

5.7 Interfaces

Ein Interface ist in Gambas eine Schnittstelle zwischen einer Klasse und dem Interpreter. Interfaces gibt es gegenwärtig nur für native Klassen – also Klassen, die nicht in Gambas geschrieben sind. Hinzu kommt, dass nur Gambas-Entwickler Interfaces nutzen können und bisher sehr wenige Klassen Interfaces implementieren.

Die Beschreibung der Effekte bei der Nutzung von Interfaces soll am Beispiel der Klasse Matrix gezeigt werden, denn die Klasse Matrix (gb.gsl) setzt intern sowohl das *_convert-Interface*, als auch das *_operator-Interface* ein.

Beispiel 1:

```
Private Sub CreateAndShowMatrix()
    mMatrix = New Matrix(Columns.Value, Rows.Value, True) ' Spalten, Zeilen
    mMatrix = [[1, 1, 1, 7 + 2i, 8], [2, 2, 2, 4, 5], [1, 0, 3, 9, 5], [-1, -2i, -3, 4, 5 + 1i]] ***
    ' Matrix-Elemente in der TableView anzeigen
    MatrixToTableView(mMatrix)
End ' CreateMatrix()
```

Das *_convert-Interface* erlaubt es, automatische Datentyp-Konvertierungen durchzuführen. Eine Klasse, die das *_convert-Interface* implementiert, akzeptiert damit bestimmte Datentypen als konvertierbar und stellt ein Verfahren bereit, um geeignete Daten zu konvertieren. In diesem Fall wird das Array von Arrays (im Hintergrund) zu einer Matrix konvertiert und die Zuweisung *** gelingt. Das *_operator-Interface* spezifiziert das Verhalten von Objekten, wenn Operatoren auf sie angewendet werden. Immer dann, wenn zwei Objekte (oder ein Objekt und einen Wert nativen Datentyps) mit einem Operator verknüpft werden, so ist das *_operator-Interface* im Spiel.

Den Gambas-Nutzer interessiert nur diese Tatsache, dass man zum Beispiel `matrix = [[z1, z2],[z3, z4]]` schreiben kann und der Interpreter das in bestimmten Zuweisungskontexten als Matrix erkennt.

Ebenfalls von der Klasse Matrix (gb.gsl) wird das *_operator-Interface* teilweise implementiert, das die Anwendbarkeit von bestimmten Operatoren auf Matrizen regelt. Damit gelingt es sicher

- eine Matrix mit einem Skalar zu multiplizieren oder
- ein Matrix-Matrix-Produkt zu berechnen – mit einer Matrix als Ergebnis.

Beispiel 2:

```
Public Sub btnMatrixMatrixProdukt_Click()
    Dim mMatrix As New Matrix(3, 3, False) ' Nur reelle Koeffizienten
    Dim mResult As Matrix

    mMatrix = [[1, 5, 6], [4, -3.3, -9], [7, 2, 8.8]]
    Print mMatrix.ToString(True) ' True → Lokale Spracheinstellung aktiv

    mMatrix = 2 * mMatrix ' Skalar-Matrix-Produkt
    Print mMatrix.ToString(True)

    mResult = mMatrix * mMatrix.Trans() ' Matrix-Matrix-Produkt
    Print mResult.ToString(True)
End
```

Die Anzeige in der Konsole der Gambas-IDE zeigt diese Ergebnisse:

```
[[1 5 6][4 -3,3 -9][7 2 8,8]]
[[2 10 12][8 -6,6 -18][14 4 17,6]]
[[248 -266 279,2][-266 431,56 -231,2][279,2 -231,2 521,76]]
```

Es gibt noch zwei weitere spezialisierte Interfaces: Paint und PaintMatrix, die etwas mit Operationen zum Zeichnen bei ausgewählten Klassen zu tun haben.