

---

Ricardo Hernández García

1. Ausgabe, April 2016

ISBN 978-3-86249-460-6

## Access 2016

**Automatisierung,  
Programmierung**

ACC2016P



**HERDT**

## 7 Steuerung des Programmablaufs

### In diesem Kapitel erfahren Sie

- ✓ wie Sie in VBA bedingte Verzweigungen programmieren
- ✓ wie Sie in VBA Schleifen programmieren

### Voraussetzungen

- ✓ Grundlagen der Programmiersprache VBA
- ✓ Kenntnisse im Umgang mit der VBA-Entwicklungsumgebung



**Beispieldatei:** Übungsdatei *Gehalt07.accdb*

### 7.1 Bedingungen für den Programmablauf

#### Kontrollstrukturen

In seiner einfachsten Form enthält ein Programm nur Anweisungen, die der Reihe nach ausgeführt werden. Solche Programme haben Sie in den bisherigen Kapiteln kennengelernt.

Häufig kommt es vor, dass ein Programm auf eine bestimmte Situation mit einer Entscheidung reagieren muss und dann von mehreren Anweisungen eine oder keine ausführt. Auch ist es oft notwendig, dass bestimmte Anweisungen nicht nur einfach, sondern mehrfach hintereinander mit verschiedenen Werten durchlaufen werden.

Programmelemente, die eine bedingte Ausführung von Anweisungen ermöglichen, heißen **Kontrollstrukturen**. Bei der Programmierung von Kontrollstrukturen müssen Bedingungen formuliert werden.

Eine Bedingung besteht meist aus zwei oder mehr Werten (zum Beispiel Variablen oder feste Werte), die mit Vergleichsoperatoren zu einem Ausdruck verknüpft werden. Als Resultat liefert sie das Ergebnis `True` oder `False`. Mehrere Teilbedingungen können mit logischen Operatoren zu einer Bedingung verbunden werden. Auch eine logische Variable kann als Bedingungsausdruck angegeben werden.

#### Vergleichsoperatoren

Bei der Formulierung von Bedingungen werden oft Werte miteinander verglichen. VBA bietet zu diesem Zweck die Vergleichsoperatoren. Das Ergebnis eines Ausdrucks mit diesen Operatoren ist entweder `True` oder `False` (wahr oder falsch). Solche Bedingungen werden vor allem bei Kontrollstrukturen verwendet, um den Programmablauf zu steuern.

Vergleichsoperator	Bedeutung	Beispiel
=	gleich	50 = 480 liefert <code>False</code> ; 771 = 771 liefert <code>True</code>
<>	ungleich	250 <> 480 liefert <code>True</code> ; 771 <> 771 liefert <code>False</code>
>	größer	5000 > (2500 * 2) liefert <code>False</code>
<	kleiner	1032 < 1076 liefert <code>True</code>
>=	größer gleich	1000 >= 1000 liefert <code>True</code>
<=	kleiner gleich	1400 <= 1000 liefert <code>False</code>
Like	wie/ähnlich Muster	"Meier" Like "?ei*" liefert <code>True</code>

Der Like-Operator vergleicht eine Zeichenkette mit einem Muster und liefert je nachdem, ob der Mustervergleich erfolgreich war oder nicht, `True` oder `False`. Im Muster können verschiedene Platzhalter verwendet werden.



Platzhalter	Bedeutung
?	Ein beliebiges Zeichen
*	Kein oder mehrere beliebige Zeichen
#	Eine beliebige einzelne Ziffer
[Zeichenliste]	Ein beliebiges einzelnes Zeichen aus der Zeichenliste, zum Beispiel [a-z, A-Z] oder [A, B, C]
[!Zeichenliste]	Ein beliebiges einzelnes Zeichen, das nicht in der Zeichenliste vorhanden ist

## Logische Operatoren

Logische Operatoren verknüpfen mehrere Wahrheitswerte miteinander und liefern als Ergebnis `True` oder `False`. Sie werden eingesetzt, um mehrere Teilbedingungen logisch miteinander zu verknüpfen. In der Tabelle finden Sie die wichtigsten logischen Operatoren:

Logischer Operator	Bedeutung	Beispiel
Not	Verneinung (invertiert einen logischen Wert)	<b>Not</b> (1000 <> 1000) liefert <code>True</code>
And	Und-Verknüpfung (liefert <code>True</code> , wenn die beiden logischen Werte <code>True</code> sind)	(200 < 400) <b>And</b> (1 > 1000) True <b>And</b> False liefert <code>False</code>
Or	Oder-Verknüpfung (liefert <code>True</code> , wenn mindestens einer der logischen Werte wahr ist)	(200 < 400) <b>Or</b> (1 > 1000) True <b>Or</b> False liefert <code>True</code>
Xor	Exklusive Oder-Verknüpfung (liefert <code>True</code> , wenn genau einer der beiden logischen Werte wahr ist)	(200 < 400) <b>Xor</b> (1000 > 1) True <b>Xor</b> True liefert <code>False</code>

## Bedingungen formulieren

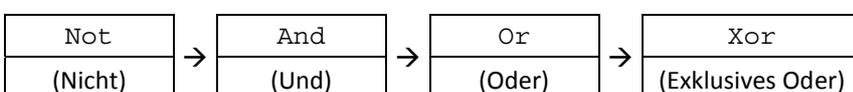
Im Folgenden finden Sie Beispiele für Bedingungen, wie sie in Kontrollstrukturen vorkommen können:

①	Summe1 <> Summe2
②	(Wert1 = 500) <b>And</b> (Name = "Meier")
③	Plz Like "#####"

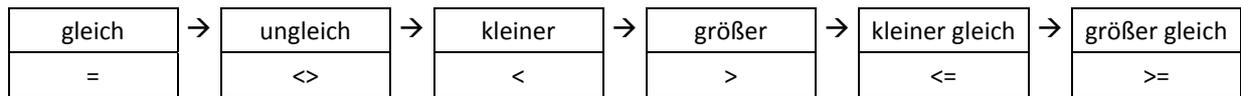
- ① Die Bedingung liefert `True` (wahr), wenn die Werte der Variablen `Summe1` und `Summe2` ungleich sind.
- ② Die Bedingung liefert `True`, wenn `Wert1` gleich 500 ist und Variable `Name` den String `Meier` enthält.
- ③ Wenn auf den Inhalt der Variablen `Plz` der Mustervergleich zutrifft, ist die Bedingung erfüllt. In diesem Fall müssen fünf Ziffern vorliegen, um dem Muster zu entsprechen.

## Prioritätsreihenfolge der Operatoren

Wenn Sie mehrere Operatoren in einer Bedingung verwenden, werden die Operatoren in einer bestimmten Reihenfolge ausgewertet. Zuerst werden die logischen Operatoren in der folgenden Reihenfolge ausgewertet:



Danach folgen die Vergleichsoperatoren in dieser Reihenfolge:



Um die Reihenfolge, in der Operatoren ausgewertet werden, gezielt zu verändern, können Sie Klammern verwenden. Ausdrücke in Klammern werden immer zuerst ausgewertet.

Als Beispiel soll eine Bedingung formuliert werden, die prüft, ob ein bestimmter Name vorhanden ist oder die Postleitzahl über einer Untergrenze liegt. Außer dieser Teilbedingung soll ein bestimmtes Alter überschritten sein. Die Variable `Name` soll `Meier` enthalten, die Variable `Plz` den Wert 54524 und die Variable `Alter` den Wert 37.

(	(Name = "Meier")	Or	(Plz >= 60000)	)	And	(Alter >= 40)
---	------------------	----	----------------	---	-----	---------------

Zuerst werden die Vergleiche in den inneren Klammern einzeln ausgewertet.

①(	True	Or	False	)	And	False
----	------	----	-------	---	-----	-------

Als Nächstes wird der geklammerte logische Ausdruck geprüft.

True	And	False
------	-----	-------

Als Letztes werden die beiden übrigen Teilbedingungen mit dem logischen Operatoren `And` verknüpft.

False
-------

Ohne Klammersetzung ① wäre das Ergebnis `True`.



Um sicher zu sein, dass Bedingungsaustrücke in der richtigen Reihenfolge geprüft werden, sollten Sie auch dann Klammern setzen, wenn sie streng genommen unnötig wären. Dies erhöht neben der Sicherheit auch die Lesbarkeit Ihres Programms.

## 7.2 Die Auswahl (Alternative)

### Verzweigung in Programmen

Der Programmablauf kann in Zweige aufgeteilt werden, von denen nur einer durchlaufen werden soll. Die Entscheidung dazu wird anhand einer Bedingung getroffen, die nur einen Wert annehmen kann: `True` oder `False`.

### Die einseitige Auswahl

Die einseitige Auswahl ist dadurch gekennzeichnet, dass aufgrund einer Bedingung eine oder mehrere Anweisungen (Auswahlblock) ausgeführt werden oder nicht. Liefert die Bedingung `True`, dann werden die Anweisungen ausgeführt. Ist das Ergebnis der Bedingung dagegen `False`, werden sie nicht ausgeführt.

### Syntax der einseitigen Auswahl

Die einseitige Auswahl wird in VBA mit der `If-Then`-Anweisung realisiert:

- ✓ Nach dem Schlüsselwort **If** steht der Bedingungsausdruck. Die Prüfung der Bedingung entscheidet, ob die Anweisungen nach dem Schlüsselwort **Then** ausgeführt werden oder nicht.
- ✓ Sind mehrere Anweisungen in Abhängigkeit von der Bedingung auszuführen, müssen sie zu einem Block zusammengefasst werden. Das Ende des Blocks wird durch **End If** markiert.
- ✓ Nach dem Ende der **If-Then**-Anweisung werden alle weiteren Anweisungen wieder unabhängig von dieser Bedingung abgearbeitet.

```
If Bedingung Then Anweisung
```

oder

```
If Bedingung Then
  Anweisung1
  Anweisung2
  ...
End If
```

### Beispiel: Datenbank *Gehalt07.accdb*, Modul **KontrollBeispiele**

Ein Beispiel für den Einsatz der einseitigen Auswahl ist ein Programm zur Rechnungsstellung eines CD-Händlers. Falls der Rechnungsbetrag größer ist als 100 €, gewährt er Rabatt, sonst nicht.

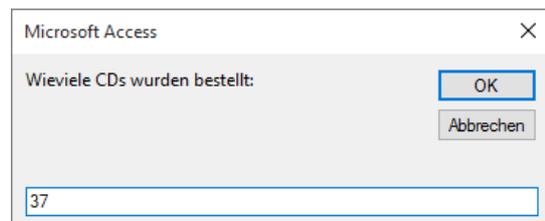
Die Prozedur **Rabatt** realisiert das Beispiel in VBA, wobei die Anzahl der verkauften CDs in einem Eingabefenster eingegeben wird. Die Ausgabe erfolgt in einem Meldungsfenster.

```

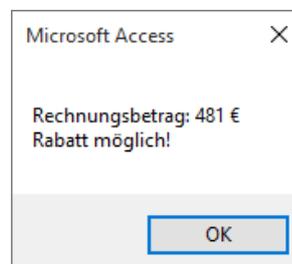
Sub Rabatt()
  Dim Anzahl As Integer, Ausgabe As String
  Dim EinzelPreis As Currency, GesamtPreis As Currency
  ① Anzahl = InputBox("Wie viele CDs wurden bestellt: ")
  EinzelPreis = 13
  ② GesamtPreis = Anzahl * EinzelPreis
  ③ Ausgabe = "Rechnungsbetrag: " & GesamtPreis & " €" & vbCrLf
  ④ If GesamtPreis >= 100 Then Ausgabe = Ausgabe & "Rabatt möglich!"
  ⑤ MsgBox Ausgabe
End Sub

```

- ① Um die Anzahl der bestellten CDs eingeben zu können, wird die VBA-Funktion `InputBox` verwendet. Sie zeigt ein Dialogfenster mit dem angegebenen Text und einem Textfeld zur Eingabe an. Die Funktion `InputBox` liefert den Wert in der Eingabezeile als Rückgabewert, der in diesem Beispiel der Variablen `Anzahl` zugewiesen wird.
- ② An dieser Stelle erfolgt die Berechnung des Gesamtpreises aus der Anzahl der Artikel und dem Einzelpreis.
- ③ Der Gesamtpreis wird in eine Zeichenkette eingefügt und der Variablen `Ausgabe` zugewiesen. Die Konstante `vbCrLf` erzeugt einen Zeilenumbruch.
- ④ Hier wird geprüft, ob der Gesamtpreis größer oder gleich 100 € ist. Dies ist die Auswahlbedingung, damit die nachfolgende Anweisung ausgeführt wird. Falls die Bedingung erfüllt ist, wird der Variablen `Ausgabe` ein Rabatthinweis angefügt.
- ⑤ Die Variable `Ausgabe` wird in einem Meldungsfenster angezeigt.



Das Eingabefenster der Prozedur



Die Ausgabe

Sie können auch innerhalb von Anweisungsblöcken einer Auswahl eine neue Auswahl angeben, denn Auswahlanweisungen lassen sich beliebig verschachteln.



Was bedeutet ...?	
zählergesteuerte Wiederholung	Es ist im Voraus bekannt, wie oft ein Anweisungsblock wiederholt werden soll. Eine Zählvariable wird dabei um den durch <code>Step</code> festgelegten Wert erhöht oder vermindert. <code>For...To...Step...Next</code>
kopfgesteuerte Wiederholung	Anweisungen sollen in Abhängigkeit von einer Bedingung einmal, mehrmals oder gar nicht ausgeführt werden. Die Bedingung wird am Anfang der Schleife geprüft. <code>Do While/Until...Loop</code>
fußgesteuerte Wiederholung	Anweisungen sollen in Abhängigkeit von einer Bedingung mindestens einmal oder mehrmals ausgeführt werden. Die Bedingung wird am Ende der Schleife geprüft. <code>Do...Loop While/Until...</code>
Schleifenabbruch	Eine Schleife soll in Abhängigkeit von einer Bedingung vorzeitig abgebrochen werden. <code>Exit Do/For</code>

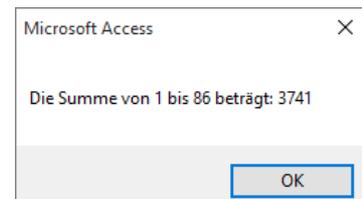
## 7.12 Übungen

### Übung 1: Die zählergesteuerte Wiederholung anwenden

Übungsdatei: **Gehalt07.accdb**

Ergebnisdatei: **Gehalt07-E.accdb**

- Erstellen Sie im Modul `KontrollBeispiele` eine SubProzedur `AddZahlen`, die alle Zahlen von eins bis zu einem vom Anwender einzugebenden Wert addieren soll.
- Benutzen Sie einen Eingabedialog, um den Endwert vom Anwender zu erfragen. Der Rückgabewert wird in einer Variablen gespeichert.
- Mit einer zählergesteuerten Wiederholung (`For Next`-Schleife) addieren Sie alle Zahlen von eins bis zur eingegebenen Zahl.
- Das Ergebnis präsentieren Sie dem Anwender in einem Meldungsfenster. Geben Sie dabei auch den eingegebenen Endwert an (vgl. nebenstehende Abbildung).



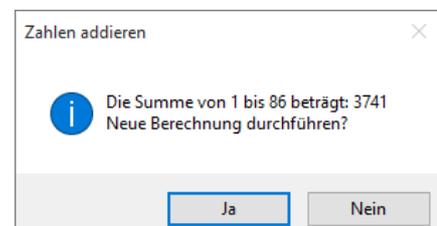
Die Ausgabe

### Übung 2: Die bedingte Wiederholung anwenden

Übungsdatei: **Gehalt07.accdb**

Ergebnisdatei: **Gehalt07-E.accdb**

- Erstellen Sie im Modul `KontrollBeispiele` eine neue SubProzedur `AddZahlenWiederholt`, indem Sie die Prozedur aus Übung 1 kopieren.
- Der Anwender soll auf Nachfrage die Berechnung wiederholen können. Erweitern Sie dazu das Meldungsfenster um eine Abfrage, ein Informationssymbol und die Schaltflächen *Ja* und *Nein* (vgl. nebenstehende Abbildung).
- Die Berechnung soll mindestens einmal durchgeführt werden und danach so oft, bis der Anwender die Schaltfläche *Nein* betätigt. Benutzen Sie dazu eine `Do Loop Until`-Struktur.



Das erweiterte Meldungsfenster

---

# Impressum

Matchcode: ACC2016P

Autor: Ricardo Hernández Garcia

Redaktion: Andrea Weikert

Produziert im HERDT-Digitaldruck

1. Ausgabe, April 2016

HERDT-Verlag für Bildungsmedien GmbH

Am Kümmerling 21-25

55294 Bodenheim

Internet: [www.herd.com](http://www.herd.com)

E-Mail: [info@herdt.com](mailto:info@herdt.com)

© HERDT-Verlag für Bildungsmedien GmbH, Bodenheim

Alle Rechte vorbehalten. Kein Teil des Werkes darf in irgendeiner Form (Druck, Fotokopie, Mikrofilm oder einem anderen Verfahren) ohne schriftliche Genehmigung des Verlags reproduziert oder unter Verwendung elektronischer Systeme verarbeitet, vervielfältigt oder verbreitet werden.

Dieses Buch wurde mit großer Sorgfalt erstellt und geprüft. Trotzdem können Fehler nicht vollkommen ausgeschlossen werden. Verlag, Herausgeber und Autoren können für fehlerhafte Angaben und deren Folgen weder eine juristische Verantwortung noch irgendeine Haftung übernehmen.

Wenn nicht explizit an anderer Stelle des Werkes aufgeführt, liegen die Copyrights an allen Screenshots beim HERDT-Verlag. Sollte es trotz intensiver Recherche nicht gelungen sein, alle weiteren Rechteinhaber der verwendeten Quellen und Abbildungen zu finden, bitten wir um kurze Nachricht an die Redaktion.

Die in diesem Buch und in den abgebildeten bzw. zum Download angebotenen Dateien genannten Personen und Organisationen, Adress- und Telekommunikationsangaben, Bankverbindungen etc. sind frei erfunden. Eventuelle Übereinstimmungen oder Ähnlichkeiten sind unbeabsichtigt und rein zufällig.

Die Bildungsmedien des HERDT-Verlags enthalten Verweise auf Webseiten Dritter. Diese Webseiten unterliegen der Haftung der jeweiligen Betreiber, wir haben keinerlei Einfluss auf die Gestaltung und die Inhalte dieser Webseiten. Bei der Bucherstellung haben wir die fremden Inhalte daraufhin überprüft, ob etwaige Rechtsverstöße bestehen. Zu diesem Zeitpunkt waren keine Rechtsverstöße ersichtlich. Wir werden bei Kenntnis von Rechtsverstößen jedoch umgehend die entsprechenden Internetadressen aus dem Buch entfernen.

Die in den Bildungsmedien des HERDT-Verlags vorhandenen Internetadressen, Screenshots, Bezeichnungen bzw. Beschreibungen und Funktionen waren zum Zeitpunkt der Erstellung der jeweiligen Produkte aktuell und gültig. Sollten Sie die Webseiten nicht mehr unter den angegebenen Adressen finden, sind diese eventuell inzwischen komplett aus dem Internet genommen worden oder unter einer neuen Adresse zu finden. Sollten im vorliegenden Produkt vorhandene Screenshots, Bezeichnungen bzw. Beschreibungen und Funktionen nicht mehr der beschriebenen Software entsprechen, hat der Hersteller der jeweiligen Software nach Drucklegung Änderungen vorgenommen oder vorhandene Funktionen geändert oder entfernt.