

Microsoft Excel

VBA - Einführung

Kontakt Daten

Organisatorisches

- Maik Ramftel
- 0172 / 81 93 687
- ramftel@ramses.de

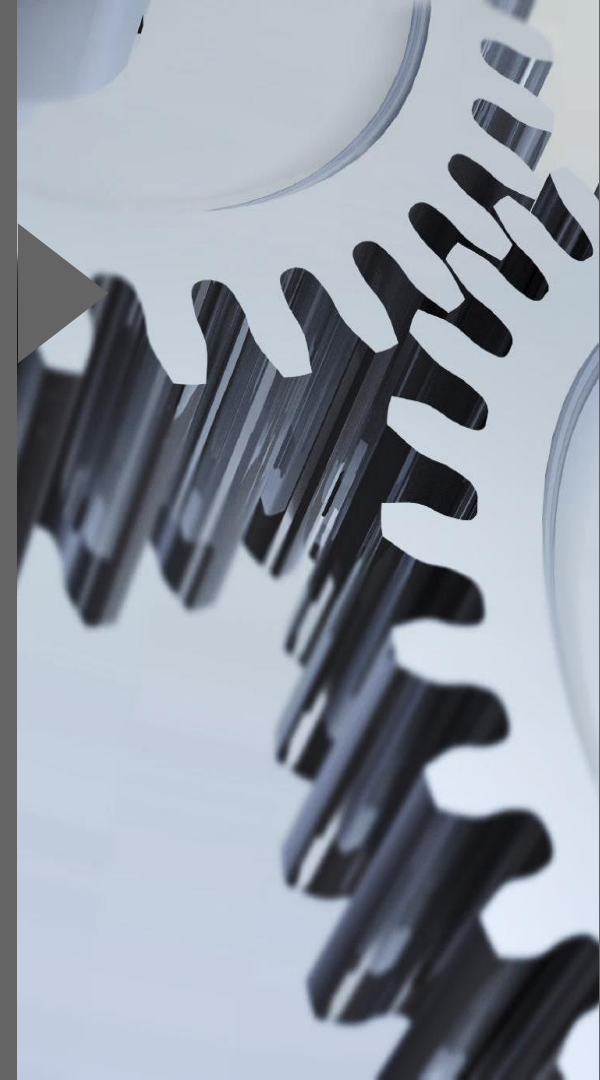
Seminarzeiten und Pausen

Organisatorisches

- 08:30 – 10:00
- 10:15 – 11:45
- 12:45 – 14:15
- 14:30 – 16:00

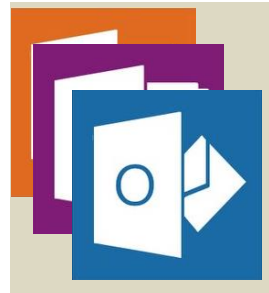
VBA

Elementare Begriffe



Programmübergreifende Sprache

Ein Office - ein Visual Basic for Application



VBA

- Das VBA ist in nahezu fast allen Office Produkten vorhanden
- Elementare Begriffe sind überall gleich
- Programmspezifische Objektmodelle
- Individuelle Besonderheiten und "Dialekte"

1

Die Grundbegriffe des VBA

- Prozeduren
- Module
- Variablen

2

Die Entscheidungsstrukturen

- Verzweigungen
- Schleifen

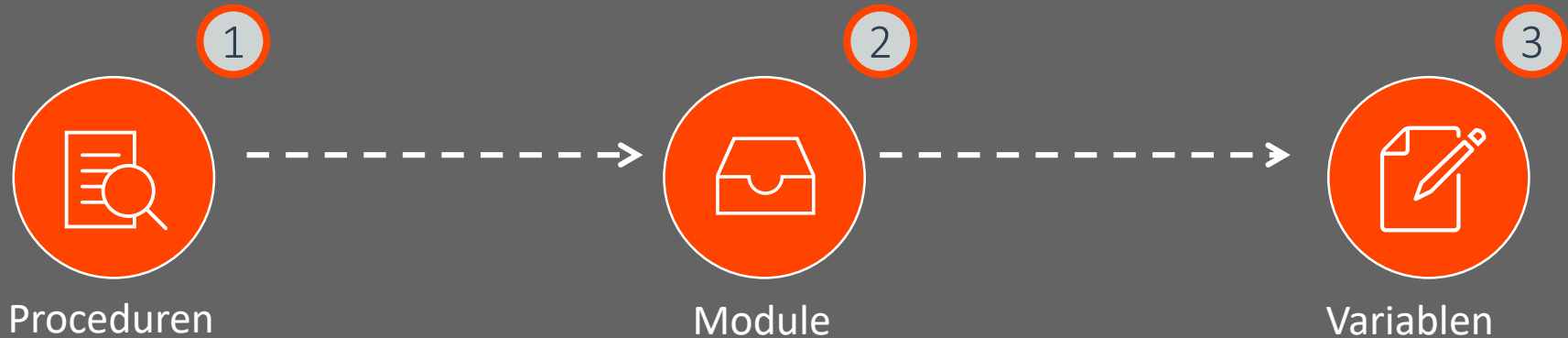
3

Das Objektmodell des Excel

- Arbeitsmappen
- Tabellen
- Spalten
- Zeilen
- Zellen

Grundbegriffe des VBA

Elementare Begriffe



Unabhängig vom Office Produkt hat das VBA
Proceduren, Module und Variablen als Grundbegriffe der Programmiersprache

Prozeduren

Grundbegriffe des VBA

- Prozeduren sind funktionell zusammengehörige Befehle, die "Programme"
- Prozeduren haben zwei verschiedene Aufgaben
 - Eine Sub beinhaltet in erster Linie Anweisungen, die Aktionen beinhalten
 - Eine Function beinhaltet eine Berechnung und gibt einen Wert zurück



Proceduren

Grundbegriffe des VBA

- Eine Sub beinhaltet in erster Linie Anweisungen, die Aktionen / Tätigkeiten beinhalten
 - Umbenennen eines Blattes
 - Festlegung einer Schriftart
- Eine Sub kann mit dem Recorder aufgezeichnet oder im Visual Basic Editor geschrieben werden
- Ausgelöst wird eine Sub durch
 - Eine Tastenkombination
 - Ein Symbol
 - Ein Shape
 - Ein Ereignis



Proceduren

Grundbegriffe des VBA

- Die Function stellt eine Alternative zu komplexen Excel Formeln dar
- Eine Function beinhaltet eine Berechnung und gibt einen Wert zurück
 - Berechnung eines Preises
 - Ermittlung einer technischen Kennziffer
- Eine Function muss im Visual Basic Editor geschrieben werden
- Genutzt wird eine Funktion
 - In einer Formel



Proceduren

Benutzerdefinierte Funktionen

	A	B	C	D	E
1	Artikel	Nettopreis	Kategorie	Bruttopreis	
2	Besen	15,30 €	A	=Bruttowert(B2;C2)	
3	Schaufel	20,50 €	A	24,40 €	
4	Apfel	2,50 €	B	2,68 €	

15,3

A

Function Bruttowert(Nettobetrag As Double, Kategorie As String) As Double

If Kategorie = "A" Then

Bruttowert = Nettobetrag * 1.19

Else

Bruttowert = Nettobetrag * 1.07

End If

End Function

18,207

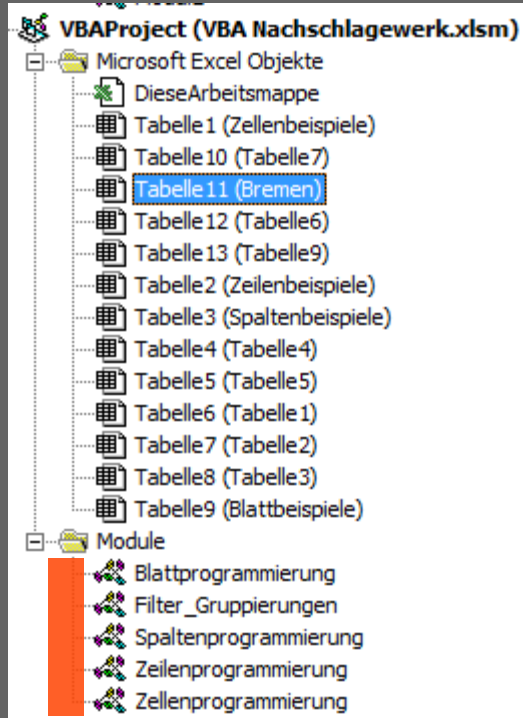
Module

Grundbegriffe des VBA

- Ein Modul ist ein Ordner in dem die Prozeduren abgespeichert werden
 - Module sind Container, die Ordnung schaffen in der Flut der Prozeduren
 - Ein Modul ist wie ein Aktenordner, in dem die verschiedenen Blätter abgeheftet werden
- Allgemeine Module beinhalten Prozeduren die manuell aufgerufen werden müssen.
- Ereignisprozeduren werden in DieseArbeitsmappe oder in der TabelleX gespeichert

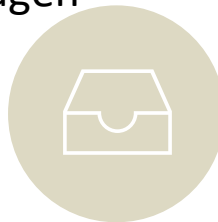


Module

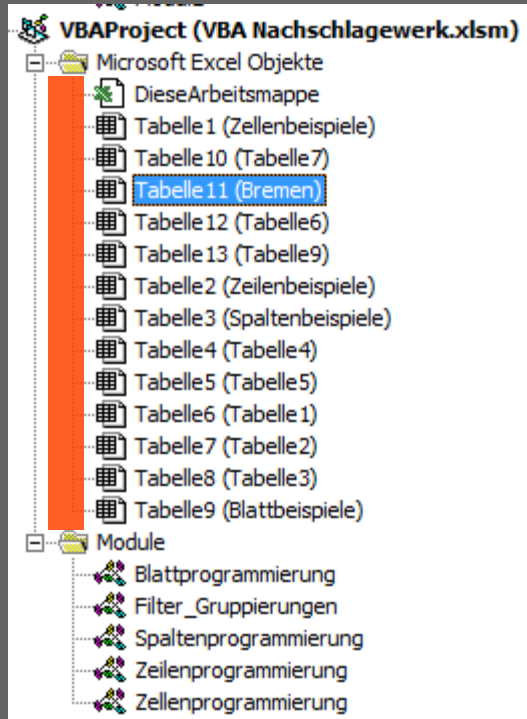


Allgemeine Module

- Manueller Aufruf von
 - Sub Procedures
 - Funktionen
- Allgemeine Module dienen der Organisation der Procedures
- Module dürfen nie den Namen einer Procedure tragen



Module



Ereignisproceduren

- Ereignisse
 - Der Arbeitsmappe Workbook
 - Eines Blattes Worksheet



Variablen - Definition

Grundbegriffe des VBA

- Variablen sind Zwischenspeicher, in denen sich ein Programm „etwas“ merken kann
 - Wertvariablen:
Variablen können Inhalte speichern wie z.B. Texte, Datums- und Zeitangaben, logische Werte oder Zahlen, also die Inhalte einer Excel Zelle
 - Objektvariablen:
Variablen können Objekte speichern wie Tabellen, Dateien oder Pivot Tabellen
- Eine Variable speichert die Inhalte während das Programm läuft
- Variablen müssen nicht initialisiert werden
- Mit dem Programmende werden die Variablen geleert



Variablen - Namenskonventionen

Beinhalten Daten die sich im Programmablauf ändern können

Länge

Der Name darf die Länge von 255 Zeichen nicht überschreiten

Reservierte Namen

Verwenden Sie keine reservierten Namen wie Left, Worksheet etc.



Benennung

Verwenden Sie einen Buchstaben als erstes Zeichen, kein Leerzeichen, keinen Punkt (.), Ausrufezeichen (!) oder die Zeichen @, &, \$, # im Namen

Einzigartigkeit

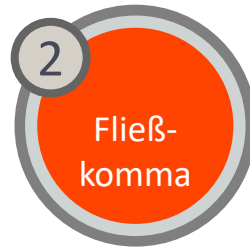
Der Name einer Variable muss im Geltungsbereich einzigartig sein

Variablentypen

Die Wertvariablen



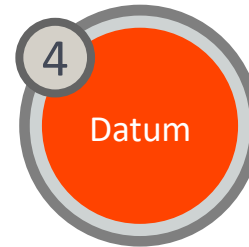
-
- Byte
 - Integer
 - Long



-
- Single
 - Currency
 - Double



-
- String



-
- Date



-
- Boolean



Variablentypen

Die Objektvariablen



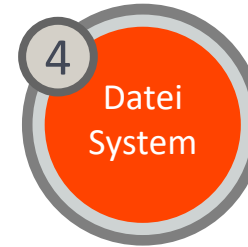
-
- Range
 - Worksheet
 - Workbook



-
- PivotCache
 - PivotTable
 - PivotField



-
- Object



-
- FileSystemObject



Variablendimensionierung

Grundbegriffe des VBA

- Variablen die Objekte beinhalten werden mit dem Befehl SET zugewiesen
 - Dim TolleMappe as Workbook
 - **Set** TolleMappe = ThisWorkbook
- Variablen die Werte speichern werden ohne Set Anweisung gefüllt
 - Dim Kindergeldsatz as long
 - Kindergeldsatz = 190



Variablendimensionierung

Grundbegriffe des VBA

- Variablen können in der Prozedur dimensioniert werden und sind nur dort gültig

Option Explicit

Function Kindergeld(Kinder As Long) As Long

Dim Kindergeldsatz As Long



Variablendimensionierung

Grundbegriffe des VBA

- Variablen können im Deklarationsteil des Moduls definiert werden und sind im gesamten Modul gültig

Option Explicit

Dim Kindergeldsatz As Long

Function Kindergeld(Kinder As Long) As Long



Variablendimensionierung

Grundbegriffe des VBA

- Wird die Variablen an die Proceedur übergeben kommt die Variable in die Klammern der Proceedur
- Diese Art der Variablen wird auch Übergabevariable oder Parameter genannt

Option Explicit

Function Kindergeld(Kinder As Long) As Long



Variablendimensionierung

Grundbegriffe des VBA

- Variablen, die nicht zwingend an die Prozedur übergeben werden müssen, werden als optional bezeichnet

Option Explicit

Function Kindergeld(Kinder As Long, Optional Alleinerziehend As Boolean) As Long



Variablendimensionierung

Grundbegriffe des VBA

- Variablen die nicht explizit deklariert werden sind Variant

Option Explicit

Function Kindergeld(Kinder As Long) As Long

Dim Kindergeldsatz



Variablen

Die Art der Übergabe

- Variablen können von einer Prozedur an eine andere übergeben werden
- ByRef-Parameter
 - Function Kindergeld(ByRef Kinderzahl as long)
 - Wird der Wert der Variable in der Prozedur geändert, so wird er auch im Original geändert
- ByVal-Parameter
 - Function Kindergeld(ByVal Kinderzahl as long)
 - Wird der Wert der Variable in der Prozedur geändert, so wird er NICHT im Original geändert
- Wenn keine Vorgabe erfolgt, wird der Parameter als ByRef behandelt



Übergabe ByRef

```
Sub Bananenbaer()
```

```
Dim x As Long, y As Long, z As Long
```

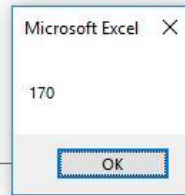
```
x = 25
```

```
y = Suppenkasper(x)
```

```
z = y + x
```

```
MsgBox z
```

```
End Sub
```



```
Function Suppenkasper(x As Long) As Long
```

```
Dim y As Long
```

```
x = 2 * 30
```

```
y = 50
```

```
Suppenkasper = x + y
```

```
End Function
```

Die Referenzübergabe

- Die Sub übergibt den Wert $x=25$ an die Function
- Der Rückgabewert der Function ist 110
 - $x = 2 * 30$
 - $y = 50$
- Weil die Übergabe des x als Referenz erfolgte, änderte sich durch das $x = 2 * 30$ auch der Wert x in der Sub auf 60
- Somit ist der Wert $z = 110 + 60$
- $z = 170$

Übergabe ByVal

```
Sub Bananenbaer()
```

```
Dim x As Long, y As Long, z As Long
```

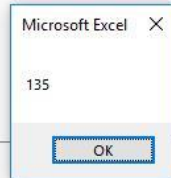
```
x = 25
```

```
y = Suppenkasper(x)
```

```
z = y + x
```

```
MsgBox z
```

```
End Sub
```



```
Function Suppenkasper(ByVal x As Long) As Long
```

```
Dim y As Long
```

```
x = 2 * 30
```

```
y = 50
```

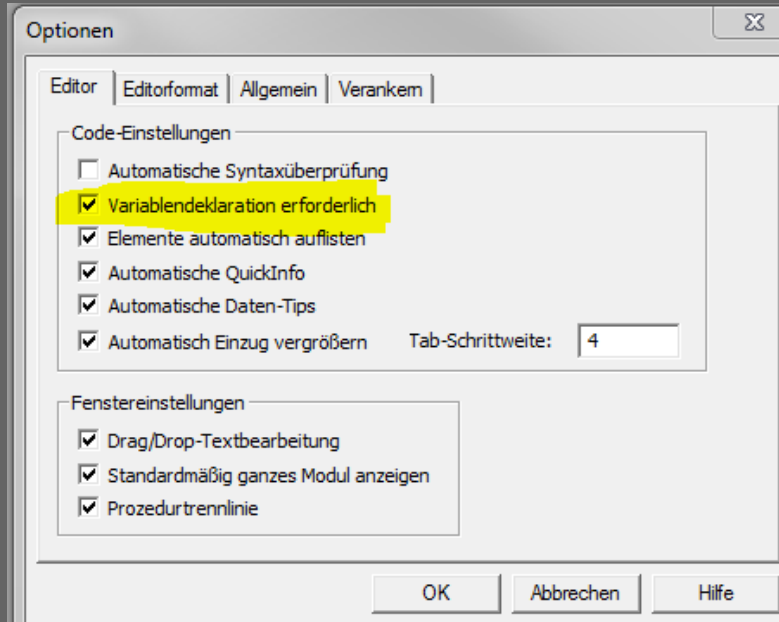
```
Suppenkasper = x + y
```

```
End Function
```

Die Wertübergabe

- Die Sub übergibt den Wert $x=25$ an die Function
- Der Rückgabewert der Function ist 110
 - $x = 2 * 30$
 - $y = 50$
- Weil die Übergabe des x als Wert erfolgte, änderte sich durch das $x = 2 * 30$ NICHT der Wert x in der Sub
- Somit ist der Wert $z = 110 + 25$
- $z = 135$

Variablen

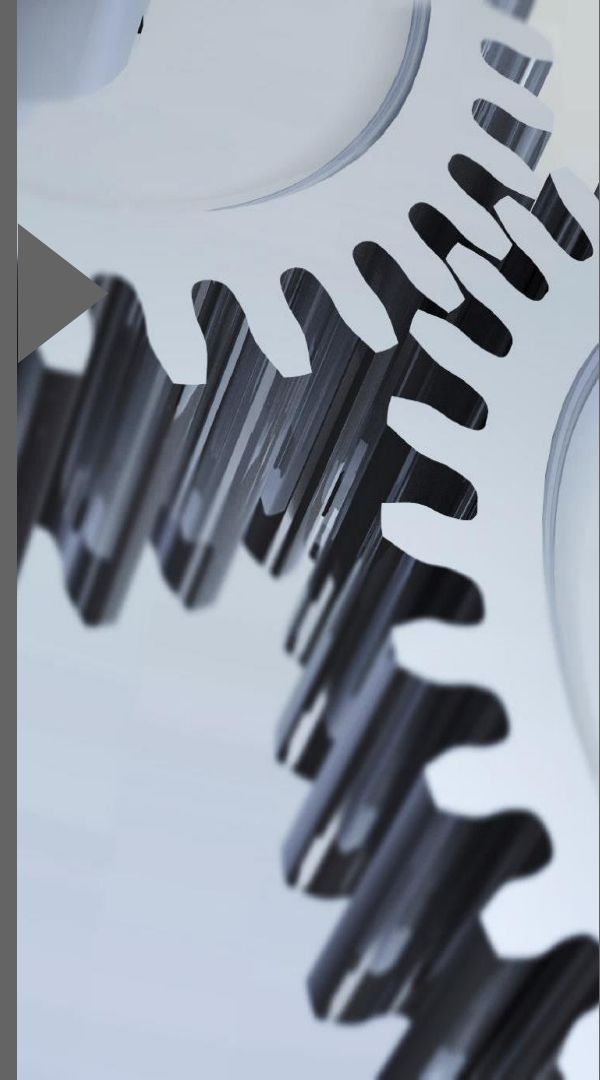


Option Explicit

- Erzwingen Sie die explizite Deklaration der Variablen
- Tragen Sie die Option Explicit als erste Anweisung im Modul ein
 - Explizite Deklaration beschleunigt das Programm
 - Schreibfehler werden verhindert

VBA

Kontrollstrukturen des VBA



VBA Grundelemente

Elementare Bestandteile der Programmierung

- Verzweigungen treffen Entscheidungen und machen....
 - A
 - A oder B
 - A, B, C oder D..
- Schleifen wiederholen Befehle, entweder....
 - Bis eine vorher definierte Menge an Wiederholungen erreicht ist
„Du zählst von 1 bis 1000“
 - Bis eine Bedingung erreicht ist oder verloren geht
„Du putzt die Teller bis alle sauber sind“
„Du putzt die Teller solange noch einer dreckig ist“

VBA Grundelemente

Elementare Bestandteile der Programmierung

Verzweigungen

- Bedingte Auswahl
 - Einseitig
 - If ... End if
 - Zweiseitig
 - If ... Else...End if
- Fallauswahl
 - Select Case ...End Select

Schleifen

- Zählergesteuert
 - For...Next
 - For Each Next
- Bedingte Wiederholung
 - Do While...Loop
 - Do Until...Loop



VBA Grundelemente

Elementare Bestandteile der Programmierung

- If Struktur
- Verzweigen wenn eine Bedingung Wahr oder Falsch ist
 - If Bedingung Then
True: Handlungsanweisung A
End If
 - If Bedingung Then Handlungsanweisung A
 - If Bedingung Then
True: Handlungsanweisung A
Else
False: Handlungsanweisung B
End If



VBA Grundelemente

Elementare Bestandteile der Programmierung

- Select Case Struktur
- Auswählen aus einer Gruppe von Bedingungen
 - Select Case Variable
 - Case is X
 - Case is Y
 - Case is Z
 - End Select



VBA Grundelemente

Elementare Bestandteile der Programmierung

- Do Loop Schleife
- Do Until
 - Mache eine Anweisung **bis** eine Bedingung Wahr ist
Putze die Monitore **bis** alle sauber sind
- Do While
 - Mache eine Anweisung **so lange** eine Bedingung Wahr ist
Putze die Monitore **so lange** einer dreckig ist



VBA Grundelemente

Elementare Bestandteile der Programmierung

- For Next
 - Bei Verwendung eines Zählers wird eine festgelegte Anzahl von Wiederholungen gemacht
 - For i = 1 to 10
Next
- Veränderung der Schrittweite mit dem Step
 - For i = 1 to 10 Step 2
Next



VBA Grundelemente

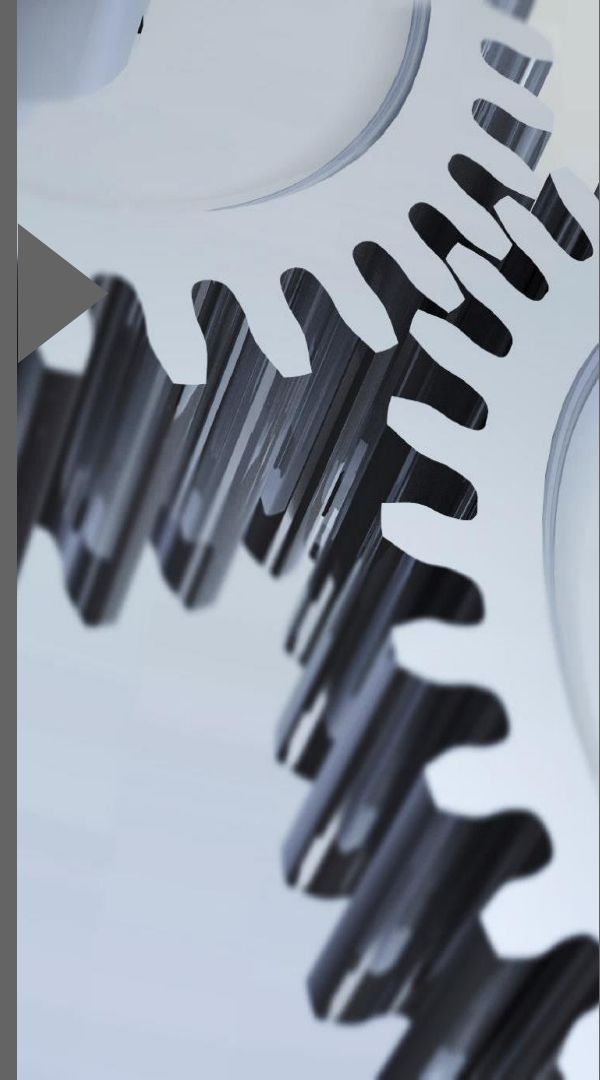
Elementare Bestandteile der Programmierung

- For Each
 - Wiederholung von Anweisungen für jedes Objekt in einer Auflistung von Objekten
 - „Putze alle Schuhe in Deinem Schrank“
 - Hierbei ist die Menge nicht zu nennen, da die Auflistung der Objekte die Menge vorgibt
 - For Each Zelle In Selection
TolleZelle.Comment.Delete
Next



DAS OBJEKTMODELL

Die Bestandteile des
Microsoft Excel



Das Excel Objektmodell

Welche Dinge gibt es im Excel...



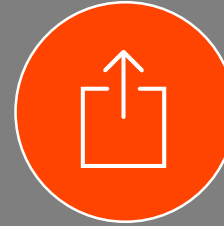
Objekte

- Objekte sind die Elemente des Excel mit denen wir im VBA arbeiten können



Methoden

- Methoden sind Aktionen die ein Objekt ausführen kann



Eigenschaften

- Eigenschaften sind Attribute, Merkmale eines Objekts

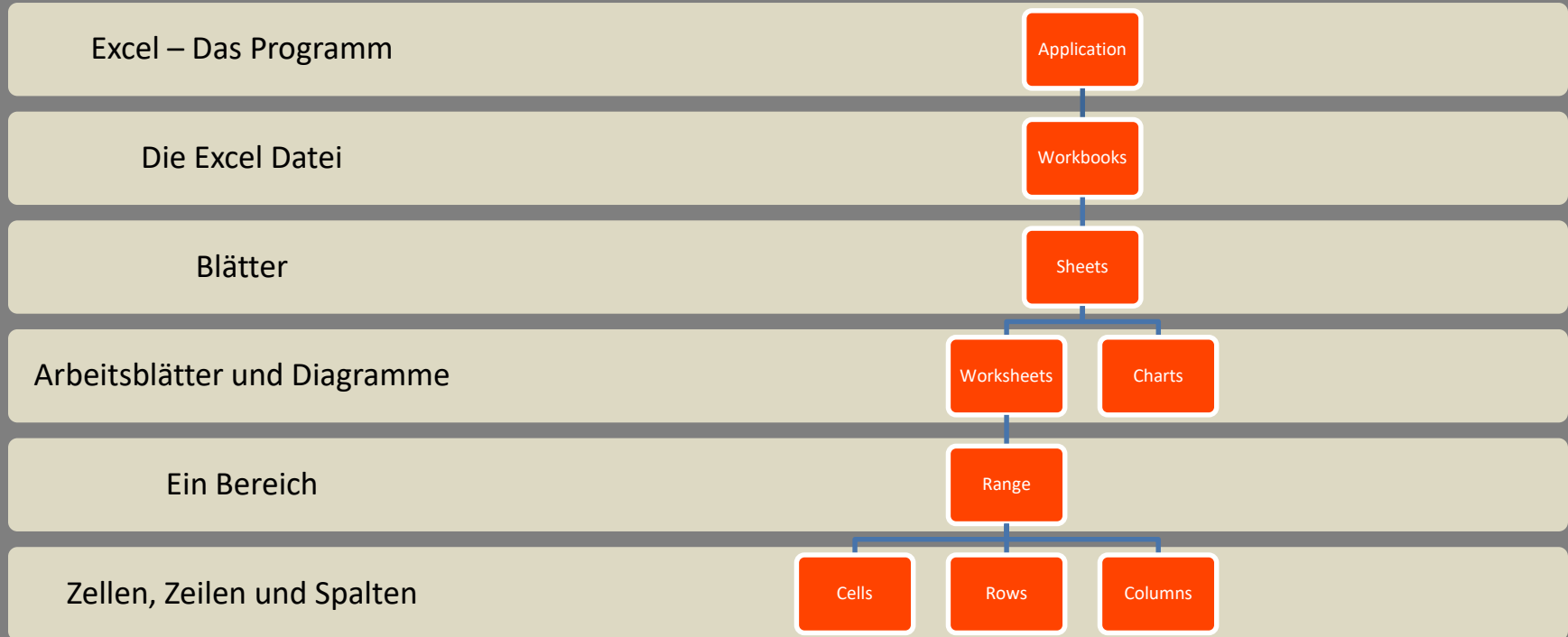


Ereignisse

- Eine Aktion auf die das Objekt reagiert

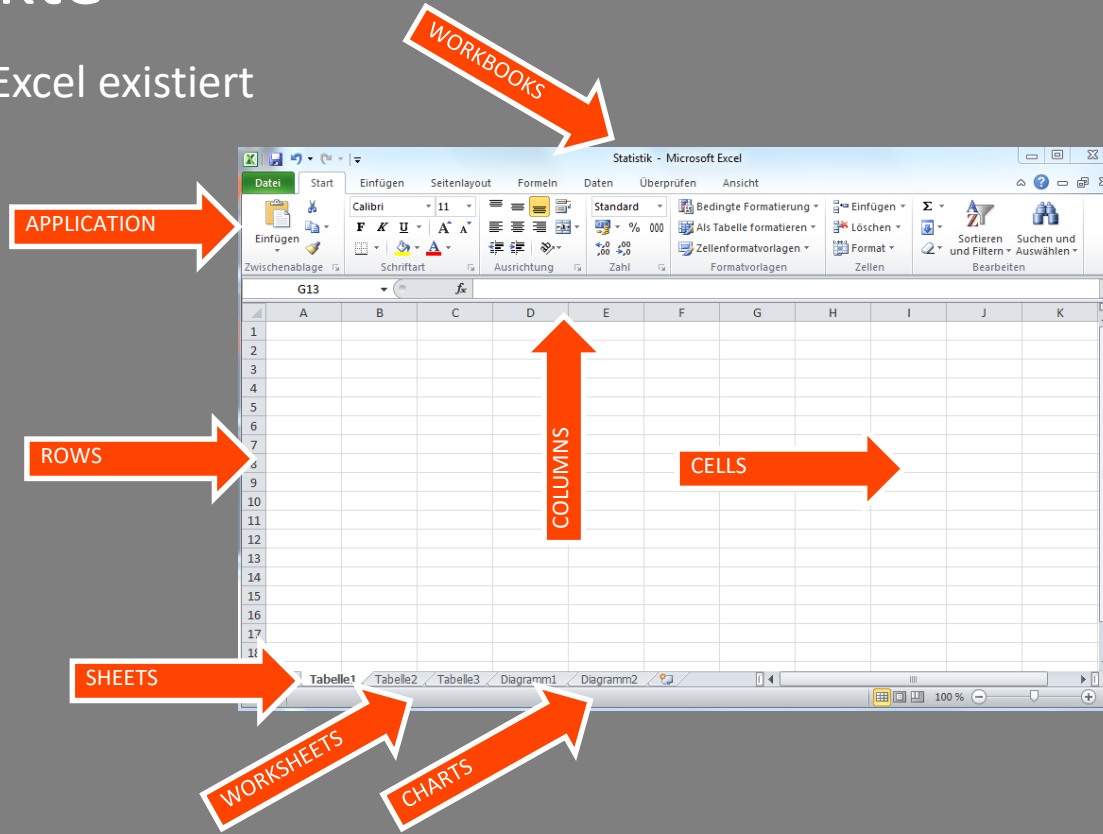
Objekte

Was im Excel existiert



Objekte

Was im Excel existiert



Objekte

Der Verweis auf ein Objekt

- Der Verweis auf ein Objekt erfolgt mit seinem Namen
 - `Workbooks("Statistik.xlsx")`
 - `Worksheets("Tabelle1")`
 - `Range("B2")`
- oder seiner Nummer in der Auflistung der Objekte
 - `Workbooks(1)`
 - `Sheets(1)`
 - `Cells(2,2)`



Objekte

Der Verweis auf das aktive Objekt

- Von einem Objekt kann es viele geben, viele Arbeitsblätter oder viele Arbeitsmappen, s.g. Auflistungen sind daher Plural
 - `Workbooks("Statistik.xlsx")`
 - `Worksheets("Tabelle1")`
 - `Cells(2,2)`
- Es kann aber nur ein Element das aktive Element sein, daher hier Singular
 - `ActiveWorkbook`
 - `ActiveSheet`
 - `ActiveCell`



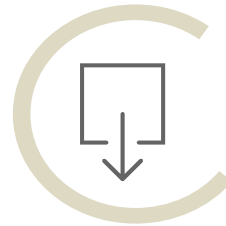
Objekte

Was kann, was hat, was passiert einem Objekt



Eigenschaften

Methoden



Ereignisse

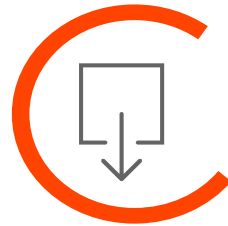


Objekte

Methoden - Die Fähigkeiten eines Objektes



Eigenschaften



Methoden



Ereignisse



Methoden

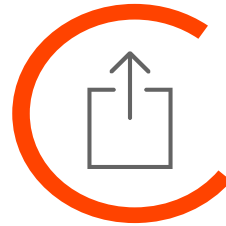
Methoden sind Aktionen die ein Objekt ausführen kann

- Die Methode eines Objektes steht hinter dem Objekt mit einem Punkt getrennt
 - Objekt.Methode
 - `Workbooks("Statistik.xlsx").Close`
 - `Worksheets("Tabelle1").PrintOut`
- Ein Objekt steht nie alleine, ebenso wird eine Methode nie ohne ein Objekt genannt
 - **`Workbooks("Statistik.xlsx")`**
 - **`Close`**



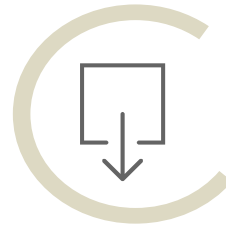
Objekte

Eigenschaften - Die Attribute eines Objektes



Eigenschaften

Methoden



Ereignisse



Eigenschaften

Eigenschaften sind Attribute eines Objekts

- Die Eigenschaft eines Objektes steht hinter dem Objekt mit einem Punkt getrennt
 - Objekt.Eigenschaft
 - ActiveWorkbook.Name
 - ActiveSheet.Name
- Ein Objekt steht nie alleine, ebenso wird eine Eigenschaft nie ohne ein Objekt genannt
 - **ActiveWorkbook**
 - **Name**



Eigenschaften

Eigenschaften sind Attribute eines Objekts

- Eigenschaften können sowohl ausgelesen oder aber verändert werden
- Beim Auslesen steht die Eigenschaft rechts vom Gleichheitszeichen
 - Variable = Objekt.Eigenschaft
 - Kundenname = ActiveCell.Value
- Beim Verändern der Eigenschaft steht die Eigenschaft links vom Gleichheitszeichen
 - Objekt.Eigenschaft = Variable
 - ActiveCell.Value = Kundenname
- Was links vom Gleichheitszeichen steht wird gefüllt mit dem was rechts vom Gleichheitszeichen steht

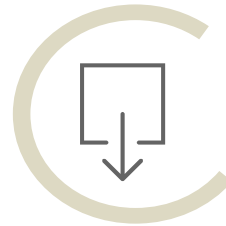


Objekte

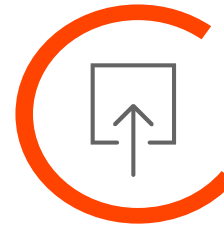
Ereignisse - Was passiert einem Objekt



Eigenschaften



Methoden



Ereignisse



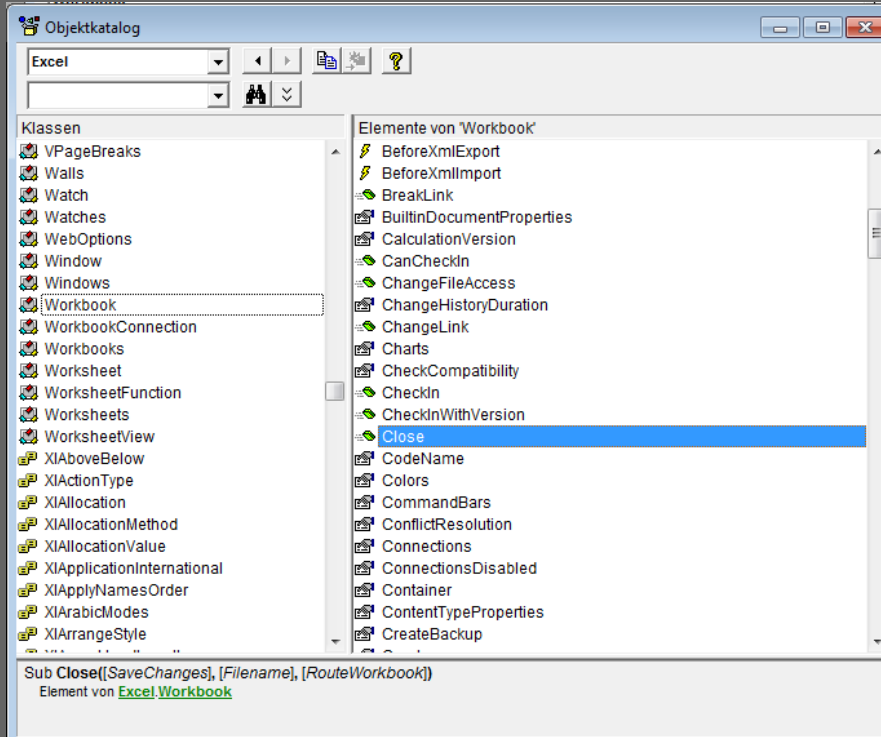
Ereignisse

Eine Aktion auf die das Objekt reagiert

- Ereignisse sind Dinge die passieren können
- Das Ereignis eines Objektes steht hinter dem Objekt mit einem Unterstrich getrennt
 - **Objekt_Ereignis**
 - Workbook_Open
 - Worksheet_Change
- Ereignisprozeduren werden nicht direkt ausgewählt sondern sind eine "Folgerscheinung" einer Aktion des Benutzers
- Die Benennung ist nicht veränderbar

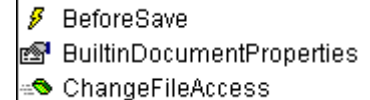


Der Objektkatalog



Das Verzeichnis der Dinge

- Aufruf des Objektkataloges im VBA mit F2
 - Der kleine Blitz ist ein Ereignis
 - Das Händchen ist eine Eigenschaft
 - Das grüne Symbol ist eine Methode



Befehls Syntax

Wie ist der grundsätzliche Aufbau eines VBA Befehls ?

- VBA Befehlsstrukturen sind unterschiedlich komplex
- Einfache Befehle bestehen aus dem Objekt und deren Eigenschaft oder Methode
 - Objekt.Methode
 - `Range("C4").Select`
 - In diesem Fall ist der Aufbau simpel, das Objekt Range wird mit seiner Methode Select verwendet
 - Objekt.Eigenschaft
 - `Columns("F:G").ColumnWidth=20`
 - Das Objekt Spalte wird mit der Eigenschaft Spaltenbreite verwendet



Befehls Syntax

Wie ist der grundsätzliche Aufbau eines VBA Befehls

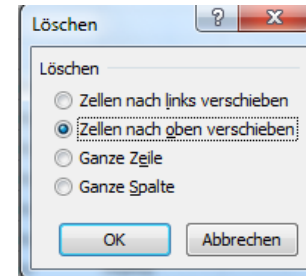
- Neben dem einfachen Aufbau Objekt und Methode kann es auch zu der Kombination der Eigenschaft UND der Methode kommen
 - Objekt.Eigenschaft.Methode
 - ActiveWorkbook.Names.Add
 - Das Objekt "Excel Datei" hat die Eigenschaft Name und ich füge eine Namen dazu



Befehls Syntax

Der grundsätzliche Aufbau eines VBA Befehls

- Einige Befehle verursachen im Excel eine Fragestellung, so wird beim Löschen einer Zelle die Frage gestellt, wie mit den vorhandenen Zellen umzugehen ist.
- Die Antwort ist hier in Form einer Option anzugeben
 - Objekt.Methode Argument:=Konstante
 - Selection.Delete Shift:=xlUp
- Die Option „nach oben“ wird mit dem Argument Shift:=xlUp angegeben
- xlUp ist eine so genannten Konstante, ich habe keine freie Eingabemöglichkeit
- Der Präfix xl zeigt, das es eine Excel Besonderheit ist



Befehls Syntax

Der grundsätzliche Aufbau eines VBA Befehls

- Beim Erstellen eines Namens im Excel wird die Frage nach dem Namen und dem Geltungsbereich gestellt.
- Im VBA werden die Argumente ebenfalls übergeben

```
ActiveWorkbook.Names.Add Name:="Umsatz", RefersTo:=Worksheets("Zellenbeispiele").Range("A$15:C$21")
```

The screenshot shows the 'Neuer Name' dialog box in Microsoft Excel. The 'Name' field is filled with 'Umsatz'. The 'Bereich' dropdown menu is set to 'Arbeitsmappe'. The 'Bezieht sich auf' field contains the formula '=Zellenbeispiele!\$A\$15:\$C\$21'. There are two red circles with white numbers: '1' is next to the 'Name' field, and '2' is next to the 'Bezieht sich auf' field. The dialog box has 'OK' and 'Abbrechen' buttons at the bottom.



Befehls Syntax

Die With Anweisung

- Werden mehrere Eigenschaften eines Objektes angesprochen so wiederholen sich einige Elemente in der Programmierung
 - Selection.Font.Name = "Arial",
Selection.Font.Size = "20",
Selection.Font.Bold = True
Selection.Font.Color = RGB(168, 1, 99)
- Zur Vereinfachung können die sich wiederholenden Elemente gebündelt werden
 - With Selection.Font
.Name = "Arial"
.Size = "20"
.Bold = True
.Color = RGB(168, 1, 99)
End With



Vielen Dank für die Aufmerksamkeit



Dipl.-Ök. Maik Ramftel
Am Brabrinke 14
30519 Hannover
0172 / 81 93 687
Ramftel@Ramses.de
www.ramses.de

