

7.4.8.2 Export und Import abgeleiteter Arrays

Das Abspeichern abgeleiteter Arrays zeichnet sich durch einige Besonderheiten aus, denn Gambas 3 kann nur *native Arrays* in einer Datei gambas-spezifisch speichern und wieder auslesen. Sie müssen die *Serialisierung* der Array-Elemente abgeleiteter Arrays beim Array-Export als auch die *De-Serialisierung* beim Array-Import selbst umsetzen. In einer geeigneten Kontrollstruktur schreiben Sie die Objekt-Eigenschaften der in den Elementen des abgeleiteten Arrays gespeicherten Objekte nacheinander in den Datei-Stream, denn Objekt-Eigenschaften von nativem Daten-Typ kann Gambas serialisieren. Ist eine Eigenschaft eines Objektes wieder ein Objekt, so kann Gambas diese Eigenschaft natürlich nicht serialisieren. Sie müssen das Verfahren dann rekursiv anwenden bis nur noch native Daten-Typen vorliegen und gespeichert werden können.

Beim Import lesen Sie die Objekt-Eigenschaften aus der Export-Datei und (re-)konstruieren das Array.

Im Projekt wird dieser Plan verfolgt:

- Zuerst wird ein abgeleitetes Array angelegt und mit Zufallsdaten gefüllt.
- Die Array-Elemente werden danach in der IDE-Konsole zur Kontrolle angezeigt.
- Dann kann man das angelegte Array exportieren (ohne Quittung im Erfolgsfall) → Datei-Speichern-Dialog (binäre Datei)
oder
- ein Array nach einem Datei-Öffnen-Dialog aus der Export-Datei importieren.
- Abschließend werden die Array-Elemente des importierten Arrays in der IDE-Konsole angezeigt.

Das im Projekt verwendete Array ist ein aus einer selbst geschriebenen Klasse abgeleitetes Array und die Elemente des Arrays sind Objekte dieser Klasse. Die Klasse *CDS* ist eine (reine) Datenstruktur ohne eigene Methoden und repräsentiert hier zum Beispiel einen Datensatz (Rekord) in einer Kursliste.

Quelltext von *CDS.class*:

```
' Gambas class file
Public JGS As Integer
Public GebDatum As Date
Public Vorname As String
Public Nachname As String
Public Kurs1 As String
Public Kurs2 As String
```

Das vollständige Projekt finden Sie im Download-Bereich und hier den Quelltext von *FMain.class*:

```
[1] ' Gambas class file
[2]
[3] Public arrayKL As New CDS[] ' Objekt mit dem Namen arrayKL der Array-Klasse CDS[] erzeugen
[4] Public hCDS As CDS ' Variable vom Typ CDS anlegen
[5]
[6] Public Sub Form_Open()
[7]     FMain.Center
[8]     FMain.Resizable = False
[9]     If Not Exist(Application.Path &/ "array.dat") Then
[10]         SetEnabled(True, False, False)
[11]     Else
[12]         SetEnabled(True, False, True)
[13]     Endif
[14] End ' Form_Open()
[15]
[16] Public Sub btnSetClassArray_Click()
[17]     Dim iCount As Integer
[18]     Dim aNames As String[] = ["Adler", "Bär", "Dachs", "Fuchs", "Meise", "Uhu"]
[19]     Dim aSurNames As String[] = ["Anna", "Bruno", "Doris", "Fred", "Maria", "Udo"]
[20]     Dim aCourses1 As String[] = ["Mathematik", "Geschichte", "Astronomie", "Biologie", "Informatik"]
[21]     Dim aCourses2 As String[] = ["Deutsch", "Physik", "Englisch", "Kunst", "Musik"]
[22]
[23]     Randomize
[24]
[25]     For iCount = 0 To 9
[26]         hCDS = New CDS
[27]         hCDS.JGS = CInt(Rnd(11, 13)) ' 11..12
[28]         ' Zufallsdatum: Jahr: passend zur JGS, Monat: 1..12, Tag: 1..28 (das passt immer...)
[29]         hCDS.GebDatum = Date(Int(Rnd(Year(Now())-18, Year(Now())-16)), Int(Rnd(1,13)), Int(Rnd(1,29)))
```

```

[30]     hCDS.Vorname = aSurNames[Int(Rnd(0, aSurNames.Count))]
[31]     hCDS.Nachname = aNames[Int(Rnd(0, aNames.Count))]
[32]     hCDS.Kurs1 = aCourses1[Int(Rnd(0, aCourses1.Count))]
[33]     hCDS.Kurs2 = aCourses2[Int(Rnd(0, aCourses2.Count))]
[34]     arrayKL.Add(hCDS)
[35] Next ' iCount
[36]
[37] ' Anzeige aller Datensätze (Konsole der IDE) zur Kontrolle
[38] ShowCDSArrayElements(arrayKL)
[39] SetEnabled(True, True, False)
[40]
[41] End ' btnClassArray_Click()
[42]
[43] Public Sub btnArrayDatenExport_Click()
[44]     Dim iCount As Integer
[45]     Dim hFile As File
[46]
[47] ' Speichern aller Elemente des abgeleiteten Arrays 'arrayKL' in einer Datei
[48] Dialog.Title = "Array-Export in eine binäre Datei"
[49] Dialog.Filter = ["*.dat", "Daten-Dateien"]
[50] Dialog.Path = Application.Path & "array.dat"
[51]
[52] If Dialog.SaveFile() Then Return
[53] hFile = Open Dialog.Path For Write Create
[54]
[55] ' Serialisierung der Objekt-Eigenschaften
[56] For iCount = 0 To arrayKL.Max
[57]     With arrayKL[iCount]
[58]         Write #hFile, .JGS As Integer
[59]         Write #hFile, .GebDatum As Date
[60]         Write #hFile, .Vorname As String
[61]         Write #hFile, .Nachname As String
[62]         Write #hFile, .Kurs1 As String
[63]         Write #hFile, .Kurs2 As String
[64]     End With
[65] Next
[66]
[67] Close #hFile
[68] SetEnabled(True, True, True)
[69]
[70] End ' btnDatenExport_Click()
[71]
[72] Public Sub btnArrayDatenImport_Click()
[73]     Dim iCount As Integer
[74]     Dim hFile As File
[75]
[76] ' Einlesen aller Elemente des abgeleiteten Arrays 'arrayKL' aus einer Datei
[77] Dialog.Title = "Array-Import aus einer binären Datei"
[78] Dialog.Filter = ["*.dat", "Daten-Dateien"]
[79] Dialog.Path = Application.Path & "array.dat"
[80]
[81] If Dialog.OpenFile() Then Return
[82] hFile = Open Dialog.Path For Read
[83]
[84] arrayKL.Clear
[85] ' De-Serialisierung
[86] While Not Eof(hFile)
[87]     hCDS = New CDS
[88]     hCDS.JGS = Read #hFile As Integer
[89]     hCDS.GebDatum = Read #hFile As Date
[90]     hCDS.Vorname = Read #hFile As String
[91]     hCDS.Nachname = Read #hFile As String
[92]     hCDS.Kurs1 = Read #hFile As String
[93]     hCDS.Kurs2 = Read #hFile As String
[94]     arrayKL.Add(hCDS)
[95] Wend
[96]
[97] ' Anzeige aller Datensätze (IDE-Konsole)
[98] ShowCDSArrayElements(arrayKL)
[99] Close #hFile
[100] SetEnabled(True, True, True)
[101]
[102] End ' btnDatenImport_Click()
[103]
[104] Private Sub ShowCDSArrayElements(aArray As CDS[])
[105] ' Anzeige aller Datensätze (Konsole der IDE)
[106] For Each hCDS In aArray
[107]     Print hCDS.JGS,
[108]     Print Format(hCDS.GebDatum, "dd.mm.yyyy"),
[109]     Print hCDS.Vorname,
[110]     Print hCDS.Nachname,
[111]     Print hCDS.Kurs1,
[112]     Print hCDS.Kurs2
[113] Next ' hCDS

```

```

[114] Print
[115] End ' ShowCDSArrayElements(aArray As CDS[])
[116]
[117] Public Sub SetEnabled(bAR As Boolean, bEX As Boolean, bIM As Boolean)
[118]     btnSetClassArray.Enabled = bAR
[119]     btnArrayDatenExport.Enabled = bEX
[120]     btnArrayDatenImport.Enabled = bIM
[121] End ' SetEnabled(..)

```

Kommentare:

- Die Klasse CDS[] existiert eigentlich nicht. Da sie aber mit "[]" endet und es eine Klasse CDS gibt, erstellt der Interpreter automatisch *on-the-fly* eine von CDS abgeleitete Klasse CDS[], die ein Array von 'CDS'-Objekten ist. So wird zum Beispiel in der Zeile 3 ein neues Objekt dieser Array-Klasse erzeugt:

```
Public arrayKL As New CDS[]
```

- In den Zeilen 16 bis 41 wird das Array von 'CDS'-Objekten – arrayKL – mit 10 Datensätzen vom Typ CDS gefüllt. Zuerst wird in der For-Next-Kontroll-Struktur *sukzessive* Platz für ein neues Array-Element im Array arrayKL geschaffen (Zeile 27), dann ein neuer, leerer Datensatz vom Typ CDS angelegt und sofort mit Zufallswerten gefüllt. Abschließend (Zeile 34) wird der aktuelle Datensatz als Element in das Array arrayKL eingefügt. In der Zeile 38 werden alle zehn Datensätze als Inhalt des Arrays arrayKL formatiert zur Kontrolle ausgegeben und in der Konsole der Gmbas-IDE angezeigt.
- Der Export des abgeleiteten Arrays arrayKL erfolgt in den Zeilen 43 bis 70. Wird der Datei-Speichern-Dialog erfolgreich beendet, wird eine neue Datei mit dem übergebenen Pfad angelegt und zum Schreiben geöffnet.
- Für alle zehn Elemente des Arrays arrayKL vom Objekt-Typ CDS werden die 6 Objekt-Eigenschaften nacheinander in den Datei-Stream geschrieben. Dann wird die Datei geschlossen.
- In der Zeile 84 werden alle Elemente des existierenden Arrays arrayKL mit der Clear-Methode gelöscht.
- Beim Import in den Zeilen 72 bis 102 steht nach dem erfolgreichen Datei-Öffnen-Dialog der Pfad zur Datei mit dem exportierten Array zur Verfügung und die Datei wird zum Lesen geöffnet.
- Solange das Ende der Datei noch *nicht* erreicht ist, werden die Elemente des Arrays arrayKL als CDS-Objekte seriell aus den 6 Werten der Objekt-Eigenschaften generiert und in das Array arrayKL eingefügt (Zeilen 86 bis 95).
- Abschließend werden in der Zeile 104 alle zehn Datensätze als Inhalt des (importierten) Arrays arrayKL formatiert ausgegeben und in der Konsole der Gmbas-IDE angezeigt:

| | | | | | |
|----|------------|-------|-------|------------|----------|
| 11 | 17.08.1997 | Anna | Bär | Mathematik | Musik |
| 11 | 23.11.1996 | Maria | Bär | Astronomie | Physik |
| 12 | 19.02.1997 | Bruno | Bär | Astronomie | Kunst |
| 12 | 03.12.1996 | Anna | Dachs | Astronomie | Musik |
| 12 | 05.07.1997 | Maria | Bär | Geschichte | Deutsch |
| 12 | 19.09.1996 | Doris | Uhu | Geschichte | Englisch |
| 12 | 03.09.1996 | Maria | Bär | Mathematik | Englisch |
| 12 | 01.09.1996 | Udo | Meise | Informatik | Deutsch |
| 12 | 24.09.1996 | Doris | Uhu | Informatik | Deutsch |
| 11 | 24.10.1997 | Doris | Meise | Mathematik | Physik |