

10.3.3 REPEAT-UNTIL-Kontroll-Struktur

Auch die REPEAT-UNTIL-Kontroll-Struktur ist *eine* Form der Loop-Kontroll-Strukturen.

10.3.3.1 Syntax

Syntax für die REPEAT-UNTIL-Kontroll-Struktur:

```
REPEAT
  <Anweisung(en)>
UNTIL Expression
```

10.3.3.2 Hinweise zur Syntax

- Die Schleife wird solange wiederholt, bis *Ausdruck* wahr ist.
- Da die Prüfung der Abbruchbedingung am Ende der REPEAT-UNTIL-Kontroll-Struktur erfolgt, wird die Anweisungsfolge *mindestens einmal* ausgeführt.

10.3.3.3 Beispiel 1

Das Beispiel bezieht sich auf ein Projekt im Kapitel → 18.12 ListContainer. Mit der Anweisung in der vierten Zeile der u.a. Prozedur wird der HTTP-Client in die Spur geschickt, um Daten vom angegebenen Server zu laden. Danach wird in der Warte-Schleife solange gewartet, bis der Download abgeschlossen ist und der HTTP-Client (wieder) inaktiv ist:

```
Public Sub cmbFeedSource_Click()
  ListContainer.Clear
  myHttpClient.URL = Feeds[cmbFeedSource.Index]
  myHttpClient.Get
  Repeat
    Wait 0.01
  Until myHttpClient.Status = Net.Inactive
' Automatisch den ersten Feed auswählen - wenn vorhanden
If ListContainer.Count Then ListContainer.Index = 0
End ' cmbFeedSource_Click()
```

10.3.3.4 Beispiel 2

Die näherungsweise Bestimmung einer Nullstelle einer Funktion $f(x)$ nach dem Sekanten-Näherungsverfahren verwendet die REPEAT-UNTIL-Kontroll-Struktur. Fehlerhafte Startwerte für die beiden Argumente x_a und x_e werden erkannt und angezeigt. Es sollte gesichert werden, dass die Funktion im abgeschlossenen Intervall $[x_a | x_e]$ genau eine Nullstelle besitzt.

Nummer	Argument	Funktionswert $y = f(x) = x^2 - 4$
8	1,9708231	-0,11585642980765
9	1,9863309	-0,05448946535669
10	1,9936095	-0,02552135997737
11	1,9970152	-0,01193029029143
12	1,9986065	-0,00557190285743
13	1,9993496	-0,00260118781812
14	1,9996965	-0,0012140982134
15	1,9998583	-0,00056662503137
16	1,9999339	-0,00026443500393
17	1,9999691	-0,00012340517737
18	1,9999856	-5,7589556565762E-5

Abbildung 10.3.3.4.1: Näherungsverfahren mit einstellbarer Genauigkeit

Der Quelltext wird vollständig angegeben:

```
' Gambas class file
Public Sub Form_Open()
  FMain.Center
  FMain.Resizable = False
End ' Form_Open()
```

```

Public Function f(x_wert As Float) As Float
    Return (x_wert * x_wert) - 4
End ' Function f(x_wert As Float)

Public Sub btnNullstelleBerechnenUndAusgeben_Click()
    Dim iCount As Integer
    Dim x1 As Float      ' Linke Intervallgrenze
    Dim x2 As Float      ' Rechte Intervallgrenze
    Dim fEpsilon As Float ' Abbruchkriterium
    Dim y1, y2, fNullstelle, fDifferenzenquotient As Float

    x1 = CFloat(txb_x1.Text)
    x2 = CFloat(txb_x2.Text)
    fEpsilon = CFloat(txb_epsilon.Text)
    If f(x1) * f(x2) > 0 Then
        Message.Error("Die Startwerte x1 und x2 sind nicht zulässig!")
        txb_x2.SetFocus
        Return
    Endif ' f(x1) * f(x2) > 0
    txaTabelle.Clear

    Repeat
        y1 = f(x1)
        y2 = f(x2)
        fDifferenzenquotient = (x2 - x1) / (y2 - y1)
        fNullstelle = x1 - (y1 * fDifferenzenquotient)
        txaTabelle.Text = txaTabelle.Text & " " & Str(iCount) & Chr$(9) & Format$(fNullstelle, "0.000000")
            & Chr$(9) & " " & Str(f(fNullstelle)) & Chr$(10)
        If f(x1) * f(fNullstelle) > 0 Then
            x1 = fNullstelle
        Else
            x2 = fNullstelle
        Endif
        Inc (iCount)
    Until Abs(f(fNullstelle)) < fEpsilon

End ' btnNullstelleBerechnenUndAusgeben_Click()

Public Sub xGroup_Change()
    txaTabelle.Clear
End ' xGroup_Change()

```