

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

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

- Objektorientierte Skriptsprache.
- Eingebettete Programmiersprache in Microsoft Office.
- Interpretative Sprache.

Einsatzmöglichkeiten

- Automatisierung von Berechnungen.
- Automatisierung von immer wiederkehrenden Befehlsfolgen.
- Manipulation von Objekten in einer Microsoft Office Anwendung.
- Benutzergesteuerte Eingabe von Daten.

Literatur

- Verschiedene Bücher zur Programmierung von Microsoft Office Anwendungen in Abhängigkeit der Version
- Handbücher des IT Services

Informationen im Web

- <https://msdn.microsoft.com/de-de/library/office/ee814735%28v=office.14%29.aspx>
- <https://msdn.microsoft.com/en-us/library/office/mt346046.aspx>
- <https://msdn.microsoft.com/de-de/library/office/ff192120.aspx>
- <http://www.vba-tutorial.de/>

Öffnen der Office-Anwendung

- Klick auf das dazugehörige Icon auf dem Desktop.
- Öffnen der Office-Anwendung entsprechend der Windows-Version.

Öffnen des VBA-Editors in Microsoft Excel

- Menüband *Entwicklertools*.
- *Visual Basic* in der Gruppe *Code*.

Öffnen des VBA-Editors in Microsoft Access

- Menüband *Datenbanktools*. *Visual Basic* in der Gruppe *Makros*.
- Menüband *Erstellen*. *Visual Basic* in der Gruppe *Makros und Code*.

Schließen des VBA-Editors

- Klick auf die Schließen-Schaltfläche in der Titelleiste.
- Andere Möglichkeit: Menü *Datei – Schließen und zurück zu ...*

Zwischen VBA-Editor und Wirt wechseln

- Über die Taskleiste von Windows.
- Andere Möglichkeit: *Ansicht – Microsoft ...* am unteren Rand des Menüs.

VBA-Editor (Beispiel)

The screenshot shows the Microsoft Visual Basic for Applications editor. The title bar reads "Microsoft Visual Basic for Applications - frontend_Handbuchdatenbank - [mySQLSetting (Code)]". The menu bar includes "Datei", "Bearbeiten", "Ansicht", "Einfügen", "Debuggen", "Ausführen", "Extras", "Add-Ins", and "Fenster". The toolbar contains various icons for file operations, editing, and execution. The left pane shows a project tree for "Projekt - frontend_Handbuch" with a sub-project "frontend_Handbuch (frontend_Handbuchdatenbank)". Under "Module", there is a list of modules including "dsDictionaryMethod", "DeklarationDLL", "globaleVariablen", "myBinding", "myEmailSetting", "myFormSetting", "myRibbonCallback", "myRibbonSetting", "mySQLSetting", "myTemplatesFind", "startenDatenbank", and several "StorageProcedure_..." modules. The main window displays the code for "mySQLSetting (Code)" in the "Deklarationen" tab. The code includes:

```

Option Compare Database
Option Explicit

Enum enmSQLTyp
    auswaehlen = 1
    einfuegen = 2
    loeschen = 3
    bearbeiten = 4
End Enum

Function getBoolean(ByVal boolWert As Boolean) As Integer
    getBoolean = CInt(boolWert)
End Function

Function getText(strWert As String) As String
    getText = "" & strWert & ""
End Function

Function getCurrency(curWert As Currency) As String
    getCurrency = Str(curWert)
End Function

Function getDate(dateWert As Date) As String
    getDate = Format(dateWert, "\#mm\~dd\~yyyy\#")
End Function

Function executesQL(strSQL As String, Optional strMode As enmSQL
    ' Rückgabe: Anzahl Datensätze | DatensatzID bei nur einem ge

    Dim dbs As DAO.Database
    Dim rs As DAO.Recordset

    Err.Clear
    
```

Titelleiste

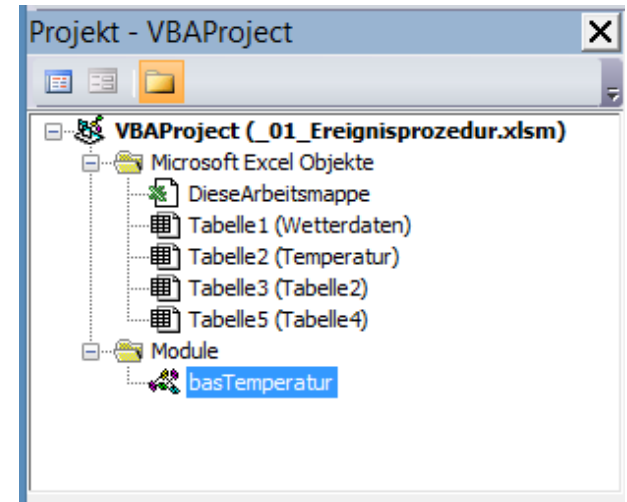
- Icon *Systemmenü* am linken Rand. Sammlung von Befehlen zum Vergrößern und Verkleinern des Fensters.
- Die Icons *Minimieren*, *Verkleinern / Maximieren* und *Schließen* am rechten Rand.
- Als Wirt eines VBA-Editors wird immer eine Datei in einer Microsoft Office – Anwendung geöffnet. Der Name des Wirts wird angezeigt.
- Der Name des, im VBA-Editors aktiven Moduls wird angezeigt. Module sammeln Code zu einem bestimmten Thema.

Menüleiste und Symbolleiste

- Sammlung von Befehlen in Text oder als Symbol, die in einem VBA-Editor genutzt werden können.
- Menü *Datei* enthält Befehle zum Speichern des Wirts und dem Export / Import von Modulen.
- Menü *Bearbeiten* zeigt Befehle zum Kopieren und Einfügen von Code an.
- Menü *Debuggen* sammelt Befehle zum Testen und Starten eines VBA-Programms.

Fenster

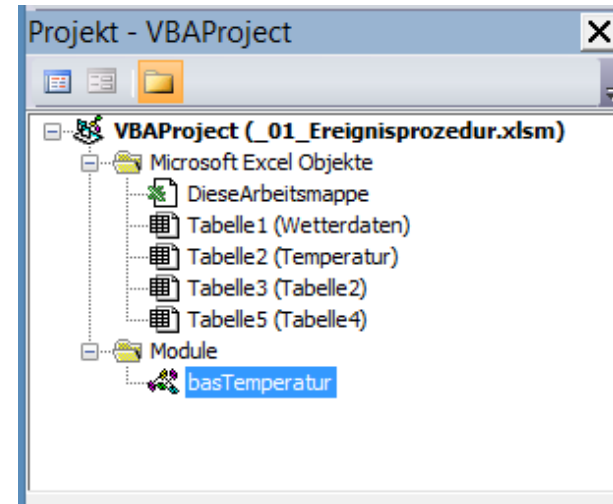
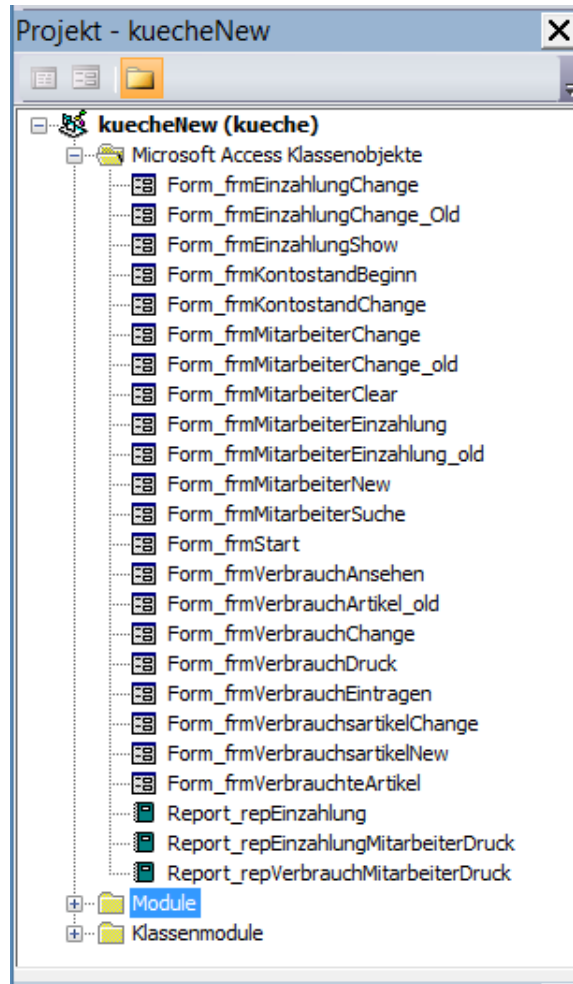
- Mit Hilfe von Drag & Drop kann ein Fenster unterhalb der Symbolleiste an jeder beliebigen Stelle positioniert werden.
- Mit Hilfe der Schaltfläche in der rechten oberen Ecke der Titelleiste wird ein Fenster geschlossen.
- Fenster werden mit Hilfe des passenden Befehls im Menü *Ansicht* eingeblendet.



Projekt-Explorer

- Struktur eines VBA-Projekts in einer Microsoft Office Anwendung.
- Sammlung von allen Modulen in einem Wirt.
- Angedockt am linken Rand des VBA-Editors.

Beispiel



Aufbau

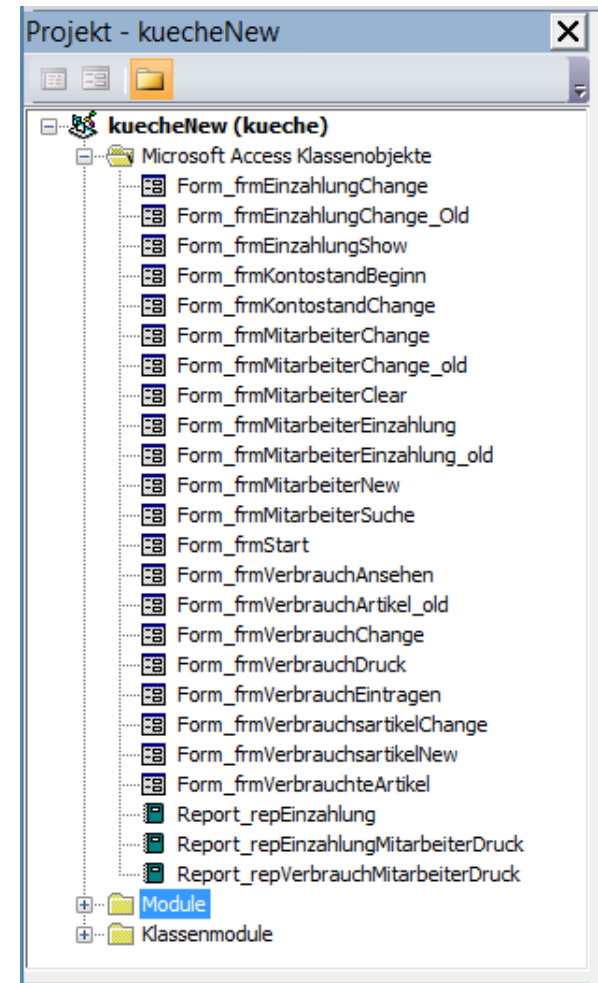
- In der Titelleiste wird der Name des Projekts sowie die Schließen-Schaltfläche angezeigt.
- Darunter befindet sich die Symbolleiste mit den Icons
 - „Zeige zum gewählten Element den Code an“.
 - „Zeige das passende Objekt zum Code an“.
 - „Sortierung mit Hilfe von Ordnern“.
- Unterhalb der Symbolleiste wird der Code alphabetisch oder in Abhängigkeit von Ordnern sortiert angezeigt.

„Ordner“ im Projekt-Explorer

- Wurzelordner. Name des Wirts.
- *Microsoft Access Klassenobjekte* oder *Microsoft ... Objekte* enthält eine Auflistung von Objekten in Bezug auf die Microsoft Office Anwendung. Die aufgelisteten Objekte führen automatisch in Abhängigkeit eines Ereignisses VBA-Code aus. Links vom Objektnamen wird die dazugehörige Objektkategorie als Symbol angezeigt.
- *Module*. Code, der unabhängig von einem Microsoft Objekt lauffähig ist.
- *Klassenmodule*. Benutzerdefinierte Klassen.

„Ordner“ öffnen oder schließen

- Klick auf das Pluszeichen: Ein Ordner wird geöffnet. Der Inhalt wird angezeigt.
- Klick auf das Minuszeichen: Ein Ordner wird geschlossen.



Codefenster

- Anzeige von VBA-Code aus einem Modul.
- Angedockt am rechten Rand des VBA-Editors.
- Schaltflächen am unteren linken Rand: Im Codefenster wird nur die aktive Prozedur oder das gesamte Modul angezeigt.

The screenshot shows the VBA Code Window with the 'Declarations' tab selected. The code is as follows:

```

Option Compare Database
Option Explicit

Enum enmSQLTyp
    auswaehlen = 1
    einfuegen = 2
    loeschen = 3
    bearbeiten = 4
End Enum

Function getBoolean(ByVal boolWert As Boolean) As Integer
    getBoolean = CInt(boolWert)
End Function

Function getText(strWert As String) As String
    getText = "" & strWert & ""
End Function

Function getCurrency(curWert As Currency) As String
    getCurrency = Str(curWert)
End Function

Function getDate(dateWert As Date) As String
    getDate = Format(dateWert, "\#mm\ -dd\ -yyyy\#")
End Function

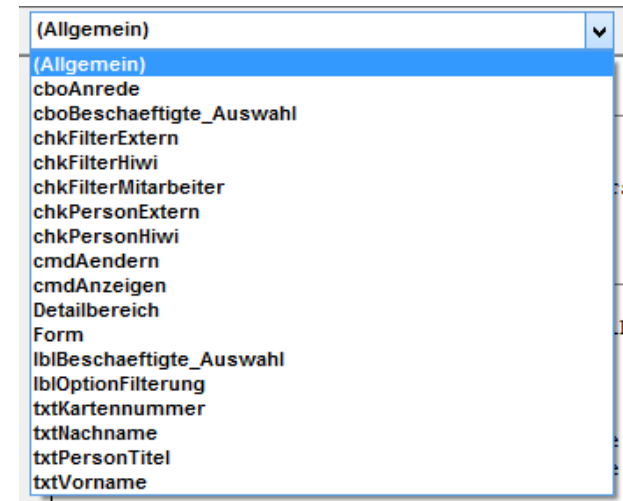
Function executeSQL(strSQL As String, Optional strMode As enmSQLTyp) As Integer
    ' Rückgabe: Anzahl Datensätze | DatensatzID bei nur einem Datensatz

    Dim dbs As DAO.Database
    Dim rs As DAO.Recordset

    Err.Clear
    
```

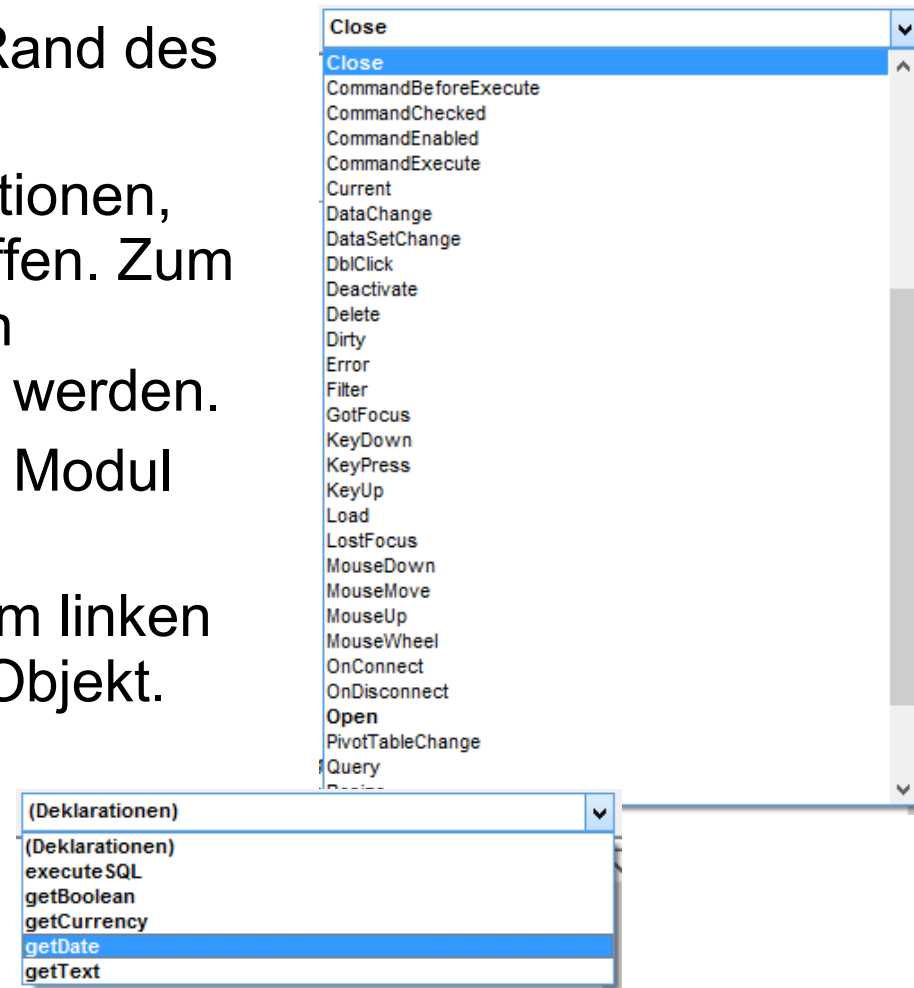
DopDown-Feld „Objekt“

- Anzeige am linken, oberen Rand des Codefensters.
- Element (*Allgemein*). Code, der das gesamte Modul betrifft. Nicht gebundene Module haben nur diesen Eintrag.
- Auflistung der verschiedenen Elemente zu einem Objekt. Zum Beispiel werden die Namen der Steuerelemente eines Formulars oder Berichts in Microsoft Access angezeigt.



DropDown-Feld „Prozedur“

- Anzeige am rechten oberen Rand des Codefensters.
- Eintrag (*Deklaration*). Deklarationen, die das gesamte Modul betreffen. Zum Beispiel Variablen, die in allen Prozeduren im Modul genutzt werden.
- Alle Prozeduren, die in einem Modul vorhanden sind.
- Ereignisprozeduren zu dem, im linken Kombinationsfeld gewählten Objekt.



Module

- Beschreibung eines Projekts mit Hilfe der Befehle einer Programmiersprache.
- Zusammenfassung von Code in der Programmiersprache VBA.

Module

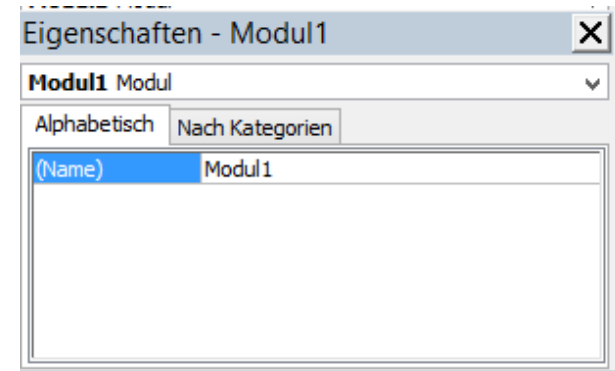
- Standardmodule sind Behälter für Prozeduren. Standardmodule sind Dateien mit der Endung „.bas“.
Standardmodule sind an kein Microsoft Office Objekt gebunden.
- Beschreibungen von Objekten mit Hilfe von Klassen werden in einer Datei mit der Endung „.cls“ gespeichert.

Standardmodul einfügen

- ... in Microsoft Access: Menüband *Erstellen*. Icon *Modul* in der Gruppe *Makros und Code*.
- ... im VBA-Editor: *Einfügen – Modul*.

Name des Standardmoduls

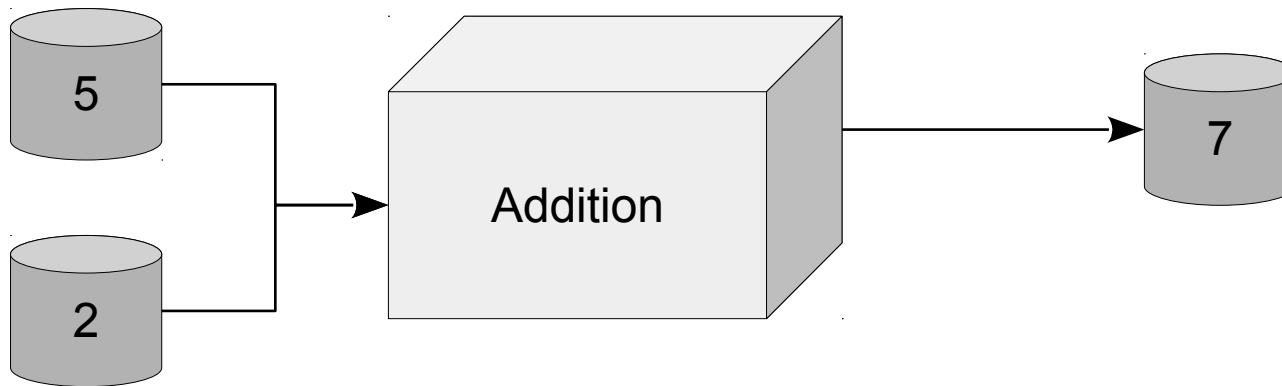
- In die Zeile (*Name*) des Eigenschaftenfensters wird ein eindeutiger Name für das Modul geschrieben.
- Die angezeigten Eigenschaften sind abhängig von dem, im Projekt-Explorer gewählten Objekt.



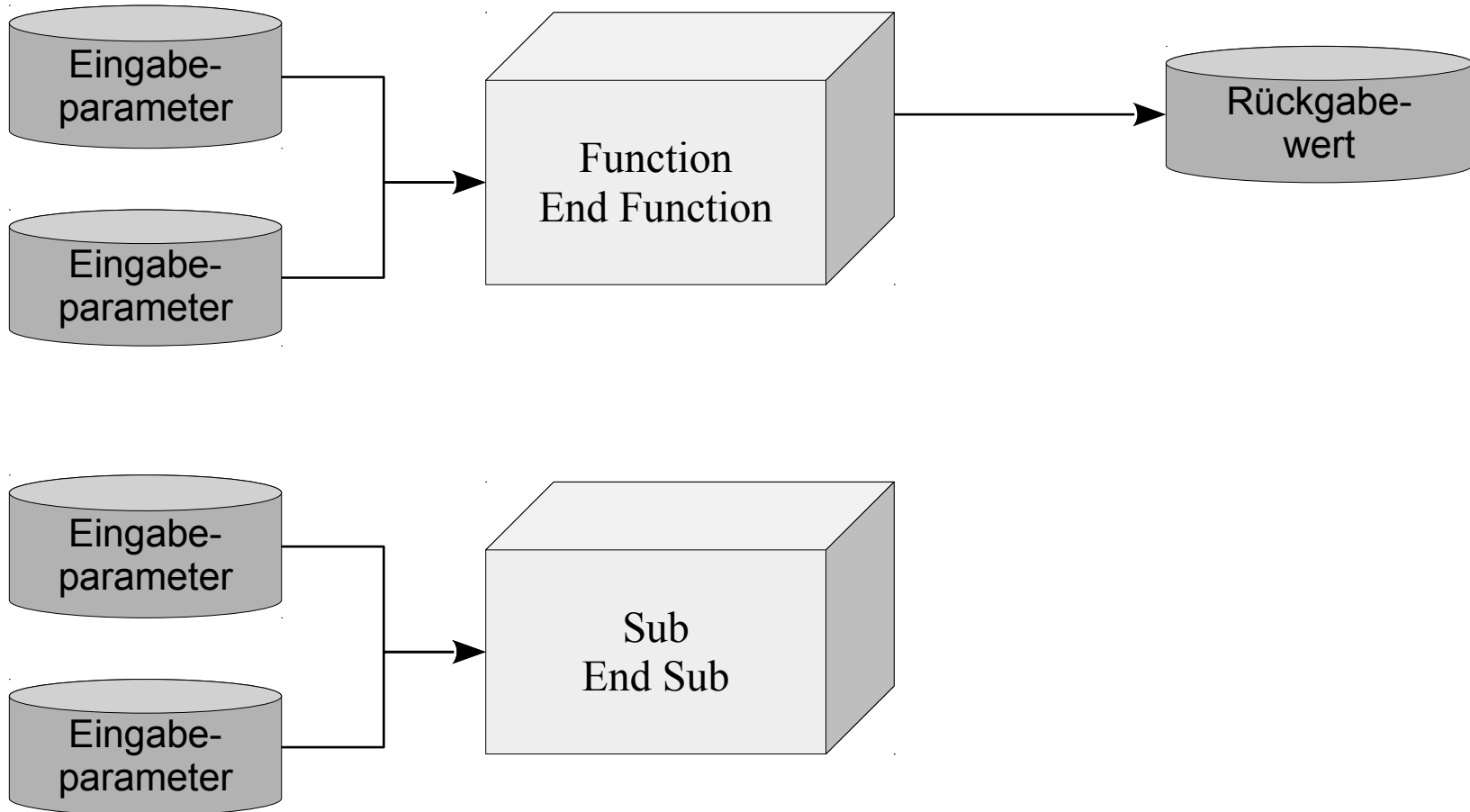
Prozeduren

- Darstellung der verschiedenen Aufgaben in einem Projekt.
- Kapselung von Anweisungen, die zu einer in sich abgeschlossenen Aufgabe gehören.
- Beschreibung einer bestimmten Funktionalität mit Hilfe einer Programmiersprache.

Beispiel: Addition von zwei Zahlen



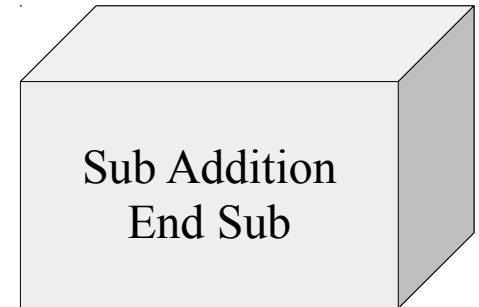
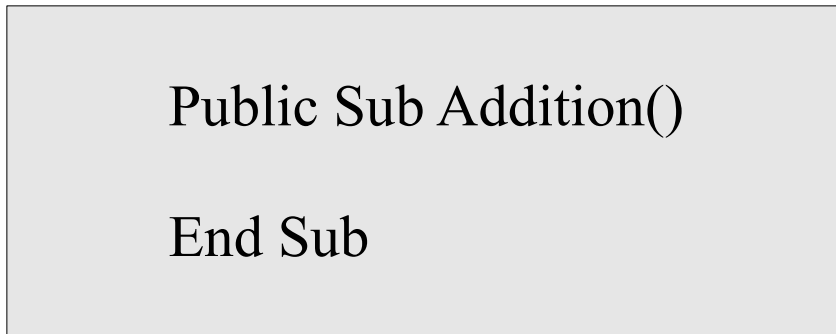
Implementierung in VBA



... einfügen

- Ein Modul ist geöffnet.
- *Einfügen – Prozedur*. In das Textfeld *Name* wird eine Bezeichnung für die Prozedur eingegeben. Die Option *Sub* ist ausgewählt. Das Gerüst einer Prozedur wird durch Klick auf die Schaltfläche *OK* eingefügt.
- Andere Möglichkeit: Die Einfügemarke befindet sich nicht innerhalb des Code `Sub ... End Sub`. Das Gerüst der Prozedur wird mit Hilfe der Tastatur eingegeben.

Grundgerüst einer Prozedur



- Zwischen den Schlüsselworten Sub ... End Sub wird Code gekapselt.
- Der Code (die Prozedur) wird mit dem Namen der Prozedur (Addition) gestartet.
- Für den Anwender ist die Prozedur eine Blackbox. Wie die Daten in der Prozedur verarbeitet werden, weiß er nicht. Der Anwender hat nur Informationen darüber, wie die Prozedur aufgerufen wird.

Erläuterung

Public Sub Addition()

End Sub

- Die Schlüsselworte Sub ... End Sub kennzeichnen eine Prozedur, die keinen Wert an den Aufrufer zurückgibt.
- Kopf der Prozedur: Public Sub Addition().
- Fuß der Prozedur: End Sub.

Prozedur-Kopf

```
Public Sub Addition()
```

```
End Sub
```

```
[Zugriff] Sub [Name]()
```

```
End Sub
```

- Der Kopf beginnt mit dem Schlüsselwort Sub.
- Dem Schlüsselwort folgt der Name der Prozedur. Der Name ist in einem Modul eindeutig.
- Dem Namen folgen die runden Klammern. In den runden Klammern können der Prozedur Parameter für den Start übergeben werden.
- Das Schlüsselwort Public kennzeichnet eine öffentliche Prozedur, die jeder aufrufen kann.

Name einer Prozedur

- Der Name ist frei wählbar.
- Der Name ist in einem Modul eindeutig.
- Der Name spiegelt die implementierte Aufgabe wieder.

Regeln für benutzerdefinierte Namen

- Jeder benutzerdefinierter Name beginnt mit einem Buchstaben.
- Die Bezeichnung besteht nur aus den Buchstaben a..z, A..Z, dem Unterstrich und den Zahlen 0..9.
- Die Groß- und Kleinschreibung von benutzerdefinierten Namen wird nicht beachtet.
- Ein benutzerdefinierter Namen kann maximal 255 Zeichen haben.
- VBA-Schlüsselwörter oder von VBA, definierte Funktionsnamen können nicht als benutzerdefinierte Namen genutzt werden.

Einfügung von Code in einer Prozedur

- Klick auf eine freie Zeile zwischen *Sub ... End Sub*.
- An der Position der Einfügemarke wird mit Hilfe der Tastatur Code eingegeben.

Beispiel

Public Sub Addition()

Deklaration von
Konstanten und
Variablen

```
Const zahlL As Integer = 5  
Const zahlR As Integer = 2  
Dim summe As Integer
```

Anweisungen

```
summe = zahlL + zahlR  
Debug.Print summe
```

End Sub

Anweisungen

- Befehle für den Computer in einer Programmiersprache.
- Anweisungen in VBA enden mit der Zeile.
- Operanden, Operatoren und Schlüsselwörter werden nach bestimmten Regeln verbunden.

Schlüsselwörter

- Befehlssatz der Programmiersprache VBA.
- Standardmäßig werden Schlüsselwörter durch die blaue Schriftfarbe gekennzeichnet.

Operanden

- Variablen, die mit `Dim [name] As [datentyp]` definiert sind. Platzhalter für einen beliebigen Wert von einem bestimmten Datentyp. Der Wert des Platzhalters kann durch Anweisungen verändert werden.
- Konstanten, die mit `Const [name] As [datentyp] = [wert]` definiert sind. Der Wert in dem Platzhalter kann nicht durch Anweisungen geändert werden.
- Literale als Ganzzahlen (4), Dezimalzahlen (4.5), Zeichenkette ("abcd") oder als Datums- / Zeitwert (#2/14/2016 10:15:23#).

Operatoren

- Arithmetische Operatoren berechnen einen Wert aus ein oder zwei Operanden.
- Vergleichsoperatoren vergleichen zwei Werte.
- Logische Operatoren verknüpfen verschiedene Ausdrücke.

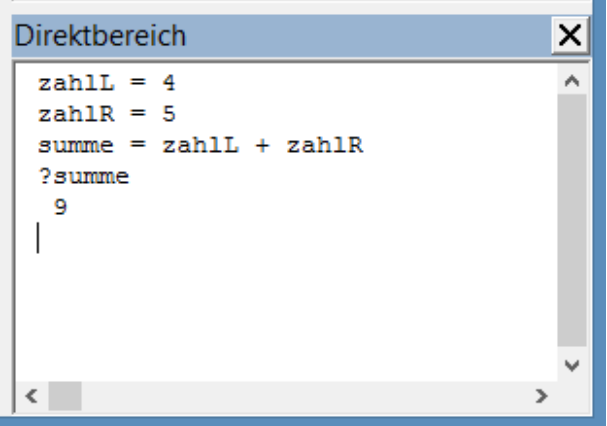
Ausgabe in das Direktfenster

```
Debug.Print summe
```

- Mit Hilfe des Befehls `Debug.Print` wird ein Wert (`summe`) in das Direktfenster geschrieben.

Direktfenster

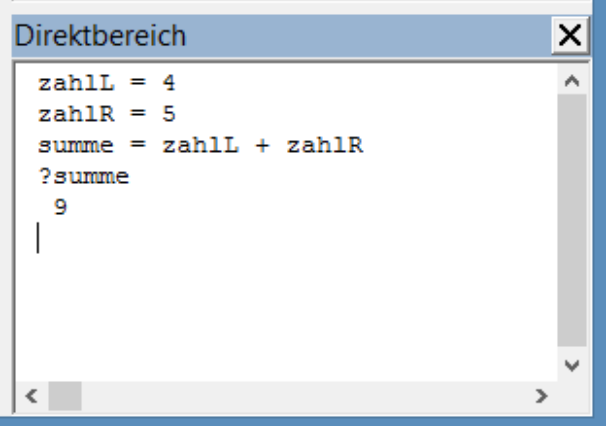
- Testen von Anweisungen.
- Überprüfen von Variablen, die in einer Prozedur definiert sind.



```
Direktbereich
zahlL = 4
zahlR = 5
summe = zahlL + zahlR
?summe
9
|
```

... nutzen

- In einer Prozedur werden mit Hilfe von `Debug.Print` Werte von Variablen und Konstanten im Direktbereich ausgegeben.
- Anweisungen können direkt zum Testen in den Direktbereich eingegeben werden.
- `?[variable]` zeigt den Wert einer Variablen, die im Direktbereich definiert wurde, an.



```
Direktbereich
zahlL = 4
zahlR = 5
summe = zahlL + zahlR
?summe
9
```

Starten der Prozedur

- Die Einfügemarke befindet sich zwischen den Schlüsselwörtern Sub ... End Sub der zu startenden Prozedur.
- <F5> startet die Prozedur. Die Anweisungen in der Prozedur werden vollständig von oben nach unten abgearbeitet.
- Oder: *Ausführen* – *Sub/UserForm ausführen* im VBA-Editor.