Wirtschaftsinformatik 2

Aufruf eigener Prozeduren, Arrays

Bernd Blümel, Christian Bockermann, Christian Metzger

Hochschule Bochum

Sommersemester 2023

Inhalt

Aufruf eigener Prozeduren

Nutzung eigener Funktionen

Eigene Funktionen haben wir bereits benutzt, z.B:

```
Dim brutto As Double
Dim nettoBetrag As Double

For i = 1 To letzteBesetzteZeile
   nettoBetrag = Sheets(1).Cells(i, 2)
   brutto = berechneBrutto(nettoBetrag)

   ' Verwenden von brutto...
End For
```

Nutzung eigener Funktionen

Eigene Funktionen haben wir bereits benutzt, z.B:

```
Function berechneBrutto(netto As Double) As
                                Double
   Const steuersatz As Double = 0.19

   berechneBrutto = netto * steuersatz
End Function
```

Eigene Funktionen

- werden mit notwendigen Parametern aufgerufen
- haben einen Rückgabewert

```
brutto = berechneBrutto(nettoPreis)
```

Eigene Prozeduren

- benötigen ebenfalls Parameter
- haben allerdings keinen Rückgabewert

Eigene Prozeduren

Prozeduren haben keinen Rückgabewert

```
Sub teste(letzteBesetzteZeile As Integer)
  Dim wert As Double

  For i=1 to letzteBesetzteZeile
    wert = Sheets(1).Cells(i, 2)
    If wert < 0 Then
      MsgBox("Wert-Spalte darf nicht negativ
                                  sein!")
    End If
  End For
End Sub
```

Aufruf eigener Prozeduren

```
letzteZeile = ermittleLetzteBesetzteZeile(..)
wert = test(letzteZeile)
```

Ohne Rückgabewert - Was soll dann in `wert` stehen?

Aufruf eigener Prozeduren

```
letzteZeile = ermittleLetzteBesetzteZeile(..)
wert = test(letzteZeile)
```

Ohne Rückgabewert - Was soll dann in `wert` stehen?

Prozeduren werden in VBA mit `call` aufgerufen:

```
letzteZeile = ermittleLetzteBesetzteZeile(..)

Call teste(letzteBesetzteZeile)
```

Datentyp Array

Bisherige Datentypen

Bisherige Datentypen stellen einen Wert dar:

- Bool
- Integer
- Double
- String

Bisherige Datentypen

Bisherige Datentypen stellen einen Wert dar:

- Bool
- Integer
- Double
- String

Wie speichern wir z.B. eine Liste von Zahlen?

Primitive Lösung:

```
Dim zahl1 As Double
Dim zahl2 As Double
Dim zahl3 As Double
Dim zahl4 As Double
```

Primitive Lösung:

```
Dim zahl1 As Double
Dim zahl2 As Double
Dim zahl3 As Double
Dim zahl4 As Double
```

Aber:

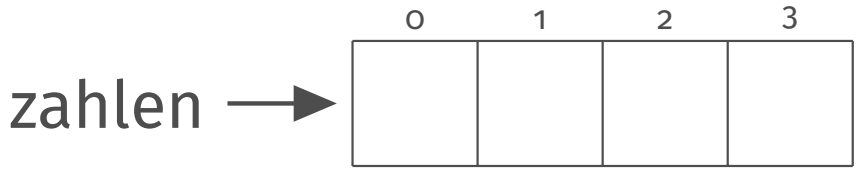
- wir wissen vorab nicht, wieviele Zahlen wir brauchen
- wir wollen Zahlen schrittweise hinzufügen

Der Datentyp Array

Arrays sind Datenstrukturen für eine Liste von Werten:

```
Dim zahlen() As Double
Redim Preserve zahlen(3)
```

Was genau passiert dabei?



Excel reserviert Plätze für 4 Double-Werte im Speicher

Arrays

Arrays enthalten mehrere Werte des gleichen Typs:

- mehrere Zahlen (Integer, Double)
- mehrere Texte, usw.
- auf jedes Element kann über den Index zugegriffen werden

Arrays

Über Index können Werte in das Array geschrieben werden:

```
Dim zahlen() As Double
ReDim Preserve zahlen(2)

zahlen(0) = 1.8
zahlen(1) = 2.3
zahlen(2) = 4.7
```

Zugriff auf Arrays

Auslesen der Wert geschieht ebenfalls über den Index:


```
Dim wertAmIndex2 As Double

wertAmIndex2 = zahlen(2)
```

Zugriff auf Arrays

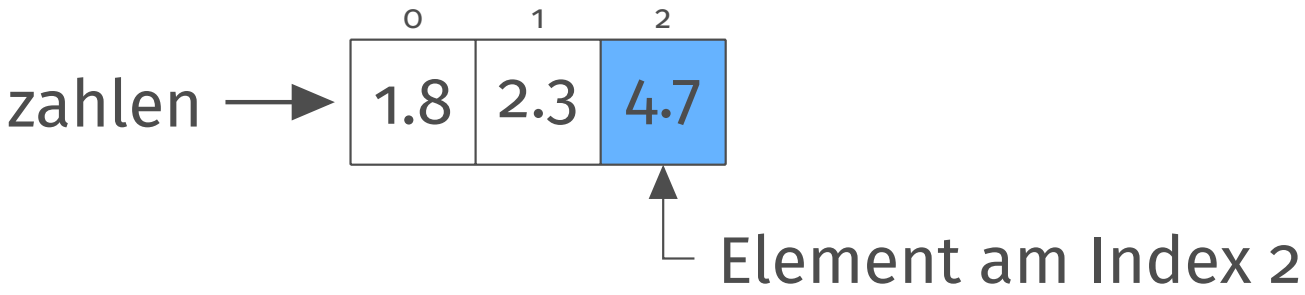
Auslesen der Wert geschieht ebenfalls über den Index:


```
Dim wertAmIndex2 As Double

wertAmIndex2 = zahlen(2)
```

Zugriff auf Arrays

Auslesen der Wert geschieht ebenfalls über den Index:




```
Dim wertAmIndex2 As Double

wertAmIndex2 = zahlen(2)
```

Was ist der derzeit höchste gültige Index im Array?

- UBound berechnet höchsten zugreifbaren Index

```
Dim letzterPlatz As Integer

letzterPlatz = UBound(zahlen)
```

Beispiel: Mittelwert von Zahlen berechnen

```
Function mittelwert(zahlen() As Double) As
                                  Double
  Dim summe As Double
  Dim anzahl As Double
  summe = 0
  anzahl = 0

  For i = 0 TO UBound(zahlen)
    summe = summe + zahlen(i)
    anzahl = anzahl + 1.0
  Next i

  mittelwert = summe / anzahl
End Function
```

Vergrößern von Arrays

Ein leeres Array definieren:

```
Dim zahlen() As Integer
```

Array auf 1 Platz vergrößern:

```
ReDim Preserve zahlen(0)
```

Wieviele Elemente sind derzeit im Array?

```
Dim anzahl As Integer
anzahl = UBound(zahlen)
```

Elemente zu Array hinzufügen

```
Dim zahlen() As Double
ReDim Preserve zahlen(0)
```

Elemente zu Array hinzufügen

```
Dim zahlen() As Double
ReDim Preserve zahlen(0)
```

Wir wollen die Zahl 42 hinzufügen:

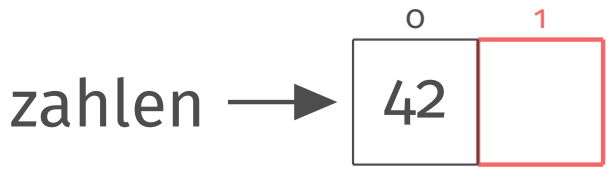
1. Wir bestimmen die aktuelle Größe
2. Wir schreiben die 42 an die letzte Position
3. Wir vergrößern das Array mit ReDim

Elemente zu Array hinzufügen

```
Dim ende As Double
ende = UBound(zahlen)
zahlen(ende) = 42
```

Elemente zu Array hinzufügen

```
Dim ende As Double
ende = UBound(zahlen)
zahlen(ende) = 42
```




```
Redim Preserve zahlen(ende + 1)
```