

3 Schleifen

In vielen Aufgabenstellungen muss man Anweisungen mehrfach wiederholen. Es wäre nun zu umständlich die Anweisungen zig -fach hinzuschreiben. Oft ist es auch zur Programmierzeit nicht absehbar, wie oft die Anweisungen wiederholt werden müssen. Wenn zum Beispiel in eine Inputbox eine Zahl eingeben werden soll und der Benutzer des Programms (versehentlich) Buchstaben eingibt, so muss die Eingabe wiederholt werden. Wie oft diese Eingabe wiederholt wird ist nicht abzusehen, da der Benutzer ja auch nach einer wiederholten Aufforderung noch immer etwas Falsches eingeben kann.

Deshalb gibt es in jeder Programmiersprache Kontrollstrukturen für Anweisungen die wiederholt werden sollen.

Jede Schleife enthält:

(Zähl)Variable

Startwert	Einer (Zähl)Variablen wird ein Startwert zugewiesen
Laufzeitbedingung	Solange diese wahr ist, wird die Schleife durchlaufen
Aktualisierung der Variable	z.B. Erhöhung des Zählers

3.1 Zählschleife

Zählschleifen werden verwendet, wenn die Anzahl der Wiederholungen vor Beginn der Schleife feststeht.

Bsp. 23: Ausgabe von 5 Message Boxen mit den Zahlen von 1 bis 5

```
Sub ausgabe1Bis5 ()
  Dim i As Integer

  For i = 1 To 5 Step 1
    MsgBox CStr(i)
  Next
End Sub
```

Zähler = 1
Wiederhole solange Zähler <=5
Gib Zähler aus
Erhöhe Zähler um 1
.....

- In unserem Beispiel heißt die **Zählvariable** i.
- Mit i=1 wird der **Startwert** festgelegt.
- to 5 gibt an, wie oft die Schleife durchlaufen werden soll, also bis i größer als 5 ist (**Endwert**).
- Bei Next wird die Zählvariable i um 1 erhöht, (Step 1) und das Programm springt wieder in die erste Anweisung unterhalb von For .

```
For Zähler = Startwert To Endwert Step Schrittweite
  Anweisungen
Next
```

Wird die Schrittweite nicht angegeben, also das Wort Step weggelassen, wird standardmäßig 1 als Schrittweite verwendet.

Sollte in der Schleife rückwärts gezählt werden, so muss Step = -1 betragen.

Wie auch schon bei der Verzweigung, wird der Code innerhalb der Schleife wegen der besseren Lesbarkeit mit einem Tabulator nach rechts gerückt.

Ü 29: Verändern Sie nun den Start und Endwert sowie die Schrittweite um folgende Werte zu erreichen: 10,11,.. 20; 1,3,5,7...15; 10,9,8.....1.

Sollten Sie aus Versehen eine „**Endlosschleife**“ programmieren, so kann man dies mit der Tastenkombination <Strg>+<Pause> unterbrechen.

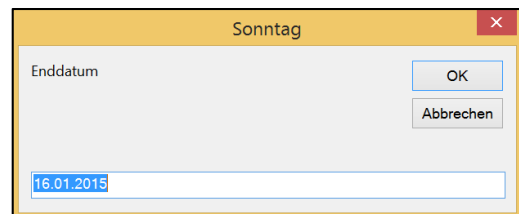
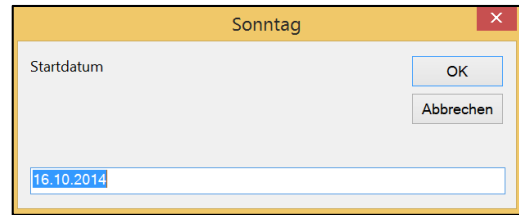
Bsp. 24: Ausgabe aller Sonntage in einem bestimmten Datumsbereich

Es sollen alle Sonntage, die zwischen dem Startdatum und dem Enddatum liegen, ausgegeben werden.

```
Sub Sonntag()
Dim i As Date, start As Date, ende As Date

start = InputBox("Startdatum", "Sonntag", date)
ende = InputBox("Enddatum", "Sonntag", dateAdd("m", 3, date))

For i = start To ende
    If Weekday(i) = vbSunday Then
        MsgBox i, , "Sonntag"
    End If
Next
End Sub
```

**Bsp. 25:** Summe der Zahlen von 1 bis 100

Es sollen die Zahlen von 1 bis zu einem bestimmten Endwert (z.B. 100) aufsummiert werden. Das ist ein „klassisches“ Beispiel. Karl Friedrich Gauß bekam diese Aufgabe im Alter von 10 Jahren von seinem Mathematiklehrer gestellt. Anstatt die mühsame Rechenoperation $1+2+3+4+\dots+100$ durchzuführen überlegte sich Gauß kurzerhand eine Formel, die das Problem allgemein löst. Wir werden es uns noch einfacher machen – wir lassen den Computer für uns rechnen.

```
Sub gaußsum()
Dim sum As Integer
Dim i As Integer

sum = 0
For i = 1 To 100
    sum = sum + i
Next

MsgBox "Die Summe ist: " & CStr(sum)
End Sub
```

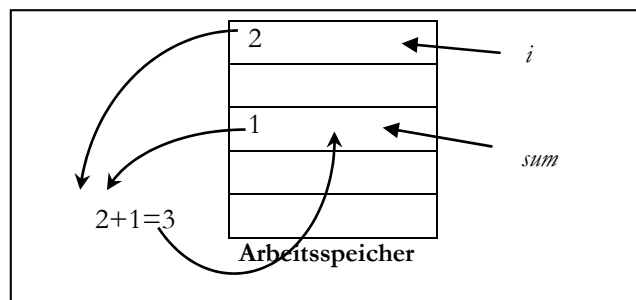
Durchlauf	i	sum+i	sum=sum+i
1	1	0+1	1
2	2	1+2	3
3	3	3+3	6
4	4	6+4	10
..

Die Zeile `sum = sum + i` zeigt



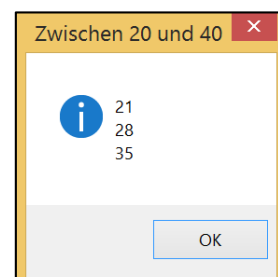
wieder deutlich die Richtung der Zuweisung von rechts nach links. Zuerst wird zur alten Summe i dazugezählt und dann das Ergebnis wieder der Variable `sum` zugewiesen.

Die nebenstehende Abbildung zeigt dies für den zweiten Durchlauf.



Ü 30: Schreiben Sie das Gaußprogramm so um, dass der Endwert eingegeben werden muss.

Ü 31: Schreiben Sie ein Programm, das alle durch 7 teilbaren Zahlen zwischen einen einzugebenen Start und Endwert ausgibt.

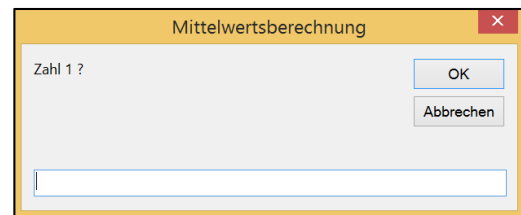


Bsp. 26: Mittelwert (=Durchschnitt) von 10 Zahlen. Eingabe : 10 Zahlen; Ausgabe: Mittelwert.
Versuchen Sie bitte zunächst das Beispiel selbstständig zu lösen.

```
Sub Mittelwert ()
    Dim i As Integer
    Dim sum As Single, zahl As Single
    Dim mittelwert As Single

    sum = 0
    For i = 1 To 10
        zahl = InputBox("Zahl " & CStr(i) & " ?", "Mittelwertsberechnung")
        sum = sum + zahl
    Next

    mittelwert = sum / 10
    MsgBox "Der Mittelwert beträgt " & CStr(mittelwert)
End Sub
```



Das Null-setzen der Summe am Anfang ist in VBA nicht notwendig; da alle Variablen nach dem Dim Befehl automatisch 0 sind. Da es jedoch in den meisten Sprachen anders ist; setzte ich hier aus didaktischen Gründen die Summe auf 0.

Debuggen und Einzelschritt

Eine gute Möglichkeit, den Programmablauf zu verstehen, ist die Debuggoption **Einzelschritt**: Mit der Taste <F8> kann das Programm Befehl für Befehl ausgeführt werden. Die aktuelle Zeile erscheint in Gelb. Sie können nun den Wert der Variablen an dieser Stelle einfach durch die **QuickInfo** ermitteln. Wenn Sie mit der Maus über die Variablen fahren so wird der momentane Wert der Variablen angezeigt.

Mit **Debuggen** (Englisch: Bug Wanze, Insekt) bezeichnet man das Beseitigen von Fehlern in Programmen. Eine Möglichkeit zur Fehlersuche ist das Anhalten des Programms an einer bestimmten Stelle (**Haltepunkt**).

Im Gegensatz zum Einzelschritt kann hier gezielt eine bestimmte Stelle im Programm betrachtet werden. An dieser Stelle kann dann der Inhalt der Variablen analysiert (und gegebenenfalls verändert) werden. Einen Haltepunkt kann man durch die Taste <F9> oder durch Klicken in der Leiste links vom Code erzeugen. Mit <F9> können Sie den Haltepunkt wieder Löschen.

Wenn Sie nun das Programm (Bsp. 26) starten, so wird es bis vor den Haltepunkt ausgeführt. Sie können den Wert der Variablen an dieser Stelle einfach durch die **QuickInfo** ermitteln. Mit <F5> wird das Programm nun bis zum nächsten Haltepunkt ausgeführt. Da unser Haltepunkt in der For – Schleife liegt wird das Programm 10 mal angehalten. Betrachten Sie nun wie sich sum, zahl und i in jedem Durchlauf ändern.

Eine weitere Möglichkeit, sich Werte von Variablen anzeigen zu lassen, ist das **Überwachungsfenster** (Ansicht -> Überwachungsfenster). Mit dem Punkt *Überwachung hinzufügen* aus dem Kontextmenü des Fensters kann man einzelne Variable, deren Inhalt man betrachten will, hinzufügen.

