

10.3.4 WHILE-WEND-Kontroll-Struktur

Auch die in den folgenden Abschnitten vorgestellte WHILE-WEND-Kontroll-Struktur ist eine weitere Form der Loop-Kontroll-Strukturen.

10.3.4.1 Syntax

Syntax für die WHILE-WEND-Kontroll-Struktur:

```
WHILE Expression
  <Anweisung(en)>
WEND
```

10.3.4.2 Hinweise zur Syntax

- Die Schleife wird wiederholt, solange der *Ausdruck* wahr ist.
- Da die Prüfung der Abbruchbedingung am Anfang erfolgt, wird die WHILE-WEND-Kontroll-Struktur *nie* ausgeführt, wenn der (Start-)Ausdruck bereits falsch ist.
- 'DO WHILE'-LOOP-Kontroll-Struktur ist eine äquivalente Struktur zur allgemeinen DO-LOOP-Kontroll-Struktur → Kapitel 10.3.1 Do..Loop.

10.3.4.3 Beispiel 1

Eine Datei wird zum Lesen geöffnet. Dann wird der Inhalt der Datei zeilenweise ausgelesen und jede Zeile in einem Array gespeichert – solange das Ende der Datei noch nicht erreicht ist und noch Zeilen ausgelesen werden können:

```
hFile = Open sRubrikPfad For Input
While NOT Eof(hFile)
  Line Input #hFile, sZeile
  aSuchDateiMatrix.Add(sZeile)
Wend
Close #hFile
aSuchDateiMatrix.Sort(0)
```

10.3.4.4 Beispiel 2

Die Aufgabe: 'Solange noch ein Block von zwei aufeinander folgenden Leerzeichen in einer Zeichenkette existiert, sollen alle Blöcke von zwei Leerzeichen durch ein einziges Leerzeichen ersetzt werden' hat eine einfache Lösung:

```
While InStr(sTestString, " ")
  sTestString = Replace$(sTestString, " ", " ")
Wend
```

10.3.4.5 Beispiel 3

Das Beispiel 3 demonstriert den Einsatz von vier unterschiedlichen Loop-Kontroll-Strukturen für die Bearbeitung der gleichen Aufgabe:

- FOR..TO..NEXT-Kontroll-Struktur
- REPEAT..UNTIL-Kontroll-Struktur
- WHILE..WEND-Kontroll-Struktur
- DO..WHILE..LOOP-Kontroll-Struktur (Alternative)

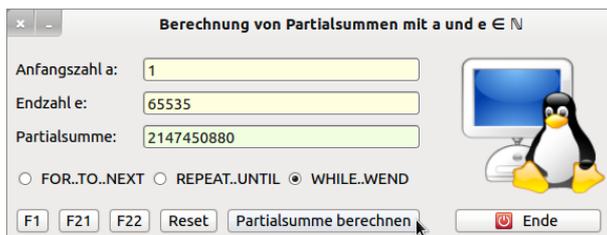


Abbildung 10.3.4.5.1: Berechnung von Partialsummen nach 3 Varianten

Der Quelltext wird nur in den relevanten Auszügen angegeben:

```

...
Public Sub btnPartialsommeBerechnen_Click()
    If KontrolleEingabedaten() = "Dateneingabe fehlerfrei!" Then
        If optVariante1.Value = True Then
            Variante_1(iAnfangszahl, iEndzahl) ' Kontrollstruktur FOR..TO..NEXT
        Else If optVariante2.Value = True
            Variante_2(iAnfangszahl, iEndzahl) ' Kontrollstruktur REPEAT..UNTIL
        Else
            Variante_3(iAnfangszahl, iEndzahl) ' Kontrollstruktur WHILE..WEND
        Endif
    Endif ' KontrolleEingabedaten()
End ' btnPartialsommeBerechnen

Private Sub Variante_1(iAnfang As Integer, iEnde As Integer) ' FOR..TO..NEXT
    Dim iPartialsomme As Integer = 0
    Dim iSummand As Integer = iAnfang
    Dim iCount As Integer = 0

    For iCount = 1 To (iEnde - iAnfang + 1) Step 1
        iPartialsomme = iPartialsomme + iSummand
        ' Überlauf wegen Bereichsüberschreitung für Integer?
        If (IsIntegerRange(iPartialsomme) = False) And If (bSpezialfall22 = False) Then
            txtPartialsomme.Text = "FEHLER! SUMME " & String.Chr(8713) & " INTEGER"
            Return
        Endif ' Partialsomme-Überlauf ?
        Inc iSummand
    Next ' iCount

    txtPartialsomme.Text = Str(iPartialsomme)
End ' Variante_1 - Kontrollstruktur FOR..TO..[STEP]..NEXT

Private Sub Variante_2(iAnfang As Integer, iEnde As Integer) ' REPEAT..UNTIL
    Dim iPartialsomme As Integer = 0
    Dim iSummand As Integer = iAnfang

    Repeat
        iPartialsomme = iPartialsomme + iSummand
        If (IsIntegerRange(iPartialsomme) = False) And If (bSpezialfall22 = False) Then
            txtPartialsomme.Text = "FEHLER! SUMME " & String.Chr(8713) & " INTEGER"
            Return
        Endif ' Partialsomme-Überlauf ?
        Inc iSummand
    Until (iSummand > iEnde)

    txtPartialsomme.Text = Str(iPartialsomme)
End ' Variante_2 - Kontrollstruktur REPEAT..UNTIL

Private Sub Variante_3(iAnfang As Integer, iEnde As Integer) ' WHILE..WEND
    Dim iPartialsomme As Integer = 0
    Dim iSummand As Integer = iAnfang

    While (iSummand < iEndzahl + 1)
        iPartialsomme = iPartialsomme + iSummand
        If (IsIntegerRange(iPartialsomme) = False) And If (bSpezialfall22 = False) Then
            txtPartialsomme.Text = "FEHLER! SUMME " & String.Chr(8713) & " INTEGER"
            Return
        Endif ' Partialsomme-Überlauf ?
        Inc iSummand
    Wend ' While End

    txtPartialsomme.Text = Str(iPartialsomme)

' Alternative Variante 3:
' Do While (iSummand < iEnde + 1)
'     iPartialsomme = iPartialsomme + iSummand
'     If (IsIntegerRange(iPartialsomme) = False) And If (bSpezialfall22 = False) Then
'         txtPartialsomme.Text = "FEHLER! SUMME " & String.Chr(8713) & " INTEGER"
'         Return
'     Endif ' Partialsomme-Überlauf ?
'     Inc iSummand
' Loop
' txtPartialsomme.Text = Str(iPartialsomme)
End ' Variante_3 - Kontrollstruktur WHILE..WEND

```