

7.4.8.1 Export und Import nativer Arrays

Um ein `natives Array` in einer Datei abzuspeichern oder den Inhalt einer derartigen Datei wieder in ein Array einzulesen (Daten-Import) können Sie ein gambas-spezifisches und binäres Datei-Format einsetzen. Gambas 3 kann einen einfachen `Array-Export` und `Array-Import` realisieren, wenn das Array über `Write` serialisiert in einen Datei-Stream geschrieben oder mit `Read` aus diesem gelesen wird:

```
If Dialog.SaveFile() Then Return
hFile = Open Dialog.Path For Write Create
Write #hFile, aNames As ARRAY
Close #hFile
```

```
If Dialog.OpenFile() Then Return
hFile = Open Dialog.Path For Read
aNames = Read #hFile As ARRAY
Close #hFile
```

Im Projekt wird folgende Strategie verfolgt:

- Zuerst wird ein natives Array angelegt und mit festen Daten 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, gambas-spezifische 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.

Im Quelltext wird die Strategie umgesetzt. Das vollständige Projekt finden Sie im Download-Bereich.

```
' Gambas class file

Public aNames As String[]

Public Sub Form_Open()
  FMain.Center
  FMain.Resizable = False
  If Not Exist(Application.Path &/ "array.dat") Then
    SetEnabled(True, False, False)
  Else
    SetEnabled(True, False, True)
  Endif
End ' Form_Open()

Public Sub btnSetArray_Click()
  aNames = ["Bär", "Meise", "Adler", "Uhu", "Dachs", "Fuchs"]
  ShowArrayElements(aNames)
  SetEnabled(True, True, False)
End ' btnClassArray_Click()

Public Sub btnArrayDatenExport_Click()
  Dim hFile As File

  ' Speichern aller Elemente des Arrays 'aNames' in einer binären Datei
  Dialog.Title = "Array-Export in eine binäre Datei"
  Dialog.Filter = ["*.dat", "Daten-Dateien"]
  Dialog.Path = Application.Path &/ "array.dat"

  If Dialog.SaveFile() Then Return
  hFile = Open Dialog.Path For Write Create
  Write #hFile, aNames As ARRAY

  Close #hFile

  SetEnabled(True, True, True)
End ' btnDatenExport_Click()

Public Sub btnArrayDatenImport_Click()
  Dim hFile As File

  ' Einlesen aller Elemente des Arrays 'aNames' aus einer binären Datei
  Dialog.Title = "Array-Import aus einer binären Datei"
  Dialog.Filter = ["*.dat", "Daten-Dateien"]
  Dialog.Path = Application.Path &/ "array.dat"

  If Dialog.OpenFile() Then Return
```

```
hFile = Open Dialog.Path For Read
aNames = Read #hFile As ARRAY

Close #hFile

ShowArrayElements(aNames)
SetEnabled(True, True, True)

End ' btnDatenImport_Click()

Private Sub ShowArrayElements(aArray As String[])
    Dim sElement As String

    ' Anzeige aller Datensätze (Konsole IDE)
    For Each sElement In aArray
        Print sElement,
    Next ' sElement
    Print

End ' ShowArrayElements(aArray As String[])

Public Sub SetEnabled(bAR As Boolean, bEX As Boolean, bIM As Boolean)

    btnSetArray.Enabled = bAR
    btnArrayDatenExport.Enabled = bEX
    btnArrayDatenImport.Enabled = bIM

End ' SetEnabled(..)
```

Ausgaben in der Konsole der Gambas-IDE:

```
Bär    Meise  Adler  Uhu    Dachs  Fuchs  ' Elemente Original-Array
Bär    Meise  Adler  Uhu    Dachs  Fuchs  ' Elemente Import-Array
```