7.4.7 Array-Kopie

Im folgenden Abschnitt geht es um vergleichende Betrachtungen zu Arrays unter dem Aspekt 'Zuweisung eines Arrays' und 'Kopie eines Arrays'.

7.4.7.1 Zuweisung

- Zuerst werden ein *Array_A* und ein *Array_B* von gleichem Typ generiert und *Array_A* wird typgerecht mit Daten gefüllt.
- Dann wird *Array_A* einem *Array_B* zugewiesen und der Inhalt von *Array_B* ausgegeben.
- Danach wird *Array_A* sortiert.
- Abschließend wird der Inhalt von *Array_B* noch einmal ausgegeben.

```
Dim aArray_A As New Integer[3]
Dim aArray_B As Integer[]
Dim iCount As Integer

aArray_A = [4, -3, 7] ' [4, -3, 7] ist ein Inline-Array (ohne Namen)
aArray_B = aArray_A ' aArray_B erhält Referenz auf aArray_A

For iCount = 0 To aArray_B.Max ' Ausgabe Array_B
  Print aArray_B[iCount]
Next

aArray_A.Sort(gb.Descent)

For iCount = 0 To aArray_A.Max ' Ausgabe Array_A
  Print aArray_A[iCount],
Next
Print
For iCount = 0 To aArray_B.Max ' Ausgabe Array_B
  Print aArray_B[iCount]
Next
```

Ausgabe in der IDE-Konsole:

```
4 -3  7   ' Inhalt von aArray_B ist Inhalt von aArray_A
7  4 -3   ' Inhalt von aArray_A nach dem Sortieren
7  4 -3   ' Inhalt von aArray_B (ohne Sortierung!)
```

Die angezeigten Inhalte von Array_A und Array_B nach dem Sortieren von Array_A sind gleich! Die verallgemeinerte Erklärung bezogen auf Objekt_A und Objekt_B ist einfach: Es wird in der Variable Objekt_B eine Referenz auf die Daten von Objekt_A hinterlegt. Das bedeutet: Objekt_A und Objekt_B zeigen auf das gleiche Objekt im Speicher. Wenn Sie das Objekt_A ändern – beispielsweise den Inhalt des Objekts sortieren – so zeigt auch Objekt_B auf diese geänderten Daten.

7.4.7.2 Array-Kopie

Native Arrays besitzen die Array.Copy()-Methode und gehören daher zu den Objekten, die sich problemlos kopieren lassen. Diese Methode gibt Ihnen ein *neues* Array-Objekt 'Array_B' als 1:1-Kopie von 'Array_A' mit dem gleichen Inhalt zurück. Die beiden Objekte Array_A und Array_B sind jedoch völlig *unabhängig* von einander. Spätere Änderungen am Original Array_A haben zum Beispiel *keine* Auswirkung auf die Kopie Array_B und umgekehrt.

```
aArray_B = aArray_A.Copy() ' aArray_B als 1:1-Kopie von aArray_A

aArray_A[1] = 55 ' Änderung von aArray_A im 2. Element

For iCount = 0 To aArray_A.Max ' Ausgabe Array_A
  Print aArray_A[iCount],
Next

Print
For iCount = 0 To aArray_B.Max ' Ausgabe Array_B
  Print aArray_B[iCount],
Next
```

Ausgabe in der IDE-Konsole:

```
7  55  -3   ' Inhalt von aArray_A
7  4   -3   ' Inhalt von aArray_B
```

7.4.7.3 Beispiel für die Verwendung einer Array-Kopie

Um in einem Demonstrationsprojekt den Effekt der Sortierung von Array-Elementen in einer GridView mit der _compare()-Methode *abwechselnd* zu zeigen, werden das originale Daten-Array und das Array der sortierten Daten vom gleichen Array-Typ benötigt. Das vollständig beschriebene Projekt finden Sie im → Kapitel 7.4.9 Array – Sortierung.

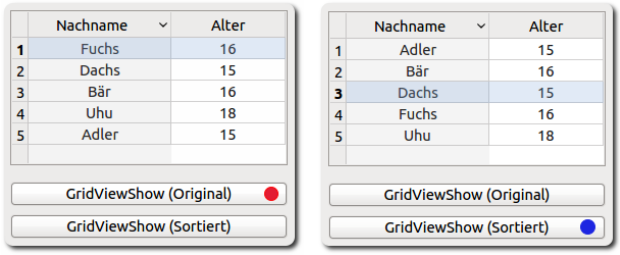

Abbildung 7.4.7.3.1: GridView-Daten (Originale Daten vs. sortierte Daten (→ Feld Nachname))

In der Zeile 4 des folgenden Quelltext-Auszugs wurde zuerst dieser Code verwendet:

```
[4] aGArray = $2DGridArray
```

Beim Test ergaben sich – erst Original und dann Sortierung nach den Nachnamen im ersten Feld – nacheinander die beiden oben abgebildeten Bilder → Abbildung 7.4.7.3.1. So sollte es sein!

Effekt erreicht? Nein, denn jeder weitere Klick auf einen der beiden Button zeigte nur noch eine Darstellung wie im 2. Bild ●. Die Erklärung für dieses Verhalten finden Sie im oberen Abschnitt über die Zuweisung von Arrays und deren Besonderheiten.

Korrekt ist der folgende Quelltext-Ausschnitt, in dem mit zwei verschiedenen Array-Objekten gearbeitet wird, wobei das Array *aGArray* eine 1:1-Kopie von Array *$2DGridArray* ist (Zeile 4). Die Sortierung der Elemente von aGArray in der Zeile 7 beeinflusst das Array *$2DGridArray* in keiner Weise, so dass in den beiden Prozeduren in Zeile 15 und 19 stets die richtigen Ansichten generiert werden:

```
[1]  Public Sub GridViewShow(Optional bSorted As Boolean)
[2]    Dim aGArray As New String[]
[3]
[4]    aGArray = $2DGridArray.Copy()
[5]
[6]    If bSorted = True Then
[7]       aGArray.Sort
[8]       ArrayToGrid(aGArray)
[9]    Else
[10]     ArrayToGrid($2DGridArray)
[11]   Endif ' bSorted = True ?
[12]
[13] End ' GridViewShow(..)
[14]
[15] Public Sub btnGridViewShow_Click()
[16]   GridViewShow()
[17] End ' btnGridViewShow_Click()
[18]
[19] Public Sub btnGridViewShowS_Click()
[20]   GridViewShow(True)
[21] End ' btnGridViewShowS_Click()
```