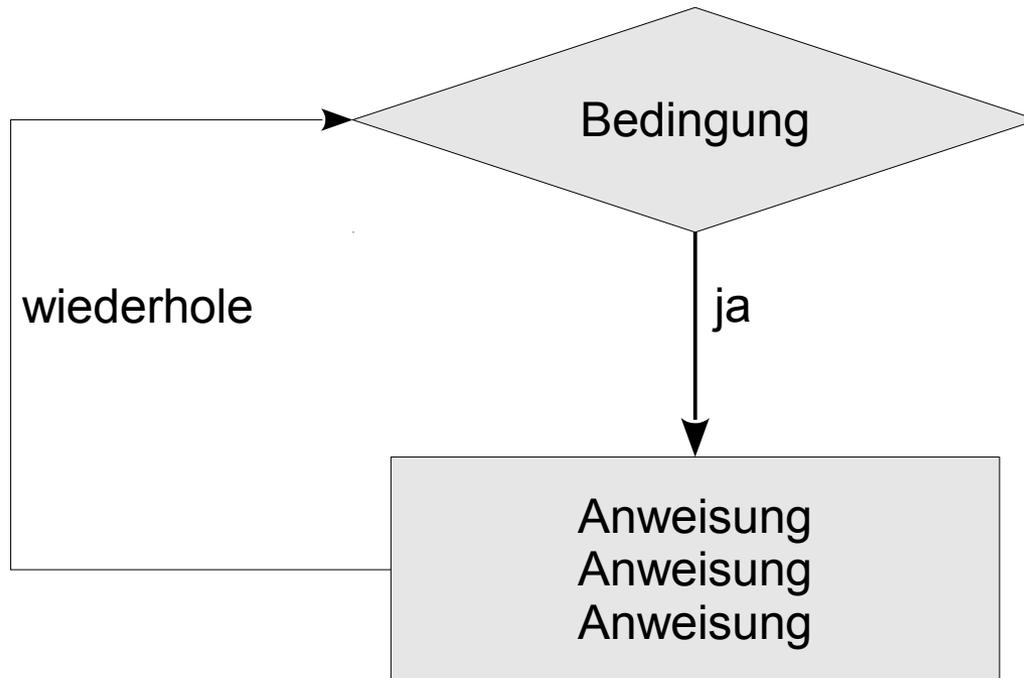


V(isual) B(asic for) A(pplication) Schleifen



Schleifen

- Iterationsanweisungen.
- Wiederholung von Anweisungen.
- Zählschleifen
- Schleifen können vorzeitig abgebrochen werden.
- Schleifen können verschachtelt werden.

Beispiel

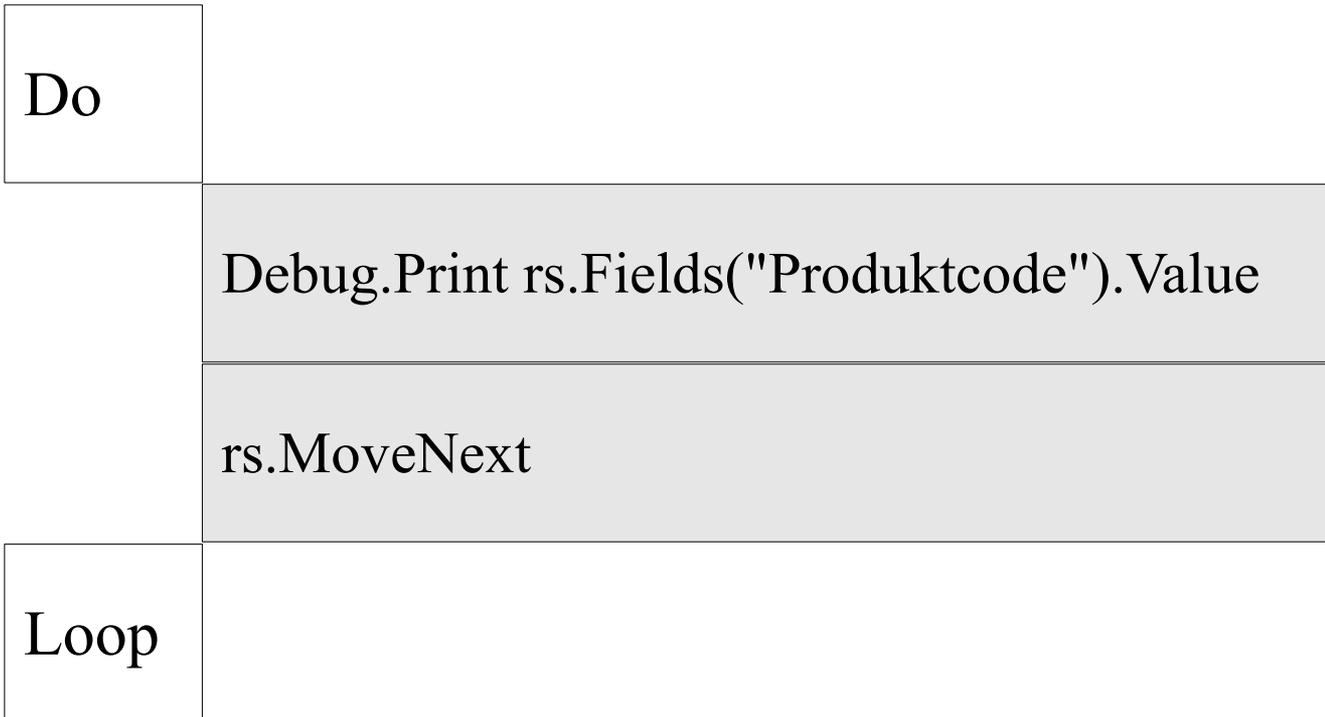
```
Set dbs = Application.CurrentDb
Set rs = dbs.OpenRecordset("Artikel", dbOpenTable)

rs.MoveFirst

Do While Not (rs.EOF)
    Debug.Print rs.Fields("Produktcode").Value
    rs.MoveNext
Loop

rs.Close
Set rs = Nothing
Set dbs = Nothing
```

Do ... Loop



Erläuterung

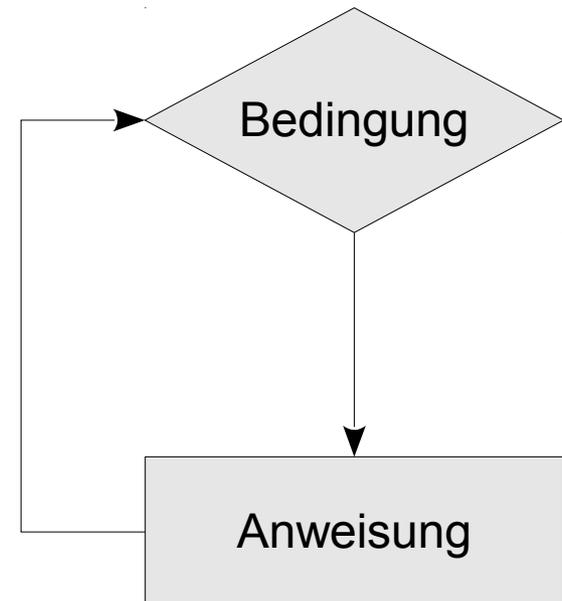
- Die Schleife beginnt mit dem Schlüsselwort Do und endet mit dem Schlüsselwort Loop.
- Die Schleife Do ... Loop wiederholt Anweisungen endlos.

Kopfgesteuerte Do ... Loop - Schleife

Do	While Until	zaehler < 10
Debug.Print zaehler		
zaehler = zaehler + 1		
Loop		

Arbeitsablauf

- In Abhängigkeit von While oder Until wird die Bedingung überprüft.
- Wenn die Bedingung zu trifft, werden die Anweisungen ausgeführt. In den Anweisungen muss die zu überprüfende Variable so verändert werden, dass die Schleife nach x Durchläufen abbricht.
- Das Schlüsselwort Loop verweist auf den nächsten Schleifendurchlauf.

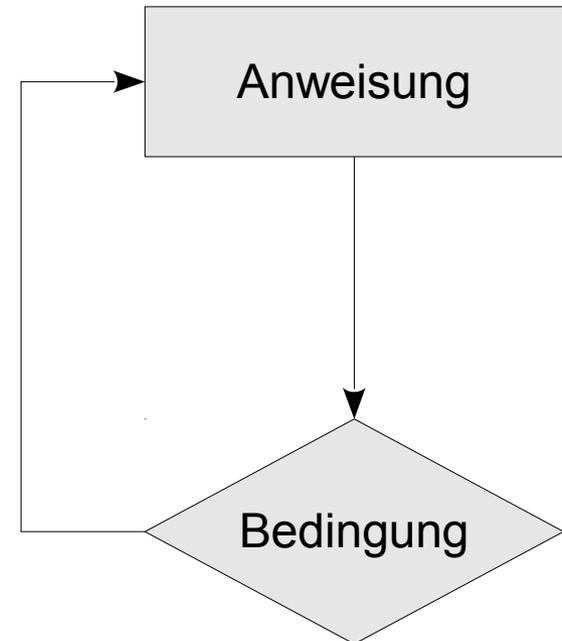


Fußgesteuerte Do ... Loop - Schleife

Do	Debug.Print zaehler	
	zaehler = zaehler + 1	
Loop	While Until	zaehler < 10

Arbeitsablauf

- Die Anweisungen werden ausgeführt.
- In Abhängigkeit von While oder Until wird die Bedingung im Fuß der Schleife überprüft. Der Fuß beginnt mit dem Schlüsselwort Loop. Die Bedingung wird vor jeden neuen Schleifendurchlauf überprüft.
- Wenn die Bedingung zu trifft, werden die Anweisungen nochmals ausgeführt.



Bedingungen

- Ausdrücke, die einen boolschen Wert zurückgeben.
- Vergleich von Werten.
- Beantwortung von Fragen wie zum Beispiel „Ist das Ende des Recordsets erreicht?“.

Beispiel

```
zaehler = 1
```

```
Do While zaehler < 10
```

```
Do Until zaehler >= 10
```

- Mit Hilfe von Vergleichsoperatoren werden zwei Werte verglichen. In diesem Beispiel wird der Wert in der Variablen zaehler mit den Wert 10 verglichen.
- Vergleichsoperatoren können aus ein oder zwei Zeichen bestehen. Zwischen den Zeichen darf kein Leerzeichen stehen.

Vergleichsoperatoren

Ist kleiner

5	<	2	=	var false
---	---	---	---	--------------

Ist kleiner gleich

5	<=	2	=	var false
---	----	---	---	--------------

Ist größer

5	>	2	=	var true
---	---	---	---	-------------

Ist größer gleich

5	>=	2	=	var true
---	----	---	---	-------------

Vergleichsoperatoren „Ist gleich“ und „Ist ungleich“

Ist gleich

5	=	2	=	var
				false

Ist nicht gleich

5	<>	2	=	var
				true

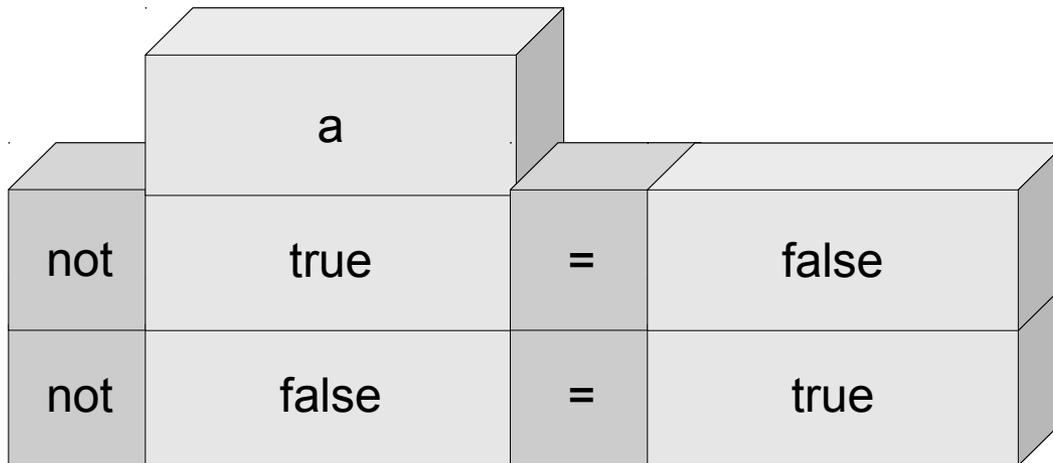
Negation einer Bedingung

rs.MoveFirst

Do While Not (rs.EOF)

- Das Schlüsselwort **Not** negiert eine Bedingung.
- Falls die Bedingung als Wert falsch zurückliefert wird sie wahr und umgekehrt.
- In diesem Beispiel: Wenn das Ende (rs.EOF) des Recordsets nicht erreicht ist.

Nicht-Verknüpfung



Kopfgesteuerte While-Schleife

Do	While	Not (rs.EOF)
		Debug.Print rs.Fields("Produktcode").Value
		rs.MoveNext
Loop		

Erläuterung

- Die Schleife beginnt mit dem Schlüsselwort `Do while` und endet mit dem Schlüsselwort `Loop`.
- Kopfgesteuerte `while`-Schleifen werden solange durchlaufen, wie die Bedingung wahr ist.
- In diesem Beispiel wird die Schleife solange durchlaufen, bis das Ende des Recordsets erreicht ist.

Kopfgesteuerte Until-Schleife

Do	Until	(rs.EOF)
		Debug.Print rs.Fields("Produktcode").Value
		rs.MoveNext
Loop		

Erläuterung

- Die Schleife beginnt mit dem Schlüsselwort `Do until` und endet mit dem Schlüsselwort `Loop`.
- Kopfgesteuerte `until`-Schleifen werden solange durchlaufen, bis die Bedingung erfüllt ist.
- In diesem Beispiel wird die Schleife nicht einmal durchlaufen, wenn das Ende des Recordsets erreicht ist,

Fußgesteuerte While-Schleife

Do	Debug.Print tblDef.Fields(zaeher).Name	
	zaehler = zaehler + 1	
Loop	While	zaehler < tblDef.Fields.Count

Erläuterung

- Die Schleife beginnt mit dem Schlüsselwort `Do`.
- Der Fuß beginnt mit dem Schlüsselwort `Loop While`.
- In diesen Beispiel werden die Anweisungen so lange durchlaufen, wie die Variable `zaehler` kleiner als die Anzahl der Gesamtfelder der Tabelle sind.

Fußgesteuerte Until-Schleife

Do		
	Debug.Print tblDef.Fields(zaeher).Name	
	zaehler = zaehler - 1	
Loop	Until	zaehler < 0

Erläuterung

- Die Schleife beginnt mit dem Schlüsselwort `Do`.
- Der Fuß beginnt mit dem Schlüsselwort `Loop Until`.
- In diesen Beispiel werden die Anweisungen so lange durchlaufen, wie die Variable `zaehler` größer als 0 ist. Sobald die Bedingung zutrifft, wird die Schleife abgebrochen.

Zählschleife

For	zaehler = 0	To	tblDef.Fields.Count - 1
	Debug.Print tblDef.Fields(zaehler).Name		
Next	zaehler		

Erläuterung

- Der Kopf einer Zählschleife beginnt mit dem Schlüsselwort For.
- Der nächste Schleifendurchlauf beginnt mit Next [variable].
- Bei Zählscheifen wird die Anzahl der Durchläufe im Kopf festgelegt.

Kopf der Zählschleife

For	zaehler = 0	To	tblDef.Fields.Count - 1
For	[Initialisierung]	To	[Endwert]

- Dem Schlüsselwort For folgt die Initialisierung der Zählvariablen. Mit Hilfe des Zuweisungsoperators wird der Zählvariablen ein Wert zugewiesen.
- Dem Schlüsselwort To folgt der Endwert. Sobald der Endwert erreicht ist, wird die Schleife abgebrochen.
- Standardmäßig wird die Zählvariable bei jedem Schleifendurchlauf inkrementiert (plus 1).

Schrittweite

For	zaehler = 0	To	10	Step	1
For	zaehler = 0	To	10	Step	0.5
For	zaehler = 10	To	0	Step	-2

- Der Datentyp der Zählvariablen entspricht der Angabe der Schrittweite. Eine Schrittweite „Double“ benötigt eine Zählvariable vom Datentyp Double.
- Mit Hilfe einer negativen Schrittweite kann die Zählvariable rückwärts durchlaufen werden.

Verschachtelte Schleifen

```
Dim tag As Integer
Dim monat As Integer

For monat = 1 To 12
    For tag = 1 To 31
        Debug.Print tag & "." & monat
    Next tag
Next monat
```

Abbruch von Schleifen

- Mit Hilfe des Schlüsselwortes `Exit For` kann eine For-Schleife vorzeitig abgebrochen werden.
- Mit Hilfe des Schlüsselwortes `Exit Do` kann eine Do ... Loop-Schleife vorzeitig abgebrochen werden.

foreach-Schleife

```
Dim dbs As DAO.Database
Dim tblDef As DAO.TableDef
Dim feld As DAO.Field

Set dbs = Application.CurrentDb
Set tblDef = dbs.TableDefs("Artikel")

For Each feld In tblDef.Fields
    Debug.Print feld.Name
Next
```

Erläuterung

- Mit Hilfe von for each – Schleifen können Auflistungen von Objekten durchlaufen werden.
- Für jedes Objekt in der Auflistung...
- Auflistungen von Objekten enden immer mit dem Buchstaben S.