



Visual Basic for Applications

Prof. Dr. Ralf Laue



Gib die ersten Potenzen der Zahl 3 aus

- Wir beginnen mit der Ausgabe der Zahl 1.
- Danach geben wir fünfmal das Dreifache der zuvor ausgegebenen Zahl aus.
- Die Ausgabe startet also mit 1, 3, 9, 27, ...



Name		Inhalt
x		1



Name		Inhalt
x		3



Name		Inhalt
x		9



Name		Inhalt
x		27



Name		Inhalt
x		81



Name		Inhalt
x		243



Rechenaufgabe

- Wir beginnen mit der Ausgabe der Zahl 1.

x erhält den Wert 1
x ausgeben

```
x = 1  
debug.print(x)
```

- Danach geben wir fünfmal das Dreifache der zuvor ausgegebenen Zahl aus.

fünfmal:
x verdreifachen
x ausgeben

```
' fünfmal:  
x = x * 3  
debug.print(x)
```



Zuweisung

- x wird der Wert 1 zugewiesen:

$$x = 1$$

- x wird verdreifacht:

$$x = x * 3$$

neuer Wert von x

wird

bisheriger Wert von x

mal 3



Programm für unsere Aufgabenstellung

```
Sub maldrei()  
    x = 1  
    Debug.Print (x)  
    x = x * 3  
    Debug.Print (x)  
    x = x * 3  
    Debug.Print (x)  
    x = x * 3  
    Debug.Print (x)  
    x = x * 3  
    Debug.Print (x)  
    x = x * 3  
    Debug.Print (x)  
End Sub
```



Operatoren zwischen Zahlen

Grundrechenarten:

+ - * /

\ (ganzzahlige Div.)

(Rest wird ignoriert)

$$10 \setminus 3 = 3$$

mod (der Rest)

$$10 \bmod 3 = 1$$

^ (Potenz)

$$5^3 = 125$$

Wurzel

$$125^{(1/3)}$$

~~$$125^{1/3}$$~~



Nach der Anweisung

```
x = InputBox("Wie heißt Du denn?")
```

ist der im Computer mit x benannte Speicherplatz (Variable) mit den eingegebenen Zeichen (in der Regel Buchstaben) belegt.

Name	Nummer (Adresse)	Inhalt
	300	
x		"Anna"
	304	
		1000001011011100110111001100001



Nach der Anweisung

$$\mathbf{x = 1}$$

ist der im Computer mit x benannte Speicherplatz mit der Zahl 1 belegt.

Name	Nummer (Adresse)	Inhalt
	300	
x		1
	...	



Dimensionierung – Dim-Anweisung

Option Explicit

Sub Addition

Dim summand1 As Integer, summand2 As Integer

Dim ergebnis As Integer

summand1 = 5

summand2 = 3

ergebnis = summand1 + summand2

Debug.Print(ergebnis)

End Sub

sagt dem Computer:
summand1 und
summand2 sind
ganze Zahlen



Dimensionierung – Dim-Anweisung

Falsch:

~~Dim summand1, summand 2 As Integer~~

Richtig:

Dim summand1 As Integer, summand 2 As Integer

oder:

Dim summand1 As Integer

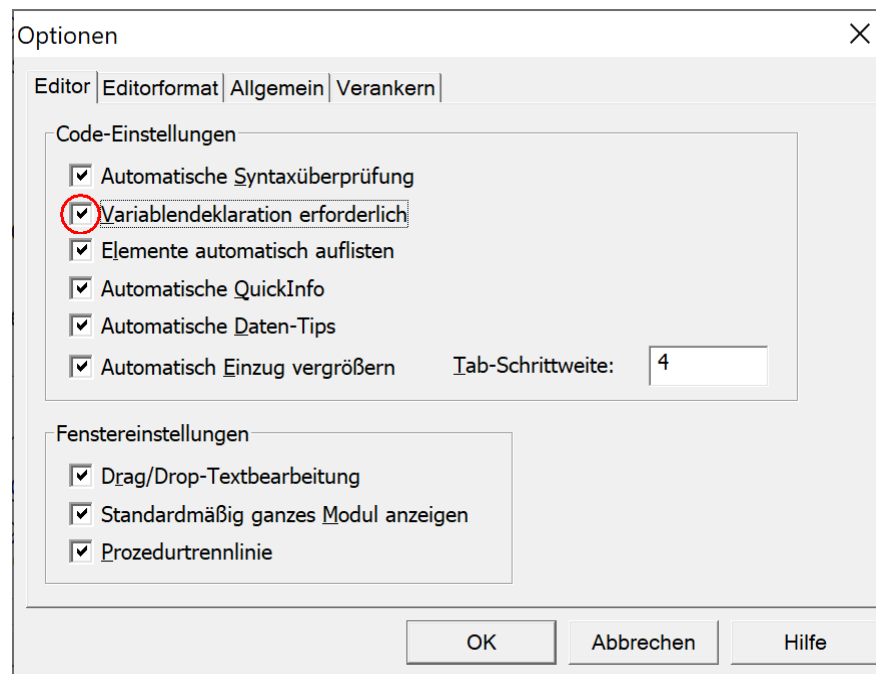
Dim summand2 As Integer



Option Explicit

Legt fest, dass jede Variable dimensioniert werden muss (sehr empfohlen!)

Kann auch unter Extras-Optionen dauerhaft gefordert werden:





Dimensionierung, Eingabe und Ausgabe

Option Explicit

Sub ein_und_ausgabe

Dim x as String

x = InputBox("Wie heißt Du denn?")

Debug.Print(x)

End Sub

Ein- und ausgegeben wird eine Zeichenkette (String).



Dimensionierung, Eingabe und Ausgabe

Option Explicit

Sub ein_und_ausgabe

 Dim x as Integer

 x = Val(InputBox("Bitte gib eine Zahl ein"))

 Debug.Print(x)

End Sub

Ein- und ausgegeben wird eine ganze Zahl (Integer).



Val und Str

- Val macht aus einer Zeichenkette eine Zahl:

```
eingabe = InputBox("Wie alt sind Sie?")  
alter = Val(eingabe)
```

- Str macht aus einer Zahl eine Zeichenkette:

```
zinsatz = 2.5  
MsgBox("Der Zinssatz ist: " & Str(zinsatz))
```

+ oder &: Aneinanderfügen von Zeichenketten



Wir merken uns...

`x = Val(TextBox("Zahl eingeben:"))`

`y = TextBox("Zeichenkette eingeben:"))`

Belegung einer String-Variablen im Nutzerdialog: ohne Val

Belegung einer Zahlen-Variablen im Nutzerdialog: mit Val



Fehler im Programm



Syntax und Semantik von Programmiersprachen

- Vor dem Abarbeiten wird geprüft, ob das Programm den Regeln der Programmiersprache entspricht.
- → Nur syntaktisch korrekte Programme werden abgearbeitet
- Syntax einer Programmiersprache - Struktur, Form der erlaubten Zeichenfolgen
- Semantik einer Programmiersprache - Bedeutung der erlaubten Zeichenfolgen (Wie reagiert ein Programm auf die Befehle in einem Programm?)



Syntaxfehler

Die fehlende Klammer in
MsgBox ("Summe=" & Str(s1 + s2)
wird bereits während der Eingabe erkannt:

```
Sub Addition()  
Dim s1 As Integer, s2 As Integer  
s1 = Val(InputBox("Erste Zahl eingeben:"))  
s2 = Val(InputBox("Zweite Zahl eingeben:"))  
MsgBox ("Summe=" + Str(s1 + s2)
```





Syntaxfehler

Ein Vergessen der zweiten Vereinbarung der Variablen s2

Dim s1 As Integer, s2 As Integer

wird beim Aufruf des Programms erkannt:

```
Sub Addition()  
Dim s1 As Integer  
s1 = Val(TextBox("Erste Zahl eingeben:"))  
s2 = Val(TextBox("Zweite Zahl eingeben:"))  
MsgBox ("Summe=" + Str(s1 + s2))
```





Debug-Modus

- Nach Drücken von "OK" befindet sich das Programm im Debug-Modus:



```
Sub Addition()  
Dim s1 As Integer  
s1 = Val(TextBox("Erste Zahl eingeben:"))  
s2 = Val(TextBox("Zweite Zahl eingeben:"))  
MsgBox ("Summe=" + Str(s1 + s2))
```

Beenden durch Klicken auf das Quadrat
(Debuggen - Zurücksetzen), dann Fehler korrigieren.



Fehlersuche im laufenden Program

- Debug-Modus: Schrittweises Testen eines Programms
- F8 – nächster Schritt
- F9 – Haltepunkt setzen
- Mit *Ansicht - Lokal-Fenster* wird ein Fenster geöffnet, das die Werte von allen benutzten Variablen anzeigt.



Logische Fehler

MsgBox ("Summe=" & s1)

statt

MsgBox ("Summe=" & (s1 + s2))

Logische Fehler müssen wir selber erkennen!

Dazu muss ein Programm mit gut durchdachten Testwerten getestet werden.



Programm zur Division

```
Sub Division()
```

```
    Dim s1 As Integer, s2 As Integer
```

```
    s1 = Val(InputBox("Erste ganze Zahl eingeben:"))
```

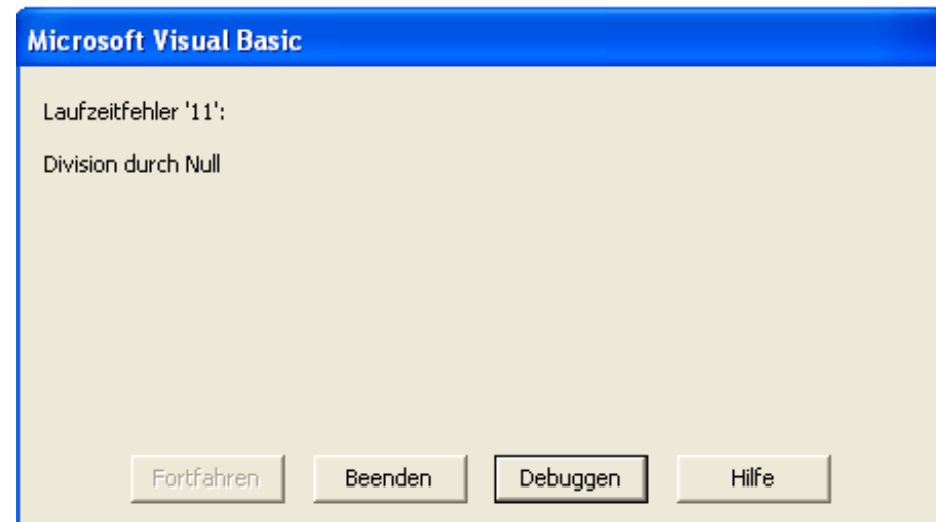
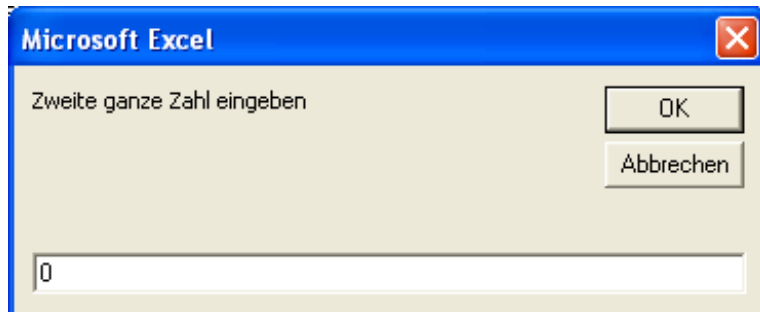
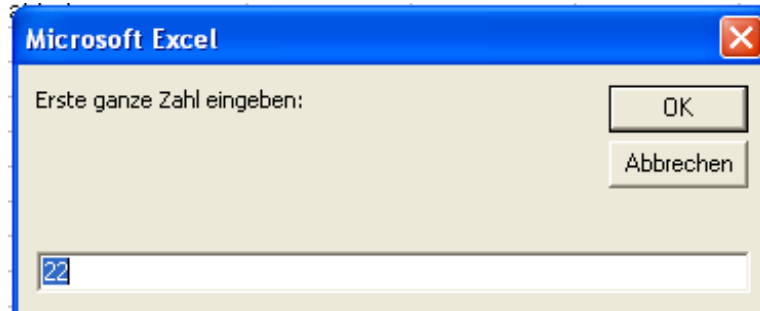
```
    s2 = Val(InputBox("Zweite ganze Zahl eingeben"))
```

```
    MsgBox ("Quotient=" & (s1 / s2))
```

```
End Sub
```



Laufzeitfehler



Division durch 0!



Wieder Addition... Was ist passiert? Wer ist schuld?

Microsoft Excel

Erste ganze Zahl eingeben:

OK

Abbrechen

1.5

Microsoft Excel

Zweite ganze Zahl eingeben:

OK

Abbrechen

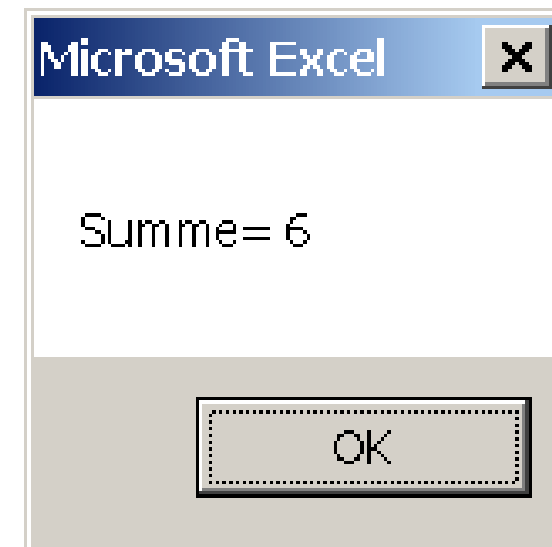
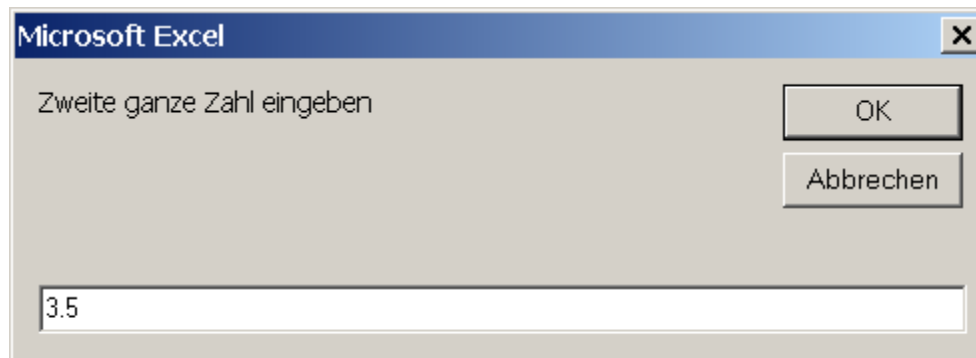
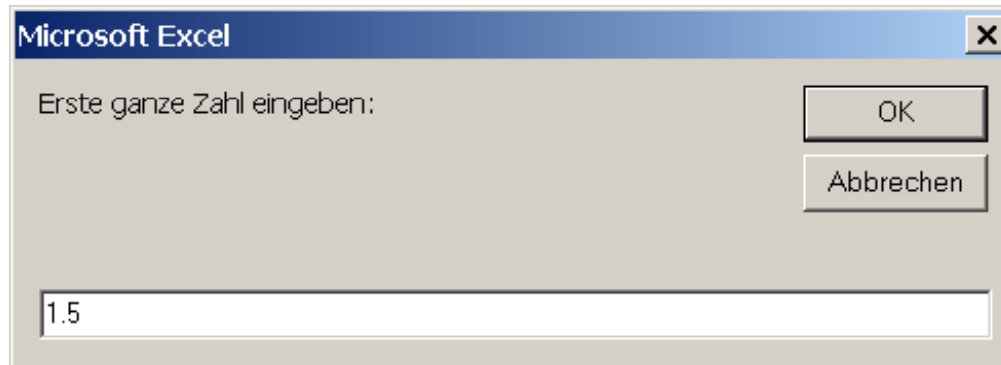
3.5

Microsoft Excel

Summe= 6

OK

Es ist immer die Aufgabe des Programmierers, mögliche Fehleingaben korrekt zu behandeln!



Zur Erinnerung:
Es gibt verschiedene Arten von Speicherplätzen (Variablen).

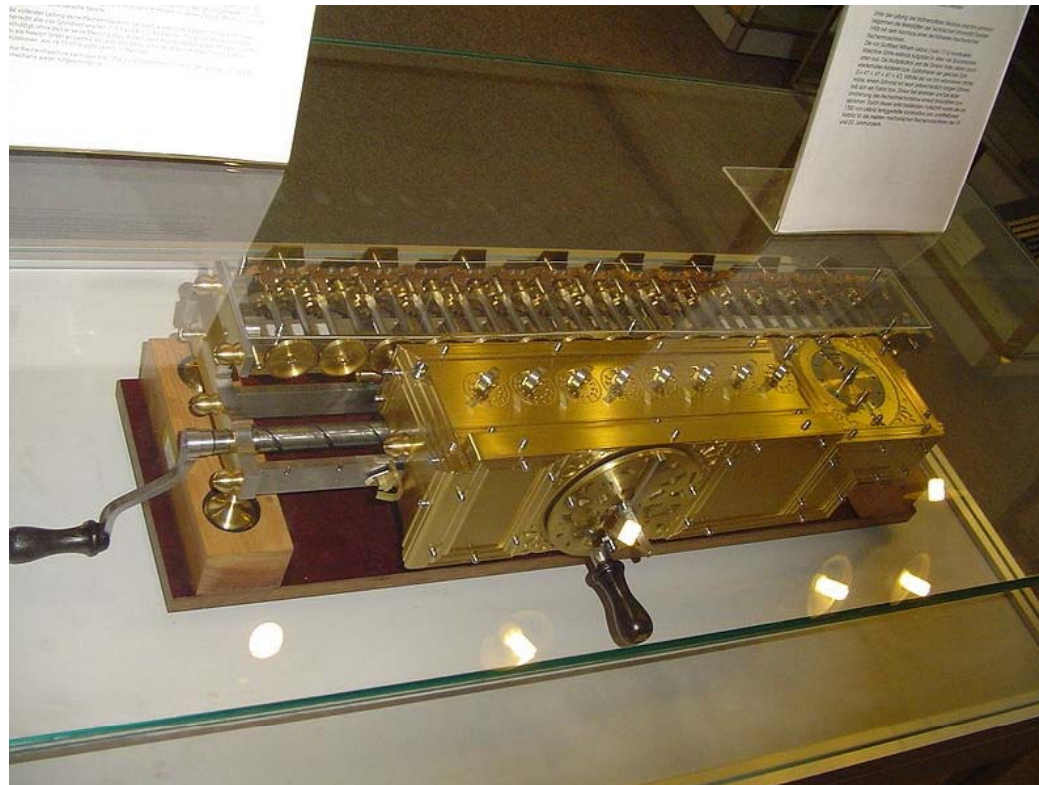


Daten und Informationen

Ein Rechner versteht nur Nullen und Einsen, also geht es zunächst um die Frage, wie man mit Nullen und Einsen Informationen repräsentiert.



- Leibniz entdeckte im 17. Jahrhundert, dass sich mechanische Rechenmaschinen leichter konstruieren lassen, wenn sie auf einer binären Kodierung beruhen.





Zahlendarstellung

im üblichen Dezimalsystem (Ziffern: 0 bis 9):

$$438 = 4 \cdot 100 + 3 \cdot 10 + 8 \cdot 1$$

$$438 = 4 \cdot 10^2 + 3 \cdot 10^1 + 8 \cdot 10^0$$



Zahlendarstellung im Dualsystem (Binärsystem)



- 10010 heißt:
 $1 \cdot 2^4 + 0 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0$
- Ziffern: 0 und 1
- Bit = eine Null bzw. eine Eins
(eine Ziffer im Dualsystem)



Eins



00001

$$0 \cdot 2^4 + 0 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0$$



Zwei



00010

$$0 \cdot 2^4 + 0 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0$$



Drei



00011

$$0 \cdot 2^4 + 0 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0$$



Vier



00100

$$0 \cdot 2^4 + 0 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 0 \cdot 2^0$$



There are 10 kinds of people...





Darstellung ganzer Zahlen in VBA

- VBA bietet für das Speichern von ganzen Zahlen Speicherplätze (Variable) mit 2 Byte und 4 Byte an.
- Zu Beginn des Programms wird vereinbart, welche Art der Speicherung genutzt werden soll:

Dim x as Integer	2 Byte	16 Bit	-32768 bis 32767
Dim x as Long	4 Byte	32 Bit	-2147483648 bis 2147483647



Vereinbarung der Speicherplätze

- Der Programmierer muss sich VORHER überlegen, welche Zahlen auf den einzelnen Speicherplätzen zu erwarten sind.

```
Sub So_gehts_schief()  
  Dim x As Integer  
  x = 50000  
  MsgBox ("x hat den Wert" & x)  
End Sub
```



Darstellung gebrochener Zahlen

719,243127

Wie in Darstellung mit Bits überführen?

Wir brauchen Bits für die folgenden Informationen:

- Ist die Zahl positiv oder negativ?
- Wo steht das Komma?
- Wie ist die Folge der Ziffern?



Darstellung gebrochener Zahlen

Dim y As **Single**

Benutzt 4 Byte = 32 Bit

- Ist die Zahl positiv oder negativ? }
 - Wo steht das Komma? }
 - Wie ist die Folge der Ziffern?
- 1 Byte
3 Byte



Darstellung gebrochener Zahlen

Dim y As **Double**

Benutzt 8 Byte = 64 Bit

- Ist die Zahl positiv oder negativ? }
 - Wo steht das Komma? }
 - Wie ist die Folge der Ziffern? }
- 11 Bit
- 52 Bit



Darstellung ist ungenau!

- Überzählige Stellen werden gerundet

```
Sub Zehn_durch_drei()
```

```
Dim quotient1 As Single
```

```
Dim quotient2 As Double
```

```
quotient1 = 10 / 3
```

```
quotient2 = 10 / 3
```

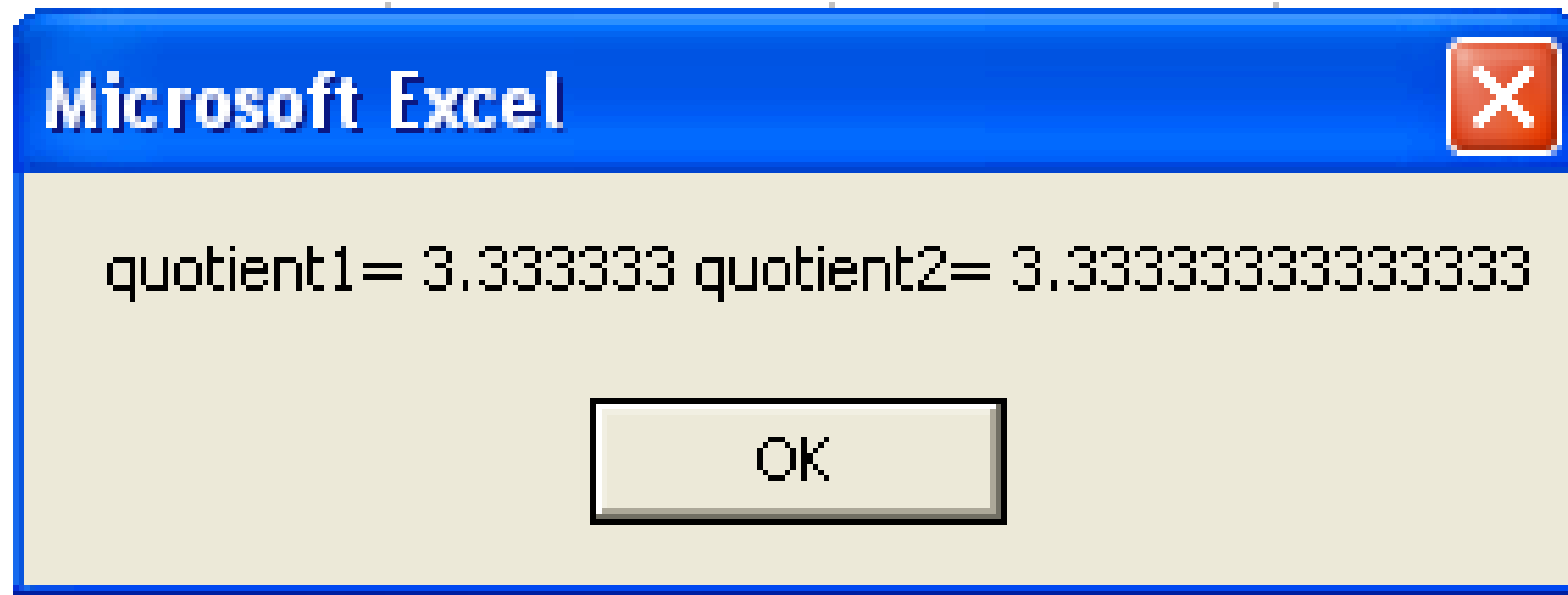
```
MsgBox ("quotient1=" & Str(quotient1) _
```

```
& " quotient2=" & Str(quotient2))
```

```
End Sub
```

 Anweisung geht über
mehrere Programmzeilen:

Leerzeichen - Unterstrich -
Enter



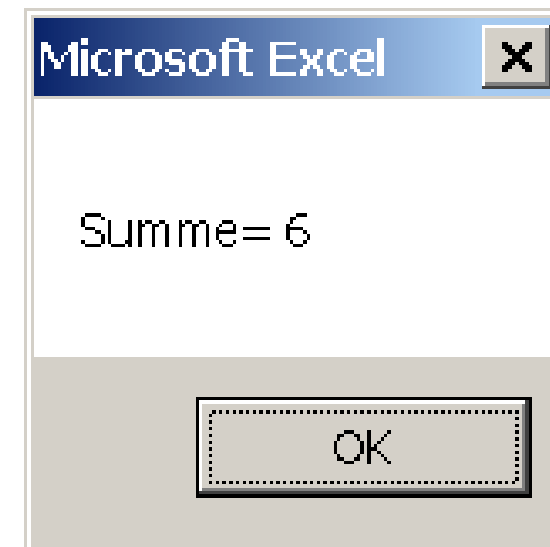
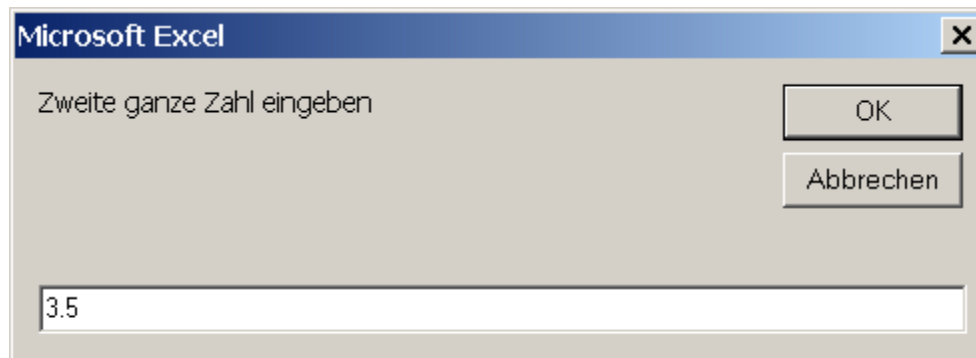
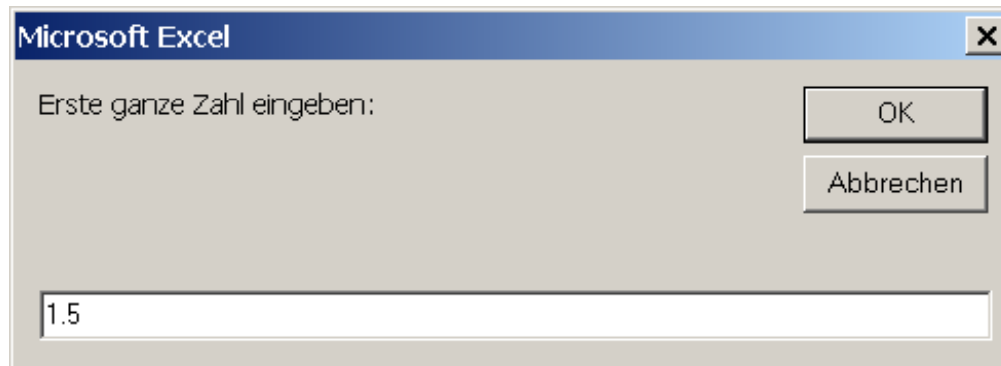
Beachte: Statt Komma wird ein Punkt verwendet!



Dim x as Single	4 Byte	32 Bit	Maximalwert ca. $3 \cdot 10^{38}$ kleinster positiver Wert ca. 10^{-45} genau auf 6 oder 7 Stellen
Dim x as Double	8 Byte	64 Bit	Maximalwert ca. $1,8 \cdot 10^{308}$ kleinster positiver Wert ca. $5 \cdot 10^{-324}$ genau auf 15 oder 16 Stellen



Was ist also vorhin passiert?





Rundung

- Beispiel:
- 21,27 € + 19 % MWSt. ist zu berechnen

```
Sub MWST_ungerundet()
```

```
Dim preis As Single
```

```
preis = 21.27
```

```
preis = preis * 1.19
```

```
Debug.Print (preis)
```

```
End Sub
```

Ausgabe: 25,3113



Kleine Verbesserung - Konstanten

```
Sub MWST_ungerundet()
```

```
  Dim preis As Single
```

```
  Const MWSt as Integer = 19
```

```
  preis = 21.27
```

```
  preis = preis * (100+MWSt) / 100
```

```
  Debug.Print (preis)
```

```
End Sub
```

Ausgabe: 25,3113



Runden "in Handarbeit"

```
Sub MWST_runden_manuell()
```

```
Dim preis As Single, ausgabe As Single
```

```
Dim hilfsvARIABLE As Integer
```

```
Const MWSt as Integer = 19
```

```
preis = 21.27
```

```
preis = preis * (100 + MWSt) / 100
```

```
preis = preis * 100
```

```
hilfsvARIABLE = preis      ' Hier wird gerundet!
```

```
ausgabe = hilfsvARIABLE / 100
```

```
Debug.Print (ausgabe)
```

```
End Sub
```



Die Funktion Round

```
Sub MWST_runden_mit_rundungsfunktion()
```

```
    Dim preis As Single, ausgabe As Single
```

```
    Const MWSt As Integer = 19
```

```
    preis = 21.27
```

```
    preis = preis * (100 + MWSt) / 100
```

```
    ausgabe = Round(preis, 2)
```

```
    Debug.Print (ausgabe)
```

```
End Sub
```

Val(x)

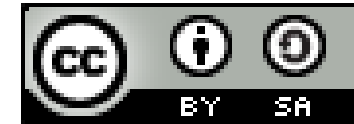


Informationen zu Urhebern und Lizenzen der verwendeten Bilder finden Sie unter den angegebenen Links:

Nachbau der Rechenmaschine von Leibnitz

Urheber: Wikimedia-Nutzer Colossos

<https://commons.wikimedia.org/wiki/File:Leibnitzrechenmaschine.jpg>



Dieses Werk bzw. Inhalt steht unter einer "Creative Commons Namensnennung - Weitergabe unter gleichen Bedingungen 3.0 Deutschland Lizenz".

Sie dürfen:

- das Werk vervielfältigen, verbreiten und öffentlich zugänglich machen
- Bearbeitungen des Werkes anfertigen

zu den folgenden Bedingungen:

- Namensnennung — Sie müssen den Namen des Autors/Rechteinhabers in der von ihm festgelegten Weise nennen.
- Weitergabe unter gleichen Bedingungen — Wenn Sie das lizenzierte Werk bzw. den lizenzierten Inhalt bearbeiten oder in anderer Weise erkennbar als Grundlage für eigenes Schaffen verwenden, dürfen Sie die daraufhin neu entstandenen Werke bzw. Inhalte nur unter Verwendung von Lizenzbedingungen weitergeben, die mit denen dieses Lizenzvertrages identisch oder vergleichbar sind.
- Weiteres dazu: <http://creativecommons.org/licenses/by-sa/3.0/de/>