

10.3.5 BREAK und CONTINUE

Break und Continue sind *besondere* Anweisungen *innerhalb* von Loop-Kontroll-Strukturen.

- Mit *Break* wird die aktuelle Loop-Kontroll-Struktur sofort abgebrochen und verlassen. Die Ausführung des Programms wird mit der nächsten Anweisung nach der Loop-Kontroll-Struktur fortgesetzt.
- Die Anweisung *Continue* bewirkt, dass der aktuelle Schleifen-Durchlauf abgebrochen wird und ein neuer Durchlauf begonnen wird.

10.3.5.1 Beispiel 1

Eine Datei wird zum Lesen geöffnet. Dann wird der Inhalt der Datei zeilenweise ausgelesen und jede Zeile – unter bestimmten Voraussetzungen – in einem Array gespeichert:

```
hFile = OPEN sFilePath For Input
WHILE NOT Eof(hFile)           ' Solange noch nicht das Datei-Ende erreicht ist ...
  LINE INPUT #hFile, sLine     ' ... wird eine Zeile aus der Datei ausgelesen
  sLine = Trim(sLine)         ' ... werden Leerzeichen am Ende und am Anfang der Zeile entfernt
  IF NOT sLine THEN CONTINUE   ' ... wird eine leere Zeile übergangen → weiter mit neuem Schleifen-Durchlauf
  IF InStr("#;", Left$(sLine)) THEN CONTINUE ' ... wird ein Kommentar übergangen → weiter ...
  ...
  aFileArray.Add(sLine)
WEND ' NOT Eof(hFile)
CLOSE #hFile
```

- Es ist sichergestellt, dass keine Leerzeilen oder Kommentar-Zeilen dem FileArray hinzugefügt werden.
- Für diese beiden Fälle wird der Schleifen-Durchlauf noch vor dem Befehl `aFileArray.Add(sLine)` abgebrochen und eine neue Zeile aus der Datei ausgelesen – solange noch Zeilen ausgelesen werden können.

10.3.5.2 Beispiel 2

Das Beispiel 2 greift einen Fall auf, in dem *zwei verschachtelte* For-Kontroll-Strukturen bei der Erfüllung einer bestimmten Bedingung verlassen werden sollen:

```
Dim iN, iI As Integer
Dim bBreak As Boolean = False

For iN = 0 To aArray.Max
  For iI = 0 To Array[iN].Max
    If iX = aArray[iN][iI] Then
      bBreak = True
      Break
    Endif
  Next
  If bBreak Then Break
Next
```

- Zuerst wird die boolsche Variable `bBreak` auf `True` gesetzt und mit `Break` die innere For-Kontroll-Struktur verlassen.
- In der äußeren For-Kontroll-Struktur wird der Wert der Variablen `bBreak` ausgewertet.
- Für den Fall `bBreak=True` wird auch die äußere For-Kontroll-Struktur mit `Break` verlassen.

10.3.5.3 Beispiel 3

Im dritten Beispiel werden aus einem Array nur die Zahlen in einem anderen Array gespeichert, die ganzzahlige Kubikwurzeln sind:

```
Private Function CubicNumbers(aArray As Integer[]) As Integer[]
  Dim iNumber As Integer
  Dim aResult As New Integer[]

  For Each iNumber In aArray
    If Frac(Cbr(iNumber)) <> 0 Then CONTINUE ' Zahlen überspringen, deren Kubikwurzel keine Ganzzahl ist
    aResult.Add(iNumber)
  Next
  Return aResult
End
```