

BASIC

Die Übersetzung des Programms in BASIC war bezüglich der Initialisierung der Arrays etwas abweichend. Uninitialisierte Arrays wie etwa `used` oder `b` werden mit dem richtigen Typ deklariert. Bei den Arrays, die einen Anfangswert erfordern (`von`, `bi`s) mußte man die Anfangswerte im Rahmen einer `For-Next`-Schleife setzen, was nicht besonders elegant aussieht. In diesem Beispiel wurden daher die Variablen `von` und `bi`s als Typ Variant gewählt, der die Initialisierung mit der Funktionsarray erlaubt. Für optimale Laufgeschwindigkeit sollte man aber den Typ Variant vermeiden.

Als Testausgabe wurde das Objekt `Debug` mit der Methode `Print` verwendet. Zur Beschleunigung des Ablaufs wurde aber die laufende Ausgabe jedes Versuchs ausgeklammert. Es werden nur die gültigen Lösungen ausgegeben.

```
' VISUAL BASIC for APPLICATIONS (ACCESS)
Option Explicit
Const maxb As Integer = 9
Dim von As Variant
Dim bi s As Variant
Dim used(0 To 9) As Boolean
Dim b(1 To maxb) As Long
Dim i As Byte
```

```
Public Sub test(i As Integer)
Dim j, k As Byte
For j = von(i) To bis(i)
If Not used(j) Then
b(j) = j
used(j) = True
If i < maxb Then test (i + 1)
Else
' For k = 1 To maxb
' Debug.Print (b(k));
' Next k
' Debug.Print
If ((b(1) * 10 + b(2)) * 10 + b(3)) * 10 + b(4) + _
((b(5) * 10 + b(6)) * 10 + b(7)) * 10 + b(2) = _
(((b(5) * 10 + b(6)) * 10 + b(3)) * 10 + b(2)) * 10 + b(8) Then
For k = 1 To maxb
Debug.Print (b(k));
Next k
Debug.Print " OK"
End If
End If
used(b(i)) = False
Next j
End Sub

Public Sub testtest()
von = Array(0, 1, 0, 0, 0, 1, 0, 0, 0, 0)
bi s = Array(9, 9, 9, 9, 9, 9, 9, 9, 9, 9)
test (1)
End Sub
□
```

CDROM „Delphi Programmierertools“

International Thomson Publishing, ISBN 3-8266-0230-7, öS 233,-

Hans Lohninger

Das CDROM „Delphi Programmierertools“ enthält eine bunte Mischung aus Free- und Shareware und einigen wenigen Demos („Crippleware“), insgesamt 80 Produkte. Die enthaltene Software kann man grob in zwei Kategorien einteilen: zum einen Delphi-Komponenten und -Erweiterungen, und zum anderen allgemeine Zubehör-Software für Programmierer.

Mit den 80 Produkten (die übrigens das CDROM nur zu 8 % belegen, der Rest ist leeres Plastik) wird ein 'CD-Manager' mitgeliefert, der helfen soll, einen besseren Überblick zu den angebotenen Produkten zu bekommen. Man kann sich mit Hilfe des CD-Managers die jeweiligen (deutschen) Hilfetexte ansehen und bei Interesse das jeweilige Software-Produkt auf die Harddisk kopieren.

Dieser CD-Manager ist im Prinzip eine gute Sache. Allerdings ist er wohl etwas zu mager geraten, sodaß der genervte Benutzer sich schnell ein kleines Batch-Programm bastelt, das alle Hilfetexte in ein File kopiert - womit der Überblick wieder da ist. Störend ist auch, daß der CD-

Manager anscheinend nicht auf Systemen mit verschiedenen Grafik-Auflösungen getestet worden ist: Bei einer Auflösung von 1024x768 passen einige Elemente der Oberfläche des CD-Managers nicht mehr zusammen.

Die Auswahl der auf der CD enthaltenen Produkte erfolgte meiner Ansicht nach eher nachlässig. So gibt es insgesamt zwar 4 verschiedene Texteditoren im Angebot, allerdings ist zum Beispiel der sehr gute PFE nur in der 32-Bit Version enthalten. Verglichen mit dem Platz, der noch auf der CD frei ist (immerhin runde 600 MB), hätten die Produzenten durchaus noch das eine oder andere Delphi-Juwel mit auf das CDROM nehmen können.

Der Wert des CDROMs wird auch insofern relativiert, als man sich für dessen stolzen Preis von 233,- öS bei entsprechender Hardware-Ausstattung ca. 60 MB aus dem Internet holen kann - also um circa 10 MB mehr als auf der CD vorhanden ist. □

Wie die verschiedenen Programmierer ihre Fahrräder bauen

ADA-Programmierer bauen ein viereckiges Rad und passen alle Straßen an.

ALGOL-Programmierer weigern sich, Räder zu bauen, weil es sie irgendwohin bringen könnte.

APL-Programmierer schweben in höheren Sphären, sie brauchen keine Räder.

ASSEMBLER-Programmierer bauen tausende von Rädern, keine 2 passen auf eine Achse.

BASIC-Programmierer bauen nur ein Rad, aber finden keine Achse dazu.

COBOL-Programmierer bauen TAUSEND-RÄDRIGE-TRANSPORT-MODULE und verbieten das Gehen.

FUNKTIONAL-Programmierer rufen eine Funktion HOLZ auf und hoffen, ein Rad zu bekommen.

FORTH-Programmierer bauen Räder und vergessen, wo sie sie gestapelt haben.

FORTRAN-Programmierer werden wahnsinnig bei der Suche nach Rädern, die mit "I" beginnen.

LOGO-Programmierer bauen kleine rote Autos.

LISP-Programmierer (bauen Räder mit (Rädern mit/Rädern mit (Rädern mit (dem was LISP-Programmierer bauen)))).

MICRO-Programmierer wissen nicht, daß Räder existieren.

PASCAL-Programmierer erklären das Gehen zur Tugend.

PL/I-Programmierer setzen ein Team ein, um eine Räderfabrik zu entwerfen, die Räder der falschen Größe bauen wird.

RPG-Programmierer haben ein Rad, schade, daß es viereckig ist.

SYSTEM-Analytiker sind viel zu beschäftigt, Räder zu suchen, um eins zu bauen.

TEXTVERARBEITUNGS-Benutzer bauen Räder.

TOS-Benutzer bleiben wo sie sind, sie sind das Warten gewohnt.

UNIX hat unter irgendeiner Schale sicher irgendwo ein Verzeichnis von Rädern.