

**Veranstaltung
Pr.-Nr.: 10 1023 V**

**Wirtschaftsinformatik für
Wirtschaftswissenschaftler**

Algorithmen und ihre Programmierung

- Teil 1 -

**Dr. Chris Bizer
WS 2007/2008**

Freie Universität Berlin – Bizer: Wirtschaftsinformatik – WS07/08 (Version vom 4. 1.08)

Einführung

■ Definition: Algorithmus

Ein Algorithmus ist eine endliche Folge von Anweisungen, deren schrittweise Ausführung eine gestellte Aufgabe löst.

■ Spricht an:

- Aufgabe
- Anweisungen
- Schrittweise Ausführung
- Endliche Folge

■ Anders gesagt:

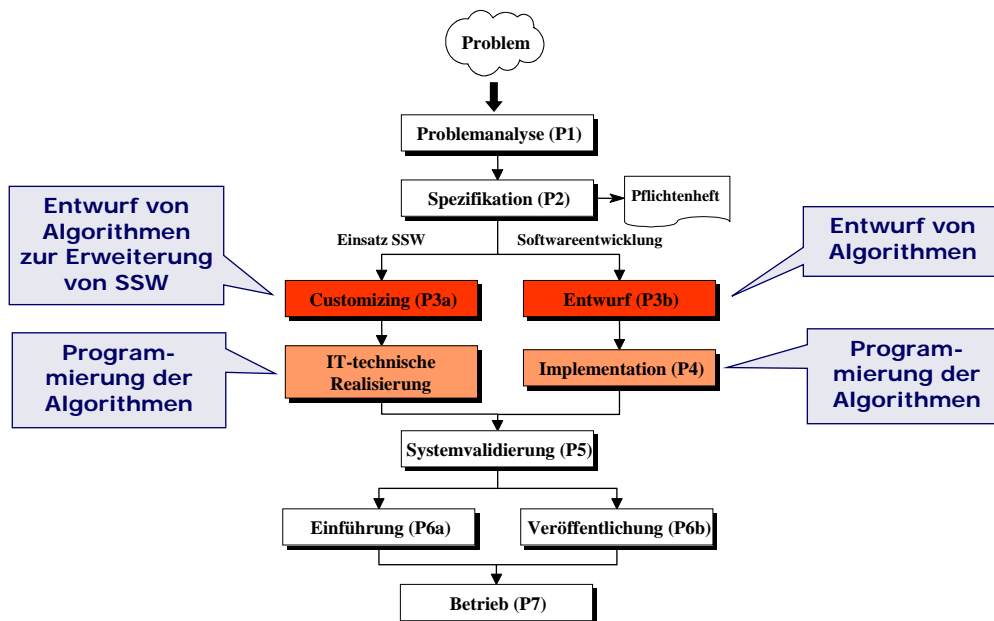
- Ein Algorithmus ist eine exakte und vollständig beschriebene Vorgehensweise zur Lösung einer Aufgabe.

■ Definition: Programm

Ein Programm ist die Umsetzung (Implementierung) eines Algorithmus mit einer konkreten Programmiersprache.

Freie Universität Berlin – Bizer: Wirtschaftsinformatik – WS07/08 (Version vom 4. 1.08)

Rolle von Algorithmen im Systementwicklungsprozess



Freie Universität Berlin – Bizer: Wirtschaftsinformatik – WS07/08 (Version vom 4. 1.08)

Beispiel: Zinsrechnungs-Algorithmus

■ Aufgabe:

- Es soll berechnet werden, wie hoch ein Guthaben inklusive Verzinsung nach einem Jahr ist.

■ Algorithmus:

1. Frage den Benutzer nach Betrag des Guthabens in Euro!
2. Frage den Benutzer nach Zinssatz in Prozent!
3. Berechne das Guthaben nach einem Jahr durch Multiplikation des Betrags mit $(100 + \text{Zinssatz})$ geteilt durch 100!
4. Zeige das Ergebnis an!

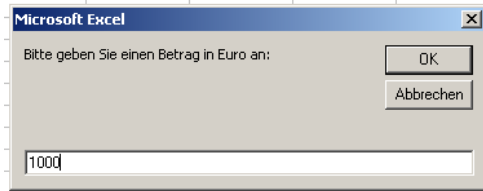
■ Programm:

1. `x = Inputbox("Bitte geben Sie einen Betrag in Euro an:")`
2. `i = Inputbox("Bitte geben Sie einen Zinssatz in Prozent an:")`
3. `e = x * (100+i)/100`
4. `MsgBox "Ihr jetziges Guthaben von " & x & " Euro wird sich in einem Jahr auf " & e & " Euro belaufen."`

Freie Universität Berlin – Bizer: Wirtschaftsinformatik – WS07/08 (Version vom 4. 1.08)

Ausführung des Zinsrechnungsprogramms

1.



Microsoft Excel

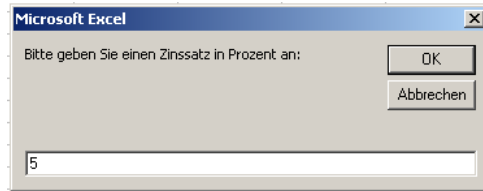
Bitte geben Sie einen Betrag in Euro an:

OK

Abbrechen

1000

2.



Microsoft Excel

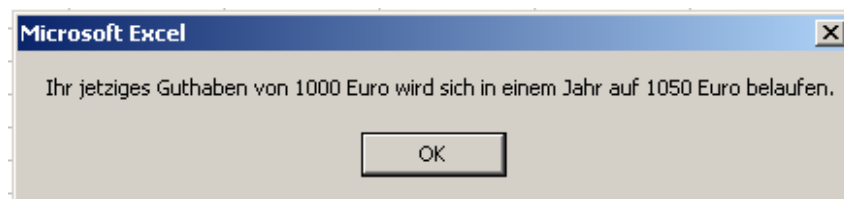
Bitte geben Sie einen Zinssatz in Prozent an:

OK

Abbrechen

5

3.



Microsoft Excel

Ihr jetziges Guthaben von 1000 Euro wird sich in einem Jahr auf 1050 Euro belaufen.

OK

Freie Universität Berlin – Bizer: Wirtschaftsinformatik – WS07/08 (Version vom 4. 1.08)

Kapitel 8: Algorithmen und ihre Programmierung

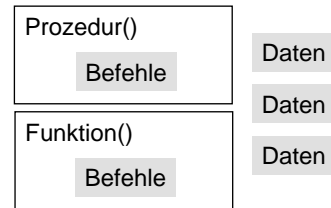
1. Einführung
2. Programmiersprachen
3. Programmkonstrukte von VBA
 1. Variablen und Datentypen
 2. Operatoren
 3. Ein- und Ausgabebefehle
 4. Entscheidungsstrukturen
 5. Schleifenstrukturen
 6. Prozeduren und Funktionen
 7. Arrays

Freie Universität Berlin – Bizer: Wirtschaftsinformatik – WS07/08 (Version vom 4. 1.08)

2. Strukturierte versus objektorientierte Programmierung

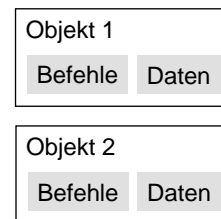
■ Strukturierte Programmierung

- Befehle werden zu Prozeduren und Funktionen zusammengefasst.
- Daten und Befehle werden getrennt betrachtet.



■ Objektorientierte Programmierung

- Daten und Befehle werden gemeinsam als Objekte betrachtet.
- Objektorientierte Programmiersprachen sind sehr verbreitet, z.B. Java, C#



Freie Universität Berlin – Bizer: Wirtschaftsinformatik – WS07/08 (Version vom 4. 1.08)

Programmiersprachen

- Einzelne Programmiersprachen sind für verschiedene Einsatzgebiete besonders gut geeignet:

Programmiersprache	Einsatzgebiet
C++, C# und Java	Universalsprachen für viele Einsatzgebiete, z.B. Programmierung von Windows, Word, SAP
FORTRAN	Mathematische Problemstellungen, da besonders schnelle Rechenoperationen, z.B. MOPS
PHP, Python, Ruby, Javascript	Web-basierte Anwendungssysteme, da gute Zusammenarbeit mit Web-Server und Browser, z.B. Homebanking, Amazon, Ebay
Visual Basic for Applications	Programmiersprache zur Erweiterung der Funktionalität von MS Office 2003

Freie Universität Berlin – Bizer: Wirtschaftsinformatik – WS07/08 (Version vom 4. 1.08)

Visual Basic for Applications (VBA)

- Wir benutzen Visual Basic for Applications (VBA), da
 - VBA eine sehr einfache Programmiersprache ist.
 - VBA in Microsoft Office 2003 integriert ist und Office uns im Pool zur Verfügung steht.
- Microsoft Visual Studio Tools for Office
 - Nachfolger von VBA für Office 2007
 - basierend auf dem .Net Framework
- Skript der Fernuni Hagen zu VBA
 - VBA – Programmierung mit Excel (125 Seiten)
 - <ftp://ftp.fernuni-hagen.de/pub/pdf/urz-broschueren/broschueren/b0129911.pdf>

Freie Universität Berlin – Bizer: Wirtschaftsinformatik – WS07/08 (Version vom 4. 1.08)

Die VBA Entwicklungsumgebung

The screenshot displays the Microsoft Visual Basic for Applications (VBA) development environment. The main window shows a VBA code editor with the following code:

```
Private Sub CommandButton1_Click()  
    x = InputBox("Bitte geben Sie einen Betrag in Euro an")  
    i = InputBox("Bitte geben Sie einen Zinssatz in Prozent e")  
    e = x * (100 + i) / 100  
    MsgBox "Ihr jetziges Guthaben von " & x & " Euro wird sic  
End Sub
```

A callout box labeled "Steuer-element" points to a button labeled "Programm starten" on an Excel spreadsheet. Another callout box labeled "Programm-code" points to the code in the editor. The Properties window on the right shows the properties for the CommandButton1, including its Name, Caption, and Font.

Property	Value
(Name)	CommandButto
Accelerator	
AutoLoad	False
AutoSize	False
BackColor	&H8000000
BackStyle	1 - fmBackStyle
Caption	Programm star
Enabled	True
Font	Arial

Freie Universität Berlin – Bizer: Wirtschaftsinformatik – WS07/08 (Version vom 4. 1.08)

3. Programmkonstrukte von VBA

1. Variablen und Datentypen
2. Operatoren
3. Ein- und Ausgabebefehle
4. Entscheidungsstrukturen
5. Schleifenstrukturen
6. Prozeduren und Funktionen
7. Arrays

Freie Universität Berlin – Bizer: Wirtschaftsinformatik – WS07/08 (Version vom 4. 1.08)

1. Variablen und Datentypen

Variablen sind Platzhalter, die zum Speichern von Werten verwendet werden; sie haben Namen und Datentypen.

- Technisch gesehen stellen Variablen symbolische Namen zum Ansprechen eines Bereichs des Hauptspeicher dar.
- Der Name einer Variablen kann frei gewählt werden.
- Der Datentyp einer Variable bestimmt
 - den Gültigkeitsbereich, also was für Werte die Variable aufnehmen kann.
 - wie die Variable im Hauptspeicher des Computers gespeichert wird.

Datentyp	Erklärung	Beispiel
String	Zeichenketten	Hallo
Integer, Long	Ganzzahl	12345
Single, Double	Gleitkommazahl	243,657
Boolean	Wahrheitswert	Wahr, Falsch
Variant	Nimmt Daten jedes Typs auf. Wird von VBA standardmäßig verwendet. Vermeiden, da langsame Verarbeitung.	

Freie Universität Berlin – Bizer: Wirtschaftsinformatik – WS07/08 (Version vom 4. 1.08)

Variablen und Datentypen

■ Deklaration

■ **Syntax:** Dim [Variablenname] as [Datentyp]

■ Beispiele:

- Dim a As Integer
- Dim a, b, c As String

■ Wertzuweisung

■ Werte werden Variablen mit einem einfachen Gleichheitszeichen zugewiesen.

■ **Syntax:** [Variablenname] = [Wert]

■ Beispiele:

- a = 1
- ausgabe = "Ich bin ein String"

■ **Zur Beschleunigung des Programmablaufs sollten Variablen in VBA immer vor ihrer Verwendung deklariert werden.**

Freie Universität Berlin – Bizer: Wirtschaftsinformatik – WS07/08 (Version vom 4. 1.08)

Beispiel: Wertzuweisung und Gültigkeitsbereiche

```
Dim a, zahl As Integer
Dim zahl2 As Double
Dim jn, d As Boolean
Dim c, assage As String

a = 124
d = 4
zahl = 123.4
zahl2 = 34.5
jn = False
assage = "Hallo"
c = assage
a = assage
jn = zahl2
```

■ **An welchen Stellen im Programmcode werden Variablen Werte außerhalb ihres Gültigkeitsbereich zugewiesen?**

■ **Nicht zulässig sind:**

- d = 4; aber d ist Boolean
- zahl = 123.4; aber zahl ist Integer
- a = assage; aber a ist Integer
- jn = zahl2; aber zahl2 ist Double und jn Boolean.

Freie Universität Berlin – Bizer: Wirtschaftsinformatik – WS07/08 (Version vom 4. 1.08)

2. Operatoren

- Addition: + Strings verbinden: &
- Subtraktion: -
- Multiplikation: *
- Division: / Ganzzahlige Division: \
- Potenzierung : ^ Rest der ganzzahligen Division: mod
- Klammerung : ()

- Syntax: [Variablenname][Operator][Variablenname oder Wert]

■ Beispiele:

```
Dim a, b, c, d As Integer
a = 1
b = 2
a = a * 3 + b
b = (a + b) / 3,5
c = (a + b) \ 3
d = (a + b) mod 3
```

```
Dim vorname, nachname, anzeige As String
vorname = "Chris"
nachname = "Bizer"
anzeige = "Guten morgen" &
         vorname & " " &
         nachname & "!"
```

Freie Universität Berlin – Bizer: Wirtschaftsinformatik – WS07/08 (Version vom 4. 1.08)

3. Ein- und Ausgabebefehle

1. Ein- und Ausgaben mittels Popup-Fenstern

- Einlesen einer Variablen
 - Syntax: [Variablenname] = Inputbox([Text der Eingabeaufforderung])
 - Beispiel: x = Inputbox("Bitte geben Sie eine Zahl ein:")
- Ausgeben einer Variablen
 - Syntax: MsgBox [Text der ausgegeben werden soll]
 - Beispiel: MsgBox "Ihr Guthaben beträgt " & guthaben

2. Ein- und Ausgaben direkt in Excel Arbeitsblatt

- Einlesen einer Variablen
 - Syntax: [Var] = Worksheets([Tabellenname]).Range([Zellenname]).Value
 - Beispiel: x = Worksheets("Tabelle1").Range("D2").Value
- Ausgeben einer Variablen
 - Syntax: Worksheets([Tabellenname]).Range([Zellenname]).Value = [Var]
 - Beispiel: Worksheets("Tabelle1").Range("D2").Value = x

3. Ein-Ausgaben mittels Formularen (UserForms)

Freie Universität Berlin – Bizer: Wirtschaftsinformatik – WS07/08 (Version vom 4. 1.08)

4. Entscheidungsstrukturen

- Die Ausführung von Anweisungen kann an Bedingungen geknüpft werden.
 - Ist die Bedingung WAHR, wird der Then-Anweisungsblock ausgeführt.
 - Ist die Bedingung FALSCH, wird der Else-Anweisungsblock ausgeführt.
- Syntax:

```
if [Bedingung] Then
  [Anweisung oder Anweisungsblock]
Else
  [Anweisung oder Anweisungsblock]
End If
```

- Beispiele:

```
if a > 6 Then
  a = a - 1
Else
  a = a - 1
  b = a + 1
End If
```

```
if a = b Then
  b = 0
  c = 5
End If
```

```
if stunde < 20 Then
  MsgBox "Guten Tag"
Else
  MsgBox "Gute Nacht"
End If
```

Freie Universität Berlin – Bizer: Wirtschaftsinformatik – WS07/08 (Version vom 4. 1.08)

Vergleichsoperatoren und logische Operatoren

- Vergleichsoperatoren

Operator	Bedeutung	Operator	Bedeutung
=	Ist gleich	>=	Größer oder gleich; mindestens
>	Größer als	<=	Kleiner oder gleich; höchstens
<	Kleiner als	<>	Ungleich

- logische Operatoren

- mit logischen Operatoren lassen sich zusammengesetzte Bedingungen formulieren.
- Logisches Und:
 - If [Bedingung1] **And** [Bedingung2] Then ...
 - Beide Bedingungen müssen erfüllt sein.
- Logischer Oder:
 - If [Bedingung1] **Or** [Bedingung2] Then ...
 - Eine der Bedingungen oder beide Bedingungen müssen erfüllt sein.
- Exklusives Oder:
 - If [Bedingung1] **Xor** [Bedingung2] Then ...
 - Eine der Bedingungen, aber nicht beide Bedingungen müssen erfüllt sein.

Freie Universität Berlin – Bizer: Wirtschaftsinformatik – WS07/08 (Version vom 4. 1.08)

Beispiel: Entscheidungsstrukturen

```
a = 4
b = 10
If a < b Then
  If a <> 4 Then
    a = a + 2
  Else
    a = a - 2
  End If
Else
  If b > 5 Then
    b = a
  End If
End If
MsgBox "Ergebnis: a=" & a
& " b=" & b
```

- Welche Werte erscheinen in der MessageBox?

Tischtest

a	b
4	10
2	
2	10

Freie Universität Berlin – Bizer: Wirtschaftsinformatik – WS07/08 (Version vom 4. 1.08)

Beispiel: Entscheidungsstrukturen

```
a = 20
b = 6
If a < b Then
  If a <> 4 Then
    a = a + 2
  Else
    a = a - 2
  End If
Else
  If b > 5 Then
    b = a
  End If
End If
MsgBox "Ergebnis: a=" & a
& " b=" & b
```

- Welche Werte erscheinen in der MessageBox?

Tischtest

a	b
20	6
	20
20	20

Freie Universität Berlin – Bizer: Wirtschaftsinformatik – WS07/08 (Version vom 4. 1.08)

Elseif

- Mit `ElseIf` lassen sich mehrere Bedingungen nacheinander prüfen.

- **Syntax:**

```
if [Bedingung1] Then
  [Anweisung oder Anweisungsblock]
ElseIf [Bedingung2] Then
  [Anweisung oder Anweisungsblock]
Else
  [Anweisung oder Anweisungsblock]
End If
```

- **Beispiele:**

```
If a < b Then
  a = a + 1
ElseIf a = b Then
  a = a + 2
  b = b - 2
Else
  a = a - 1
End If
```

1. Bedingung

2. Bedingung wird geprüft, wenn
1. Bedingung nicht erfüllt ist.

Wird ausgeführt, wenn 1. und 2.
Bedingung nicht erfüllt sind.

Freie Universität Berlin – Bizer: Wirtschaftsinformatik – WS07/08 (Version vom 4. 1.08)

Beispiel: Elseif

```
a=3
b=10
c=20

If c - 2 <= a Then
  a = a + 2
ElseIf c > a Then
  c = 3
Else
  c = a + b
End If

If c < a and c < b Then
  a = c + b
End If

If c = a or b < a Then
  b = a + b
End If
```

- **Welches Ergebnis liefert nebenstehendes Programm?**

Tischtest

a	b	c
3	10	20
		3
	13	
3	13	3

Freie Universität Berlin – Bizer: Wirtschaftsinformatik – WS07/08 (Version vom 4. 1.08)

5. Schleifenstrukturen

- Schleifen ermöglichen es, Anweisungen oder Anweisungsblöcke mehrfach ausführen zu lassen.
- Wir behandeln folgende Schleifentypen:

Schleifentypen	Erklärung
For .. Next	Schleife mit feststehender Anzahl an Durchläufen.
While [Bedingung] .. Wend	Schleife wird durchlaufen, solange die Abbruch-Bedingung wahr ist.

- **Gemeinsame Elemente aller Schleifentypen**
 - Initialisierung einer Zählervariablen
 - Zählfunktion
 - Abbruchbedingung

Freie Universität Berlin – Bizer: Wirtschaftsinformatik – WS07/08 (Version vom 4. 1.08)

For .. Next Schleife

- Schleife mit einer feststehenden Anzahl von Durchläufen.
- Bei jedem Durchlauf wird der Schleifenzähler um 1 erhöht.
- Syntax

```
For [Schleifenzähler] = [Startwert] to [Endwert]
    [Anweisung oder Anweisungsblock]
Next
```

- Beispiel

```
b = 0
For a = 1 To 3
    b = b + a
Next
MsgBox b
```

Tischtest zum Beispiel

a	b
	0
1	1
2	3
3	6
4	

- **Vorgehen von Visual Basic**
 1. b=0 setzen.
 2. Zähler a auf den Startwert setzen.
 3. Es wird geprüft, ob der Zähler a größer als der Endwert ist. In diesem Fall wird die Schleife beendet.
 4. $b = b + a$ wird berechnet.
 5. Der Zähler a wird um 1 erhöht.
 6. Sprung zurück zu Schritt 3

Freie Universität Berlin – Bizer: Wirtschaftsinformatik – WS07/08 (Version vom 4. 1.08)

For .. Next Schleife mit STEP Anweisung

- Normalerweise wird der Schleifenzähler immer um 1 erhöht.
- Mit STEP lassen sich auch andere Schrittweiten definieren.
z.B. For a = 1 To 10 Step 2

```
Dim a, b as Integer
b = 4
For a = 12 To 1 Step -3
    If a > 6 Then
        b = b + 2
    Else
        b = b - 1
    End If
Next a
MsgBox "a=" & a & " b=" & b
```

Tischtest

a	b
	4
12	
	6
9	
	8
6	
	7
3	
	6
0	

Freie Universität Berlin – Bizer: Wirtschaftsinformatik – WS07/08 (Version vom 4. 1.08)

While .. Wend Schleife

- Bei der While .. Wend Schleife ist die Anzahl der Durchläufe von einer Bedingung abhängig. Die Schleife wird so lange durchlaufen, wie die Bedingung wahr ist.
- Syntax

```
[Initialisierung des Zählers]
While [Bedingung]
    [Anweisung oder Anweisungsblock]
    [Zählfunktion]
Loop
```

- Beispiel

```
a = 1
b = 0
While a < 3
    a = a + 1
    b = b + a
Wend
```

Tischtest zum Beispiel

a	b
1	0
2	
	2
3	
	5

- Vorgehen von Visual Basic

1. Bedingung wird geprüft.
2. Wenn Bedingung WAHR, wird Anweisungsblock ausgeführt.
3. Sprung zu Schritt 1.

- Durchläufe: Die Schleife wird einmal oder mehrmals durchlaufen.

Freie Universität Berlin – Bizer: Wirtschaftsinformatik – WS07/08 (Version vom 4. 1.08)

Beispiel: While .. Wend Schleife

```
Dim a, b as Integer
a = 4
For b = 1 To 3
  a = a + 1
  While a / 2 > b
    a = a - 2
  Wend
Next b
```

- Welches Ergebnis liefert ein Tischtest des nebenstehenden Codes?

Tischtest

a	b
4	
	1
5	
3	
1	
	2
2	
	3
3	
	4
3	4