Prof. Dr. Hans-Christian Walter

Access

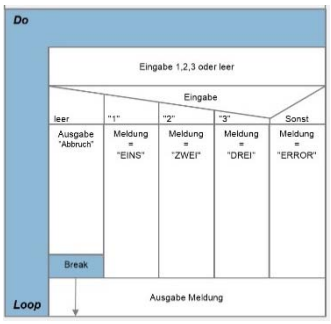
Teil 2
Programmierung in VBA

Begleitheft

Inhalt

Einführung

Im Videobuch "Access Programmierung in VBA" werden alle wichtigen Anweisungen, die Sie für die Programmierung benötigen, ausführlich anhand von Programmbeispielen dargestellt. Sie lernen Sie die die Entwicklungsumgebung des **VBA-Editors** und **Debuggers** in Access kennen, um Ihre Programme schnell und zuverlässig zu erstellen und zu testen.



Struktogramm

Nach einer Einführung in die Werkzeuge und Verweise von Access wird auf Variable, Konstante und **Elementardatentypen** eingegangen sowie auf das **Array**, das Ihnen vielfältige Möglichkeiten für die Programmierung bietet. Zu den Grundlagen gehören die **Boolesche Operationen,** um Bedingungen zu formulieren, und **String-Operationen**, um auf Teile eines Textes zugreifen und Veränderungen vornehmen zu können.

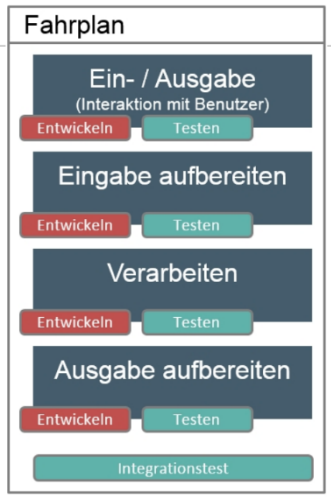
Weiterhin werden die einzelnen Anweisungen zur **Auswahl** und **Wiederholung** dargestellt, mit denen sich die Programmlogik darstellen lässt. Selbstverständlich folgt die Programmentwicklung den allgemein anerkannten Regeln der **Strukturierten Programmierung**.

Anwendungsbeispiel

Für komplexere Anwendungen ist das Wissen um **Module**, **Prozeduren** und den **Gültigkeitsbereich von Variablen** erforderlich, wozu auch der Einsatz **öffentlicher Prozeduren und Variablen** gehört.

Im Teil Prozeduren im Formular wird die **Verwendung von VBA in Formularen** behandelt, wobei an einem ausführlichen Beispiel die Zusammenhänge zwischen dem Formular, den **Ereignisprozeduren** und dem VBA-Code gezeigt werden.

Zum Abschluss gehe ich auf die Vorgehensweise bei der Programmentwicklung ein. Die Aufgabenstellung wird – beginnend bei der Benutzerschnittstelle – zerlegt und schrittweise entwickelt und getestet. So kommt der Entwickler sicher ans Ziel und erhält ein fehlerfreies Programm.



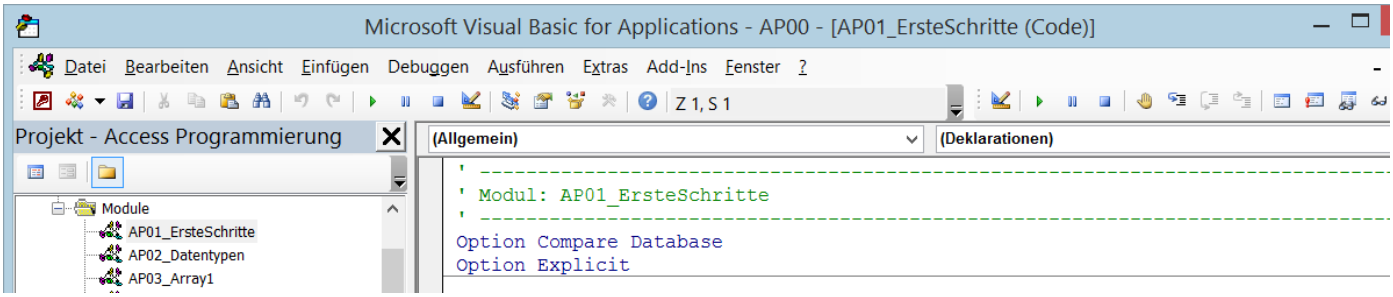
Vorgehensweise

Ergänzt wird der Kurs durch Begleitmaterial mit Programmbeispielen, das als Beispieldatenbank heruntergeladen werden kann.

Das Videobuch ist sehr umfassend und stellt in systematischer Abfolge die Inhalte in 16 Lektionen innerhalb von ca. 3 Stunden dar.

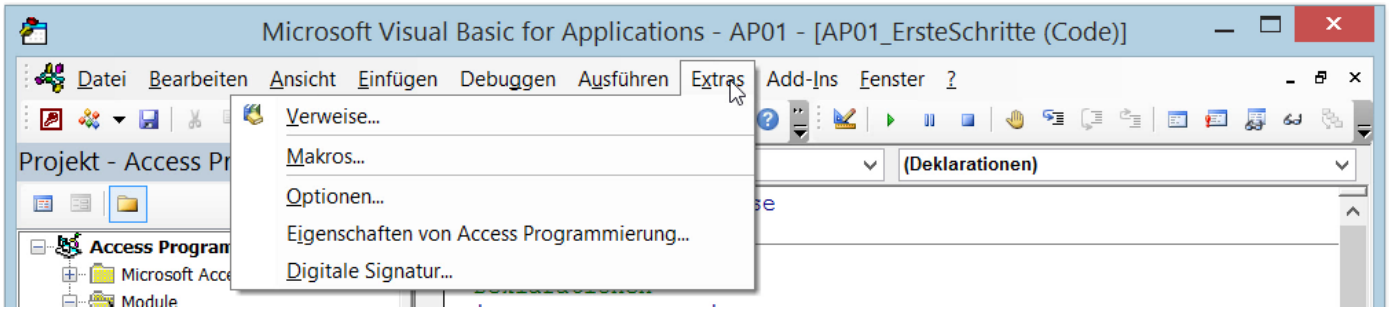
Das vorliegende Videobuch Access Programmieren in VBA (Teil 2) erscheint in der Reihe Access, in der auch die Videobücher Access Grundlagen (Teil 1) und Access Tabellenverarbeitung in VBA (Teil 3) erschienen sind. Es ist möglich, den Teil 2 auch ohne den Teil 1 durchzuarbeiten.

Lektion 1 Werkzeuge - Datenbank AP00.accdb

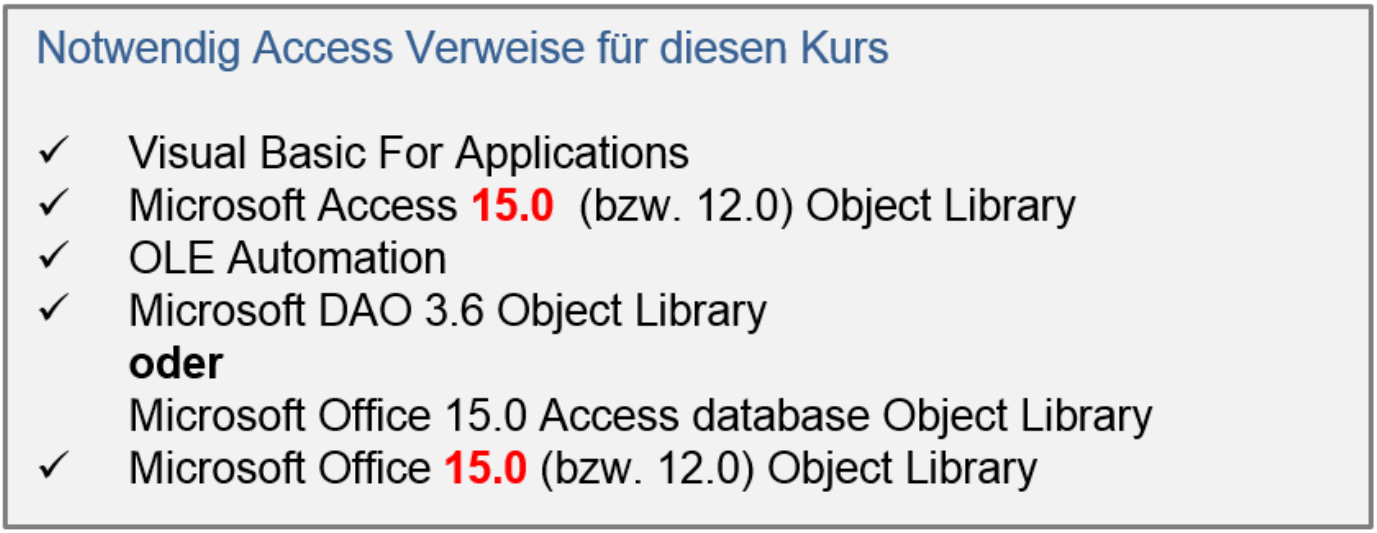


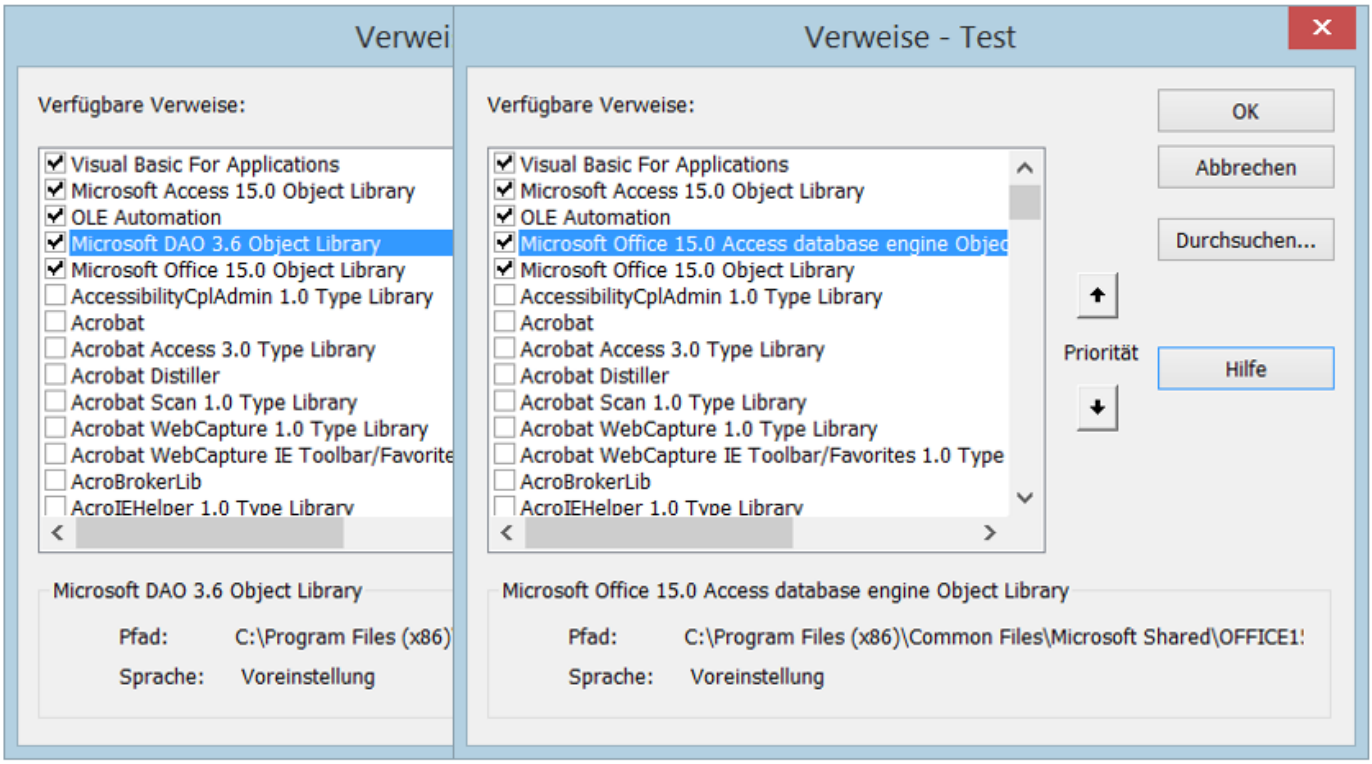

```
' ---------------------------------------------------------------------------------------
' Modul: AP01_ErsteSchritte
' ---------------------------------------------------------------------------------------
Option Compare Database
Option Explicit

Sub ErsteSchritte()
' ---------------------------------------------------------------------------------------
' Deklarationen
' ---------------------------------------------------------------------------------------

Dim strText As String

' ---------------------------------------------------------------------------------------
' Anweisungen
' ---------------------------------------------------------------------------------------

Let strText = "Hallo Welt"
Debug.Print strText

Let strText = "Erste Schritte!"
Debug.Print strText

Debug.Print "Ende"

End Sub
```

Lektion 2 Verweise - Datenbank AP00.accdb



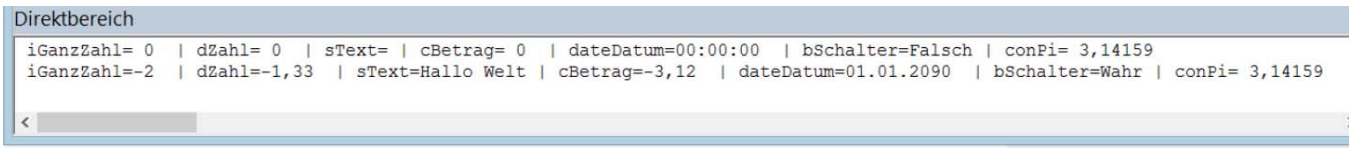
Verweise werden im VBA-Editor gesetzt





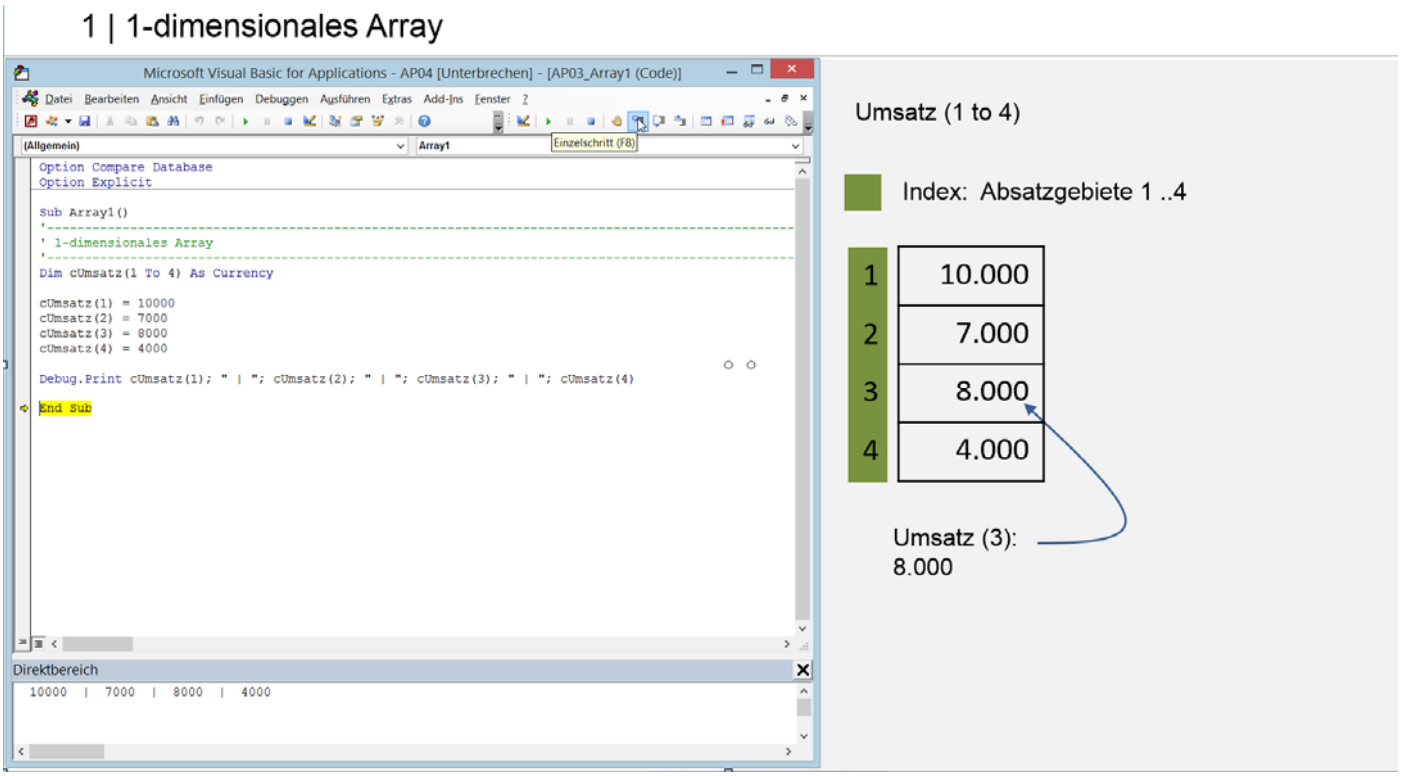
Lektion 3 Variablen, Konstanten - Datenbank AP00.accdb

```
' ----------------------------------------------------------------------------------------------------
' Modul: AP02_Datentypen
' ----------------------------------------------------------------------------------------------------
Option Compare Database
Option Explicit

Sub Datentypen()
' ----------------------------------------------------------------------------------------------------
' Deklaration der Variablen
' ----------------------------------------------------------------------------------------------------
Dim lngGanzZahl As Long                     'Ganzzahlig: 0,1,2 ... -1,-2,...
Dim dblZahl As Double                       'Dezimalzahl:  1.3, 0.00345 .... Gleitkommazahl
Dim strText As String                       'Text: "Hallo Welt" (Stets in Hochkommata einschließen)
Dim curBetrag As Currency                   'Währung: -3.12(€)
Dim dateDatum As Date                       'Datum: 01.01.2090
Dim boolSchalter As Boolean                 'Boolesche Variable: True/False

' ----------------------------------------------------------------------------------------------------
' verkürzte Schreibweise, wird nachfolgend verwendet
' ----------------------------------------------------------------------------------------------------
Dim iGanzZahl As Long                       'Ganzzahlig: 0,1,2 ... -1,-2,...
Dim dZahl As Double                         'Dezimalzahl:  1.3, 0.00345 .... Gleitkommazahl
Dim sText As String                         'Text: "Hallo Welt" (Stets in Hochkommata einschließen)
Dim cBetrag As Currency                     'Währung: -3.12(€)
'Dim dateDatum As Date                      'Datum: 01.01.2090, bereits oben definiert
Dim bSchalter As Boolean                    'Boolesche Variable: True/False

'Datentyp [Integer] kann durch [Long] , Datentyp [Single] kann durch [Double] ersetzt werden.

' ----------------------------------------------------------------------------------------------------
' Deklaration der Konstanten
' ----------------------------------------------------------------------------------------------------
Const conPi = 3.14159                       'Fester Wert: PI

' ----------------------------------------------------------------------------------------------------
' Ausgabe der Werte im Direktbereich
' ----------------------------------------------------------------------------------------------------
' Unterstrich ist Fortsetzungszeichen einer Anweisung)

Debug.Print "iGanzZahl="; iGanzZahl; " | "; "dZahl="; dZahl; " | "; "sText="; sText; " | "; _
        "cBetrag="; cBetrag; " | "; "dateDatum="; dateDatum; " | "; "bSchalter="; bSchalter; " | "; "conPi="; conPi

' ----------------------------------------------------------------------------------------------------
' Initialisierung
' ----------------------------------------------------------------------------------------------------
iGanzZahl = -2
dZahl = -1.33
sText = "Hallo Welt"
cBetrag = -3.12
dateDatum = "01.01.2090"
bSchalter = True

' ----------------------------------------------------------------------------------------------------
' Ausgabe der Werte im Direktbereich
' ----------------------------------------------------------------------------------------------------
Debug.Print "iGanzZahl="; iGanzZahl; " | "; "dZahl="; dZahl; " | "; "sText="; sText; " | "; _
        "cBetrag="; cBetrag; " | "; "dateDatum="; dateDatum; " | "; "bSchalter="; bSchalter; " | "; "conPi="; conPi
End Sub
```



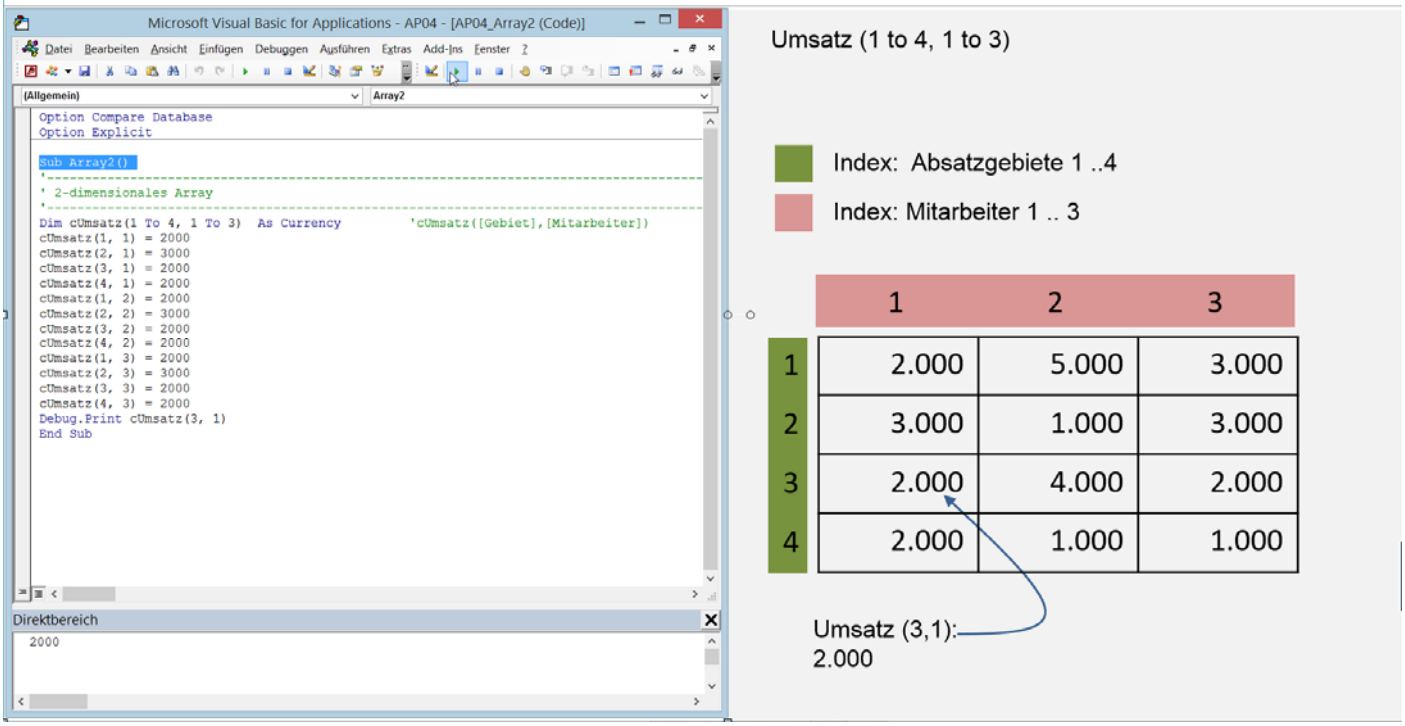
Ergebnis im Direktbereich

Lektion 4 Arrays - Datenbank AP00.accdb

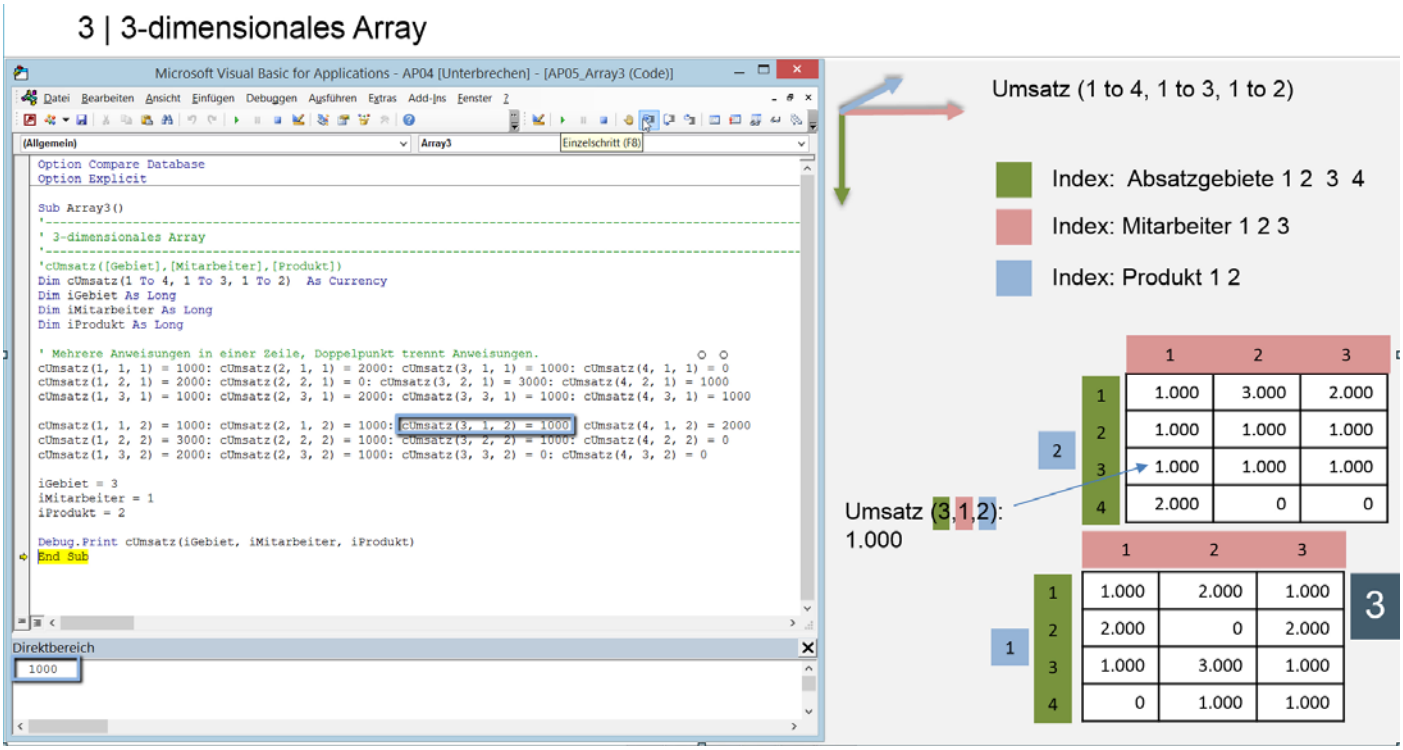



```
' ---------------------------------------------------------------------------------------
' Modul: AP03_Array1
' ---------------------------------------------------------------------------------------
Option Compare Database
Option Explicit

Sub Array1()
'-----------------------------------------------------------------------------------------------------------
' 1-dimensionales Array
'-----------------------------------------------------------------------------------------------------------
Dim cUmsatz(1 To 4) As Currency

cUmsatz(1) = 10000
cUmsatz(2) = 7000
cUmsatz(3) = 8000
cUmsatz(4) = 4000

Debug.Print cUmsatz(1); " | "; cUmsatz(2); " | "; cUmsatz(3); " | "; cUmsatz(4)

End Sub
```

2 | 2-dimensionales Array

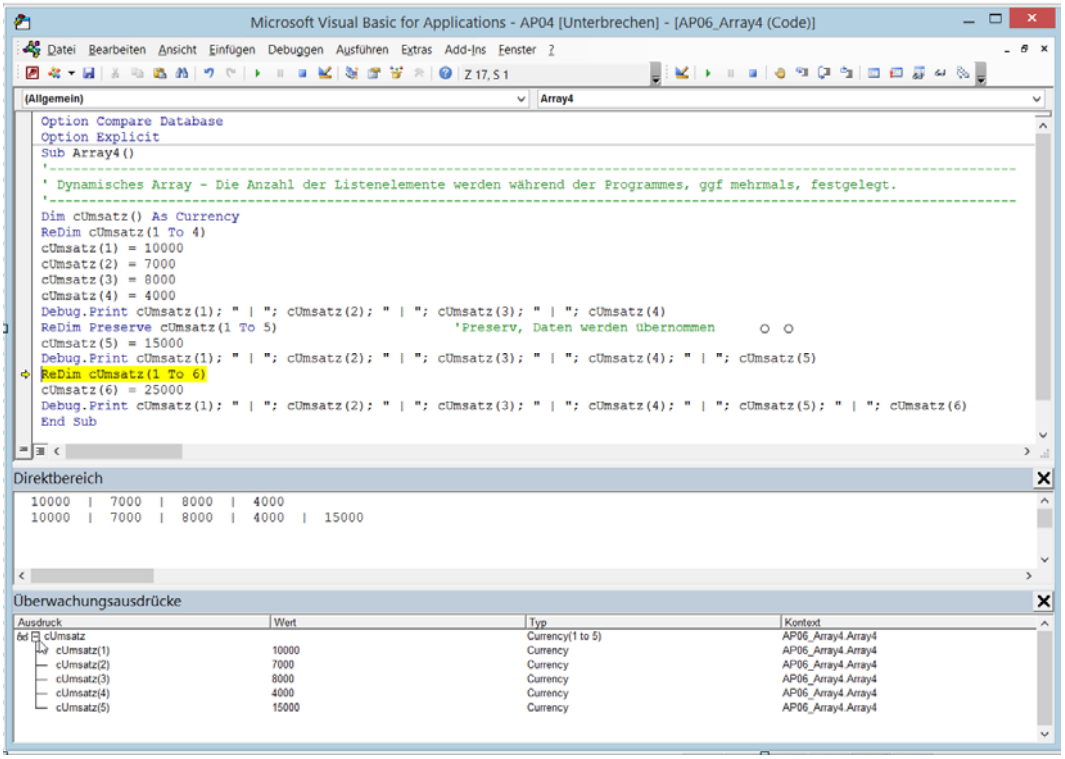



```
' ---------------------------------------------------------------------------------------------------------------
' Modul: AP04_Array2
' ---------------------------------------------------------------------------------------------------------------
Option Compare Database
Option Explicit

Sub Array2()
'---------------------------------------------------------------------------------------------------------------
' 2-dimensionales Array
'---------------------------------------------------------------------------------------------------------------
Dim cUmsatz(1 To 4, 1 To 3)  As Currency         'cUmsatz([Gebiet],[Mitarbeiter])
cUmsatz(1, 1) = 2000
cUmsatz(2, 1) = 3000
cUmsatz(3, 1) = 2000
cUmsatz(4, 1) = 2000
cUmsatz(1, 2) = 2000
cUmsatz(2, 2) = 3000
cUmsatz(3, 2) = 2000
cUmsatz(4, 2) = 2000
cUmsatz(1, 3) = 2000
cUmsatz(2, 3) = 3000
cUmsatz(3, 3) = 2000
cUmsatz(4, 3) = 2000
Debug.Print cUmsatz(3, 1)
End Sub
```




```
' ---------------------------------------------------------------------------------------------------
' Modul: AP05_Array3
' ---------------------------------------------------------------------------------------------------
Option Compare Database
Option Explicit

Sub Array3()
'---------------------------------------------------------------------------------------------------------------------
' 3-dimensionales Array
'---------------------------------------------------------------------------------------------------------------------
'cUmsatz([Gebiet],[Mitarbeiter],[Produkt])
Dim cUmsatz(1 To 4, 1 To 3, 1 To 2)  As Currency
Dim iGebiet As Long
Dim iMitarbeiter As Long
Dim iProdukt As Long

' Mehrere Anweisungen in einer Zeile, Doppelpunkt trennt Anweisungen.
cUmsatz(1, 1, 1) = 1000: cUmsatz(2, 1, 1) = 2000: cUmsatz(3, 1, 1) = 1000: cUmsatz(4, 1, 1) = 0
cUmsatz(1, 2, 1) = 2000: cUmsatz(2, 2, 1) = 0: cUmsatz(3, 2, 1) = 3000: cUmsatz(4, 2, 1) = 1000
cUmsatz(1, 3, 1) = 1000: cUmsatz(2, 3, 1) = 2000: cUmsatz(3, 3, 1) = 1000: cUmsatz(4, 3, 1) = 1000

cUmsatz(1, 1, 2) = 1000: cUmsatz(2, 1, 2) = 1000: cUmsatz(3, 1, 2) = 1000: cUmsatz(4, 1, 2) = 2000
cUmsatz(1, 2, 2) = 3000: cUmsatz(2, 2, 2) = 1000: cUmsatz(3, 2, 2) = 1000: cUmsatz(4, 2, 2) = 0
cUmsatz(1, 3, 2) = 2000: cUmsatz(2, 3, 2) = 1000: cUmsatz(3, 3, 2) = 0: cUmsatz(4, 3, 2) = 0

iGebiet = 3
iMitarbeiter = 1
iProdukt = 2

Debug.Print cUmsatz(iGebiet, iMitarbeiter, iProdukt)
End Sub
```

4 | Dynamisches Array




```
' ------------------------------------------------------------------------------------------------------------------------
' Modul: AP06_Array4
' ------------------------------------------------------------------------------------------------------------------------
Option Compare Database
Option Explicit

Sub Array4()
'------------------------------------------------------------------------------------------------------------------------
' Dynamisches Array
'------------------------------------------------------------------------------------------------------------------------
Dim cUmsatz() As Currency
ReDim cUmsatz(1 To 4)
cUmsatz(1) = 10000
cUmsatz(2) = 7000
cUmsatz(3) = 8000
cUmsatz(4) = 4000

Debug.Print cUmsatz(1); " | "; cUmsatz(2); " | "; cUmsatz(3); " | "; cUmsatz(4)

ReDim Preserve cUmsatz(1 To 5)                  'Preserv, Daten werden übernommen
cUmsatz(5) = 15000
Debug.Print cUmsatz(1); " | "; cUmsatz(2); " | "; cUmsatz(3); " | "; cUmsatz(4); " | "; cUmsatz(5)

ReDim cUmsatz(1 To 6)
cUmsatz(6) = 25000
Debug.Print cUmsatz(1); " | "; cUmsatz(2); " | "; cUmsatz(3); " | "; cUmsatz(4); " | "; cUmsatz(5); " | "; cUmsatz(6)


End Sub
```

Lektion 5 Boolesche Operationen - Datenbank AP00.accdb

Aussagelogik

Aussage 1:	A ist ein Buchstabe	Richtig
Aussage 2:	9 ist ein Buchstabe	Falsch
Aussage 3:	A ist ein Buchstabe UND 9 ist ein Buchstabe	Falsch
Aussage 4:	A ist ein Buchstabe ODER 9 ist ein Buchstabe	Richtig

Boolesche Variaben nehmen die Werte True (Richtig) oder False (Falsch) an.
Sie können für den Wahrheitsgehalt von Aussagen stehen und über boolesche Operatoren verknüpft werden.

Boolesche Operatoren
And (UND)
Or (ODER)
XOR (entweder – oder)
Not(NICHT)

Die oben genannten Aussagen können wie folgt, abgebildet werden:
b1, b2, b3, b4 seinen boolesche Variablen:
b1 = True
b2 = False
b3 = b1 And b2 = False
b4 = b1 Or b2 = True

Boolesche Variablen und Operationen

1. Grundregeln für Boolesche Operationen

```
True   And  True    = True
True   And  False   = False
False  And  True    = False
False  And  False   = False

True   Or   True    = True
True   Or   False   = True
False  Or   True    = True
False  Or   False   = False
```

2. Prioritätsregeln:

- Klammerausdruck geht vor Operatoren
- Not geht vor And
- And geht vor Or

Beispiele:

```
b1    = True And False = False
b2    = True Or False = True.

b3    = False And False Or True
      = -------False------- Or True
      = True

b4    = False And (False Or True)
      = False And ------True--------
      = False

b5    = (True Or False) And (True And False) And True Or True
      = -------True------- And -------False-------- And True Or True
      = ----------------------------False---------------------------- Or True
      = True

b6    = (b1 Or b2 Or b3) And (b4 Or b5)
      = (False Or True Or True) And (False Or True)
      = -------------True------------- And -------True -------
      = True

b7    = Not (Not b1 And b2) And b3
      = Not (True And True) And True
      = Not --------True -------- And True
      = ----------False--------- And True
      = False
```

```
' ---------------------------------------------------------------------------------------------
' Modul: AP07_BoolescheOperationen
' ---------------------------------------------------------------------------------------------------------
Option Compare Database
Option Explicit

Sub BoolescheOperationen()
' ---------------------------------------------------------------------------------------------------------
' Beispiele für Boolesche Operationen
' ---------------------------------------------------------------------------------------------------------

Dim b1 As Boolean
Dim b2 As Boolean
Dim b3 As Boolean
Dim b4 As Boolean
Dim b5 As Boolean
Dim b6 As Boolean
Dim b7 As Boolean
b1 = True And False                                    'Ergebnis False
b2 = True Or False                                     'Ergebnis True
b3 = True And False Or True                            'Ergebnis True
b4 = False And (False Or True)                         'Ergebnis Falsch
b5 = (True Or False) And (True And False) And True Or True   'Ergebnis Wahr
b6 = (b1 Or b2 Or b3) And (b4 Or b5)                   'Ergebnis Wahr
b7 = Not (Not b1 And b2) And b3

Debug.Print "b1="; b1; "  b2="; b2; "  b3="; b3; "   b4="; b4; "  b5="; b5; "  b6="; b6; "  b7="; b7;

End Sub
```

Direktbereich

```
b1=Falsch  b2=Wahr  b3=Wahr   b4=Falsch  b5=Wahr  b6=Wahr  b7=Falsch
```

Lektion 6 Stringoperationen - Datenbank AP00.accdb

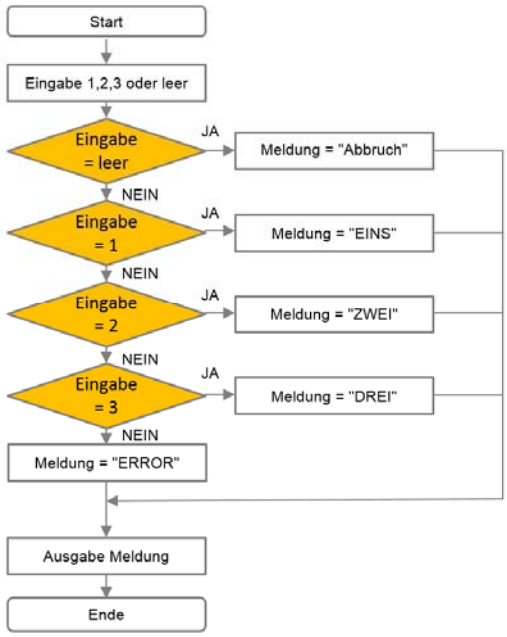
Access-Methode	Beschreibung
IsEmpty("Hallo" *String*) → False *Boolean*	Funktion prüft, ob Ausdruck leer ist
IsEmpty("" *String*) → True *Boolean*	Funktion prüft, ob Ausdruck leer ist
Trim(" Hallo " *String*) → "Hallo" *String*	Funktion trennt führende und endende Leerzeichen ab.
Mid("Hallo" *String*,2 *Integer*,3 *Integer*)→ "allo" *String*	Funktion gibt Teilstring eines Text ab einer Stelle in einer Länge zurück
Len("Hallo" *String*) → 5 *Integer*	Funktion gibt Anzahl der Zeichen eines Textes zurück
LCase("Hallo" *String*) → "hallo" *String*	Funktion gibt Text in Kleinbuchstaben zurück
UCase("Hallo" *String*) → "HALLO" *String*	Funktion gibt Text in Großbuchstaben zurück
InStr(2 *Integer*, "Ha Ha" *String*, "Ha" *String*) → 4 *Integer*	Funktion gibt Position eines Suchtextes in einem Text ab einer Stelle an. Rückgabewert = 0, wenn Suchstring nicht gefunden wird
IsNumeric("12" *String*) → True *Boolean*	Funktion prüft, ob Ausdruck numerisch ist
IsNumeric("Hallo" *String*) → False *Boolean*	Funktion prüft, ob Ausdruck numerisch ist
CLng("12" *String*) → 12 *Long*	Funktion konvertiert eine Zahl in den Datentyp Long
CDbl("12" *String*) → 12 *Double*	Funktion konvertiert eine Zahl in den Datentyp Double
CStr(12 *Zahl*) → "12" *String*	Funktion konvertiert eine Zahl in den Datentyp String

Access-Methoden für Stringoperationen

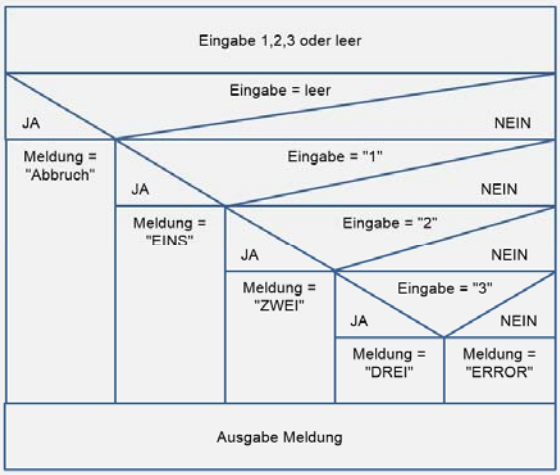
```
' ----------------------------------------------------------------------------------------------------
' Modul: AP08_StringOperationen
' ----------------------------------------------------------------------------------------------------
Option Compare Database
Option Explicit

Sub StringOperationen()
' ----------------------------------------------------------------------------------------------------
' Beispiele für String-Operationen
' ----------------------------------------------------------------------------------------------------
Dim sText, sTeil, sZahl As String
Dim iPos, iLang, iZahl As Long
Dim b1 As Boolean
Dim dZahl As Double

sText = Empty                       'sText auf Leerstring setzen, wie sText = ""
b1 = IsEmpty(sText)                 'Prüfen, ob Variable sText ein Leerstring ist
sText = "Welt"
sText = " Hallo " & sText & " 24   "  'Mit dem Zeichen & werden Teilstrings verbunden
sText = Trim(sText)                 'führende und endende Leerzeichen abtrennen
sTeil = Mid(sText, 12, 2)           'Teilstring von sText an der Stelle 12 in der Länge 2
sText = sText & " " & sTeil         'Zeichen an der Stelle 5 in der Länge 2: 24
sText = LCase(sText)                'Kleinschreibung
sText = UCase(sText)                'Großschreibung
iPos = InStr(1, sText, "24")        'Position, beginnend bei 1, in sText, des Suchstrings "24"
iPos = InStr(iPos + 1, sText, "24") 'Position, beginnend bei iPos + 1, in sText, des Suchstrings "24"
Mid(sText, iPos, 2) = "12"          'Teilstring von sText, beginnend bei iPos, in der Länge 2
iLang = Len(sText)                  'Länge von sText

'Teilstring von sText, beginnend bei iPos, in der Länge: Länge von sText - iPos + 1
sZahl = Mid(sText, iPos, Len(sText) - iPos + 1)

b1 = IsNumeric(sText)               'Abfrage auf numerisch
b1 = IsNumeric(sZahl)               'Abfrage auf numerisch
iZahl = CLng(sZahl) + 1             'Umwandlung in Datentyp Long
dZahl = CDbl(sZahl) + 0.1           'Umwandlung in Datentyp Double
sZahl = CStr(dZahl - 0.1)           'Umwandlung in Datentyp String

End Sub
```

Lektion 7 Auswahl (IF) - Datenbank AP00.accdb

Programmablaufplan



Struktogramm

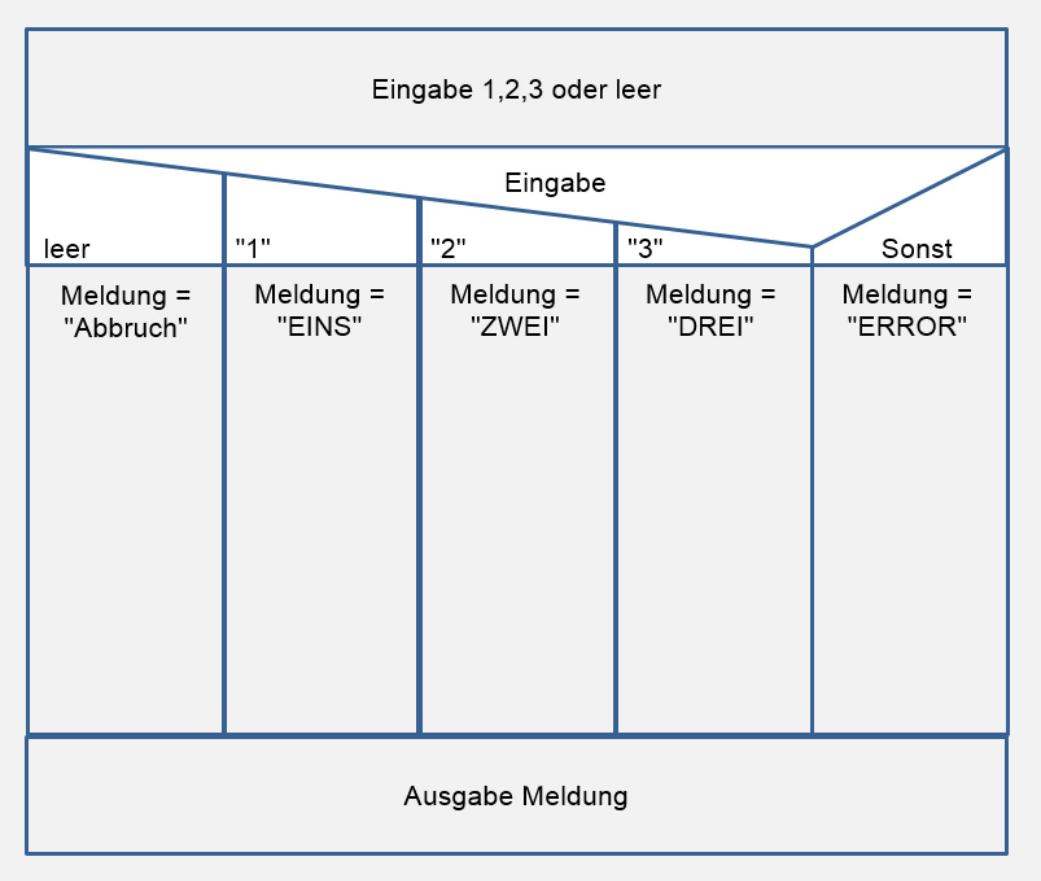


Graphische Darstellung der Programmlogik

```
' ---------------------------------------------------------------------------------------------------------
' Modul: AP09_Auswahl
' ---------------------------------------------------------------------------------------------------------
Option Compare Database
Option Explicit

Sub Auswahl()
'---------------------------------------------------------------------------------------------------------
' Geschachtelte If-Anweisungen
'---------------------------------------------------------------------------------------------------------
Dim sEingabe As String
Dim sMeldung As String


sEingabe = Empty
sEingabe = InputBox("Bitte geben Sie die Zahl 1, 2 oder 3 ein.", "Eingabeaufforderung")
'---------------------------------------------------------------------------------------------------------
If sEingabe = "" Then                                   'If  1
   sMeldung = "Eingabe ist nicht erfolgt oder wurde abgebrochen!"
Else                                                    'Else
   If sEingabe = "1" Then                               'If  2
      sMeldung = "Sie haben eine EINS eingegeben!"
   Else                                                 'Else
      If sEingabe = "2" Then                            'If  3
         sMeldung = "Sie haben eine ZWEI eingegeben!"
      Else                                              'Else
         If sEingabe = "3" Then                         'If  4
            sMeldung = "Sie haben eine DREI eingegeben!"
         Else                                           'Else
            sMeldung = "ERROR!"
         End If                                         'End 4
      End If                                            'End 3
   End If                                               'End 2
End If                                                  'End 1
'---------------------------------------------------------------------------------------------------------
MsgBox sMeldung, , "Ergebnis"
End Sub
```

```
' ---------------------------------------------------------------------------------------------------------------
' Modul: AP11_AuswahlBeispiele
' ---------------------------------------------------------------------------------------------------------------
Option Compare Database
Option Explicit

Sub AuswahlBeispiele()
'---------------------------------------------------------------------------------------------------------------
' Unterschiedliche Formen der If-Notation
'---------------------------------------------------------------------------------------------------------------
Dim dZahl As Double
Dim iZahl As Long
Dim b1 As Boolean

dZahl = 1.1
iZahl = 1
b1 = True

If dZahl >= iZahl Then MsgBox dZahl & " ist >= " & iZahl      'Beispiel 1, bedingte Anweisung

If dZahl >= iZahl Then                                        'Beispiel 2, klassische Schreibweise ohne Else-Zweig
   MsgBox dZahl & " ist >= " & iZahl
End If

If b1 Then MsgBox "b1 ist " & b1                              'Beispiel 3, Boolesche Variable

If b1 And dZahl >= 1.1 Then MsgBox "b1 ist Wahr und dZahl ist >= 1.1"   'Beispiel 4, Boolesche Operation

If b1 Then b1 = False: MsgBox "Wert von b1 wurde auf Falsch gesetzt"    'Beispiel 5, mehrere Anweisungen in einer Zeile

If b1 Then                                                    'Beispiel 6, nur Else-Zweig
Else
   MsgBox "b1 hat den Wert " & b1
End If

If b1 Then                                                    'Beispiel 7, nur Else-Zweig, mehrere Anweisungen in einer Zeile
Else: MsgBox "b1 hat den Wert " & b1
End If

If b1 Then: Else: MsgBox "b1 hat den Wert " & b1               'Beispiel 8, Nur eine Zeile mit mehreren Anweisungen

End Sub
```

Lektion 8 Fallauswahl - Datenbank AP00.accdb



Struktogramm mit Fallauswahl

```
' ---------------------------------------------------------------------------------------------------------
' Modul: AP12_Fallauswahl
' ---------------------------------------------------------------------------------------------------------
Option Compare Database
Option Explicit

Sub Fallauswahl()
'---------------------------------------------------------------------------------------------------------
' Select...Case Anweisung
'---------------------------------------------------------------------------------------------------------
Dim sEingabe As String
Dim sMeldung As String

sEingabe = Empty
sEingabe = InputBox("Bitte geben Sie die Zahl 1, 2 oder 3 ein.", "Eingabeaufforderung")

Select Case sEingabe
Case ""
   sMeldung = "Eingabe ist nicht erfolgt oder wurde abgebrochen!"
Case "1"
   sMeldung = "Sie haben eine EINS eingegeben!"
Case "2"
   sMeldung = "Sie haben eine ZWEI eingegeben!"
Case "3"
   sMeldung = "Sie haben eine DREI eingegeben!"
Case Else
   sMeldung = "ERROR!"
End Select

MsgBox sMeldung, , "Ergebnis"
End Sub
```

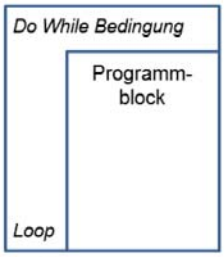
```
' ---------------------------------------------------------------------------------------------------------------
' Modul: AP13_Fallauswahl2
' ---------------------------------------------------------------------------------------------------------------
Option Compare Database
Option Explicit

Sub Fallauswahl2()
'---------------------------------------------------------------------------------------------------------------
' Select...Case Anweisung - kompakte Schreibweise
'---------------------------------------------------------------------------------------------------------------
Dim sEingabe As String
Dim sMeldung As String
sEingabe = Empty
sEingabe = InputBox("Bitte geben Sie die Zahl 1, 2 oder 3 ein.", "Eingabeaufforderung")

Select Case sEingabe
Case "":    sMeldung = "Eingabe ist nicht erfolgt oder wurde abgebrochen!"
Case "1":   sMeldung = "Sie haben eine EINS eingegeben!"
Case "2":   sMeldung = "Sie haben eine ZWEI eingegeben!"
Case "3":   sMeldung = "Sie haben eine DREI eingegeben!"
Case Else:  sMeldung = "ERROR!"
End Select

MsgBox sMeldung, , "Ergebnis"
End Sub
```

```
' ---------------------------------------------------------------------------------------------------------------
' Modul: AP14_Fallauswahl3
' ---------------------------------------------------------------------------------------------------------------
Option Compare Database
Option Explicit

Sub Fallauswahl3()
'---------------------------------------------------------------------------------------------------------------
' Select...Case Anweisung
'---------------------------------------------------------------------------------------------------------------
Dim sEingabe As String
Dim sMeldung As String
sEingabe = Empty
sEingabe = InputBox("Bitte geben Sie die Zahl 1, 2 oder 3 ein.", "Eingabeaufforderung")

Select Case sEingabe
Case ""
   sMeldung = "Eingabe ist nicht erfolgt oder wurde abgebrochen!"
Case "1", "3"  'Es kann eine Werteliste für den Fall angegeben werden, getrennt durch Kommata
   sMeldung = sEingabe & " ist eine ungerade Zahl!"
Case "2"
   sMeldung = sEingabe & " ist eine gerade Zahl!"
Case Else
   sMeldung = "ERROR!"
End Select

MsgBox sMeldung, , "Ergebnis"
End Sub
```

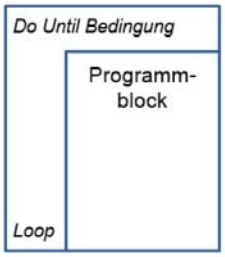
```
' ---------------------------------------------------------------------------------------------------------------
' Modul: AP15_Fallauswahl4
' ---------------------------------------------------------------------------------------------------------------
Option Compare Database
Option Explicit

Sub Fallauswahl4()
'---------------------------------------------------------------------------------------------------------------
' Select...Case Anweisung
'---------------------------------------------------------------------------------------------------------------
Dim sEingabe As String
Dim sMeldung As String
sEingabe = Empty
sEingabe = InputBox("Bitte geben Sie die Zahl 1, 2 oder 3 ein.", "Eingabeaufforderung")

Select Case False
Case sEingabe = ""
   sMeldung = "Eingabe ist nicht erfolgt oder wurde abgebrochen!"
Case sEingabe = "1" Or sEingabe = "3"
   sMeldung = "Sie haben eine ungerade Zahl eingegeben!"
Case sEingabe = "2"
   sMeldung = "Sie haben eine gerade Zahl eingegeben!"
Case Else
   sMeldung = "ERROR!"
End Select

MsgBox sMeldung, , "Ergebnis"
End Sub
```

```
' ---------------------------------------------------------------------------------------------------------------
' Modul: AP16_FallauswahlErweitert
' ---------------------------------------------------------------------------------------------------------------
Option Compare Database
Option Explicit

Sub FallauswahlErweitert()
'---------------------------------------------------------------------------------------------------------------
' Select...Case Anweisung
'---------------------------------------------------------------------------------------------------------------
Dim dZahl As Double
Dim iZahl As Long
Dim b1 As Boolean

dZahl = 0.9             ' Test mit    1.1     1     0.9
iZahl = 1
b1 = True

Select Case True
Case b1 And dZahl >= 1.1                    'Case 1
   MsgBox "b1 ist Wahr und dZahl ist >= 1.1"
Case dZahl > 1.1                            'Case 2
   MsgBox "dZahl ist >= 1.1"
Case dZahl >= iZahl                         'Case 3
   MsgBox dZahl & " ist >= " & iZahl
Case b1                                     'Case 4
  b1 = False
  MsgBox "Wert von b1 wurde auf Falsch gesetzt"
Case Not b1                                 'Case 5
   MsgBox "b1 hat den Wert " & b1
Case Else                                   'Case Else
   MsgBox "Sonst"
End Select

End Sub
```

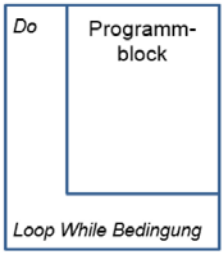
Lektion 9 Schleifen - Datenbank AP00.accdb



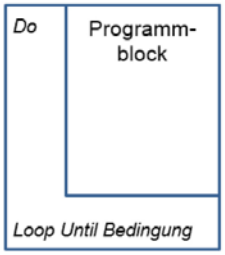
Typ 1:
Vorprüfende Schleife
mit Laufbedingung



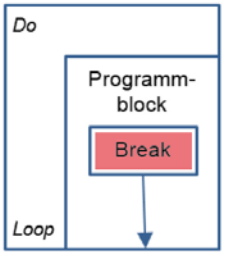
Typ 2:
Vorprüfende Schleife
mit Abbruchbedingung



Typ 3:
Nachprüfende Schleife
mit Laufbedingung



Typ 4:
Nachprüfende Schleife
mit Abbruchbedingung



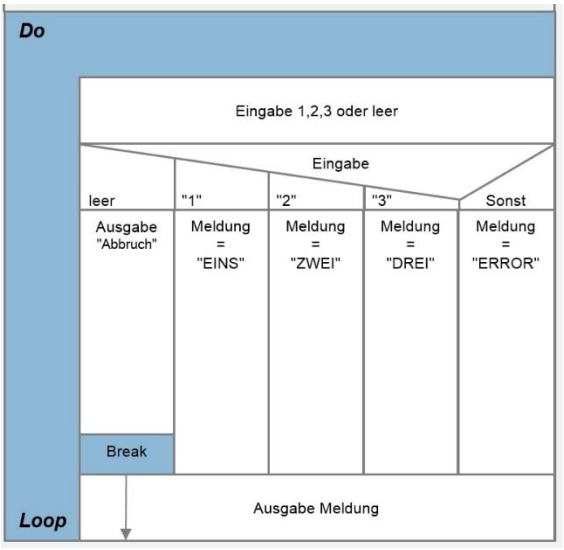
Typ 5
Schleife
mit Abbruch-Anweisung

```
' ----------------------------------------------------------------------------------------------------------------
' Modul: AP17_PreCheckLoopWhile
' ----------------------------------------------------------------------------------------------------------------
Option Compare Database
Option Explicit

Sub PreCheckLoopWhile()
'----------------------------------------------------------------------------------------------------------------
' Vorprüfende Schleife mit Laufbedingung
'----------------------------------------------------------------------------------------------------------------
Dim sEingabe As String
Dim sMeldung As String
Dim bEnde As Boolean

bEnde = False
Do While bEnde = False              'Vorprüfende Schleife mit Laufbedingung
   sEingabe = Empty
   sEingabe = InputBox("Bitte geben Sie die Zahl 1, 2 oder 3 ein.", "Eingabeaufforderung")

   Select Case sEingabe
   Case ""
      sMeldung = "Eingabe ist nicht erfolgt oder wurde abgebrochen!"
      bEnde = True
   Case "1"
      sMeldung = "Sie haben eine EINS eingegeben!"
   Case "2"
      sMeldung = "Sie haben eine ZWEI eingegeben!"
   Case "3"
      sMeldung = "Sie haben eine DREI eingegeben!"
   Case Else
      sMeldung = "ERROR!"
   End Select

   MsgBox sMeldung, , "Ergebnis"
Loop
End Sub
```

```
' ---------------------------------------------------------------------------------------------------------------
' Modul: AP18_PreCheckLoopUntil
' ---------------------------------------------------------------------------------------------------------------
Option Compare Database
Option Explicit

Sub PreCheckLoopUntil()
'---------------------------------------------------------------------------------------------------------------
' Vorprüfende Schleife mit Abbruchbedingung
'---------------------------------------------------------------------------------------------------------------
Dim sEingabe As String
Dim sMeldung As String
Dim bEnde As Boolean

bEnde = False
Do Until bEnde = True                    'Vorprüfende Schleife mit Abbruchbedingung
   sEingabe = Empty
   sEingabe = InputBox("Bitte geben Sie die Zahl 1, 2 oder 3 ein.", "Eingabeaufforderung")

   Select Case sEingabe
   Case ""
      sMeldung = "Eingabe ist nicht erfolgt oder wurde abgebrochen!"
      bEnde = True
   Case "1"
      sMeldung = "Sie haben eine EINS eingegeben!"
   Case "2"
      sMeldung = "Sie haben eine ZWEI eingegeben!"
   Case "3"
      sMeldung = "Sie haben eine DREI eingegeben!"
   Case Else
      sMeldung = "ERROR!"
   End Select

   MsgBox sMeldung, , "Ergebnis"
Loop

 End Sub
```

```
' ---------------------------------------------------------------------------------------------------------------
' Modul: AP19_PostCheckLoopWhile
' ---------------------------------------------------------------------------------------------------------------
Option Compare Database
Option Explicit

Sub PostCheckLoopWhile()
'---------------------------------------------------------------------------------------------------------------
' Nachprüfende Schleife mit Laufbedingung
'---------------------------------------------------------------------------------------------------------------
Dim sEingabe As String
Dim sMeldung As String
Dim bEnde As Boolean

bEnde = False
Do                          'Nachprüfende Schleife mit Laufbedingung
   sEingabe = Empty
   sEingabe = InputBox("Bitte geben Sie die Zahl 1, 2 oder 3 ein.", "Eingabeaufforderung")

   Select Case sEingabe
   Case ""
      sMeldung = "Eingabe ist nicht erfolgt oder wurde abgebrochen!"
      bEnde = True
   Case "1"
      sMeldung = "Sie haben eine EINS eingegeben!"
   Case "2"
      sMeldung = "Sie haben eine ZWEI eingegeben!"
   Case "3"
      sMeldung = "Sie haben eine DREI eingegeben!"
   Case Else
      sMeldung = "ERROR!"
   End Select

   MsgBox sMeldung, , "Ergebnis"
Loop While bEnde = False
End Sub
```

```
' ---------------------------------------------------------------------------------------------------------------
' Modul: AP19_PostCheckLoopUntil
' ---------------------------------------------------------------------------------------------------------------
Option Compare Database
Option Explicit

Sub PostCheckLoopUntil()
'---------------------------------------------------------------------------------------------------------------
' Nachprüfende Schleife mit Abbruchbedingung
'---------------------------------------------------------------------------------------------------------------
Dim sEingabe As String
Dim sMeldung As String
Dim bEnde As Boolean

bEnde = False
Do                                  'Nachprüfende Schleife mit Abbruchbedingung
   sEingabe = Empty
   sEingabe = InputBox("Bitte geben Sie die Zahl 1, 2 oder 3 ein.", "Eingabeaufforderung")

   Select Case sEingabe
   Case ""
      sMeldung = "Eingabe ist nicht erfolgt oder wurde abgebrochen!"
      bEnde = True
   Case "1"
      sMeldung = "Sie haben eine EINS eingegeben!"
   Case "2"
      sMeldung = "Sie haben eine ZWEI eingegeben!"
   Case "3"
      sMeldung = "Sie haben eine DREI eingegeben!"
   Case Else
      sMeldung = "ERROR!"
   End Select

   MsgBox sMeldung, , "Ergebnis"
Loop Until bEnde = True

End Sub
```



Schleife mit Abbruchanweisung (Break)

```
' ---------------------------------------------------------------------------------------------------------------
' Modul: AP21_LoopExit
' ---------------------------------------------------------------------------------------------------------------
Option Compare Database
Option Explicit

Sub LoopExit()
'---------------------------------------------------------------------------------------------------------------
' Schleife mit Abbruchanweisung (Break)
'---------------------------------------------------------------------------------------------------------------
Dim sEingabe As String
Dim sMeldung As String

Do
   sEingabe = Empty
   sEingabe = InputBox("Bitte geben Sie die Zahl 1, 2 oder 3 ein.", "Eingabeaufforderung")

   Select Case sEingabe
   Case ""
      MsgBox "Eingabe ist nicht erfolgt oder wurde abgebrochen!", , "Ergebnis"
      Exit Do    'Break
   Case "1"
      sMeldung = "Sie haben eine EINS eingegeben!"
   Case "2"
      sMeldung = "Sie haben eine ZWEI eingegeben!"
   Case "3"
      sMeldung = "Sie haben eine DREI eingegeben!"
   Case Else
      sMeldung = "ERROR!"
   End Select

   MsgBox sMeldung, , "Ergebnis"
Loop
End Sub
```

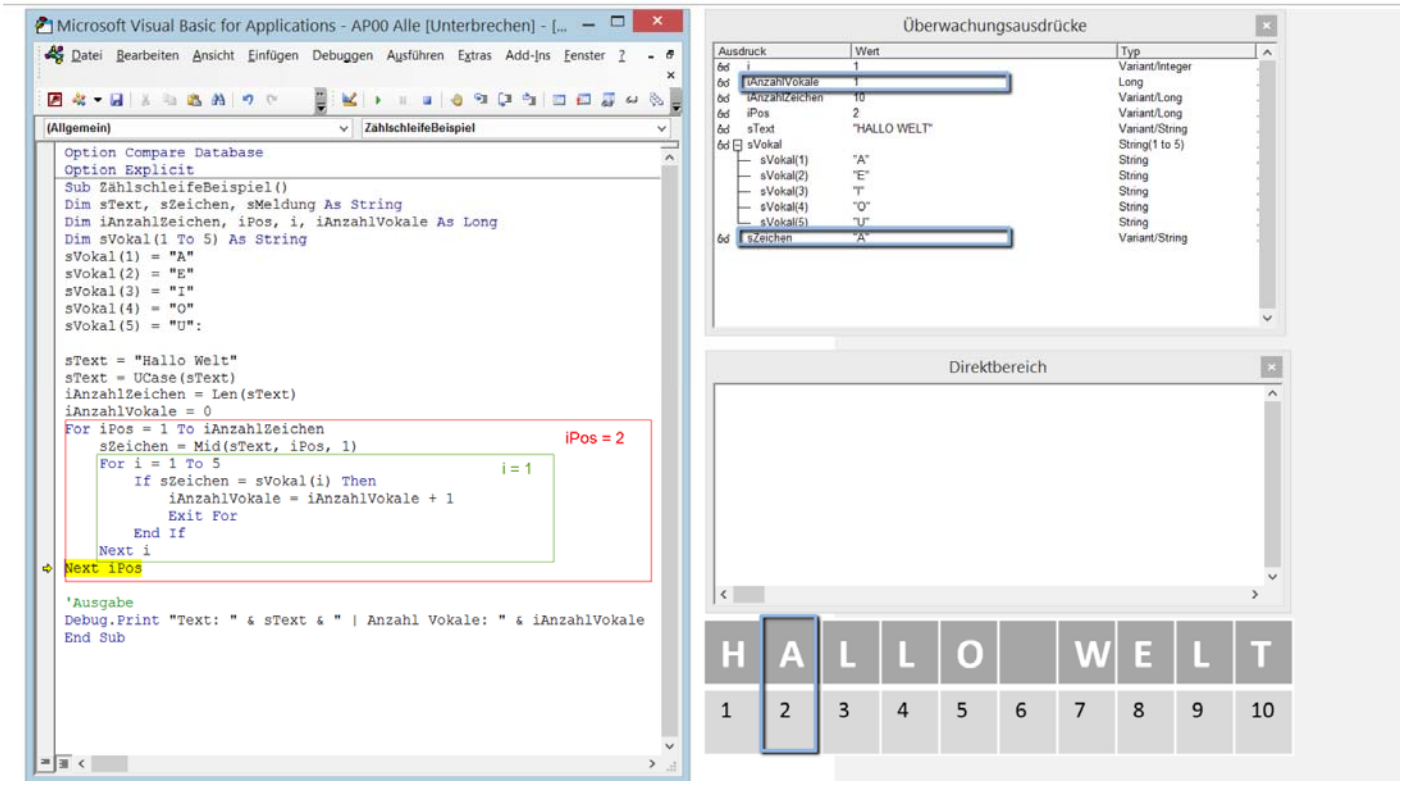
Lektion 10 Zählschleifen - Datenbank AP00.accdb

Zählschleife

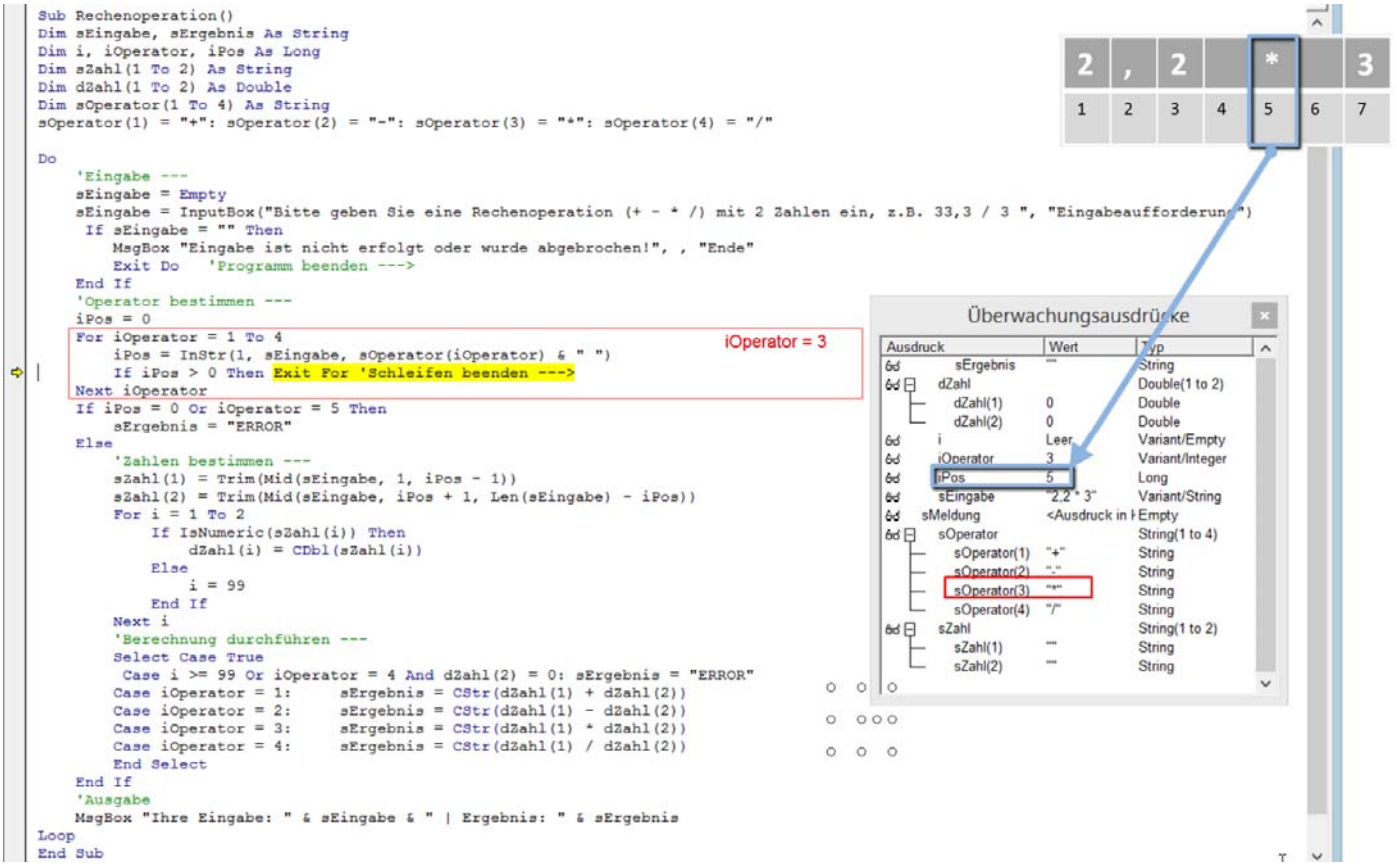

```
' ---------------------------------------------------------------------------------------------------------
' Modul: AP22_Zählschleife
' ---------------------------------------------------------------------------------------------------------
Option Compare Database
Option Explicit

Sub Zählschleife()
'---------------------------------------------------------------------------------------------------------
' Beispiele für Zählschleifen
'---------------------------------------------------------------------------------------------------------
Dim i, iStart, iEnde, iSchritt As Long

'Beispiel 1
iStart = 1: iEnde = 3: iSchritt = 1
For i = iStart To iEnde Step iSchritt
   Debug.Print "Beispiel 1: "; i
Next i

'Beispiel 2
iStart = 3: iEnde = 1: iSchritt = -1
For i = iStart To iEnde Step iSchritt
   Debug.Print "Beispiel 2: "; i
Next i

'Beispiel 3
iStart = 4: iEnde = 3: iSchritt = 1
For i = iStart To iEnde Step iSchritt
   Debug.Print "Beispiel 3: "; i
Next i


'Beispiel 4
For i = 1 To 3 Step 1
   Debug.Print "Beispiel 4: "; i
Next i

'Beispiel 5
For i = 1 To 3
   Debug.Print "Beispiel 5: "; i
Next i

End Sub
```

Direktbereich

```
Beispiel 1:  1
Beispiel 1:  2
Beispiel 1:  3
Beispiel 2:  3
Beispiel 2:  2
Beispiel 2:  1
Beispiel 4:  1
Beispiel 4:  2
Beispiel 4:  3
Beispiel 5:  1
Beispiel 5:  2
Beispiel 5:  3
```




```
' ---------------------------------------------------------------------------------------------------------------
' Modul: AP22_ZählschleifeBeispiel
' ---------------------------------------------------------------------------------------------------------------
Option Compare Database
Option Explicit

Sub ZählschleifeBeispiel()
' ---------------------------------------------------------------------------------------------------------------
' Es werden die Vokale in der Eingabe gezählt
' ---------------------------------------------------------------------------------------------------------------
Dim sText, sZeichen, sMeldung As String
Dim iAnzahlZeichen, iPos, i, iAnzahlVokale As Long
Dim sVokal(1 To 5) As String
sVokal(1) = "A"
sVokal(2) = "E"
sVokal(3) = "I"
sVokal(4) = "O"
sVokal(5) = "U":

sText = "Hallo Welt"
sText = UCase(sText)
iAnzahlZeichen = Len(sText)
iAnzahlVokale = 0

For iPos = 1 To iAnzahlZeichen
   sZeichen = Mid(sText, iPos, 1)
   For i = 1 To 5
      If sZeichen = sVokal(i) Then
         iAnzahlVokale = iAnzahlVokale + 1
         Exit For
      End If
   Next i
Next iPos

'Ausgabe
Debug.Print "Text: " & sText & " | Anzahl Vokale: " & iAnzahlVokale
End Sub
```


Sub Rechenoperation()
Dim sEingabe, sErgebnis As String
Dim i, iOperator, iPos As Long
Dim sZahl(1 To 2) As String
Dim dZahl(1 To 2) As Double
Dim sOperator(1 To 4) As String
sOperator(1) = "+": sOperator(2) = "-": sOperator(3) = "*": sOperator(4) = "/"
Do
'Eingabe ---
sEingabe = Empty
sEingabe = InputBox("Bitte geben Sie eine Rechenoperation (+ - * /) mit 2 Zahlen ein, z.B. 33,3 / 3 ", "Eingabeaufforderung")
If sEingabe = "" Then
MsgBox "Eingabe ist nicht erfolgt oder wurde abgebrochen!", , "Ende"
Exit Do 'Programm beenden --->
End If
'Operator bestimmen ---
iPos = 0
For iOperator = 1 To 4
iPos = InStr(1, sEingabe, sOperator(iOperator) & " ")
If iPos > 0 Then Exit For 'Schleifen beenden --->
Next iOperator
If iPos = 0 Or iOperator = 5 Then
sErgebnis = "ERROR"
Else
'Zahlen bestimmen ---
sZahl(1) = Trim(Mid(sEingabe, 1, iPos - 1))
sZahl(2) = Trim(Mid(sEingabe, iPos + 1, Len(sEingabe) - iPos))
For i = 1 To 2
If IsNumeric(sZahl(i)) Then
dZahl(i) = CDbl(sZahl(i))
Else
i = 99
End If
Next i
'Berechnung durchführen ---
Select Case True
Case i >= 99 Or iOperator = 4 And dZahl(2) = 0: sErgebnis = "ERROR"
Case iOperator = 1: sErgebnis = CStr(dZahl(1) + dZahl(2))
Case iOperator = 2: sErgebnis = CStr(dZahl(1) - dZahl(2))
Case iOperator = 3: sErgebnis = CStr(dZahl(1) * dZahl(2))
Case iOperator = 4: sErgebnis = CStr(dZahl(1) / dZahl(2))
End Select
End If
'Ausgabe
MsgBox "Ihre Eingabe: " & sEingabe & " | Ergebnis: " & sErgebnis
Loop
End Sub
iOperator = 3
2 , 2 * 3
1 2 3 4 5 6 7
Überwachungsausdrücke
Ausdruck Wert Typ
sErgebnis "" String
dZahl Double(1 to 2)
dZahl(1) 0 Double
dZahl(2) 0 Double
i Leer Variant/Empty
iOperator 3 Variant/Integer
iPos 5 Long
sEingabe "2,2 * 3" Variant/String
sMeldung <Ausdruck in F Empty
sOperator String(1 to 4)
sOperator(1) "+" String
sOperator(2) "-" String
sOperator(3) "*" String
sOperator(4) "/" String
sZahl String(1 to 2)
sZahl(1) "" String
sZahl(2) "" String

```
'---------------------------------------------------------------------------------------------------------------------------
' Modul: AP24_Rechenoperation
'---------------------------------------------------------------------------------------------------------------------------
Option Compare Database
Option Explicit

Sub Rechenoperation()
'---------------------------------------------------------------------------------------------------------------------------
' Rechenoperation mit zwei Zahlen
'---------------------------------------------------------------------------------------------------------------------------
Dim sEingabe, sErgebnis As String
Dim i, iOperator, iPos As Long
Dim sZahl(1 To 2) As String
Dim dZahl(1 To 2) As Double
Dim sOperator(1 To 4) As String
sOperator(1) = "+": sOperator(2) = "-": sOperator(3) = "*": sOperator(4) = "/"

Do
   'Eingabe ---
   sEingabe = Empty
   sEingabe = InputBox("Bitte geben Sie eine Rechenoperation (+ - * /) mit 2 Zahlen ein, z.B. 33,3 / 3 ", "Eingabeaufforderung")
    If sEingabe = "" Then
      MsgBox "Eingabe ist nicht erfolgt oder wurde abgebrochen!", , "Ende"
      Exit Do   'Programm beenden --->
   End If
   'Operator bestimmen ---
   iPos = 0
   For iOperator = 1 To 4
      iPos = InStr(1, sEingabe, sOperator(iOperator) & " ")
      If iPos > 0 Then Exit For 'Schleifen beenden --->
   Next iOperator
   If iPos = 0 Or iOperator = 5 Then
      sErgebnis = "ERROR"
   Else
      'Zahlen bestimmen ---
      sZahl(1) = Trim(Mid(sEingabe, 1, iPos - 1))
      sZahl(2) = Trim(Mid(sEingabe, iPos + 1, Len(sEingabe) - iPos))
      For i = 1 To 2
         If IsNumeric(sZahl(i)) Then
            dZahl(i) = CDbl(sZahl(i))
         Else
            i = 99
         End If
      Next i
      'Berechnung durchführen ---
      Select Case True
      Case i >= 99 Or iOperator = 4 And dZahl(2) = 0: sErgebnis = "ERROR"
      Case iOperator = 1:    sErgebnis = CStr(dZahl(1) + dZahl(2))
      Case iOperator = 2:    sErgebnis = CStr(dZahl(1) - dZahl(2))
      Case iOperator = 3:    sErgebnis = CStr(dZahl(1) * dZahl(2))
      Case iOperator = 4:    sErgebnis = CStr(dZahl(1) / dZahl(2))
      End Select
   End If
   'Ausgabe
   MsgBox "Ihre Eingabe: " & sEingabe & " | Ergebnis: " & sErgebnis
Loop
End Sub
```

Lektion 11 Prozeduren - Datenbank AP00.accdb

Sub: Diese Prozedur gibt keinen Wert zurück

sText → AusgabeMeldung

Name: AusgabeMeldung
Übergabe- Parameter:
sText

```
Sub AusgabeMeldung(sText As String)
MsgBox sText, , "Meldung"
End Sub
```

Function: Diese Prozedur gibt einen Wert zurück

dZahl1
dZahl2 → dAdd → dAdd

Name: dAdd, d gibt Datentyp an.
Übergabe- Parameter:
dZahl1 dZahl2

```
Function dAdd(dZahl1 As Double, dZahl2 As Double) As Double
Dim d3 As Double
d3 = dZahl1 + dZahl2
dAdd = d3
End Function
```

Sub und Function

```
Sub Main01()
'Hauptprozedur
Dim d1 As Double
Dim d2 As Double
Dim d3 As Double

AusgabeMeldung "Es wird eine Berechnung durchgeführt"
Call AusgabeMeldung("Es wird eine Berechnung durchgeführt")'alternativ
d1 = 1
d2 = 1
d3 = dAdd(d1, d2)
MsgBox "Ergebnis: " & CStr(d3) & " * " & CStr(d2) & " = " & CStr(d3)
End Sub
```

Name: Main01
Hauptprozedur

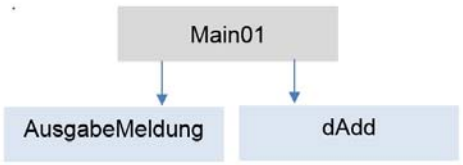
Hier erfolgt der Aufruf der Prozeduren

Der Name der Übergabeparameter
kann, aber muss nicht übereinstimmen.

```
Sub AusgabeMeldung(sText As String)
MsgBox sText, , "Meldung"
End Sub
```

```
Function dAdd(d1 As Double, dZahl2 As Double) As Double
Dim d3 As Double
d3 = dZahl1 + dZahl2
dAdd = d3
End Function
```

Aufrufhierarchie



Aufruf und Aufrufhierarchie

```
' ------------------------------------------------------------------------------------------------------------------------
' Modul: AP25_Main01
' ------------------------------------------------------------------------------------------------------------------------
Option Compare Database
Option Explicit

Sub Main01()
'------------------------------------------------------------------------------------------------------------------------
' Beispiel für Unterprozeduren
'------------------------------------------------------------------------------------------------------------------------
Dim d1 As Double
Dim d2 As Double
Dim d3 As Double

AusgabeMeldung "Es wird eine Berechnung durchgeführt"
Call AusgabeMeldung("Es wird eine Berechnung durchgeführt")
d1 = 1
d2 = 1
d3 = dAdd(d1, d2)
MsgBox "Ergebnis: " & CStr(d3) & " * " & CStr(d2) & " = " & CStr(d3)
End Sub

Sub AusgabeMeldung(sText As String)
    MsgBox sText, , "Meldung"
End Sub

Function dAdd(d1 As Double, dZahl2 As Double) As Double
'Unterprozedur(Stufe 2)
Dim d3 As Double
d3 = d1 + dZahl2
dAdd = d3
End Function
```

```
Sub Rechenoperation()
Dim sEingabe, sErgebnis As String
Dim i, iOperator, iPos As Long
Dim sZahl(1 To 2) As String
Dim dZahl(1 To 2) As Double
Dim sOperator(1 To 4) As String
sOperator(1) = "+": sOperator(2) = "-": sOperator(3) = "*": sOperator(4) = "/"

Do
    'Eingabe ---
    sEingabe = Empty
    sEingabe = InputBox("Bitte geben Sie eine Rechenoperation (+ - * /) mit 2 Zahlen
     If sEingabe = "" Then
        MsgBox "Eingabe ist nicht erfolgt oder wurde abgebrochen!", , "Ende"
        Exit Do   'Programm beenden --->
    End If
    'Operator bestimmen ---
    iPos = 0
    For iOperator = 1 To 4
        iPos = InStr(1, sEingabe, sOperator(iOperator) & " ")
        If iPos > 0 Then Exit For 'Schleifen beenden --->
    Next iOperator
    If iPos = 0 Or iOperator = 5 Then
        sErgebnis = "ERROR"
    Else
        'Zahlen bestimmen ---
        sZahl(1) = Trim(Mid(sEingabe, 1, iPos - 1))
        sZahl(2) = Trim(Mid(sEingabe, iPos + 1, Len(sEingabe) - iPos))
        For i = 1 To 2
            If IsNumeric(sZahl(i)) Then
                dZahl(i) = CDbl(sZahl(i))
            Else
                i = 99
            End If
        Next i
        'Berechnung durchführen ---
        Select Case True
         Case i >= 99 Or iOperator = 4 And dZahl(2) = 0: sErgebnis = "ERROR"
        Case iOperator = 1:     sErgebnis = CStr(dZahl(1) + dZahl(2))
        Case iOperator = 2:     sErgebnis = CStr(dZahl(1) - dZahl(2))
        Case iOperator = 3:     sErgebnis = CStr(dZahl(1) * dZahl(2))
        Case iOperator = 4:     sErgebnis = CStr(dZahl(1) / dZahl(2))
        End Select
    End If
    'Ausgabe
    MsgBox "Ihre Eingabe: " & sEingabe & " | Ergebnis: " & sErgebnis
Loop
End Sub
```

```
Sub Rechenoperation2()
Dim sEingabe As String
Dim sMeldung As String
Do
    'Eingabe ---
    sEingabe = Empty
    sEingabe = InputBox("Bitte geben Sie eine Rechenoperation (+ - * /)" & "mit 2 Za
    If sEingabe = "" Then
        MsgBox "Eingabe ist nicht erfolgt oder wurde abgebrochen!", , "Ende"
        Exit Do   'Programm beenden --->
    End If
    sMeldung = sRechenoperation(sEingabe)
    'Ausgabe
    MsgBox "Ihre Eingabe: " & sEingabe & " | Ergebnis: " & sMeldung
Loop
End Sub

Function sRechenoperation(sEingabe As String) As String
Dim sErgebnis As String
Dim i, iOperator, iPos As Long
Dim sZahl(1 To 2) As String
Dim dZahl(1 To 2) As Double
Dim sOperator(1 To 4) As String
sOperator(1) = "+": sOperator(2) = "-": sOperator(3) = "*": sOperator(4) = "/"
'Operator bestimmen ---
iPos = 0
For iOperator = 1 To 4
    iPos = InStr(1, sEingabe, sOperator(iOperator) & " ")
    If iPos > 0 Then Exit For 'Schleifen beenden --->
Next iOperator
If iPos = 0 Or iOperator = 5 Then
    sErgebnis = "ERROR"
Else
    'Zahlen bestimmen ---
    sZahl(1) = Trim(Mid(sEingabe, 1, iPos - 1))
    sZahl(2) = Trim(Mid(sEingabe, iPos + 1, Len(sEingabe) - iPos))
    For i = 1 To 2
        If IsNumeric(sZahl(i)) Then
            dZahl(i) = CDbl(sZahl(i))
        Else
            i = 99
        End If
    Next i
    'Berechnung durchführen ---
    Select Case True
    Case i >= 99 Or iOperator = 4 And dZahl(2) = 0: sErgebnis = "ERROR"
    Case iOperator = 1:     sErgebnis = CStr(dZahl(1) + dZahl(2))
    Case iOperator = 2:     sErgebnis = CStr(dZahl(1) - dZahl(2))
    Case iOperator = 3:     sErgebnis = CStr(dZahl(1) * dZahl(2))
    Case iOperator = 4:     sErgebnis = CStr(dZahl(1) / dZahl(2))
    End Select
End If
sRechenoperation = sErgebnis
End Function
```

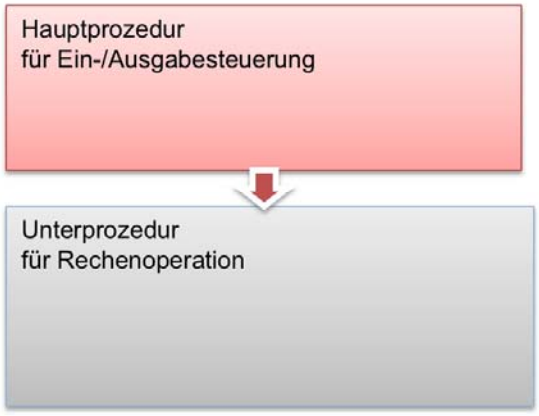
Neustrukturierung des Programmes in Haupt- und Unterprozeduren

```
' ------------------------------------------------------------------------------------------------------------------------
' Modul: AP27_Rechenoperation2
' ------------------------------------------------------------------------------------------------------------------------
Option Compare Database
Option Explicit

Sub Rechenoperation2()
'------------------------------------------------------------------------------------------------------------------------
' Rechenoperation mit zwei Zahlen - Verwendung einer Funktion
'------------------------------------------------------------------------------------------------------------------------
Dim sEingabe As String
Dim sMeldung As String
Do
   'Eingabe ---
   sEingabe = Empty
   sEingabe = InputBox("Bitte geben Sie eine Rechenoperation (+ - * /)" & "mit 2 Zahlen ein, z.B. 33,3 / 3", "Eingabeaufforderung")
   If sEingabe = "" Then
      MsgBox "Eingabe ist nicht erfolgt oder wurde abgebrochen!", , "Ende"
      Exit Do   'Programm beenden --->
   End If
   sMeldung = sRechenoperation(sEingabe)
   'Ausgabe
   MsgBox "Ihre Eingabe: " & sEingabe & " | Ergebnis: " & sMeldung
Loop
End Sub
```

```
Function sRechenoperation(sEingabe As String) As String
Dim sErgebnis As String
Dim i, iOperator, iPos As Long
Dim sZahl(1 To 2) As String
Dim dZahl(1 To 2) As Double
Dim sOperator(1 To 4) As String
sOperator(1) = "+": sOperator(2) = "-": sOperator(3) = "*": sOperator(4) = "/"
'Operator bestimmen ---
iPos = 0
For iOperator = 1 To 4
   iPos = InStr(1, sEingabe, sOperator(iOperator) & " ")
   If iPos > 0 Then Exit For 'Schleifen beenden --->
Next iOperator
If iPos = 0 Or iOperator = 5 Then
   sErgebnis = "ERROR"
Else
   'Zahlen bestimmen ---
   sZahl(1) = Trim(Mid(sEingabe, 1, iPos - 1))
   sZahl(2) = Trim(Mid(sEingabe, iPos + 1, Len(sEingabe) - iPos))
   For i = 1 To 2
      If IsNumeric(sZahl(i)) Then
         dZahl(i) = CDbl(sZahl(i))
      Else
         i = 99
      End If
   Next i
   'Berechnung durchführen ---
   Select Case True
   Case i >= 99 Or iOperator = 4 And dZahl(2) = 0: sErgebnis = "ERROR"
   Case iOperator = 1:    sErgebnis = CStr(dZahl(1) + dZahl(2))
   Case iOperator = 2:    sErgebnis = CStr(dZahl(1) - dZahl(2))
   Case iOperator = 3:    sErgebnis = CStr(dZahl(1) * dZahl(2))
   Case iOperator = 4:    sErgebnis = CStr(dZahl(1) / dZahl(2))
   End Select
End If
sRechenoperation = sErgebnis
End Function
```

Vorteil der Modularisierung:

1. Die Komplexität eines Programmes wird reduziert. Dadurch steigt die Transparenz.
2. Prozeduren können getrennt entwickelt und getestet werden.
3. Fehler können einfacher gefunden werden (Wartbarkeit).
4. Einzelne Prozeduren können ausgetauscht werden.



Lektion 12 Gültigkeit von Variablen - Datenbank AP00.accdb

```
' ---------------------------------------------------------------------------------------------------------------------
' Modul: AP28_BeispielLokaleVariable
' ---------------------------------------------------------------------------------------------------------------------
Option Compare Database
Option Explicit

Sub BeispielLokaleVariable()      'Main-Prozedur
'---------------------------------------------------------------------------------------------------------------------
' Gültigkeitbereich für lokale Variable
'---------------------------------------------------------------------------------------------------------------------
Dim d1 As Double                  'Gültigkeitsbereich: lokal innerhalb der Prozedur
Dim d2 As Double
Dim d3 As Double
d1 = 1
d2 = 1
d3 = 1
dAdd d1, d2
Debug.Print "Beispiel dAdd: d3=" & d3
End Sub

Sub dAdd(d1 As Double, d2 As Double)
Dim d3 As Double                  'Gültigkeitsbereich: lokal innerhalb des Prozedur
d3 = d1 + d2
End Sub
```

```
' ---------------------------------------------------------------------------------------------------------------------
' Modul: AP29_BeispielByRefVal
' ---------------------------------------------------------------------------------------------------------------------
Option Compare Database
Option Explicit

Sub BeispielByRefVal()  'Main-Prozedur
'---------------------------------------------------------------------------------------------------------------------
' Beispiele für Call by Referenz | Call by Value
'---------------------------------------------------------------------------------------------------------------------
Dim d1 As Double
Dim d2 As Double
d1 = 1
d2 = 1
Debug.Print "Anfangswerte d1,d2:  " & d1; " | "; d2
Call F01(d1, d2)
Debug.Print "ByRef d1, ByRef d2:  " & d1; " | "; d2

d1 = 1
d2 = 1
Call F02(d1, d2)
Debug.Print "ByVal d1, ByVal d2:  " & d1; " | "; d2

d1 = 1
d2 = 1
Call F03(d1, d2)
Debug.Print "ByVal d1, ByRef d2:  " & d1; " | "; d2
End Sub

Sub F01(d1 As Double, d2 As Double)
d1 = d1 + 1
d2 = d2 + 1
End Sub

Sub F02(ByVal d1 As Double, ByVal d2 As Double)
d1 = d1 + 1
d2 = d2 + 1
End Sub

Sub F03(ByVal d1 As Double, ByRef d2 As Double)
d1 = d1 + 1
d2 = d2 + 1
End Sub
```

Direktbereich

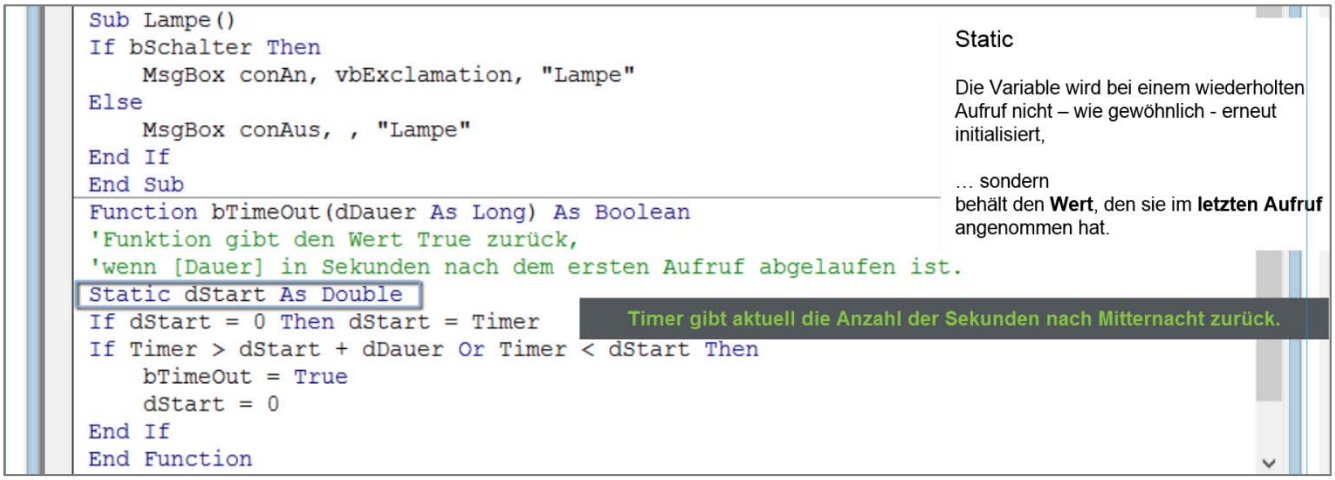
```
Anfangswerte d1,d2:  1 |  1
ByRef d1, ByRef d2:  2 |  2
ByVal d1, ByVal d2:  1 |  1
ByVal d1, ByRef d2:  1 |  2
```

```
' ---------------------------------------------------------------------------------------------------------------------------
' Modul: AP29_Schaltersteuerung
' ---------------------------------------------------------------------------------------------------------------------------
Option Compare Database
Option Explicit
'Modulkopf
Dim bSchalter As Boolean
Const conAn = ">>>>>>>>>> AN <<<<<<<<<<"
Const conAus = "__________________________________________________"
'Ende Modulkopf

Sub Schaltersteuerung()    'Main-Prozedur
' ---------------------------------------------------------------------------------------------------------------------------
' Schaltersteuerung mit Variablen auf Modulebene
' ---------------------------------------------------------------------------------------------------------------------------
Do
   If bSchalter Then
      Schalter_Aus
   Else
      Schalter_An
   End If

   Lampe
Loop
End Sub

Sub Schalter_An()
bSchalter = True
End Sub

Sub Schalter_Aus()
bSchalter = False
End Sub

Sub Lampe()
If bSchalter Then
   MsgBox conAn, vbExclamation, "Lampe"
Else
   MsgBox conAus, , "Lampe"
End If
End Sub
```

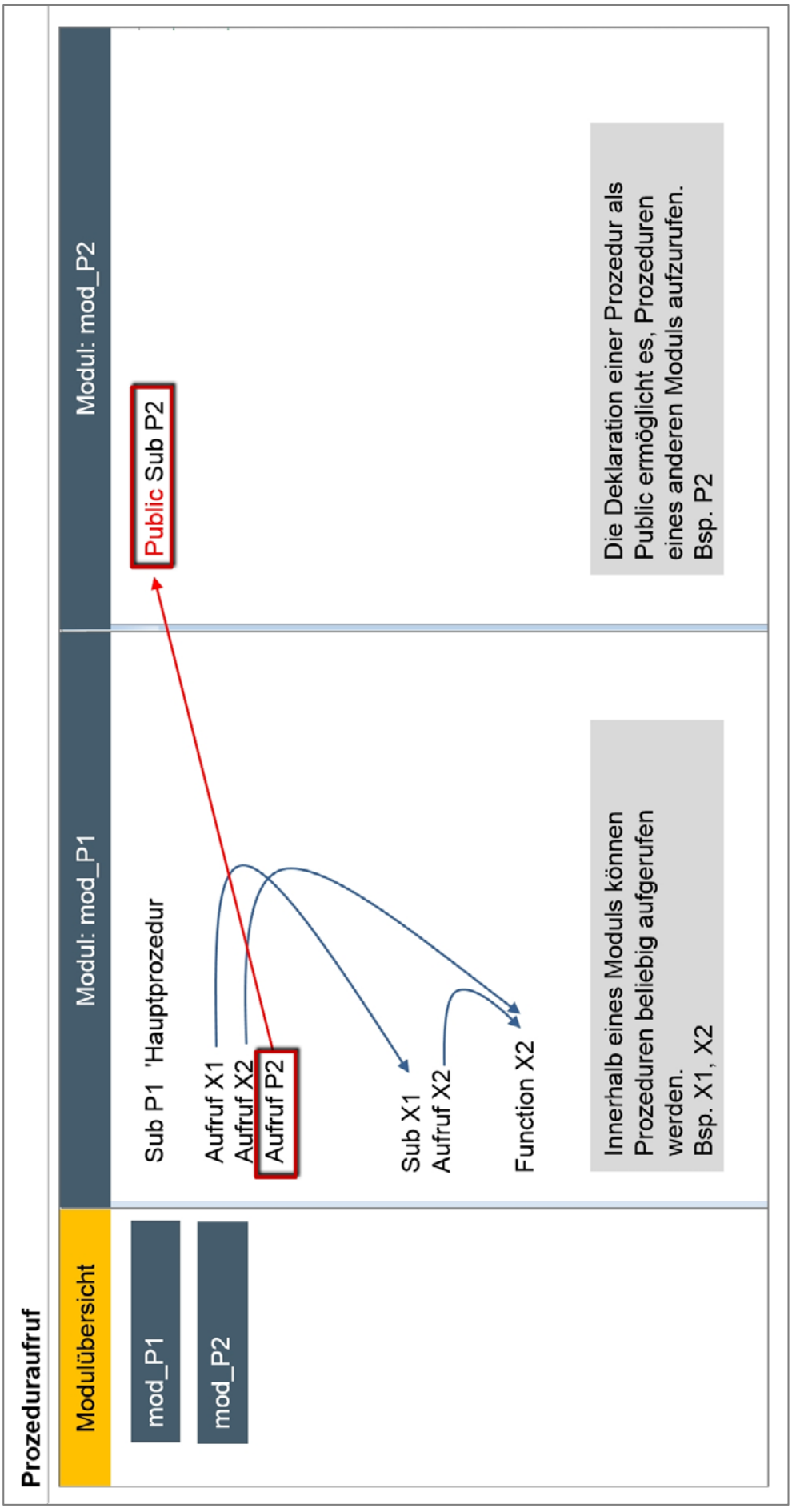
```
' ---------------------------------------------------------------------------------------------------------------------------
' Modul: AP30_Schaltersteuerung2
' ---------------------------------------------------------------------------------------------------------------------------
Option Compare Database
Option Explicit

Sub Schaltersteuerung2()    'Main-Prozedur
' ---------------------------------------------------------------------------------------------------------------------------
' Schaltersteuerung mit Variablenübergabe / Call by Referenz
' ---------------------------------------------------------------------------------------------------------------------------
Dim bSchalter As Boolean

Do While True
   If bSchalter Then
      Schalter_Aus bSchalter
   Else
      Schalter_An bSchalter
   End If
   Lampe bSchalter
Loop
End Sub

Sub Schalter_An(bSchalter As Boolean)
bSchalter = True
End Sub

Sub Schalter_Aus(bSchalter As Boolean)
bSchalter = False
End Sub

Sub Lampe(bSchalter As Boolean)
If bSchalter Then
   MsgBox ">>>>>>>>>> AN <<<<<<<<<<", vbExclamation, "Lampe"
Else
   MsgBox "__________________________________________________", , "Lampe"
End If
End Sub
```

```
' ---------------------------------------------------------------------------------------------------------------------
' Modul: AP31_Schaltersteuerung3
' ---------------------------------------------------------------------------------------------------------------------
Option Compare Database
Option Explicit

'Modulkopf
Dim bSchalter As Boolean
Const conAn = ">>>>>>>>>> AN <<<<<<<<<<"
Const conAus = "__________________________________________________"
'Ende Modulkopf

Sub Schaltersteuerung3()    'Main-Prozedur
' ---------------------------------------------------------------------------------------------------------------------
' Schaltersteuerung mit Timeout
' ---------------------------------------------------------------------------------------------------------------------
bSchalter = False
Do Until bTimeOut(15)
   If bSchalter Then
      Schalter_Aus
   Else
      Schalter_An
   End If
   Lampe
Loop
MsgBox "Ende"
End Sub

Sub Schalter_An()
bSchalter = True
End Sub

Sub Schalter_Aus()
bSchalter = False
End Sub

Sub Lampe()
If bSchalter Then
   MsgBox conAn, vbExclamation, "Lampe"
Else
   MsgBox conAus, , "Lampe"
End If
End Sub
```

```
Function bTimeOut(dDauer As Long) As Boolean
'Funktion gibt den Wert True zurück,
'wenn [Dauer] in Sekunden nach dem ersten Aufruf abgelaufen ist.
Static dStart As Double
If dStart = 0 Then dStart = Timer   ' 'Timer gibt aktuell die Anzahl der Sekunden nach Mitternacht zurück.
If Timer > dStart + dDauer Or Timer < dStart Then
   bTimeOut = True
   dStart = 0
End If
End Function
```

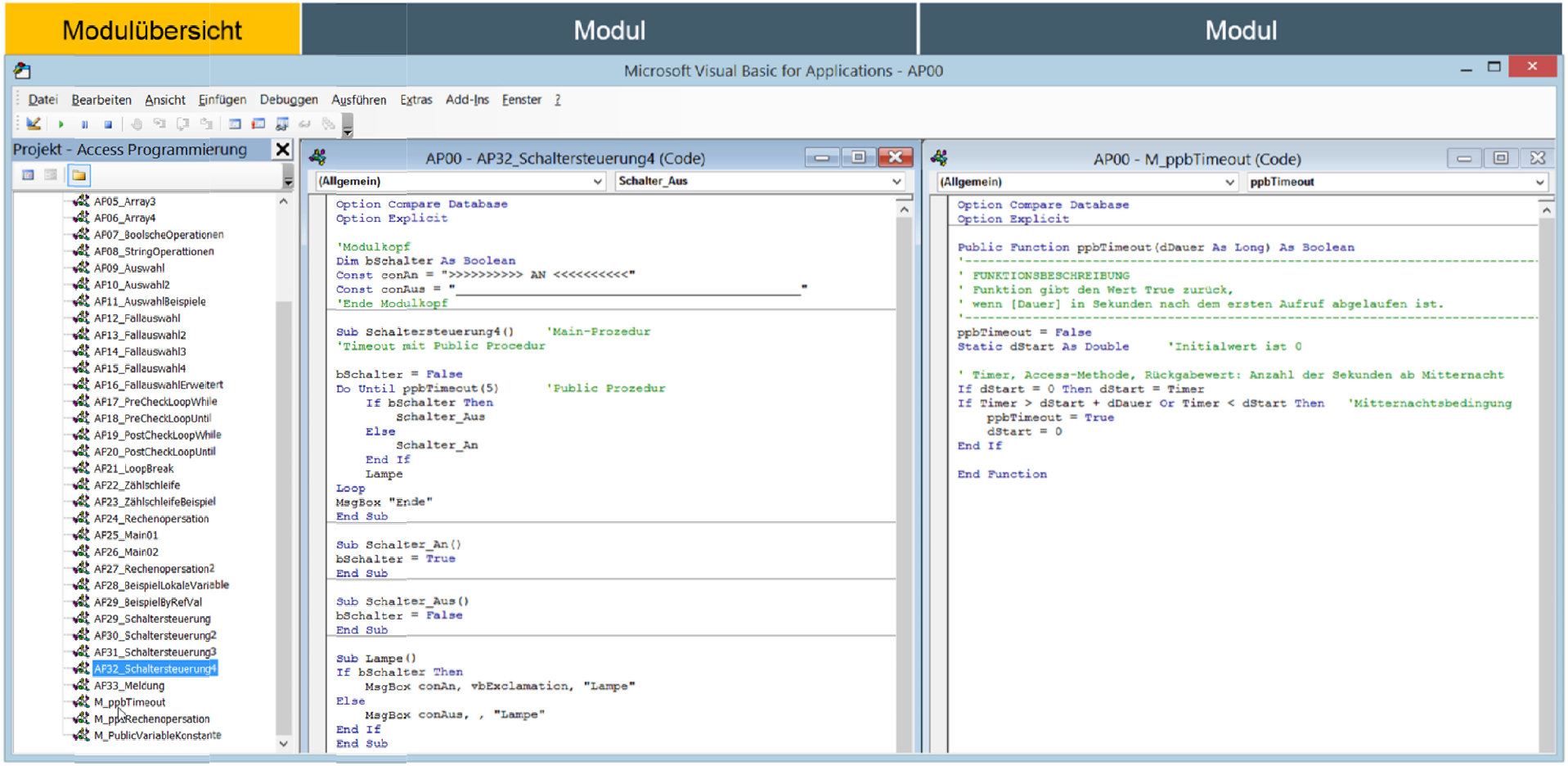


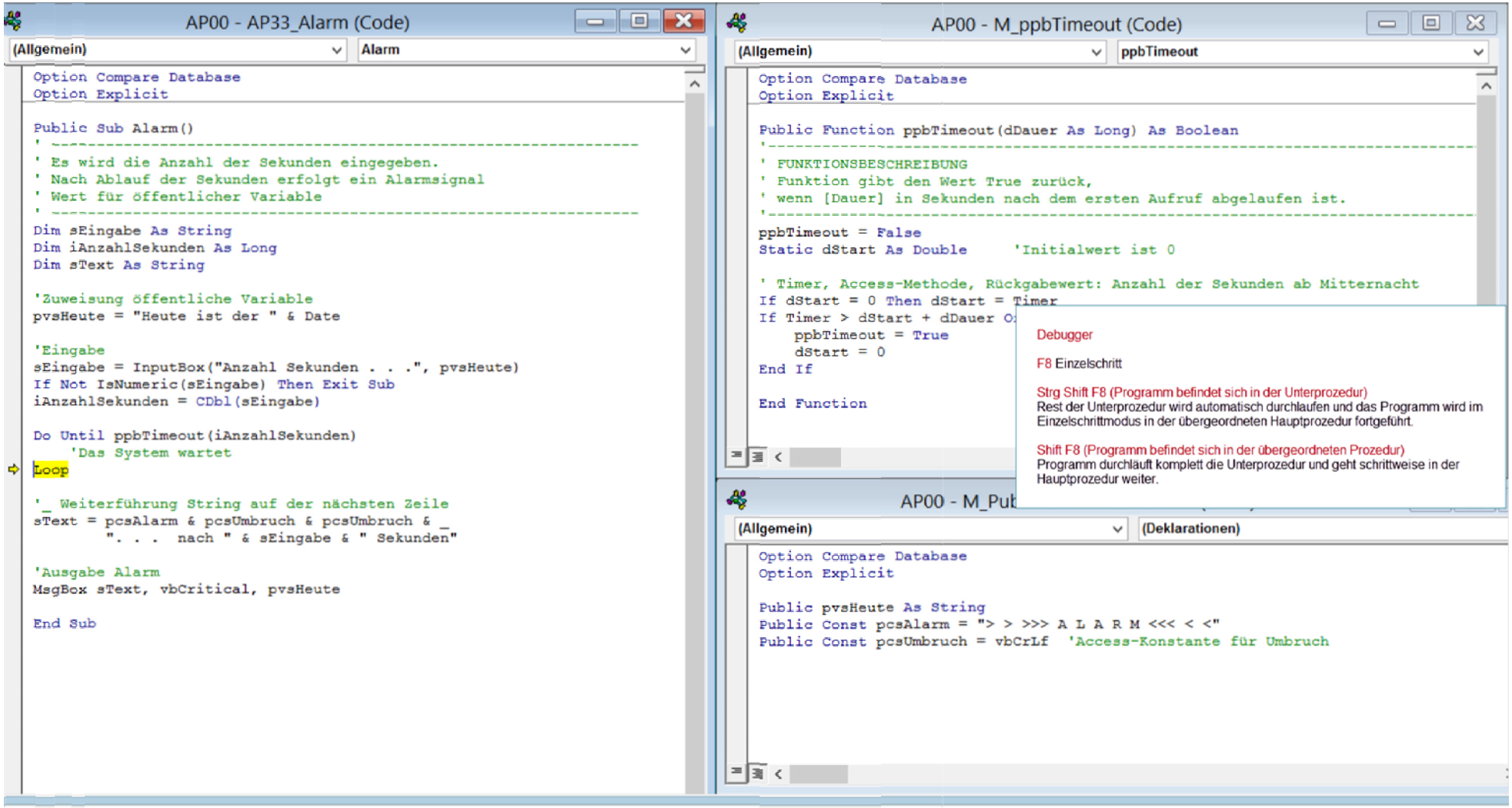
Lektion 13 Öffentliche Prozeduren - Datenbank AP00.accdb



Beispiel Schaltersteuerung

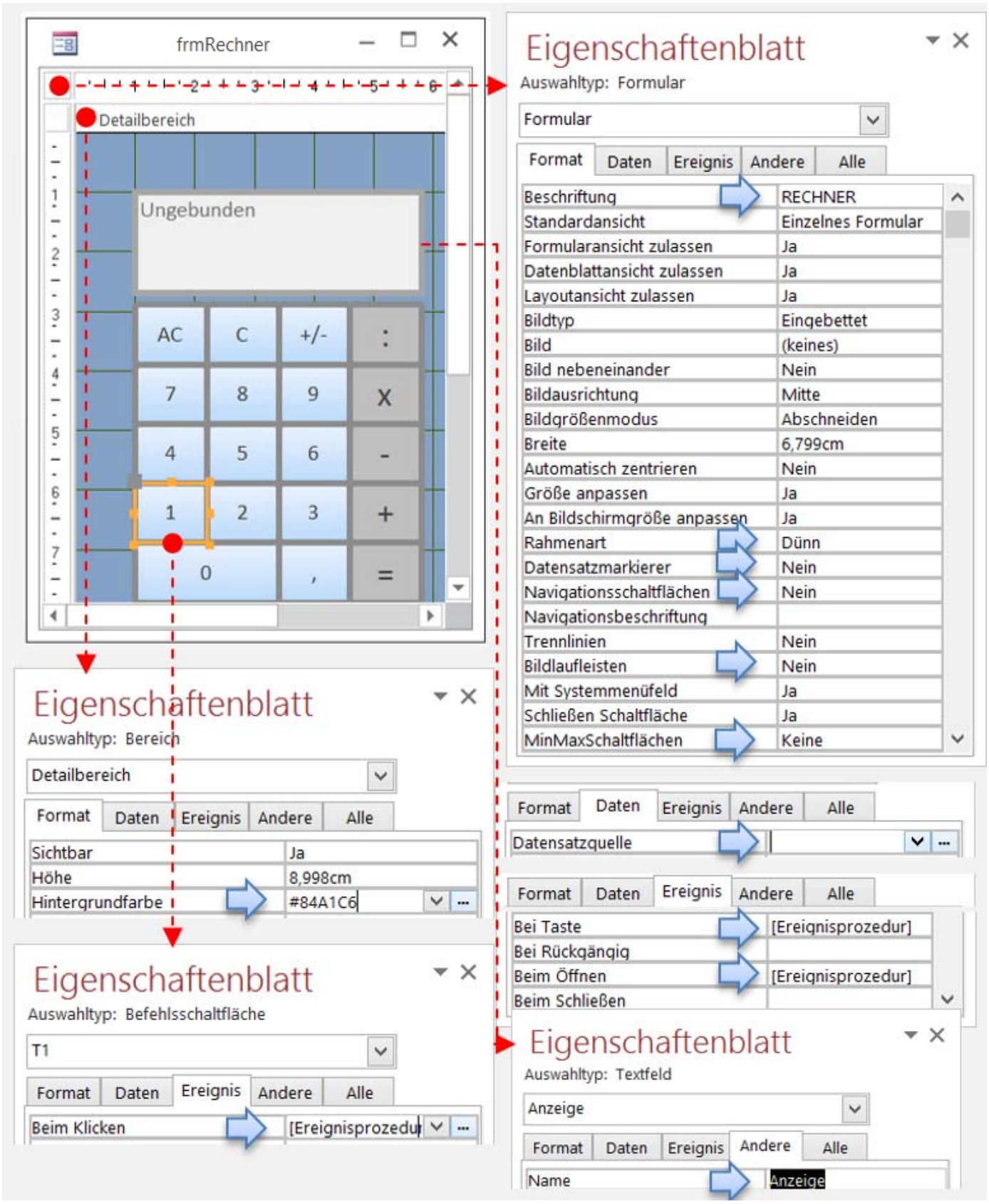
Modulübersicht | Modul | Modul




AP00 - AP33_Alarm (Code)
(Allgemein)
Alarm
Option Compare Database
Option Explicit
Public Sub Alarm()
' Es wird die Anzahl der Sekunden eingegeben.
' Nach Ablauf der Sekunden erfolgt ein Alarmsignal
' Wert für öffentlicher Variable
Dim sEingabe As String
Dim iAnzahlSekunden As Long
Dim sText As String
'Zuweisung öffentliche Variable
pvsHeute = "Heute ist der " & Date
'Eingabe
sEingabe = InputBox("Anzahl Sekunden . . .", pvsHeute)
If Not IsNumeric(sEingabe) Then Exit Sub
iAnzahlSekunden = CDbl(sEingabe)
Do Until ppbTimeout(iAnzahlSekunden)
'Das System wartet
Loop
'_ Weiterführung String auf der nächsten Zeile
sText = pcsAlarm & pcsUmbruch & pcsUmbruch & _
". . . nach " & sEingabe & " Sekunden"
'Ausgabe Alarm
MsgBox sText, vbCritical, pvsHeute
End Sub
AP00 - M_ppbTimeout (Code)
(Allgemein)
ppbTimeout
Option Compare Database
Option Explicit
Public Function ppbTimeout(dDauer As Long) As Boolean
' FUNKTIONSBESCHREIBUNG
' Funktion gibt den Wert True zurück,
' wenn [Dauer] in Sekunden nach dem ersten Aufruf abgelaufen ist.
ppbTimeout = False
Static dStart As Double 'Initialwert ist 0
' Timer, Access-Methode, Rückgabewert: Anzahl der Sekunden ab Mitternacht
If dStart = 0 Then dStart = Timer
If Timer > dStart + dDauer O
ppbTimeout = True
dStart = 0
End If
End Function
Debugger
F8 Einzelschritt
Strg Shift F8 (Programm befindet sich in der Unterprozedur)
Rest der Unterprozedur wird automatisch durchlaufen und das Programm wird im Einzelschrittmodus in der übergeordneten Hauptprozedur fortgeführt.
Shift F8 (Programm befindet sich in der übergeordneten Prozedur)
Programm durchläuft komplett die Unterprozedur und geht schrittweise in der Hauptprozedur weiter.
AP00 - M_Pub
(Allgemein)
(Deklarationen)
Option Compare Database
Option Explicit
Public pvsHeute As String
Public Const pcsAlarm = "> > >>> A L A R M <<< < <"
Public Const pcsUmbruch = vbCrLf 'Access-Konstante für Umbruch

```
' ---------------------------------------------------------------------------------------------------------------------------------------------------------------
' Modul: AP33_Alarm
' ---------------------------------------------------------------------------------------------------------------------------------------------------------------
Option Compare Database
Option Explicit

Public Sub Alarm()
' ---------------------------------------------------------------------------------------------------------------------------------------------------------------
' Alarmgeber
' ---------------------------------------------------------------------------------------------------------------------------------------------------------------
' Es wird die Anzahl der Sekunden eingegeben.
' Nach Ablauf der Sekunden erfolgt ein Alarmsignal
' Wert für öffentlicher Variable
' ---------------------------------------------------------------------------------------------------------------------------------------------------------------

Dim sEingabe As String
Dim iAnzahlSekunden As Long
Dim sText As String

'Zuweisung öffentliche Variable
pvsHeute = "Heute ist der " & Date

'Eingabe
sEingabe = InputBox("Anzahl Sekunden . . .", pvsHeute)
If Not IsNumeric(sEingabe) Then Exit Sub
iAnzahlSekunden = CDbl(sEingabe)

Do Until ppbTimeout(iAnzahlSekunden)
   'Das System wartet
Loop

sText = pcsAlarm & pcsUmbruch & pcsUmbruch & ". . .  nach " & sEingabe & " Sekunden"

'Ausgabe Alarm
MsgBox sText, vbCritical, pvsHeute
End Sub

' ---------------------------------------------------------------------------------------------------------------------------------------------------------------
' Modul: M_ppbTimeout
' ---------------------------------------------------------------------------------------------------------------------------------------------------------------
Option Compare Database
Option Explicit

Public Function ppbTimeout(dDauer As Long) As Boolean
' ---------------------------------------------------------------------------------------------------------------------------------------------------------------
' Timeout
' Funktion gibt den Wert True zurück,
' wenn [Dauer] in Sekunden nach dem ersten Aufruf abgelaufen ist.
'---------------------------------------------------------------------------------------------------------------------------------------------------------------
Static dStart As Double     'Initialwert ist 0

ppbTimeout = False

If dStart = 0 Then dStart = Timer                  ' Timer, Access-Methode, Rückgabewert: Anzahl der Sekunden ab Mitternacht
If Timer > dStart + dDauer Or Timer < dStart Then   'Mitternachtsbedingung
   ppbTimeout = True
   dStart = 0
End If
End Function

' ---------------------------------------------------------------------------------------------------------------------------------------------------------------
' Modul: M_PublicVariableKonstanten
' ---------------------------------------------------------------------------------------------------------------------------------------------------------------
Option Compare Database
Option Explicit
' ---------------------------------------------------------------------------------------------------------------------------------------------------------------
' Öffentliche Variablen und Konstanten
' ---------------------------------------------------------------------------------------------------------------------------------------------------------------
Public pvsHeute As String                          'Heute ist [aktuelles Datum]
Public Const pcsAlarm = "> > >>> A L A R M <<< < <"     'Textkonstante
Public Const pcsUmbruch = vbCrLf                   'Access-Konstante für Umbruch
```

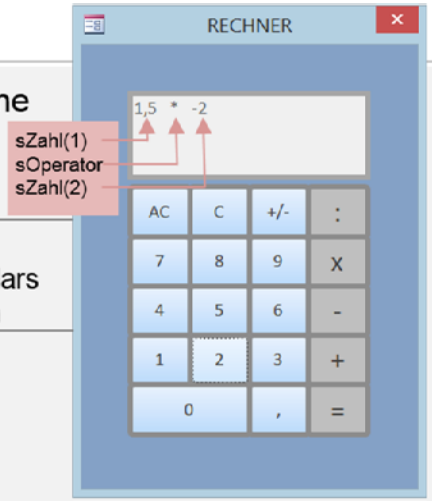
Lektion 14 Prozeduren im Formular (Teil 1) - Datenbank AP00.accdb



Formularmodul

```
'Variable auf Modulebene
Dim sZahl(1 To 2) As String
Dim sOperator As String                    ' + - * :
Dim iZahl As Long                          'Aktive Zahl: 1 oder 2
```

Variable auf Modulebene




```
Private Sub Form_Open(Cancel As Integer)
Private Sub Form_KeyPress(KeyAscii As Integer)   'Tastatureingabe
Private Sub T1_Click()                           'TASTE 1
Private Sub T2_Click()                           'TASTE 2
Private Sub T3_Click()                           'TASTE 3
Private Sub T4_Click()                           'TASTE 4
Private Sub T5_Click()                           'TASTE 5
Private Sub T6_Click()                           'TASTE 6
Private Sub T7_Click()                           'TASTE 7
Private Sub T8_Click()                           'TASTE 8
Private Sub T9_Click()                           'TASTE 9
Private Sub T0_Click()                           'TASTE 0
Private Sub TKomma_Click()                       'TASTE ,
Private Sub TAC_Click()                          'TASTE AC
Private Sub TC_Click()                           'TASTE C
Private Sub TGleich_Click()                      'TASTE =
Private Sub TPlus_Click()                        'TASTE +
Private Sub TMinus_Click()                       'TASTE -
Private Sub TMal_Click()                         'TASTE *
Private Sub TGeteilt_Click()                     'TASTE :
Private Sub TPlusMinus_Click()                   'TASTE +/-
```

Ereignisprozeduren

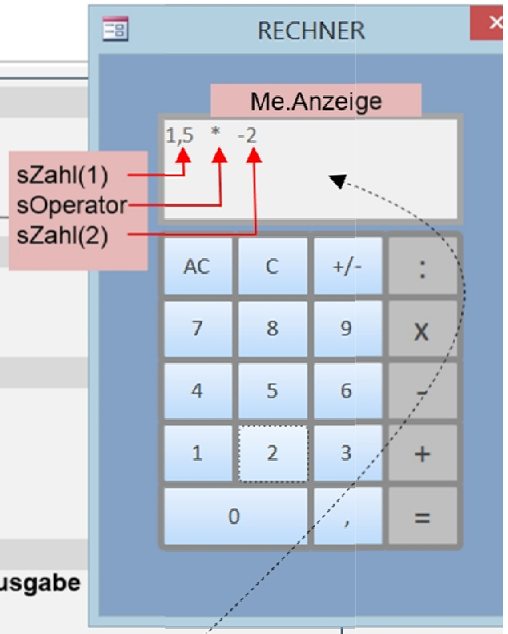
- Beim Öffnen des Formulars
- Beim Klicken der Tasten
- Bei Tastatureingabe

```
Sub Ausgabe()                              'Aufbereitung Anzeige
Function sErgebnis() As String             'Berechnung
```

Prozeduren

Formularmodul - Programmlogik

'Variable auf Modulebene		
Dim sZahl(1 To 2) As String		sZahl(1) und sZahl(2) nehmen die Ziffern und das Komma auf
Dim sOperator As String		' + - * :
Dim iZahl As Long		'Bestimmt aktive Zahl: 1 oder 2
'Formularereignisse		
Private Sub Form_Open(Cancel As Integer)	'Beim Öffnen	'iZahl = 1, sZahl(1) für Eingabe aktiv
Private Sub Form_KeyPress(KeyAscii As Integer)	'Bei Taste	'Tastatureingabe: Ereignisprozedur der entsprechenden 'Schaltfläche wird ausgelöst
'Ziffern, Komma		
Private Sub T0_Click()	'TASTE 0	Zeichen "0" an sZahl(iZahl) anhängen, **Ausgabe**
Private Sub T1_Click()	'TASTE 1	
...		
Private Sub TKomma_Click()	'TASTE ,	Zeichen "," an sZahl(iZahl) anhängen, **Ausgabe**
'Löschen		
Private Sub TAC_Click()	'TASTE AC	sZahl(1)=LEER sZahl(2)=LEER, sOperator=LEER, iZahl = 1, **Ausgabe**
Private Sub TC_Click()	'TASTE C	Rückwärts einzelnes Zeichen löschen, **Ausgabe**
'=, Operatoren		
Private Sub TGleich_Click()	'TASTE =	Berechnen **sErgebnis** und nach sZahl(1) übertragen, iZahl = 2, **Ausgabe**
Private Sub TPlus_Click()	'TASTE +	wenn iZahl = 2 (also Eingabe für 2. Zahl) dann
...		... Berechnen **sErgebnis**, Ergebnis nach sZahl(1) übertragen
		sOperator = +, iZahl = 2, **Ausgabe**
'Vorzeichen		
Private Sub TPlusMinus_Click()	'TASTE +/-	Für sZahl(iZahl) Vorzeichen umkehren, **Ausgabe**
Interne Prozeduren		
Sub **Ausgabe**()	'Anzeige	me.Anzeige = sZahl(1) & sOperator & sZahl(2)
Function **sErgebnis**() As String	'Berechnung	sErgebnis = sZahl(1) +-*/ sZahl(2)



Lektion 15 Prozeduren im Formular (Teil 2) - Datenbank AP00.accdb

Formularmodul frmRechner

```
' ----------------------------------------------------------------------------------------------------------------------------------------------------------
Option Compare Database
Option Explicit
' ----------------------------------------------------------------------------------------------------------------------------------------------------------
' R e c h n e r   mit Eingabe:    | Zahl 1 | Operator | Zahl 2 |      z.B.  -1,5  *  3
' ----------------------------------------------------------------------------------------------------------------------------------------------------------
'Variable auf Modulebene
Dim sZahl(1 To 2) As String
Dim sOperator As String
Dim iZahl As Long                                   'iZahl gibt an, welche Zahl (Zahl 1 oder Zahl 2) für die Eingabe aktiv ist

Private Sub Form_Open(Cancel As Integer)
Me.Anzeige = ""                                     'Anzeige-Feld auf LEER setzen
iZahl = 1                                           ' Beim Öffnen des Formulars wird Zahl 1 aktiviert
End Sub

Private Sub Form_KeyPress(KeyAscii As Integer)
'Tasteneingabe verweist auf Ereignisprozedur der betreffenden Schaltfläche
Select Case KeyAscii
Case 48:    T0_Click
Case 49:    T1_Click
Case 50:    T2_Click
Case 51:    T3_Click
Case 52:    T4_Click
Case 53:    T5_Click
Case 54:    T6_Click
Case 55:    T7_Click
Case 56:    T8_Click
Case 57:    T9_Click
Case 43:    TPlus_Click
Case 44:    TKomma_Click
Case 45:    TMinus_Click
Case 42:    TMal_Click
Case 58:    TGeteilt_Click                  ':
Case 47:    TGeteilt_Click                  '/ alternativ zu :
Case 61:    TGleich_Click
Case 97:    TAC_Click                       'A Display komplett löschen
Case 99:    TC_Click                        'C letztes Zeichen löschen
Case 8:     TC_Click                        'Rücklauftaste alternativ zu C
Case Else:  Me.TGleich.SetFocus             'Leerstaste und alle anderen Tasten wie "="
End Select
End Sub

Private Sub T1_Click()                      'TASTE 1
sZahl(iZahl) = sZahl(iZahl) & "1"           ' "1" wird an die aktive Zahl angehängt.
Ausgabe
End Sub

Private Sub T2_Click()                      'TASTE 2
sZahl(iZahl) = sZahl(iZahl) & "2"           ' "2" wird an die aktive Zahl angehängt.
Ausgabe
End Sub

Private Sub T3_Click()                      'TASTE 3
sZahl(iZahl) = sZahl(iZahl) & "3"           ' "3" wird an die aktive Zahl angehängt.
Ausgabe
End Sub

Private Sub T4_Click()                      'TASTE 4
sZahl(iZahl) = sZahl(iZahl) & "4"           ' "4" wird an die aktive Zahl angehängt.
Ausgabe
End Sub
```

```
Private Sub T5_Click()                          'TASTE 5
sZahl(iZahl) = sZahl(iZahl) & "5"               ' "5" wird an die aktive Zahl angehängt.
Ausgabe
End Sub
```

```
Private Sub T6_Click()                          'TASTE 6
sZahl(iZahl) = sZahl(iZahl) & "6"               ' "6" wird an die aktive Zahl angehängt.
Ausgabe
End Sub
```

```
Private Sub T7_Click()                           'TASTE 7
sZahl(iZahl) = sZahl(iZahl) & "7"               ' "7" wird an die aktive Zahl angehängt.
Ausgabe
End Sub
```

```
Private Sub T8_Click()                           'TASTE 8
sZahl(iZahl) = sZahl(iZahl) & "8"               ' "8" wird an die aktive Zahl angehängt.
Ausgabe
End Sub
```

```
Private Sub T9_Click()                          'TASTE 9
sZahl(iZahl) = sZahl(iZahl) & "9"               ' "9" wird an die aktive Zahl angehängt.
Ausgabe
End Sub
```

```
Private Sub T0_Click()                          'TASTE 0
sZahl(iZahl) = sZahl(iZahl) & "0"               ' "0" wird an die aktive Zahl angehängt.
Ausgabe
End Sub
```

```
Private Sub TKomma_Click()                      'TASTE ,
sZahl(iZahl) = sZahl(iZahl) & "," '             ' "," wird an die aktive Zahl angehängt.
Ausgabe
End Sub
```

```
Private Sub TAC_Click()                         'TASTE AC Display komplett löschen
sZahl(1) = ""                                   'Zahl 1 wird auf LEER gesetzt
sZahl(2) = ""                                   'Zahl 2 wird auf LEER gesetzt
sOperator = ""                                  'Operator wird auf LEER gesetzt
iZahl = 1                                       'Zahl 1 wird für Eingabe aktiviert
Ausgabe
End Sub
```

```
Private Sub TC_Click()                          'TASTE C Letztes Zeichen wird gelöscht
Select Case True
Case sZahl(iZahl) <> ""                         'wenn aktive Zahl nicht LEER, letztes Zeichen abschneiden
   sZahl(iZahl) = Mid(sZahl(iZahl), 1, Len(sZahl(iZahl)) - 1)
Case iZahl = 2 And sZahl(2) = ""                'wenn Zahl 2 aktiv und LEER, Operator wird LEER gesetzt
   sOperator = ""
   iZahl = 1
End Select
Ausgabe
End Sub
```

```
Private Sub TGleich_Click()                      'TASTE =
'Wenn Zahl 2 nicht leer, dann  R e c h n e n, Ergebnis wird Zahl 1 zugewiesen
If sZahl(2) <> "" Then sZahl(1) = sErgebnis()
sZahl(2) = ""
iZahl = 1                                       'Zahl 1 wird aktiviert
sOperator = ""
Ausgabe
End Sub
```

```
Private Sub TPlus_Click()                          'TASTE +
If sZahl(1) = "" Or sZahl(1) = "-" Then Exit Sub   'Keine Aktion
If iZahl = 2 Then                                  'Wenn Zahl 2 aktiv
   'Wenn Zahl 2 nicht leer, dann R e c h n e n, Ergebnis wird Zahl 1 zugewiesen
   If sZahl(2) <> "" Then sZahl(1) = sErgebnis()
   sZahl(2) = ""
End If
iZahl = 2                                          'Zahl 2 wird aktiv gesetzt
sOperator = "+"                                    'Operator wird standardmäßig auf + gesetzt
Ausgabe
End Sub
```

```
Private Sub TMinus_Click()                         'TASTE -
Select Case True
Case iZahl = 1 And sZahl(1) = ""                   'Wenn Zahl 1 aktiv und LEER
   sZahl(1) = "-"                                  'Interpretation - Vorzeichen für Zahl 1
Case iZahl = 1 And sZahl(1) = "-"                  'Wenn Zahl 1 aktiv und  -
   sZahl(1) = ""                                   'Interpretation - Vorzeichen umkehren für Zahl 1
Case iZahl = 1                                     'Interpretation als Operator
   iZahl = 2                                       'Zahl 2 wird aktiviert
   sOperator = "-"                                 'Operator wird -
Case iZahl = 2 And sZahl(2) = "" And sOperator <> ""   'Wenn Zahl 2 aktiv und LEER und Operator bereits Wert hat
   sZahl(2) = "-"                                  'Interpretation - als Vorzeichen für Zahl 2
Case iZahl = 2 And sZahl(2) = "-" And sOperator <> ""  'Wenn Zahl 2 aktiv und "-" und Operator bereits Wert hat
   sZahl(2) = ""                                   'Interpretation - Vorzeichen umkehren für Zahl 2
Case iZahl = 2                                     'Wenn Zahl 2 aktiv, nicht Leer, Operator muss gesetzt sein
   If sZahl(2) <> "" Then sZahl(1) = sErgebnis()    'R e c h n e n, Ergebnis wird Zahl 1 zugewiesen
   sZahl(2) = ""
   iZahl = 2                                       'Zahl 2 wird aktiv gesetzt
   sOperator = "-"                                 'Operator wird -
End Select
Ausgabe
End Sub
```

```
Private Sub TMal_Click()                           'TASTE *
If sZahl(1) = "" Or sZahl(1) = "-" Then Exit Sub   'Keine Aktion
If iZahl = 2 Then
   'Wenn Zahl 2 nicht leer, dann  R e c h n e n, Ergebnis wird Zahl 1 zugewiesen
   If sZahl(2) <> "" Then sZahl(1) = sErgebnis()
   sZahl(2) = ""
End If
iZahl = 2                                          'Zahl 2 wird aktiviert
sOperator = "*"                                    'Operator wird *
Ausgabe
End Sub
```

```
Private Sub TGeteilt_Click()                       'TASTE :
If sZahl(1) = "" Or sZahl(1) = "-" Then Exit Sub   'Keine Aktion
If iZahl = 2 Then                                  'Wenn Zahl 2 aktiviert
   'Wenn Zahl 2 nicht leer, dann  R e c h n e n, Ergebnis wird Zahl 1 zugewiesen
   If sZahl(2) <> "" Then sZahl(1) = sErgebnis()
   sZahl(2) = ""
End If
iZahl = 2                                          'Zahl 2 wird aktiviert
sOperator = ":"                                    'Operator wird /
Ausgabe
End Sub
```

```
Private Sub TPlusMinus_Click()                     'TASTE +/-
'Vorzeichen wird umgekehrt
If Mid(sZahl(iZahl), 1, 1) = "-" Then
   sZahl(iZahl) = Mid(sZahl(iZahl), 2, Len(sZahl(iZahl)) - 1)
Else
   sZahl(iZahl) = "-" & sZahl(iZahl)
End If
Ausgabe
End Sub
```

```
Sub Ausgabe()
'Aufbereitung Anzeige
Me.Anzeige = sZahl(1) & "   " & sOperator & "   " & sZahl(2)

'Wenn Das Ergebnis ERROR ist, dann werden Zahl 1, 2 und Operator auf LEER gesetzt
If sZahl(1) = "ERROR" Then
   sOperator = ""
   sZahl(1) = ""
   sZahl(2) = ""
   iZahl = 1
End If
End Sub
```

```
Function sErgebnis() As String
If IsNumeric(sZahl(1)) And IsNumeric(sZahl(2)) Then
   Select Case True
   Case sOperator = "+"
      sErgebnis = CStr(CDbl(sZahl(1)) + CDbl(sZahl(2)))
   Case sOperator = "-"
      sErgebnis = CStr(CDbl(sZahl(1)) - CDbl(sZahl(2)))
   Case sOperator = "*"
      sErgebnis = CStr(CDbl(sZahl(1)) * CDbl(sZahl(2)))
   Case sOperator = ":" And CLng(sZahl(2)) <> 0
      sErgebnis = CStr(CDbl(sZahl(1)) / CDbl(sZahl(2)))
   Case Else
      sErgebnis = "ERROR"
   End Select
Else
   sErgebnis = "ERROR"
End If
End Function
```

Lektion 16 Go Live - Datenbank AP00.accdb

```
' ---------------------------------------------------------------------------------------------------------------------------------------
' Modul: AP34_Namensortieren
' ---------------------------------------------------------------------------------------------------------------------------------------
Option Compare Database
Option Explicit

Sub Namensortieren()
' ---------------------------------------------------------------------------------------------------------------------------------------
' Es werden vom Benutzer Namen eingegeben und von System in sortierter Form wieder ausgegeben
' ---------------------------------------------------------------------------------------------------------------------------------------
Dim sEingabe As String            'Eingabetext
Dim sAusgabe As String            'Ausgabetext
Dim sName() As String             'Array() für Namen
Dim sNameSave As String           'Merkfeld
Dim iEnd As Long                  'Anzahl Einträge im Array
Dim i As Long
Dim iP1 As Long                   'Start der Suche nach Komma im Eingabetext
Dim iP2 As Long                   'Position Komma Im Eingabetext
Dim i1 As Long                    'Zählvariable
Dim i2 As Long                    'Zählvariable
Dim iSave As Long                 'Merkfeld
Dim sNameMin As String            'Merkfeld, kleinster Name

' ---------------------------------------------------------------------------------------------------------------------------------------
' Eingabe
' ---------------------------------------------------------------------------------------------------------------------------------------
sEingabe = InputBox("Bitte geben Sie die Namen, getrennt durch Kommata ein")
If sEingabe = "" Then Exit Sub

' ---------------------------------------------------------------------------------------------------------------------------------------
' Eingabe - Aufbereitung
' ---------------------------------------------------------------------------------------------------------------------------------------
If Mid(sEingabe, Len(sEingabe), 1) <> "," Then sEingabe = sEingabe & ","
ReDim sName(1 To 1)
iEnd = 0
iP1 = 1
Do
   iP2 = InStr(iP1, sEingabe, ",")
   If iP2 = 0 Then Exit Do
   sNameSave = Trim(Mid(sEingabe, iP1, iP2 - iP1))
   iEnd = iEnd + 1
   ReDim Preserve sName(1 To iEnd)
   sName(iEnd) = sNameSave
   iP1 = iP2 + 1
Loop

' ---------------------------------------------------------------------------------------------------------------------------------------
' Verarbeitung - Sortierung
' ---------------------------------------------------------------------------------------------------------------------------------------
For i1 = 1 To iEnd - 1
   sNameMin = sName(i1)
   iSave = i1
   For i2 = i1 To iEnd
      If sName(i2) < sNameMin Then
         sNameMin = sName(i2)
         iSave = i2
      End If
   Next i2
   sNameSave = sName(i1)
   sName(i1) = sNameMin
   sName(iSave) = sNameSave
Next i1

' ---------------------------------------------------------------------------------------------------------------------------------------
' Ausgabe - Aufbereitung
' ---------------------------------------------------------------------------------------------------------------------------------------
For i1 = 1 To iEnd
   sAusgabe = sAusgabe & sName(i1) & ", "
Next i1

' ---------------------------------------------------------------------------------------------------------------------------------------
' Ausgabe
' ---------------------------------------------------------------------------------------------------------------------------------------
MsgBox "Ihre Eingabe ..." & pcsUmbruch & sEingabe & pcsUmbruch & "... sortiert" & pcsUmbruch & sAusgabe

End Sub
```