) ' Festlegung der Sortier-Reihenfolge
[90]     ArrayToGrid(aGArray)
[91] Else
[92]     ArrayToGrid($2DGridArray)
[93] Endif
[94]
[95] End ' GridViewShow(..)
[96]
[97] Public Sub btnGridViewShow_Click()
[98]     GridViewShow()
[99] End ' btnGridViewShow_Click()
[100]
[101] Public Sub btnGridViewShowS_Click()
[102]     GridViewShow(True)
[103] End ' btnGridViewShowS_Click()

```

Kommentare:

- Die GridView dient zum Anzeigen der originalen Daten und zum Anzeigen der sortierten GridView-Daten.
- Die Funktion *GridToArray()* und die Prozedur *ArrayToGrid(aArray As CDS[])* dienen der Transformation der GridView-Daten in ein Array – das später sortiert werden kann – und aus dem originalen oder sortierten Array in GridView-Daten.
- In der Zeile 89 wird die Sortierung des abgeleiteten Arrays vom Typ *CDS[]* aufgerufen, was intern die spezielle Sortier-Methode für Objekt-Arrays in der Klasse *CDS* (Zeile 6) aufruft. Nach welcher Objekt-Variablen sortiert wird, ist dort durch den Selektor vorgegeben.

Das Ergebnis kann sich sehen lassen:

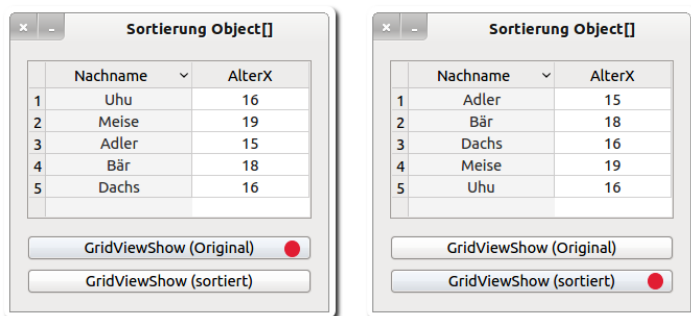


Abbildung 7.4.9.3.1: Korrekte Sortierung von GridView-Daten

Im Kapitel '7.4.9.4 Projekt zur Sortierung von Arrays' wird Ihnen ein Projekt vorgestellt, in dem das Element im Objekt dynamisch festgelegt werden kann, nach dem das Objekt-Array sortiert wird.