

VBA Grundlagen

- Basic = Programmiersprache (**B**eginners **A**ll **P**urpose **S**ymbolic **I**nstruction **C**ode). Wurde 1964 als einfache Programmiersprache entwickelt. Bestandteil vom Anfang des MS-DOS (1981) an als BASICA, später als GWBASIC, QBasic.
- VB = Visual Basic. Weiterentwicklung von Microsoft (1991), um mit Basic auch graphische Fensterprogrammierung realisieren zu können.
- VBA = **V**isual **B**asic for **A**pplications (Visual Basic für Anwendungen). Seit 1995 Makrosprache zur Programmierung bei MS-Office Anwendungen.
- Link zu Online Tutorial: http://de.wikibooks.org/wiki/VBA_in_Excel_-_Grundlagen

Um in Office 2007 VBA benutzen zu können, muß bei den Optionen unter „HÄUFIG VERWENDET“ die Entwicklerregisterkarte für die Multifunktionsleiste aktiviert werden.

Der VBA Code ist Bestandteil der Office Datei und wird mit dem Visual Basic Editor geschrieben:

- Bis Office 2003 Menü: EXTRA / MAKRO / VISUAL BASIC EDITOR
- Ab Office 2007 Registerkarte ENTWICKLERTOOLS / Gruppe CODE / Button VISUAL BASIC

VBA Grundlagen - Steuerelemente

Benötigt werden Steuerelemente. Steuerelemente (engl. Controls) dienen der Interaktionen zwischen VBA und dem Benutzer. Um sie in einer Office Anwendung einzufügen, benötigen Sie die Symbolleiste Toolbox:

- Bis Office 2007 Menü: ANSICHT / SYMBOLLEISTEN / STEUERELEMENTE-TOOLBOX (bei Access: ANSICHT / TOOLBOX).



- Ab Office 2007 Registerkarte ENTWICKLERTOOLS.

Einige ActiveX Steuerelemente:

Bezeichnung (engl. Label) zur Ausgabe:

Textfeld (engl. Textbox) für Eingaben:

Befehlsschaltfläche (eng. Commandbutton) zur Codeausführung:

In Excel können zur Ein- und Ausgabe auch Zellen verwendet werden.

VBA Grundlagen – Namensbildung, Kommentar

Die in die Anwendung eingefügten Steuerelemente können in ihren Eigenschaften verändert werden (Kontextmenü Eigenschaft). Im VBA Code wird auf ein Steuerelement über seinen Namen zugegriffen. Deshalb ist es wichtig, dem Steuerelement bei der Eigenschaft Name eine verständliche informative Bezeichnung zu vergeben.

Allgemein gelten bei VBA die folgenden Regeln bei der Namensbildung:

1. Es sind nur Buchstaben (ohne Umlaute), Ziffern und als Sonderzeichen der Unterstrich _ erlaubt.
2. Es wird nicht zwischen Groß- und Kleinschrift unterschieden.
3. Ein Name muß mit einem Buchstaben beginnen.
4. Nach dem ersten Buchstaben können weitere Buchstaben, Ziffern oder Unterstrich folgen.

Beispiele für korrekte Namen: SiebenZwerge, Oelstand, Messwert_1.

Beispiele für fehlerhafte Namen: 7Zwerge, Ölstand, Messwert 1.

Kommentare beginnen mit einem Apostroph ' und gelten bis zum Zeilenende. Ein Kommentar wird nicht als auszuführender Programmcode interpretiert.

VBA Grundlagen - Variable

Variable sind Speicherplätze des Programmcodes, die nicht in VBA vor ihrer Benutzung vereinbart (deklariert) werden müssen. Es ist aber aus folgenden Gründen zu empfehlen:

- Nicht vereinbarte Variablen sind vom Datentyp Variant und belegen bei Zahlen jeweils 22 Byte, bei vereinbarten Variablen für ganze Zahlen vom Datentyp Integer oder reelle Zahlen vom Typ Single jedoch jeweils nur 4 Byte.
- Durch die Vereinbarung von Variablen am Anfang des Codesegments behält man die Übersicht über alle vergebenen Variablennamen.

Die DIM Anweisung zur Variablendeklaration hat die Form

DIM Name [, ...] AS Datentyp

Der **Datentyp** kann sein (Grundtypen):

INTEGER (ganze Zahlen -32768 ... 32767).

LONG (ganze Zahlen -2147483648 ... 2147483647).

SINGLE (reelle Zahlen ca. $\pm 10^{-45}$... $\pm 3 \cdot 10^{38}$).

DOUBLE (reelle Zahlen ca. $\pm 5 \cdot 10^{-324}$... $\pm 10^{308}$).

STRING für Texte (max. 2 Milliarden Zeichen).

VARIANT für beliebige Datentypen.

VBA Grundlagen - Konstanten

Konstanten sind unveränderliche Werte im Programm:

- Sie können als Werte geschrieben werden, Beispiele:
Ganze Zahlen 123 oder -987
Reelle Zahlen (Dezimalpunkt statt Dezimalkomma!) 1.23 oder -9.87
Zeichen und Zeichenketten "X" oder "Guten Tag!" (Leere Zeichenkette: "")
- Sie können als benannte (mit Namen versehen) Konstanten deklariert werden:
Die CONST Anweisung hat die Form
CONST Name AS Datentyp = Wert

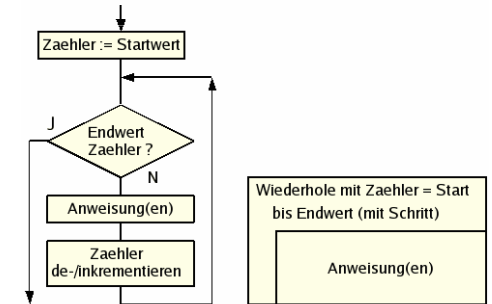
INTEGER, LONG Beispiel: CONST Maximum AS INTEGER = 10
SINGLE, DOUBLE Beispiel: CONST Preis AS SINGLE = 4.99
STRING Beispiel: CONST Gruss AS STRING = "Guten Tag!"
VARIANT Beispiel: CONST Text AS VARIANT = "Hallo"

VBA Grundlagen - FOR/NEXT Zählschleife

Die FOR / NEXT Schleifenanweisung: Werden Anweisungen wiederholt bei bekannter Anzahl der Wiederholungen ausgeführt, handelt es sich um eine Zählschleife. Die FOR Anweisung beginnt mit der Zuweisung des Startwertes an die Zählvariable. Alle Anweisungen in der Schleife bis NEXT werden nun solange wiederholt, bis der hinter TO angegebene Endwert erreicht ist. Bei NEXT (Variablenname kann entfallen) wird automatisch der Wert der Zählvariablen um 1 erhöht, falls keine andere Änderung durch STEP angegeben wurde.

FOR Variable = Startwert TO Endwert [STEP Schrittwert]
Anweisung

...
NEXT [Variable]



VBA Grundlagen – Das Problem

Das Problem: Wieviel Kapital häuft sich bei einmaliger Einzahlung auf einem Sparbuch nach einem bestimmten Zeitraum in Jahren an. Die jährlich anfallenden Zinsen werden auf dem Sparbuch gut geschrieben und in den Folgejahren mit verzinst.

Die Lösung: Nach jedem Jahr werden die Zinsen berechnet ...

$$K = K_0 \cdot \left(1 + \frac{Z}{100}\right)$$

... und auf dem Sparbuch gut geschrieben.

K = Kapital auf dem Sparbuch nach Ablauf des Zinsjahres.

K₀ = Anfangskapital des Zinsjahres.

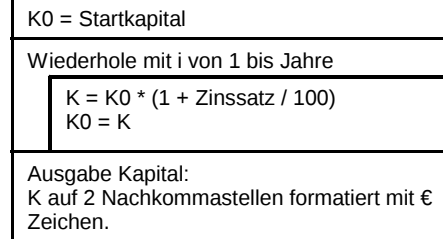
Z = Zinssatz p.a. in %.

Die Berechnung: Bei einem Startkapital von 100 und einem Zinssatz von 10%.

| Jahr | K ₀ | Berechnung von K | K |
|------|----------------|-------------------------|--------|
| 1 | 100,00 | K = 100,00 * (1+10/100) | 110,00 |
| 2 | 110,00 | K = 110,00 * (1+10/100) | 121,00 |
| 3 | 121,00 | K = 121,00 * (1+10/100) | 133,10 |
| 4 | 133,10 | K = 133,10 * (1+10/100) | 146,41 |
| 5 | 146,41 | K = 146,41 * (1+10/100) | 161,05 |

VBA Grundlagen – Das Struktogramm

Der Programmablauf als Struktogramm:



$$K = K_0 \cdot \left(1 + \frac{Z}{100}\right)$$

VBA Grundlagen – Die Lösung

Der VBA Code für den Button:

```
Private Sub BerechneButton_Click()
```

```
' Variablendeklaration:
```

```
Dim K, K0 As Single
```

```
Dim i As Integer
```

```
K0 = Startkapital
```

```
For i = 1 To Jahre ' Kapital Jahr für Jahr
```

```
    K = K0 * (1 + Zinssatz / 100)
```

```
    K0 = K
```

```
Next i
```

```
Kapital = Format(K, "#0.00 €")
```

```
End Sub
```

$$K = K_0 \cdot \left(1 + \frac{Z}{100}\right)$$

Startkapital

Zinssatz [%]

Jahre

Kapital

VBA Grundlagen - Verzweigung mit und ohne Alternative

Die Anweisung zur Verzweigung ohne Alternative:

Mit einer Programmverzweigung ohne Alternative wird gesteuert, ob überhaupt ein Anweisungsteil ausgeführt werden soll. Ist der Vergleich wahr, werden die Anweisung(en) bis zum END IF ausgeführt. Ist er falsch, werden sie übergangen.

```
IF Vergleich THEN
```

```
    Anweisung(en)
```

```
END IF
```

Ein **Vergleich** hat folgende Syntax:

Ausdruck Vergleichszeichen Ausdruck

Ein **Ausdruck** ist:

Konstante | Variable | Berechnung

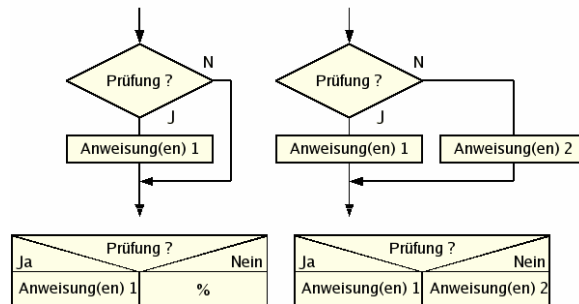
Die **Vergleichszeichen** sind: = (gleich), <> (ungleich), < (kleiner), > (größer), <= (kleiner und gleich) und >= (größer und gleich).

Mehrere Einzelvergleiche können zu einem Vergleich zusammengesetzt werden. Dazu dienen in VBA die **Verknüpfungen** AND ('und'), OR ('oder') und NOT ('nicht').

VBA Grundlagen - Verzweigung mit und ohne Alternative

Programmablauf von Verzweigungen:

Der Ablauf (oben Fluß- und unten Struktogramm) von Verzweigungen links ohne Alternative und rechts mit Alternative.



Das Problem: Die Zinsseszins Berechnung führt zu Fehlermeldungen und Stop der Ausführung des VBA Codes, wenn keine Eingaben oder keine Zahlen für Startkapital, Zinssatz und/oder Jahre vorliegen.

Die Lösung: Die komplette Berechnung des Kapitals wird nur ausgeführt, wenn alle Eingaben als numerische Werte vorliegen (keine Eingabe ist nicht numerisch!).

VBA Grundlagen – Verzweigung ohne Alternative

Der VBA Code für den Button:

```
Private Sub BerechneButton_Click()
```

```
' Variablendeklaration:
```

```
Dim K, K0 As Single
```

```
Dim i As Integer
```

```
If IsNumeric(Startkapital) And IsNumeric(Zinssatz) _
```

```
And IsNumeric(Jahre) Then ' Eingaben vorhanden
```

```
    K0 = Startkapital
```

```
For i = 1 To Jahre ' Kapital Jahr für Jahr
```

```
    K = K0 * (1 + Zinssatz / 100)
```

```
    K0 = K
```

```
Next i
```

```
Kapital = Format(K, "#0.00 €")
```

```
End If
```

```
End Sub
```

$$K = K_0 \cdot \left(1 + \frac{Z}{100}\right)$$

Startkapital

Zinssatz [%]

Jahre

Kapital

Anmerkung: eine Code-Anweisungszeile kann mit Leerzeichen gefolgt von Unterstrich (_) in einer neuen Zeile fortgesetzt werden!

VBA Grundlagen - Verzweigung mit Alternative

Die Anweisung zur Verzweigung mit Alternative:

Mit einer Programmverzweigung mit Alternative wird gesteuert, ob ein Anweisungsteil ausgeführt werden soll oder ein anderer. Ist der Vergleich wahr, werden die *Anweisung(en)1* hinter THEN ausgeführt. Ist er falsch, werden andernfalls (engl. else) andere *Anweisung(en)2* hinter ELSE ausgeführt.

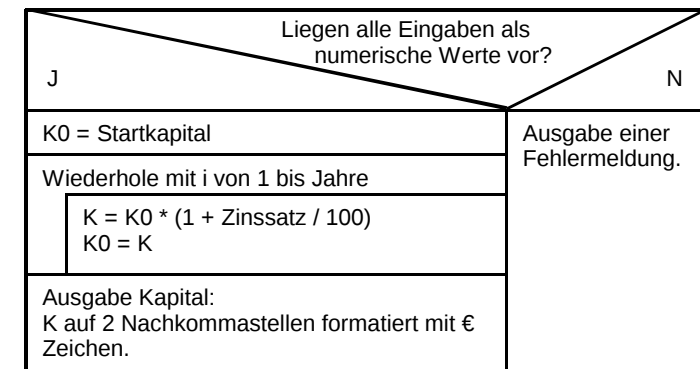
```
IF Vergleich THEN
    Anweisung(en)1
ELSE
    Anweisung(en)2
END IF
```

Das Problem: Wenn keine Eingaben oder keine Zahlen für Startkapital, Zinssatz und/oder Jahre vorliegen, wird nichts berechnet - jedoch der Benutzer im Ungewissen gelassen, woran es liegen könnte.

Die Lösung: Die komplette Berechnung des Kapitals wird weiterhin nur ausgeführt, wenn alle Eingaben als numerische Werte vorliegen (keine Eingabe ist ebenfalls nicht numerisch!). Andernfalls wird die MsgBox Funktion aufgerufen, um den Benutzer auf die fehlerhaften Eingaben aufmerksam zu machen und ein eventuell vorausgegangener Anzeigewert des Steuerelementes Kapital wird gelöscht.

VBA Grundlagen – Lösung Verzweigung mit Alternative

Der Programmablauf als Struktogramm:



VBA Grundlagen – Lösung (Verzweigung mit Alternative)

Der VBA Code für den Button:

```
Private Sub BerechneButton_Click()
```

' Variablendeklaration:

```
Dim K, K0 As Single
```

```
Dim i, Dummy As Integer
```

```
If IsNumeric(Startkapital) And IsNumeric(Zinssatz) _
And IsNumeric(Jahre) Then ' Eingaben vorhanden
    K0 = Startkapital
```

```
For i = 1 To Jahre ' Kapital Jahr für Jahr
```

```
    K = K0 * (1 + Zinssatz / 100)
```

```
    K0 = K
```

```
Next i
```

```
Kapital = Format(K, "#0.00 €")
```

```
Else ' Eingabefehler
```

```
    Kapital = ""
```

```
Dummy = MsgBox("Eingaben fehlerhaft!", 48)
```

```
End If
```

```
End Sub
```

$$K = K_0 \cdot \left(1 + \frac{Z}{100}\right)$$

| | |
|--------------|---|
| Startkapital | <input type="text" value="100"/> |
| Zinssatz [%] | <input type="text"/> |
| Jahre | <input type="text" value="3"/> |
| | <input type="button" value="BerechneButton"/> |
| Kapital | <input type="text"/> |

VBA Grundlagen - Objekte (Eigenschaften, Zugriff)

Zugriff auf Objekteigenschaften: Die Steuerelemente von VBA sind Objekte. Die Officeanwendungen bestehen ebenfalls aus Objekten (wie zum Beispiel in Excel die Zellen). **Jedes Objekt besitzt Eigenschaften** (engl. Properties) **und Methoden**, die über den **Objektzugriff** mit dem **Punktoperator** erreichbar sind:

Objektname.Eigenschaft

```
Private Sub FarbButton1_Click()
```

```
    FarbButton1.ForeColor = RGB(255, 255, 0)
```

```
    FarbButton1.BackColor = RGB(255, 0, 0)
```

```
    FarbButton2.ForeColor = RGB(255, 255, 0)
```

```
    FarbButton2.BackColor = RGB(255, 0, 0)
```

```
End Sub
```

Gelb auf Rot

Normalfarbe

Anmerkung: Warum keine RGB Farben beim Zurücksetzen? Weil Standardfarben systemweit in den Windowseinstellungen gesetzt werden können - es muß also nicht immer Schwarz auf Grau sein.

```
Private Sub FarbButton2_Click()
```

```
    FarbButton1.ForeColor = vbButtonText
```

```
    FarbButton1.BackColor = vbButtonFace
```

```
    FarbButton2.ForeColor = vbButtonText
```

```
    FarbButton2.BackColor = vbButtonFace
```

```
End Sub
```

VBA Grundlagen - Zugriffe auf Excel Zellen

In Excel lassen sich ebenfalls Steuerelemente zur Ein- und Ausgabe verwenden. Nahe liegt allerdings dazu die Verwendung von Excel-Zellen selbst.

Referenzen auf einzelne Excel Zellen in VBA (Beispiel immer für Zelle B3):

- In sogenannter **A1 Schreibweise** mit **Range** als String: **Range("B3")**.
- Mit **Indizes** als ganze Zahlen für Zeile und Spalte mit **Cells: Cells(3, 2)**.
- In **abgekürzter Schreibweise** mit eckigen Klammern: **[B3]**.
- Auf **benannte Zelle** als String mit **Range** verweisen: **Range("Name_von_B3")**.
- Auf Zellen mit **relativem Bezug** zur aktiven Zelle mit **ActiveCell.Offset** verweisen, Angabe Zeilen- und Spaltendifferenz in ganzen Zahlen. Wenn aktive Zelle A1: **ActiveCell.Offset(1, 2)**.

Befindet sich der VBA Code auf dem gleichen Arbeitsblatt, genügen obige Methoden, ein Objektname ist nicht nötig (Ausnahme ActiveCell). Befinden sich die **Zellen auf einem anderem Arbeitsblatt**, so muß als Objektname der Blattname angegeben werden und mit dem Punktoperator die Methode aufgerufen werden:

- **Sheets("Blattname").Range("B3")**
- **Worksheets("Blattname").Cells(3, 2)**
- **Worksheets("Blattname").[B3]**
- **Range("Blattname!Name_von_B3")**

VBA Grundlagen - Lösung mit Excel Zellen I

Der VBA Code für den Button „A1 Schreibweise“:

```
Private Sub BerechneButton1_Click()
' A1 Schreibweise bei Excel
Dim K, K0 As Single
Dim i, Dummy As Integer
```

```
If IsNumeric(Range("A3")) And IsNumeric(Range("B3")) _
And IsNumeric(Range("C3")) Then
K0 = Range("A3")
```

```
For i = 1 To Range("C3")
K = K0 * (1 + Range("B3") / 100)
K0 = K
Next i
```

```
Range("B8") = Format(K, "#0.00 €")
```

```
Else
Range("B8") = ""
Dummy = MsgBox("Eingaben fehlerhaft!", 48)
End If
End Sub
```

| | A | B | C |
|---|------------------------------|-------------------|-------|
| | Zinseszinsberechnung mit VBA | | |
| 1 | | | |
| 2 | Startkapital | Zinssatz p.a. [%] | Jahre |
| 3 | 100 | 10 | 2 |
| 4 | A1 Schreibweise | Mit Indizes | |
| 5 | | | |
| 6 | Kurzschreibweise | Benannte Zellen | |
| 7 | | | |
| 8 | Kapital: | 121,00 € | |

VBA Grundlagen - Lösung mit Excel Zellen II

Der VBA Code für den Button „Mit Indizes“:

```
Private Sub BerechneButton2_Click()
' Index Schreibweise bei Excel
Dim K, K0 As Single
Dim i, Dummy As Integer
```

```
If IsNumeric(Cells(3, 1)) And IsNumeric(Cells(3, 2)) _
And IsNumeric(Cells(3, 3)) Then
K0 = Cells(3, 1)
```

```
For i = 1 To Cells(3, 3)
K = K0 * (1 + Cells(3, 2) / 100)
K0 = K
Next i
```

```
Cells(8, 2) = Format(K, "#0.00 €")
```

```
Else
Cells(8, 2) = ""
Dummy = MsgBox("Eingaben fehlerhaft!", 48)
End If
End Sub
```

| | A | B | C |
|---|------------------------------|-------------------|-------|
| | Zinseszinsberechnung mit VBA | | |
| 1 | | | |
| 2 | Startkapital | Zinssatz p.a. [%] | Jahre |
| 3 | 100 | 10 | 2 |
| 4 | A1 Schreibweise | Mit Indizes | |
| 5 | | | |
| 6 | Kurzschreibweise | Benannte Zellen | |
| 7 | | | |
| 8 | Kapital: | 121,00 € | |

VBA Grundlagen - Lösung mit Excel Zellen III

Der VBA Code für den Button „Kurzschreibweise“:

```
Private Sub BerechneButton3_Click()
' Abgekürzte Schreibweise bei Excel
Dim K, K0 As Single
Dim i, Dummy As Integer
```

```
If IsNumeric([A3]) And IsNumeric([B3]) _
And IsNumeric([C3]) Then
K0 = Range("Startkapital")
```

```
For i = 1 To [C3]
K = K0 * (1 + [B3] / 100)
K0 = K
Next i
```

```
[B8] = Format(K, "#0.00 €")
```

```
Else
[B8] = ""
Dummy = MsgBox("Eingaben fehlerhaft!", 48)
End If
End Sub
```

| | A | B | C |
|---|------------------------------|-------------------|-------|
| | Zinseszinsberechnung mit VBA | | |
| 1 | | | |
| 2 | Startkapital | Zinssatz p.a. [%] | Jahre |
| 3 | 100 | 10 | 2 |
| 4 | A1 Schreibweise | Mit Indizes | |
| 5 | | | |
| 6 | Kurzschreibweise | Benannte Zellen | |
| 7 | | | |
| 8 | Kapital: | 121,00 € | |

VBA Grundlagen - Lösung mit Excel Zellen IV

Der VBA Code für den Button „Benannte Zellen“:

```
Private Sub BerechneButton4_Click()
```

```
' Benannte Zellen in Excel
```

```
Dim K, K0 As Single
```

```
Dim i, Dummy As Integer
```

```
If IsNumeric(Range("Startkapital")) And IsNumeric(Range("Zinssatz")) _
```

```
And IsNumeric(Range("Jahre")) Then
```

```
    K0 = Range("Startkapital")
```

```
    For i = 1 To Range("Jahre")
```

```
        K = K0 * (1 + Range("Zinssatz") / 100)
```

```
        K0 = K
```

```
    Next i
```

```
    Range("Kapital") = Format(K, "#0.00 €")
```

```
Else
```

```
    Range("Kapital") = ""
```

```
    Dummy = MsgBox("Eingaben fehlerhaft!", 48)
```

```
End If
```

```
End Sub
```

| | A | B | C |
|---|-------------------------------|-------------------|-------|
| 1 | Zinsseszinsberechnung mit VBA | | |
| 2 | Startkapital | Zinssatz p.a. [%] | Jahre |
| 3 | 100 | 10 | 2 |
| 4 | A1 Schreibweise | Mit Indizes | |
| 5 | | | |
| 6 | Kurzschreibweise | Benannte Zellen | |
| 7 | | | |
| 8 | Kapital: | 121,00 € | |

VBA Grundlagen - Zugriffe auf Excel Zellen mit FOR/NEXT

Der Zugriff auf Excel Zellen per Indizes eignet sich besonders für Zählschleifen mit der FOR/NEXT Anweisung.

```
Private Sub GoButton_Click()
Dim Spalte, Zeile As Integer
```

```
For Spalte = 3 To 4
```

```
    For Zeile = 15 To 20
```

```
        Cells(Zeile, Spalte) = 0
```

```
    Next Zeile
```

```
Next Spalte
```

```
End Sub
```

Vorher:

| | A | B | C | D |
|----|---------------------|----|---|---|
| 14 | Setze C15:D20 auf 0 | | | |
| 15 | | Go | | 2 |
| 16 | | | 1 | 4 |
| 17 | | | 7 | 8 |
| 18 | | | 9 | 6 |
| 19 | | | 3 | 1 |
| 20 | | | 5 | 7 |

Nachher:

| | A | B | C | D |
|----|---------------------|----|---|---|
| 14 | Setze C15:D20 auf 0 | | | |
| 15 | | Go | | 0 |
| 16 | | | 0 | 0 |
| 17 | | | 0 | 0 |
| 18 | | | 0 | 0 |
| 19 | | | 0 | 0 |
| 20 | | | 0 | 0 |

VBA Grundlagen - Zugriffe auf Eigenschaften von Excel Zellen

Auch auf die **Eigenschaften der Excel Zellen** läßt sich über den Objektzugriff per Punktoperator zugreifen. Da Objekte wieder Objekte enthalten können (anschaulich: die Milchstraße sei ein Objekt. Sie enthält weitere Objekte - die Planetensysteme. Ein Planetensystem enthält weitere Objekte - Sonne und Planeten. ...) erfolgt hier der Zugriff durch mehrmalige Anwendung des Punktoperators.

| | A | B | C |
|----|--------------|-------------|---|
| 8 | Kapital: | 121,00 € | |
| 9 | | | |
| 10 | Gelb auf Rot | Normalfarbe | |

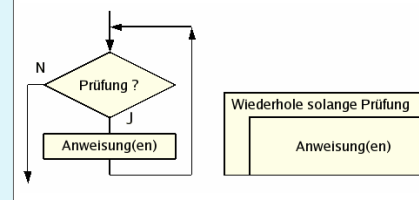
```
Private Sub FarbButton1_Click()
Range("A8:B8").Font.Color = RGB(255, 255, 0)
Range("B8").Interior.Color = RGB(255, 0, 0)
End Sub
```

```
Private Sub FarbButton2_Click()
Range("A8").Font.Color = vbWhite
Range("B8").Font.Color = vbBlack
Range("B8").Interior.Color = vbWhite
End Sub
```

VBA Grundlagen – abweisende und nicht abweisende Schleife

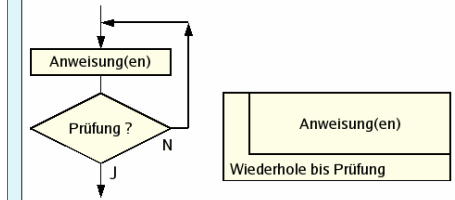
Bei der **abweisenden Schleife** steht die Anzahl der Schleifendurchläufe nicht fest. Die Schleifenanweisungen werden wiederholt, **solange die Bedingung wahr** ergibt. Die **Bedingungsprüfung erfolgt zu Schleifenbeginn** – deshalb kann es sein, daß die Schleife kein einziges Mal durchlaufen wird.

```
DO WHILE Bedingung
Anweisung(en)
LOOP
```



Bei der **nicht abweisenden Schleife** steht die Anzahl der Schleifendurchläufe nicht fest. Die Schleifenanweisungen werden wiederholt, **bis die Bedingung wahr** ergibt. Die **Bedingungsprüfung erfolgt zu Schleifenende** – deshalb wird die Schleife auf jeden Fall mindestens einmal durchlaufen.

```
DO
Anweisung(en)
LOOP UNTIL Bedingung
```



VBA Grundlagen – Code abweisende Schleife

```
Private Sub GoButton_Click()  
Dim K, K0, KE As Single  
Dim i, Dummy As Integer  
Jahre = ""  
If IsNumeric(Startkapital) And IsNumeric(Zinssatz) _  
And IsNumeric(Endkapital) Then  
    K0 = Startkapital  
    KE = Endkapital  
    i = 0  
  
    Do While KE > K0  
        i = i + 1  
        K = K0 * (1 + Zinssatz / 100)  
        K0 = K  
    Loop  
  
    Jahre = i  
Else  
    Dummy = MsgBox("Eingaben fehlerhaft!", 48)  
End If  
End Sub
```

Startkapital:

Zinssatz p.a.:

Endkapital:

Endkapital erreicht nach
 Jahren

VBA Grundlagen – Code nicht abweisende Schleife


```
Private Sub GoButton_Click()  
Dim K, K0, KE As Single  
Dim i, Dummy As Integer  
Jahre = ""  
If IsNumeric(Startkapital) And IsNumeric(Zinssatz) _  
And IsNumeric(Endkapital) Then  
    K0 = Startkapital: KE = Endkapital: i = 0  
  
    If KE > K0 Then  
        Do  
            i = i + 1  
            K = K0 * (1 + Zinssatz / 100)  
            K0 = K  
        Loop Until KE <= K0  
    End If  
  
    Jahre = i  
Else  
    Dummy = MsgBox("Eingaben fehlerhaft!", 48)  
End If  
End Sub
```

Startkapital:

Zinssatz p.a.:

Endkapital:

Endkapital erreicht nach
 Jahren

 **Anmerkung:** Mehrere Anweisungen können in eine Zeile geschrieben werden, wenn sie durch einen Doppelpunkt voneinander getrennt werden.