



BEUTH HOCHSCHULE FÜR TECHNIK BERLIN
University of Applied Sciences

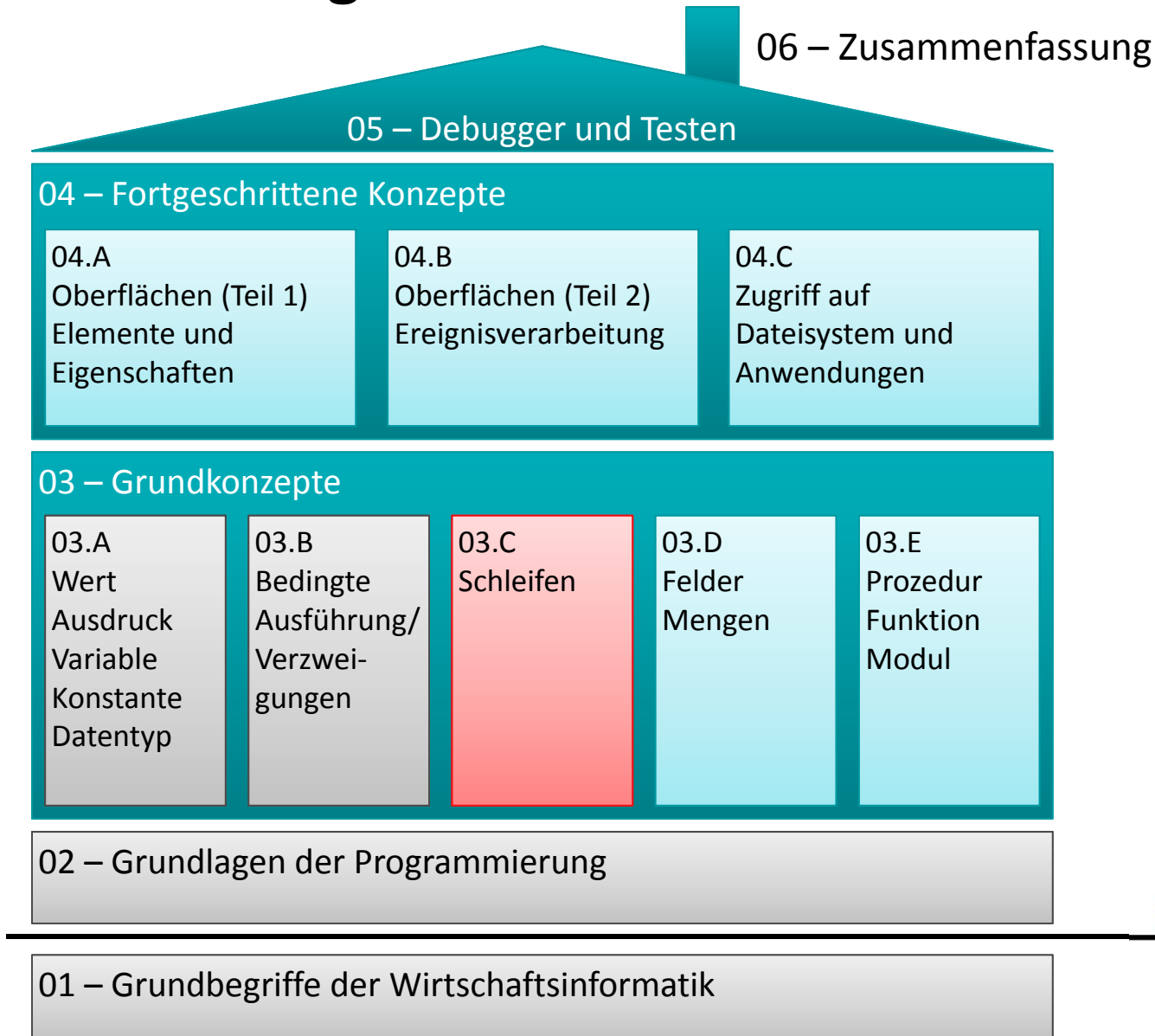
Wirtschaftsinformatik 1

LE 05 – Schleifen

Prof. Dr. Thomas Off

<http://www.ThomasOff.de/lehre/beuth/wi1>

Einordnung





Inhalt

Rückblick

Überblick

- Arten von Schleifen
- Unterschiede im Überblick

Implementierung von Schleifen mit VBA

- Vorprüfende Schleifen
- Nachprüfende Schleifen
- Zählerschleifen
- Weitere Möglichkeiten

Abschluss und Ausblick



Inhalt

Rückblick

Überblick

- Arten von Schleifen
- Unterschiede im Überblick

Implementierung von Schleifen mit VBA

- Vorprüfende Schleifen
- Nachprüfende Schleifen
- Zählerschleifen
- Weitere Möglichkeiten

Abschluss und Ausblick

Rückblick



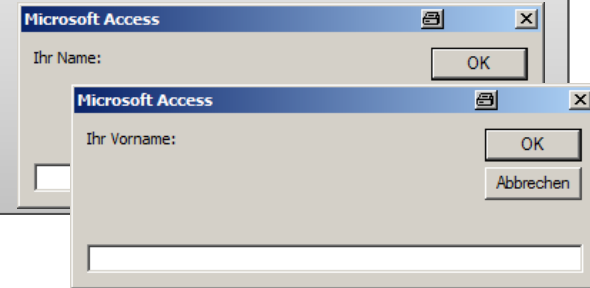
Rückblick

Eingabe und Ausgabe



– Eingabe mittels Dialog

```
' Generelle Syntax:  
' Let <Variable-vom-Typ-String> = InputBox("<Meldungstext>")  
' Beispiele  
Let strName = InputBox("Ihr Name:")  
Let strVorname = InputBox("Ihr Vorname:")  
Let bytAlter = Val(InputBox("Ihr Alter:"))
```

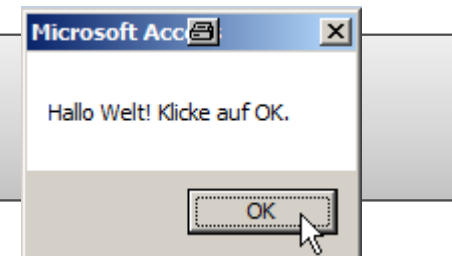


– Ausgabe im Debug-Bereich

```
' Beispiel  
Debug.Print "Hallo Welt!"
```

– Ausgabe im Meldungsfenster

```
' Generelle Syntax: MsgBox ("<Meldungstext>")  
' Beispiel:  
MsgBox ("Hallo Welt! Klicke auf OK.")
```



Rückblick



Ausgangspunkt

- Beschränkung bisheriger Programme auf linearen Programmablauf
- Notwendigkeit zu Verzweigungen

Konzepte

- Arten von Verzweigungen, z.B.

Bedingung ist wahr	
Ja	Nein
Anweisung 4/ Anweisungsblock 4	Anw 5/ Anwb 5

Variable A			
Wert 1	Wert 2	Wert 3	Sonst
Anw4/ Anwb4	Anw5/ Anwb5	Anw6/ Anwb6	A7/ Ab7

- Formulierung von Bedingungen für Verzweigungen
 - als Ausdrücke, in der Regel mit Vergleichsoperator
 - mit Wahrheitswert als Ergebnis der Auswertung
 - Einsatz logischer Operationen

Rückblick

Implementierung in VBA



- Einfach Verzweigungen in Form von If-Then-Else-Elseif und End If
- Mehrfach Verzweigungen
 - Elself-Erweiterung
 - Select-Case-Anweisung

```
' Beispiel
Select Case bytZahl
Case 2
  Debug.Print "Zwei"
Case 3
  Debug.Print "Drei"
Case Else
  Debug.Print "Etwas anderes"
End Select
```

```
' Beispiel
If intZahl > 4 Then
  Debug.Print "Größer 4"
End If
```

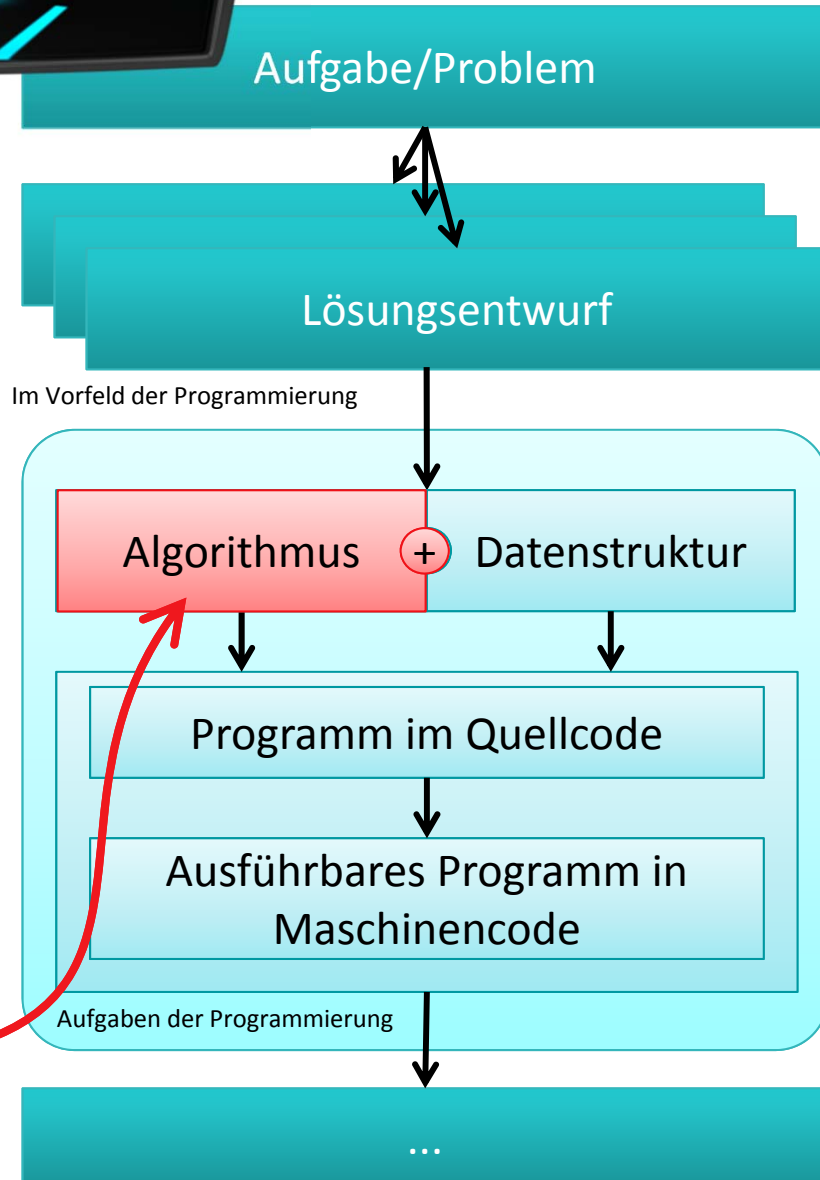
```
' Beispiel
If intZahl > 4 Then
  Debug.Print "Größer 4"
Else
  Debug.Print "Kleiner 5"
End If
```

```
' Beispiel
If intZahl > 4 Then
  Debug.Print "Größer 4"
ElseIf intZahl = 4 Then
  Debug.Print "Gleich 4"
Else
  Debug.Print "Kleiner 4"
End If
```


Rückblick

Programmierung als stufenweise Umsetzung

- von Algorithmus und
- Datenstruktur
- in Quellcode einer Programmiersprache und
- der Überführung in ein ausführbares Programm im Maschinencode



Heute: Schleifen



Inhalt

Rückblick

Überblick

- Arten von Schleifen
- Unterschiede im Überblick

Implementierung von Schleifen mit VBA

- Vorprüfende Schleifen
- Nachprüfende Schleifen
- Zählerschleifen
- Weitere Möglichkeiten

Abschluss und Ausblick





Inhalt

Rückblick

Überblick

- Arten von Schleifen
- Unterschiede im Überblick

Implementierung von Schleifen mit VBA

- Vorprüfende Schleifen
- Nachprüfende Schleifen
- Zählerschleifen
- Weitere Möglichkeiten

Abschluss und Ausblick



Inhalt

Rückblick

Überblick

- Arten von Schleifen
- Unterschiede im Überblick

Implementierung von Schleifen mit VBA

- Vorprüfende Schleifen
- Nachprüfende Schleifen
- Zählerschleifen
- Weitere Möglichkeiten

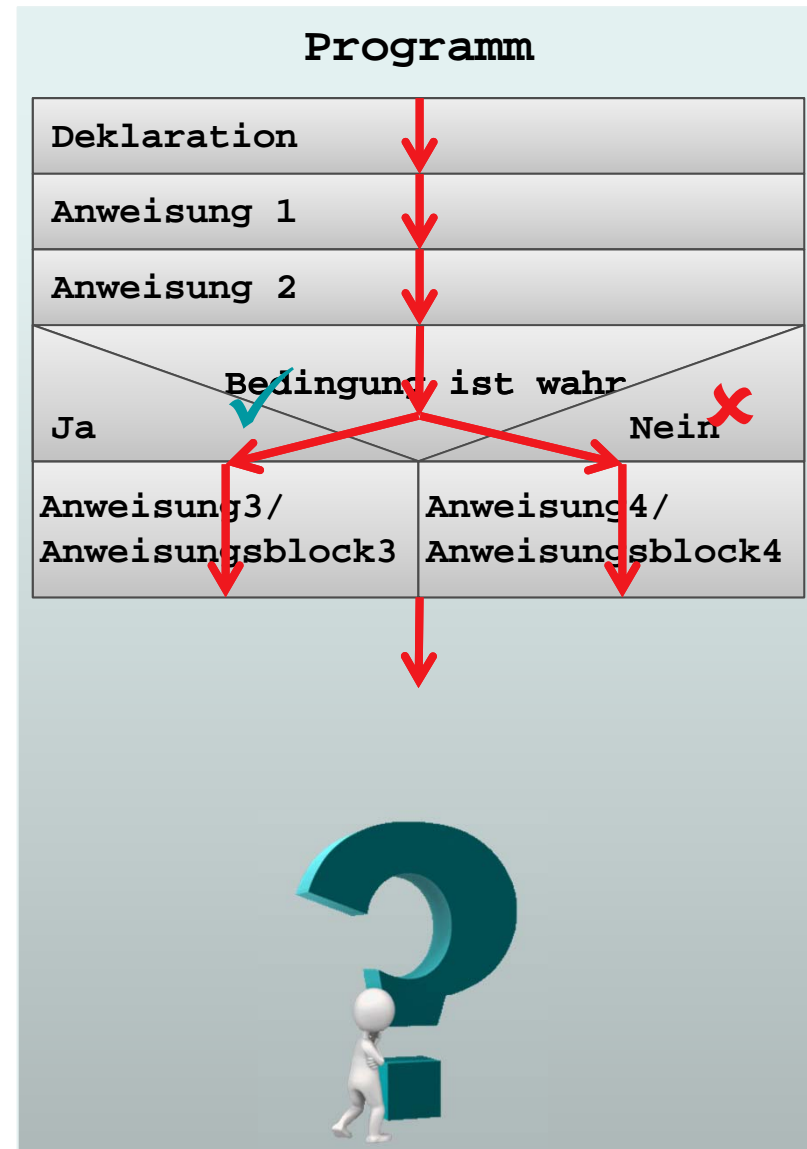
Abschluss und Ausblick

Arten von Schleifen

Bisherige Programme

- mit linearem Ablauf
- und mit Verzweigungen

Was tun wenn eine Anweisung mehrfach ausgeführt werden soll?





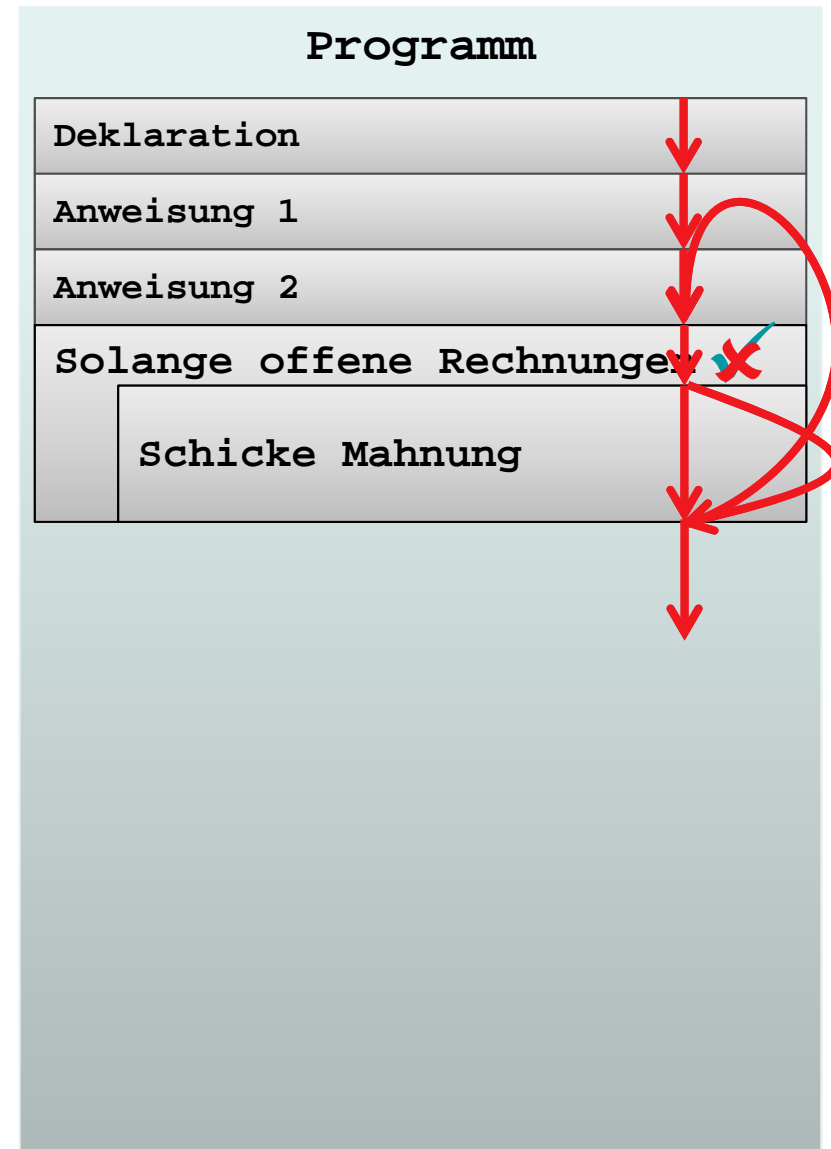
Arten von Schleifen

Bisherige Programme

- mit linearem Ablauf
- und mit Verzweigungen

Was tun wenn eine Anweisung mehrfach ausgeführt werden soll?

- solange es noch offene Rechnungen ohne Mahnung gibt, schicke eine Mahnung an den Rechnungsempfänger
- ..





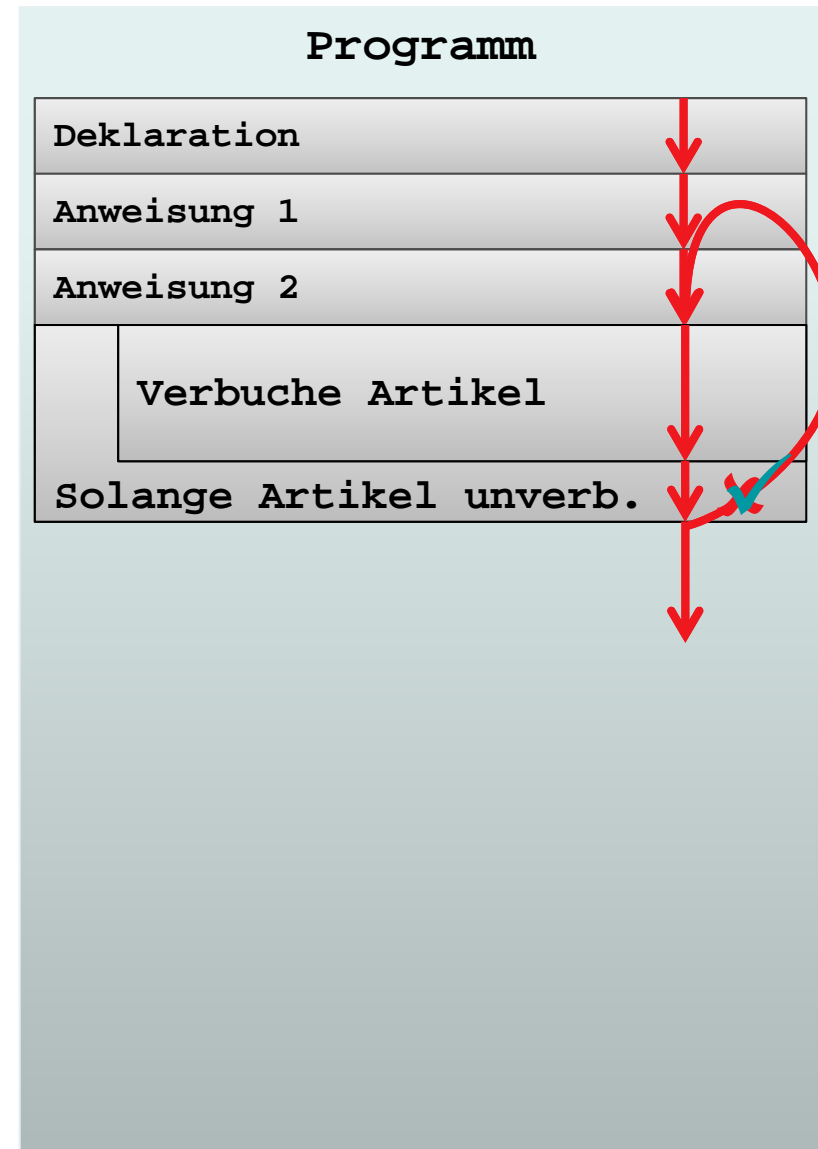
Arten von Schleifen

Bisherige Programme

- mit linearem Ablauf
- und mit Verzweigungen

Was tun wenn eine Anweisung mehrfach ausgeführt werden soll?

- ...
- verbuche einen Artikel aus der Bestellung, wiederhole solange es noch unverbuchte Artikel der Bestellung gibt
- ...





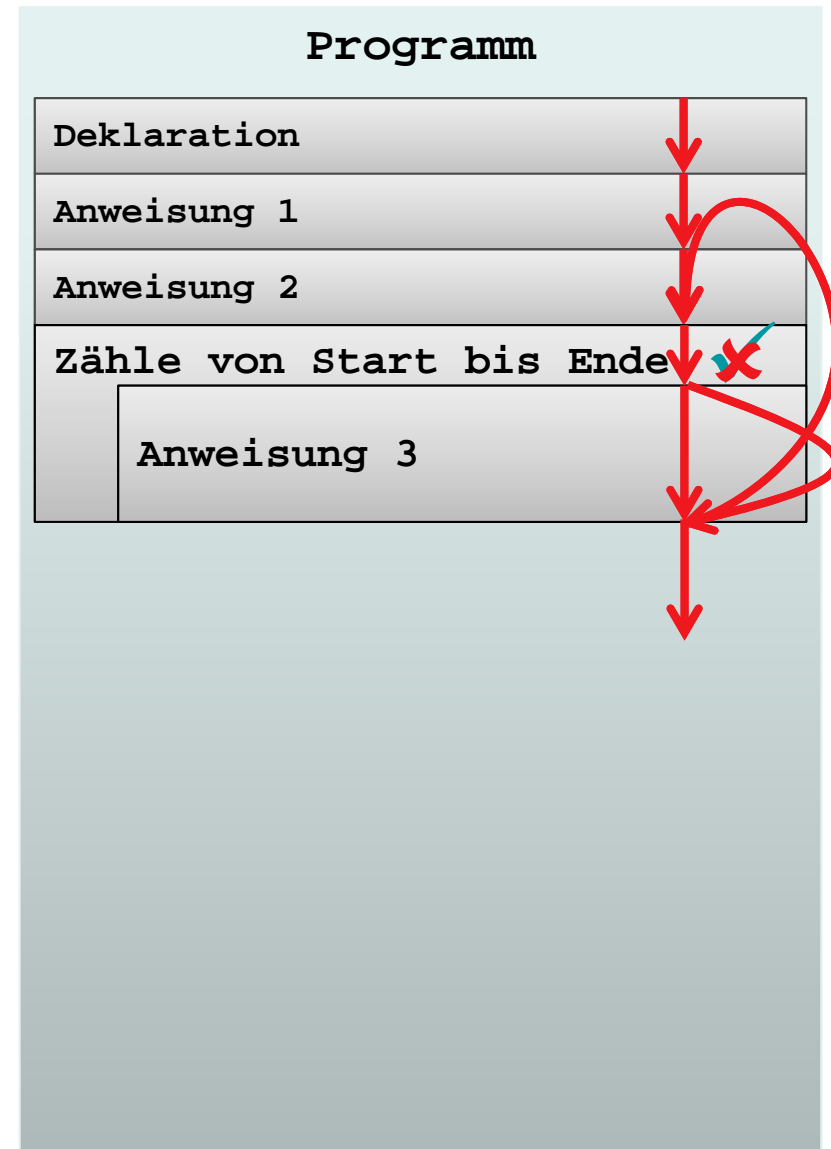
Arten von Schleifen

Bisherige Programme

- mit linearem Ablauf
- und mit Verzweigungen

Was tun wenn eine Anweisung mehrfach ausgeführt werden soll?

- ...
- Wiederhole x Mal eine *<Anweisung 3>*

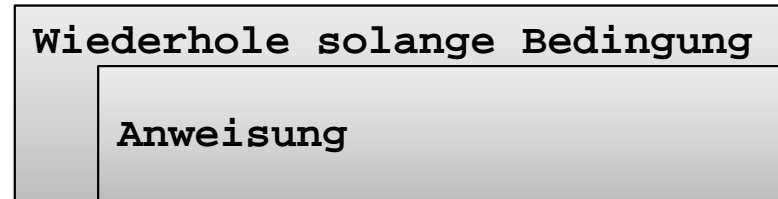


Unterschiede im Überblick



Vorprüfende/Kopfgesteuerte

- erst Bedingung prüfen
- dann ggf. Anweisung ausführen
- anschließend Wiederholung der Prüfung usw.



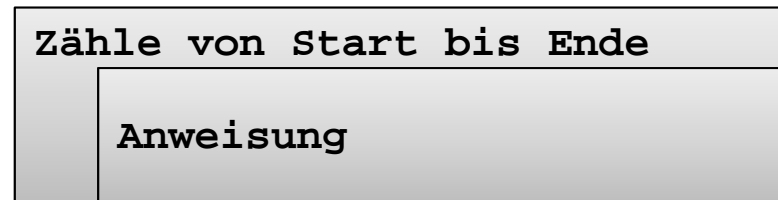
Nachprüfende/Fußgesteuerte

- erst Anweisung ausgeführt
- dann Bedingung prüfen
- anschließend Wiederholung der Anweisungsausführung usw.



Zählerschleifen

- Vorher bekannte Anzahl von Wiederholungen
- Anzahl gesteuert über Start und Ende
- Ausführung der Anweisung solange Anzahl Wiederholungen das Ende noch überschritten hat





Inhalt

Rückblick

Überblick

- Arten von Schleifen
- Unterschiede im Überblick

Implementierung von Schleifen mit VBA

- Vorprüfende Schleifen
- Nachprüfende Schleifen
- Zählerschleifen
- Weitere Möglichkeiten

Abschluss und Ausblick





Inhalt

Rückblick

Überblick

- Arten von Schleifen
- Unterschiede im Überblick

Implementierung von Schleifen mit VBA

- Vorprüfende Schleifen
- Nachprüfende Schleifen
- Zählerschleifen
- Weitere Möglichkeiten

Abschluss und Ausblick



Inhalt

Rückblick

Überblick

- Arten von Schleifen
- Unterschiede im Überblick

Implementierung von Schleifen mit VBA

- Vorprüfende Schleifen
- Nachprüfende Schleifen
- Zählerschleifen
- Weitere Möglichkeiten

Abschluss und Ausblick

Vorprüfende Schleife

Schlüsselbegriffe

- Variante 1
- Variante 2



Vorprüfende Schleife



Schlüsselbegriffe

– Variante 1

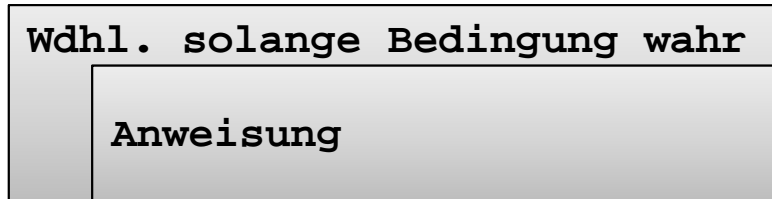
`Do While <Bedingung>`

`<Anweisung(en)>`

`Loop`

– Variante 2

- ...





Vorprüfende Schleife

Schlüsselbegriffe

- Variante 1

Do While *<Bedingung>*

<Anweisung(en)>

Loop

Wdh1. solange Bedingung wahr

Anweisung

- Variante 2

- ...



Vorprüfende Schleife

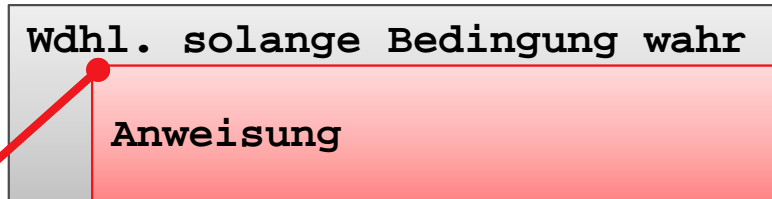
Schlüsselbegriffe

- Variante 1

Do While *<Bedingung>*

<Anweisung(en)>

Loop



- Variante 2

- ...



Vorprüfende Schleife

Schlüsselbegriffe

- Variante 1

`Do While <Bedingung>`

`<Anweisung(en)>`

Loop

Wdh1. solange Bedingung wahr

Anweisung

- Variante 2

- ...

Vorprüfende Schleife



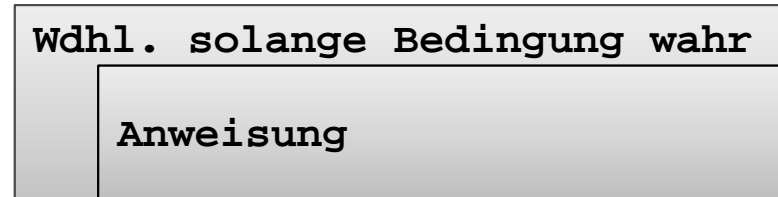
Schlüsselbegriffe

– Variante 1

Do While *<Bedingung>*

<Anweisung(en)>

Loop

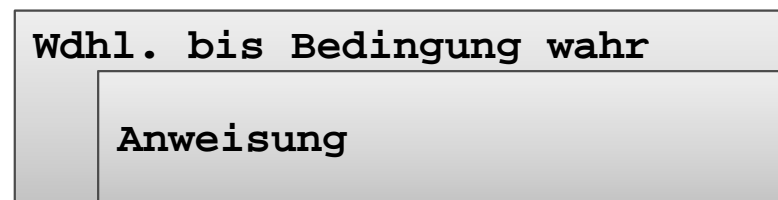


– Variante 2

Do Until *<Bedingung>*

<Anweisung(en)>

Loop



Vorprüfende Schleife

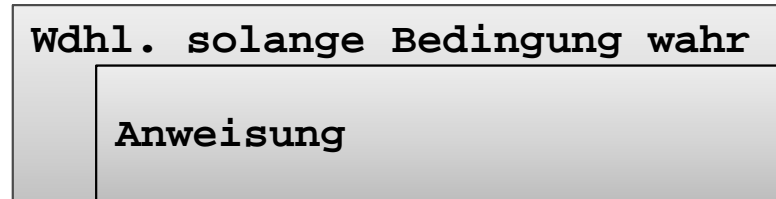


Schlüsselbegriffe

- Variante 1

`Do While <Bedingung>`
`<Anweisung(en)>`

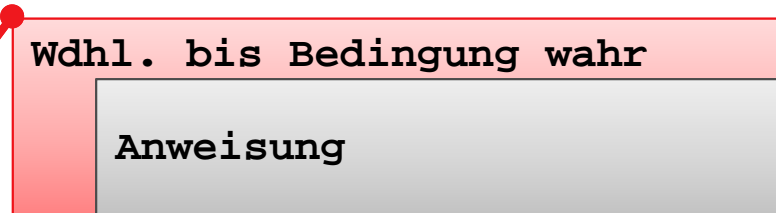
Loop



- Variante 2

`Do Until <Bedingung>`
`<Anweisung(en)>`

Loop





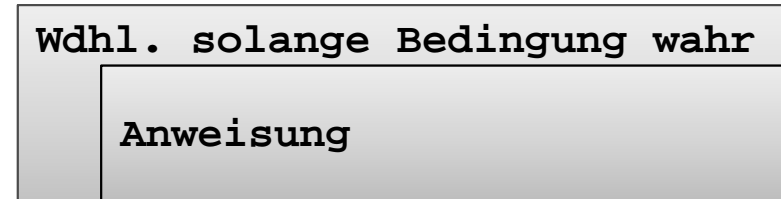
Vorprüfende Schleife

Schlüsselbegriffe

- Variante 1

`Do While <Bedingung>`
`<Anweisung(en)>`

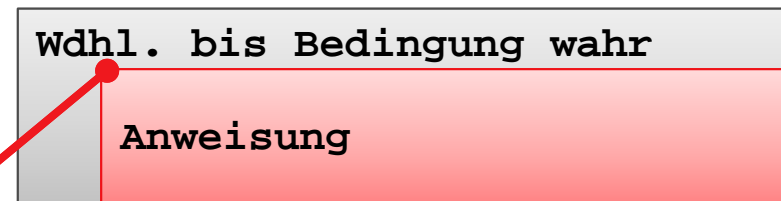
Loop



- Variante 2

`Do Until <Bedingung>`
`<Anweisung(en)>`

Loop





Vorprüfende Schleife

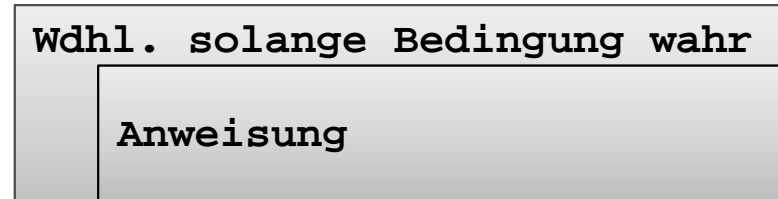
Schlüsselbegriffe

– Variante 1

Do While *<Bedingung>*

<Anweisung(en)>

Loop

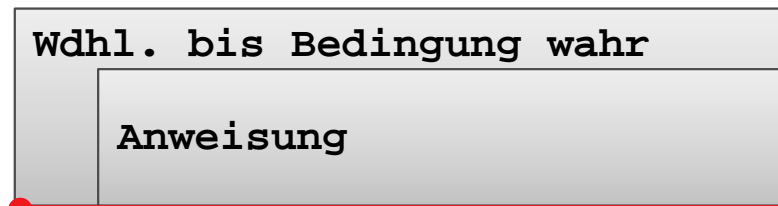


– Variante 2

Do Until *<Bedingung>*

<Anweisung(en)>

Loop



Vorprüfende Schleife



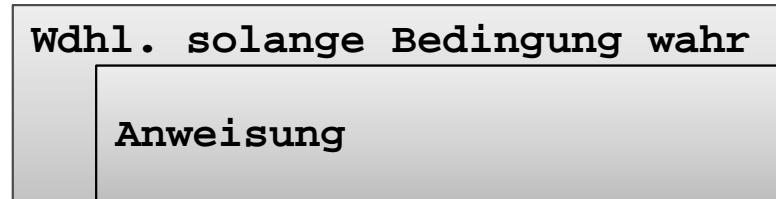
Schlüsselbegriffe

– Variante 1

Do While *<Bedingung>*

<Anweisung(en)>

Loop

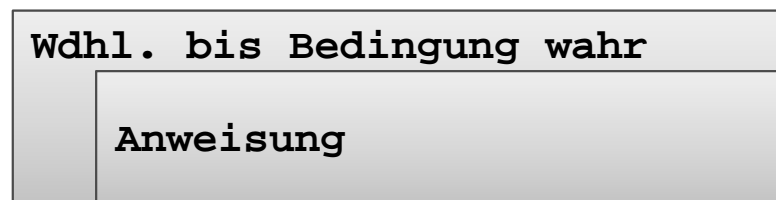


– Variante 2

Do Until *<Bedingung>*

<Anweisung(en)>

Loop





Vorprüfende Schleife

Generelle Syntax

– Variante 1

```
Do While <Bedingung>  
  <Anweisung(en)>  
Loop
```

– Variante 2

```
Do Until <Bedingung>  
  <Anweisung(en)>  
Loop
```

Beispiele

```
Dim i As Integer  
Let i = 0  
Do While i < 5  
  Let i = i + 1  
  Debug.Print i  
Loop
```

```
1  
2  
3  
4  
5
```

```
Dim j As Byte  
Let j = 0  
Do Until j > 4  
  Let j = j + 1  
  Debug.Print j  
Loop
```

```
1  
2  
3  
4  
5
```

Vorprüfende Schleife: Beispiel 05.00



Ziel

- Vorprüfende Schleife mit Bedingungen verwenden

Aufgabe

- Schreiben Sie ein Programm mit einer vorprüfenden Schleife, die
 - das die Zahlen von 1 bis 10 ausgibt
 - schrittweise die Summe dieser Zahlen bildet, indem
 - auf die bisherige Summe die nächste Zahl addiert wird und
 - das Ergebnis als neue Summe verwendet wird
 - nach der Schleife, die Gesamtsumme ausgibt



Vorprüfende Schleife: Beispiel 05.01



Ziel

- Vorprüfende Schleife mit Bedingungen verwenden

Aufgabe

- Schreiben Sie ein Programm mit einer vorprüfenden Schleife, die
 - das Datum aller verbliebenden Tage bis zur Klausur am 19.06. ausgibt
 - die Summe der Anzahl dieser Tage bildet und die Summe im Direktbereich ausgibt.



Vorprüfende Schleife: Beispiel 05.02



Ziel

- Vorprüfende Schleife verwenden

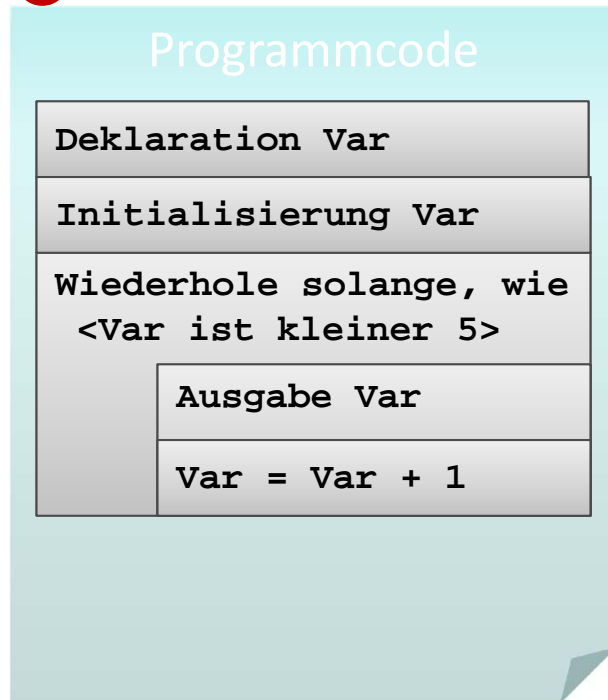
Aufgabe

- Teil 1: Vorprüfende Schleife, die von 1 bis 5 die Zahlen im Direktbereich ausgibt **0**
- Teil 2: Vorprüfende Schleife, die von 5 bis 1 die Zahlen im Direktbereich ausgibt
- Teil 3: Vorprüfende Schleife, die von 1 bis 10 die ungeraden Zahlen im Direktbereich ausgibt
- Teil 4: Vorprüfende Schleife, die von einem eingegebenen Datum die verbleibenden Tage bis Weihnachten zählt
- Teil 5: Vorprüfende Schleife, die die Summe der Zahlen 1, 2, 3, 4, ... usw. bildet, und solange läuft bis vorher eingegebene Zahl erreicht ist.



Vorprüfende Schleife: Beispiel 05.02

1



2

```
Sub Bsp24()  
  
Dim i As Integer  
i = 1  
  
Do While i < 5  
    Debug.Print i  
    Let i = i + 1  
Loop  
  
End Sub
```

- **Überführen in Programmcode**
 - Deklaration einer Variablen, die für die Bedingung verwendet wird
 - Schleife, die von 1 bis 5 zählt und eine Ausgabeanweisung enthält

- **Nutzen von Do While Loop**
- **Was passiert beim Ausführen des Programmcodes mit der Variable i?**

Vorprüfende Schleife: Beispiel 05.02

→ 3 → 4 →

```
Option Compare Database
Option Explicit

Sub Bsp24 ()

    Dim i As Integer
    i = 1

    Do While i < 5
        Debug.Print i
        Let i = i + 1
    Loop

End Sub
```

- Was ist im Direktbereich sichtbar?
- Ausführen des Programmcodes liefert im Direktbereich schrittweise die Ausgabe der Zahlen von 1 bis 5

Vorprüfende Schleife: Beispiel 05.02



→ 5 Ziel

- Vorprüfende Schleife verwenden

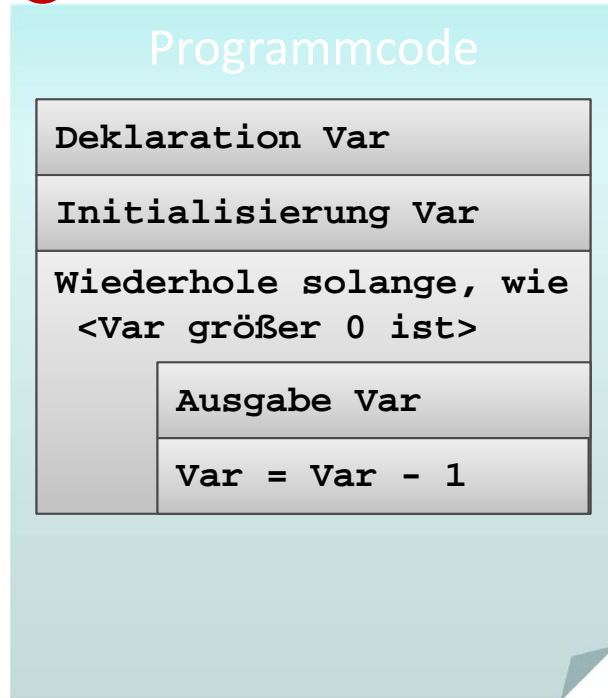
Aufgabe

- Teil 1: Vorprüfende Schleife, die von 1 bis 5 die Zahlen im Direktbereich ausgibt
- Teil 2: Vorprüfende Schleife, die von 5 bis 1 die Zahlen im Direktbereich ausgibt 0
- Teil 3: Vorprüfende Schleife, die von 1 bis 10 die ungeraden Zahlen im Direktbereich ausgibt
- Teil 4: Vorprüfende Schleife, die von einem eingegebenen Datum die verbleibenden Tage bis Weihnachten zählt
- Teil 5: Vorprüfende Schleife, die die Summe der Zahlen 1, 2, 3, 4, ... usw. bildet, und solange läuft bis vorher eingegebene Zahl erreicht ist.



Vorprüfende Schleife: Beispiel 05.02

1



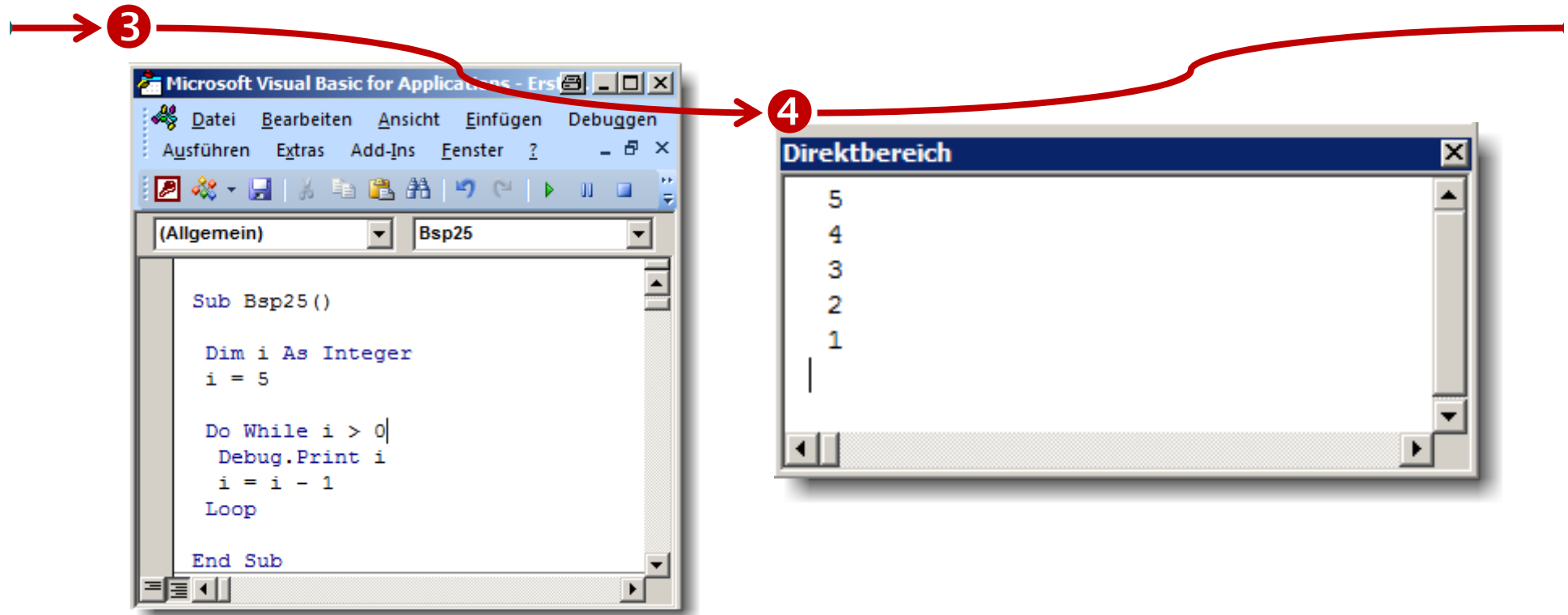
2

```
Sub Bsp25()  
  
Dim i As Integer  
i = 5  
  
Do While i > 0  
    Debug.Print i  
    i = i - 1  
Loop  
  
End Sub
```

- **Überführen in Programmcode**
 - Deklaration einer Variablen, die zum Zählen verwendet wird
 - Schleife, die von 5 bis 1 rückwärts zählt und eine Ausgabeanweisung enthält

- **Nutzung Do While/Loop**
- **Was passiert beim Ausführen des Programmcodes mit der Variable i?**

Vorprüfende Schleife: Beispiel 05.02



- Was ist im Direktbereich sichtbar?
- Ausführen des Programmcodes liefert im Direktbereich schrittweise die Ausgabe der Zahlen von 5 bis 1

Vorprüfende Schleife: Beispiel 05.02



→ 5 Ziel

- Vorprüfende Schleife verwenden

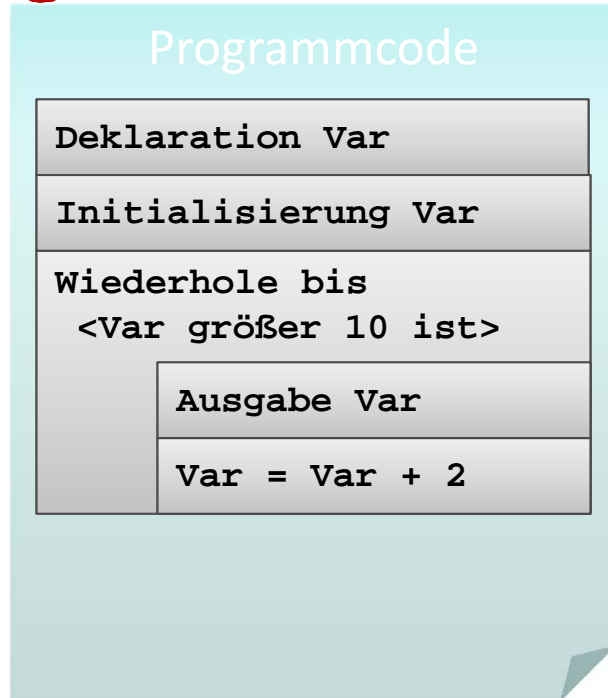
Aufgabe

- Teil 1: Vorprüfende Schleife, die von 1 bis 5 die Zahlen im Direktbereich ausgibt
- Teil 2: Vorprüfende Schleife, die von 5 bis 1 die Zahlen im Direktbereich ausgibt
- Teil 3: Vorprüfende Schleife, die von 1 bis 10 die ungeraden Zahlen im Direktbereich ausgibt 0
- Teil 4: Vorprüfende Schleife, die von einem eingegebenen Datum die verbleibenden Tage bis Weihnachten zählt
- Teil 5: Vorprüfende Schleife, die die Summe der Zahlen 1, 2, 3, 4, ... usw. bildet, und solange läuft bis vorher eingegebene Zahl erreicht ist.



Vorprüfende Schleife: Beispiel 05.02

1



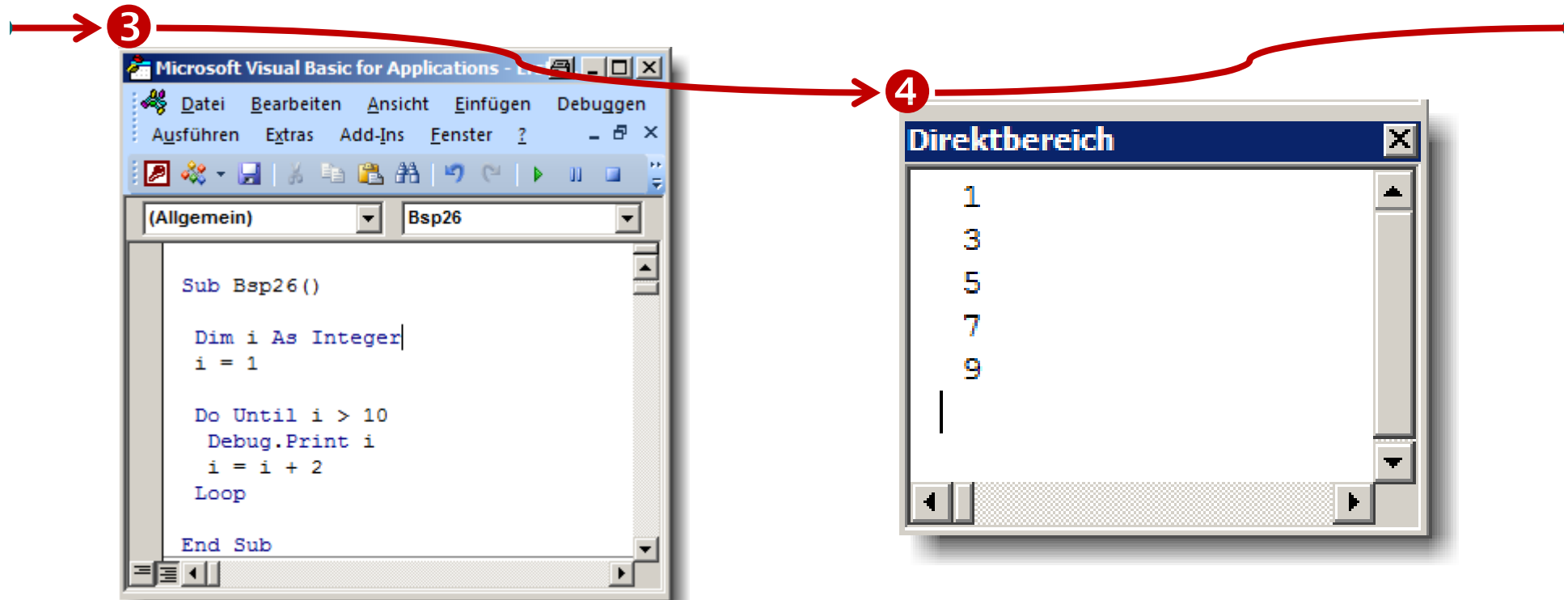
2

```
Sub Bsp26()  
  
Dim i As Integer  
i = 1  
  
Do Until i > 10  
    Debug.Print i  
    i = i + 2  
Loop  
  
End Sub
```

- **Überführen in Programmcode**
 - Deklaration einer Variablen, die zum Zählen verwendet wird
 - Schleife, die von 1 bis 10 in Zweierschritten zählt und eine Ausgabeanweisung enthält

- **Nutzung Do Until/Loop**
- **Was passiert beim Ausführen des Programmcodes mit der Variable i?**

Vorprüfende Schleife: Beispiel 05.02



- Was ist im Direktbereich sichtbar?
- Ausführen des Programmcodes liefert im Direktbereich schrittweise die Ausgabe der Zahlen 1, 3, 5, 7 und 9.

Vorprüfende Schleife: Beispiel 05.02



→ 5 Ziel

- Vorprüfende Schleife verwenden

Aufgabe

- Teil 1: Vorprüfende Schleife, die von 1 bis 5 die Zahlen im Direktbereich ausgibt
- Teil 2: Vorprüfende Schleife, die von 5 bis 1 die Zahlen im Direktbereich ausgibt
- Teil 3: Vorprüfende Schleife, die von 1 bis 10 die ungeraden Zahlen im Direktbereich ausgibt
- Teil 4: Vorprüfende Schleife, die von einem eingegebenen Datum die 0 verbleibenden Tage bis Weihnachten zählt
- Teil 5: Vorprüfende Schleife, die die Summe der Zahlen 1, 2, 3, 4, ... usw. bildet, und solange läuft bis vorher eingegebene Zahl erreicht ist.



Vorprüfende Schleife: Beispiel 05.02

1

2

Programmcode

Deklaration Var

Initialisierung Var

Wiederhole bis
<Var ist Weihnachten>

Ausgabe Var

Var = Var + 1 Tag

```
Sub Bsp27()  
  Dim i As Integer, d As Date  
  Let i = 0  
  Let d = DateValue(InputBox("Datum: "))  
  
  Do Until d >= #12/24/2011#  
    Let i = i + 1  
    Let d = d + 1  
    Debug.Print d  
  Loop  
  
  Debug.Print "Tage: " & i  
  
End Sub
```

- **Überführen in Programmcode**

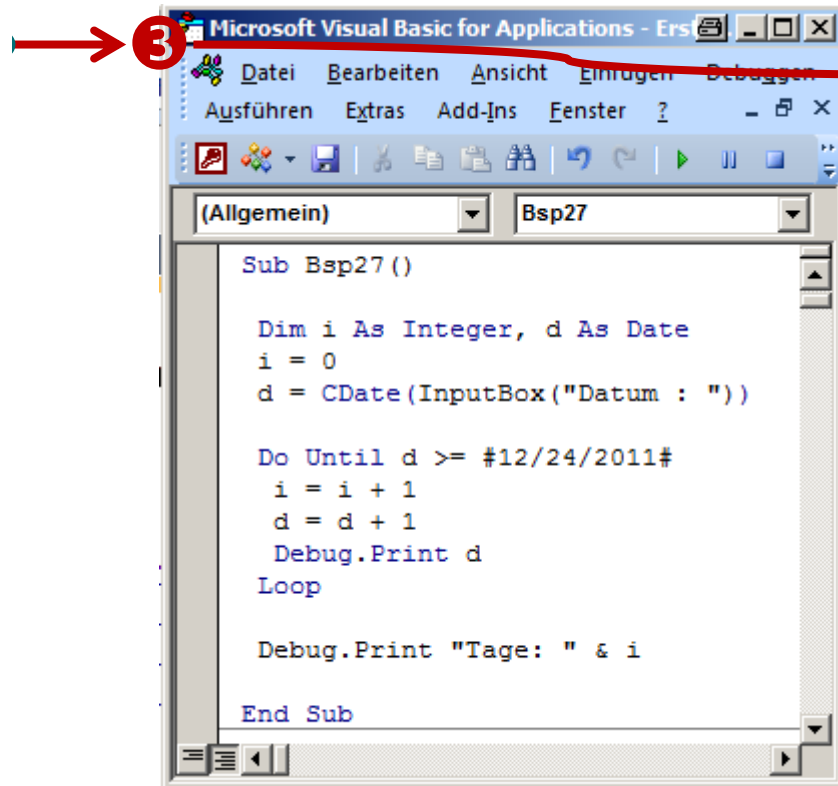
- Deklaration einer Bedingung, die das Datum prüft ("Ist Weihnachten?")
- Schleife, die auf das eingegebene Datum 1 Tag addiert und eine Ausgabeanweisung enthält

- Nach Schleifenende Ausgabe der Tage bis Weihnachten

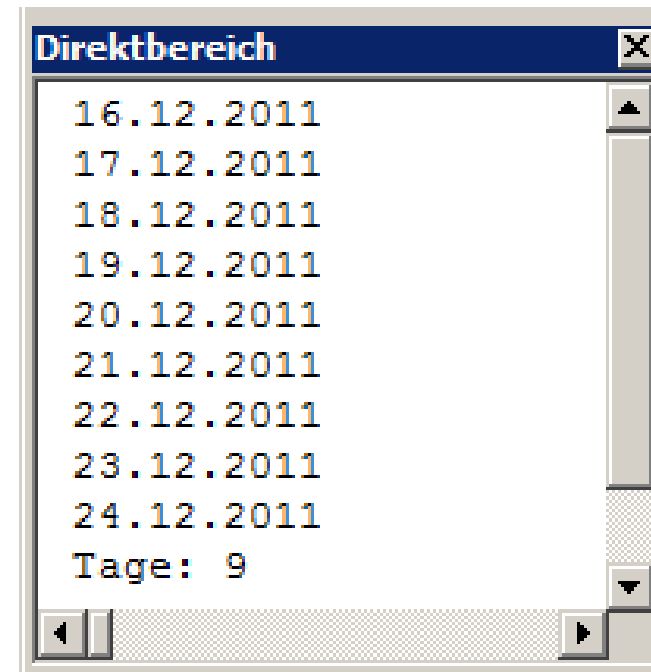
- **Nutzung Do Until/Loop**

- **Was passiert beim Ausführen des Programmcodes?**

Vorprüfende Schleife: Beispiel 05.02



4



- Was ist im Direktbereich sichtbar?
- Ausführen des Programmcodes liefert im Direktbereich schrittweise die Ausgabe der Zahlen von 5 bis 1

Vorprüfende Schleife: Beispiel 05.02



→ 5 Ziel

- Vorprüfende Schleife verwenden

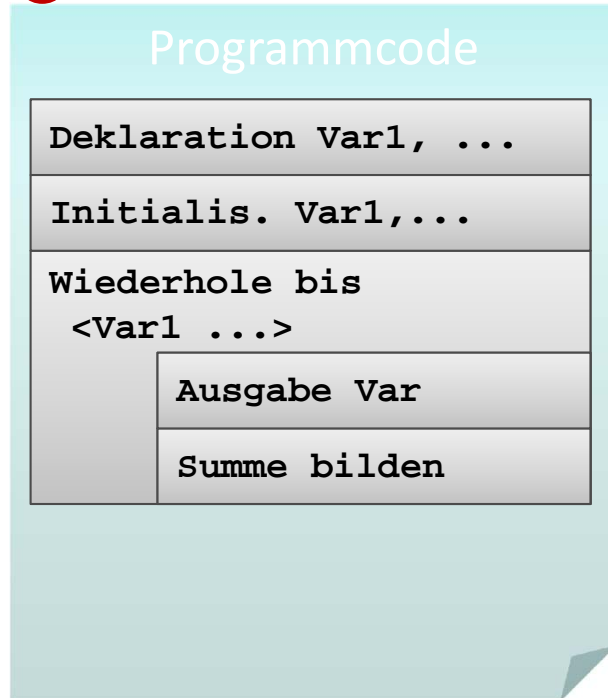
Aufgabe

- Teil 1: Vorprüfende Schleife, die von 1 bis 5 die Zahlen im Direktbereich ausgibt
- Teil 2: Vorprüfende Schleife, die von 5 bis 1 die Zahlen im Direktbereich ausgibt
- Teil 3: Vorprüfende Schleife, die von 1 bis 10 die ungeraden Zahlen im Direktbereich ausgibt
- Teil 4: Vorprüfende Schleife, die von einem eingegebenen Datum die verbleibenden Tage bis Weihnachten zählt
- Teil 5: Vorprüfende Schleife, die die Summe der Zahlen 1, 2, 3, 4, ... usw. bildet, und solange läuft bis vorher eingegebene Zahl erreicht ist. 0



Vorprüfende Schleife: Beispiel 05.02

1



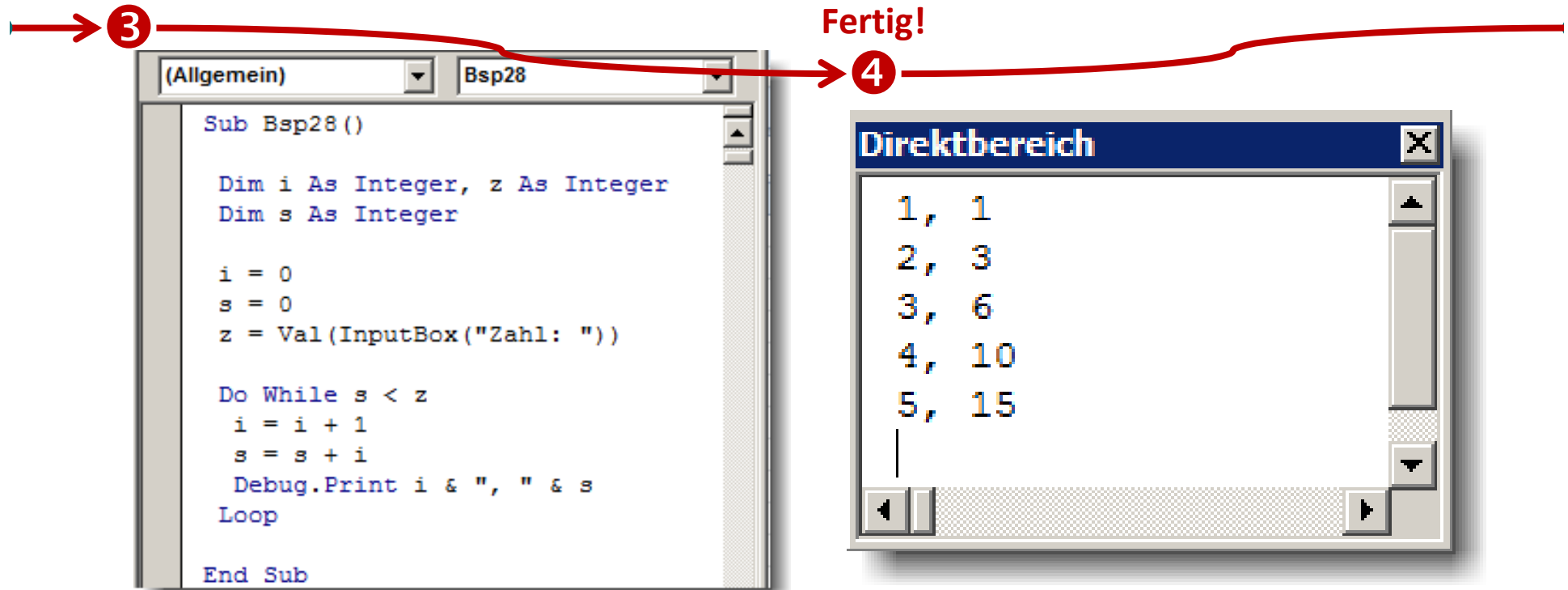
- **Überführen in Programmcode**
 - Deklaration einer Bedingung ("Ist eingeegebene Zahl erreicht?")
 - Schleife, die hochzählt, die Summe bildet und eine Ausgabeanweisung enthält

2

```
Sub Bsp28()  
  Dim i As Integer, z As Integer  
  Dim s As Integer  
  Let i = 0  
  Let s = 0  
  Let z = Val(TextBox("Zahl: "))  
  
  Do While s < z  
    Let i = i + 1  
    Let s = s + i  
    Debug.Print i & ", " & s  
  Loop  
  
End Sub
```

- **Nutzung Do While/Loop**
- **Was passiert beim Ausführen des Programmcodes?**

Vorprüfende Schleife: Beispiel 05.02



- Was ist im Direktbereich sichtbar?
- Ausführen des Programmcodes liefert im Direktbereich schrittweise die Ausgabe des Zählers und der Summe

Vorprüfende Schleife: Beispiel 05.02



→ 5 Ziel

- Vorprüfende Schleife verwenden

Aufgabe

- Teil 1: Vorprüfende Schleife, die von 1 bis 5 die Zahlen im Direktbereich ausgibt
- Teil 2: Vorprüfende Schleife, die von 5 bis 1 die Zahlen im Direktbereich ausgibt
- Teil 3: Vorprüfende Schleife, die von 1 bis 10 die ungeraden Zahlen im Direktbereich ausgibt
- Teil 4: Vorprüfende Schleife, die von einem eingegebenen Datum die verbleibenden Tage bis Weihnachten zählt
- Teil 5: Vorprüfende Schleife, die die Summe der Zahlen 1, 2, 3, 4, ... usw. bildet, und solange läuft bis vorher eingegebene Zahl erreicht ist.





Inhalt

Rückblick

Überblick

- Arten von Schleifen
- Unterschiede im Überblick

Implementierung von Schleifen mit VBA

- Vorprüfende Schleifen
- Nachprüfende Schleifen
- Zählerschleifen
- Weitere Möglichkeiten

Abschluss und Ausblick





Inhalt

Rückblick

Überblick

- Arten von Schleifen
- Unterschiede im Überblick

Implementierung von Schleifen mit VBA

- Vorprüfende Schleifen
- Nachprüfende Schleifen
- Zählerschleifen
- Weitere Möglichkeiten

Abschluss und Ausblick

Nachprüfende Schleife



Schlüsselbegriffe

- Variante 1
- Variante 2

Nachprüfende Schleife



Schlüsselbegriffe

– Variante 1

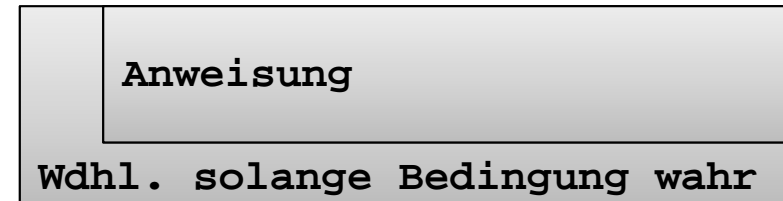
Do

<Anweisung(en)>

Loop While *<Beding.>*

– Variante 2

- ...





Nachprüfende Schleife

Schlüsselbegriffe

– Variante 1

Do

<Anweisung(en)>

Loop While *<Beding.>*

Anweisung

Wdhl. solange Bedingung wahr

– Variante 2

- ...



Nachprüfende Schleife

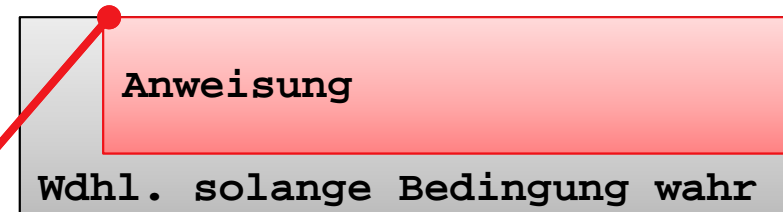
Schlüsselbegriffe

– Variante 1

Do

<Anweisung(en)>

Loop While <Beding.>



– Variante 2

- ...

Nachprüfende Schleife



Schlüsselbegriffe

– Variante 1

Do

<Anweisung(en)>

Loop While *<Beding.>*

Anweisung

Wdhl. solange Bedingung wahr

– Variante 2

- ...



Nachprüfende Schleife

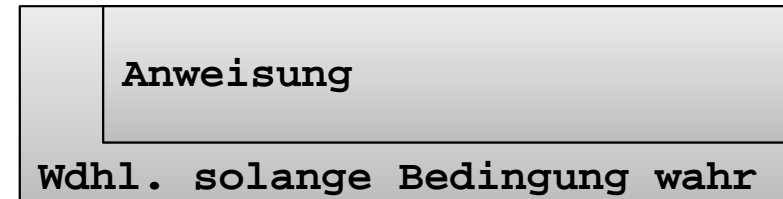
Schlüsselbegriffe

– Variante 1

Do

<Anweisung(en)>

Loop While *<Beding.>*

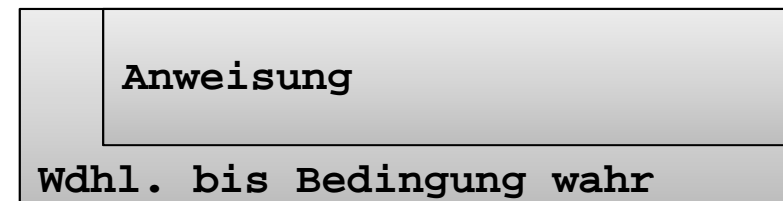


– Variante 2

Do

<Anweisung(en)>

Loop Until *<Beding.>*





Nachprüfende Schleife

Schlüsselbegriffe

– Variante 1

Do

<*Anweisung(en)*>

Loop While <*Beding.*>

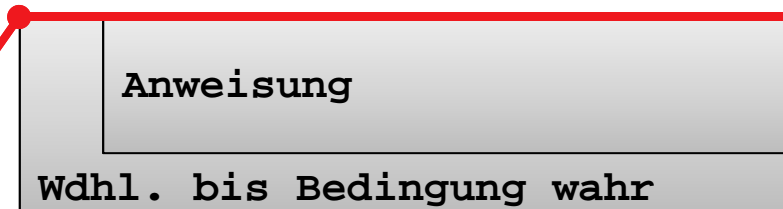


– Variante 2

Do

<*Anweisung(en)*>

Loop Until <*Beding.*>





Nachprüfende Schleife

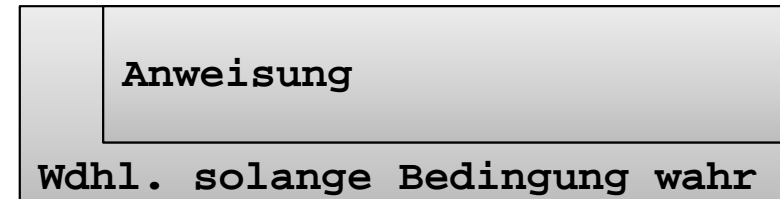
Schlüsselbegriffe

– Variante 1

Do

<Anweisung(en)>

Loop While *<Beding.>*

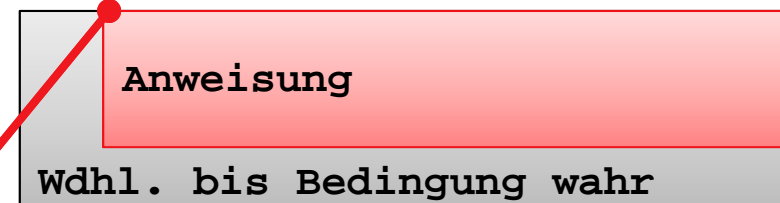


– Variante 2

Do

<Anweisung(en)>

Loop Until *<Beding.>*





Nachprüfende Schleife

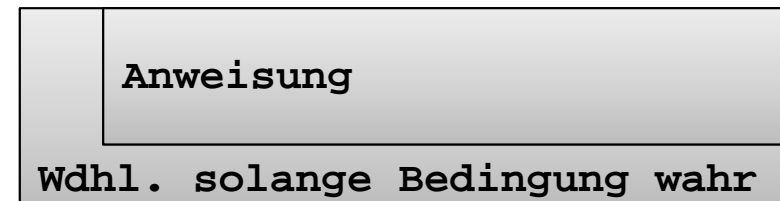
Schlüsselbegriffe

– Variante 1

Do

<Anweisung(en)>

Loop While *<Beding.>*

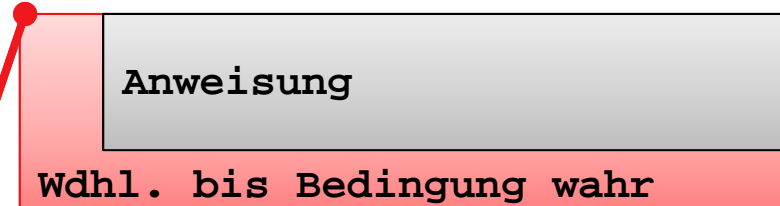


– Variante 2

Do

<Anweisung(en)>

Loop Until *<Beding.>*





Nachprüfende Schleife

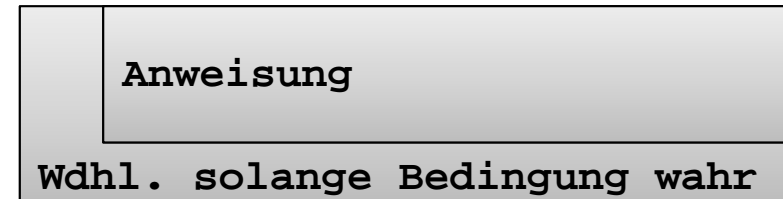
Schlüsselbegriffe

– Variante 1

Do

<Anweisung(en)>

Loop While *<Beding.>*

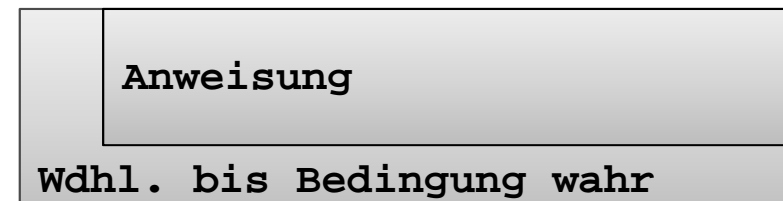


– Variante 2

Do

<Anweisung(en)>

Loop Until *<Beding.>*





Nachprüfende Schleife

Generelle Syntax

– Variante 1

```
Do  
  <Anweisung(en)>  
Loop While <Beding.>
```

– Variante 2

```
Do  
  <Anweisung(en)>  
Loop Until <Beding.>
```

Beispiele

```
Dim i As Integer  
Let i = 0  
Do  
  Let i = i + 1  
  Debug.Print i  
Loop While i < 5
```

```
1  
2  
3  
4  
5
```

```
Dim j As Byte  
Let j = 0  
Do  
  Let j = j + 1  
  Debug.Print j  
Loop Until j > 4
```

```
1  
2  
3  
4  
5
```

Nachprüfende Schleife: Beispiel 05.03



Ziel

- Nachprüfende Schleife mit Bedingungen verwenden

Aufgabe: Schreiben Sie ein Programm mit einer nachprüfenden Schleife

– Teil 1

- In der Schleife sollen alle Vielfachen von 3 ausgegeben werden
- Schleife soll enden, wenn die Vielfachen 100 überschreiten

– Teil 2

- bilden Sie die Zwischensumme der Vielfachen von 3
- Geben Sie das aktuelle Vielfache von 3 im Direktbereich aus
- Geben Sie die Zwischensumme im Direktbereich aus
- Die Schleife soll enden, sobald die Summe größer als 75 ist.



Nachprüfende Schleife: Beispiel 05.04



Ziel

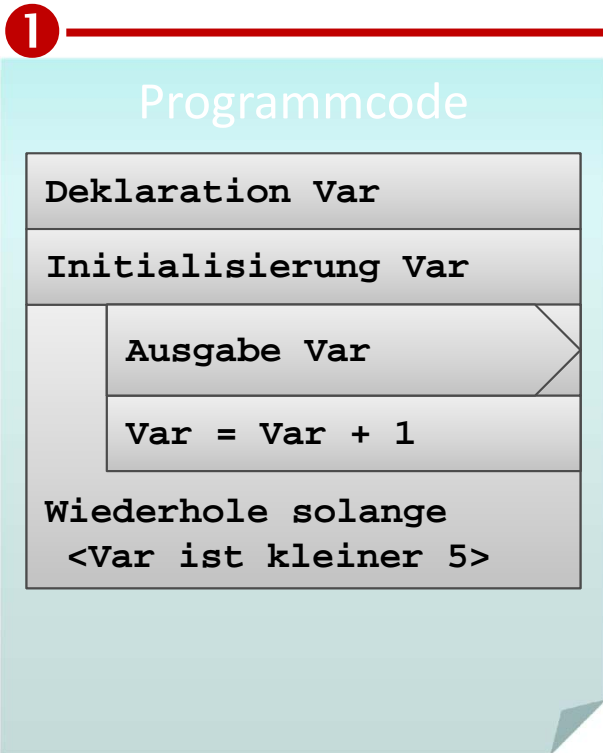
- Weitere Möglichkeiten der nachprüfenden Schleife kennenlernen

Aufgabe

- Teil 1: Nachprüfende Schleife, die von 1 bis 5 die Zahlen im Direktbereich ausgibt **0**
- Teil 2: Nachprüfende Schleife, die von 5 bis 1 die Zahlen im Direktbereich ausgibt
- Teil 3: Nachprüfende Schleife, die von 1 bis 10 die ungeraden Zahlen im Direktbereich ausgibt



Nachprüfende Schleife: Beispiel 05.04



2

```
Sub Bsp()  
  
Dim i As Integer  
Let i = 1  
  
Do  
    Debug.Print i  
    Let i = i + 1  
Loop While i < 5  
  
End Sub
```

- **Überführen in Programmcode**
 - Deklaration einer Variablen, die für die Bedingung verwendet wird
 - Schleife, die von 1 bis 5 zählt und eine Ausgabeanweisung enthält

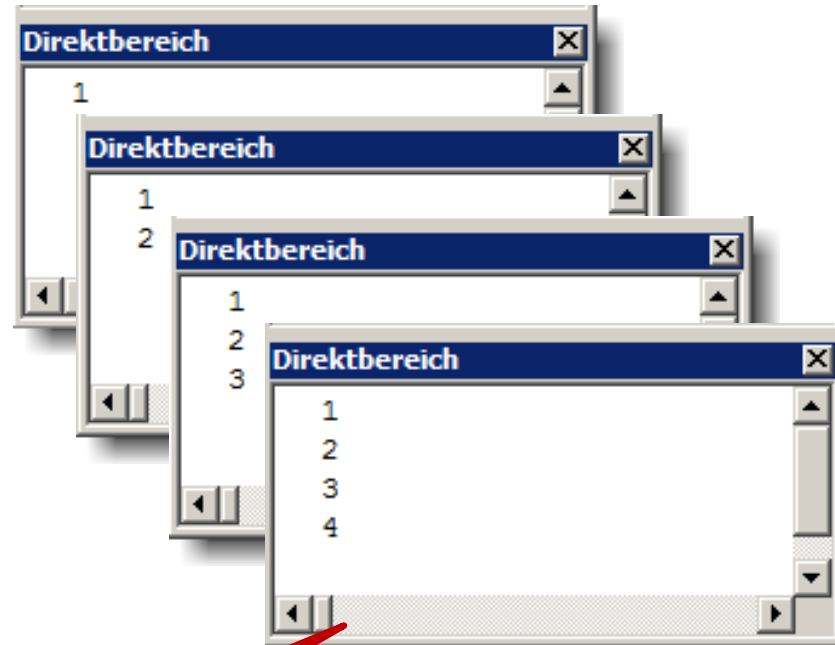
- **Nutzen von Do/Loop While**
- **Was passiert beim Ausführen des Programmcodes mit der Variable i?**

Nachprüfende Schleife: Beispiel 05.04

3

4

```
(Allgemein) Bsp29  
  
Sub Bsp29()  
  
    Dim i As Integer  
    i = 1  
  
    Do  
        Debug.Print i  
        i = i + 1  
    Loop While i < 5  
  
End Sub
```



- Was ist im Direktbereich sichtbar?

Huch!?
Wo ist die 5?

Ausführen des Programmcodes liefert im Direktbereich schrittweise die Ausgabe der Zahlen von 1 bis 4

Nachprüfende Schleife: Beispiel 05.04



➤ 5 Ziel

- Weitere Möglichkeiten der nachprüfenden Schleife kennenlernen

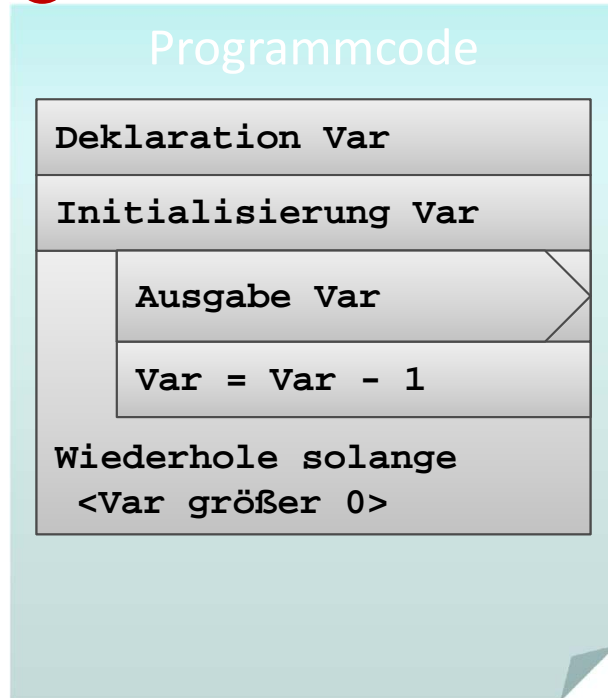
Aufgabe

- Teil 1: Nachprüfende Schleife, die von 1 bis 5 die Zahlen im Direktbereich ausgibt
- Teil 2: Nachprüfende Schleife, die von 5 bis 1 die Zahlen im Direktbereich ausgibt
- Teil 3: Nachprüfende Schleife, die von 1 bis 10 die ungeraden Zahlen im Direktbereich ausgibt



Nachprüfende Schleife: Beispiel 05.04

1



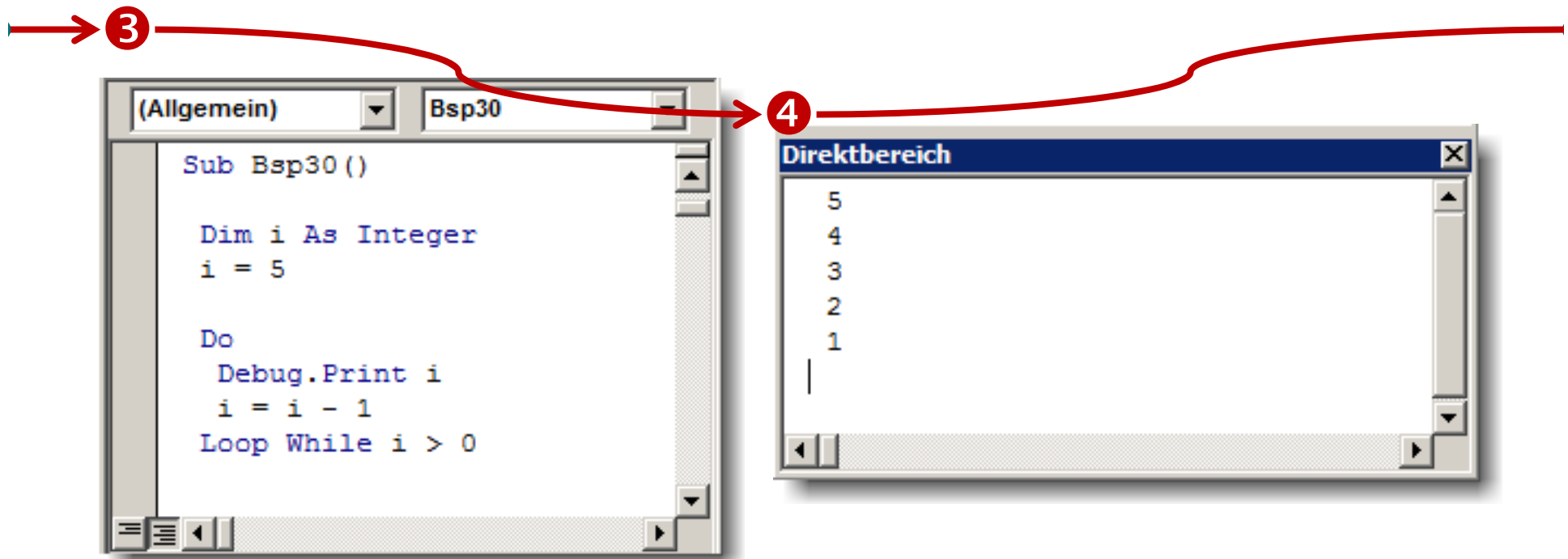
2

```
Sub Bsp30()  
  
Dim i As Integer  
i = 5  
  
Do  
    Debug.Print i  
    i = i - 1  
Loop While i > 0  
  
End Sub
```

- **Überführen in Programmcode**
 - Deklaration einer Variablen, die zum Zählen verwendet wird
 - Schleife, die von 5 bis 1 rückwärts zählt und eine Ausgabeanweisung enthält

- **Nutzung Do While/Loop**
- **Was passiert beim Ausführen des Programmcodes mit der Variable i?**

Nachprüfende Schleife: Beispiel 05.04



- Was ist im Direktbereich sichtbar?
- Ausführen des Programmcodes liefert im Direktbereich schrittweise die Ausgabe der Zahlen von 5 bis 1

Nachprüfende Schleife: Beispiel 05.04



→ 5 Ziel

- Weitere Möglichkeiten der nachprüfenden Schleife kennenlernen

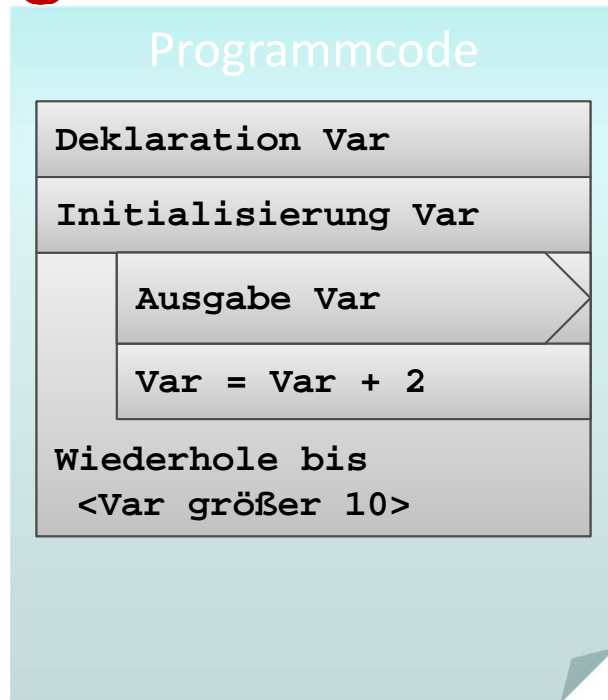
Aufgabe

- Teil 1: Nachprüfende Schleife, die von 1 bis 5 die Zahlen im Direktbereich ausgibt
- Teil 2: Nachprüfende Schleife, die von 5 bis 1 die Zahlen im Direktbereich ausgibt
- Teil 3: Nachprüfende Schleife, die von 1 bis 10 die ungeraden Zahlen im Direktbereich ausgibt → 0



Nachprüfende Schleife: Beispiel 05.04

1



2

```
Sub Bsp31()  
  
Dim i As Integer  
i = 1  
  
Do  
    Debug.Print i  
    i = i + 2  
Loop Until i > 10  
  
End Sub
```

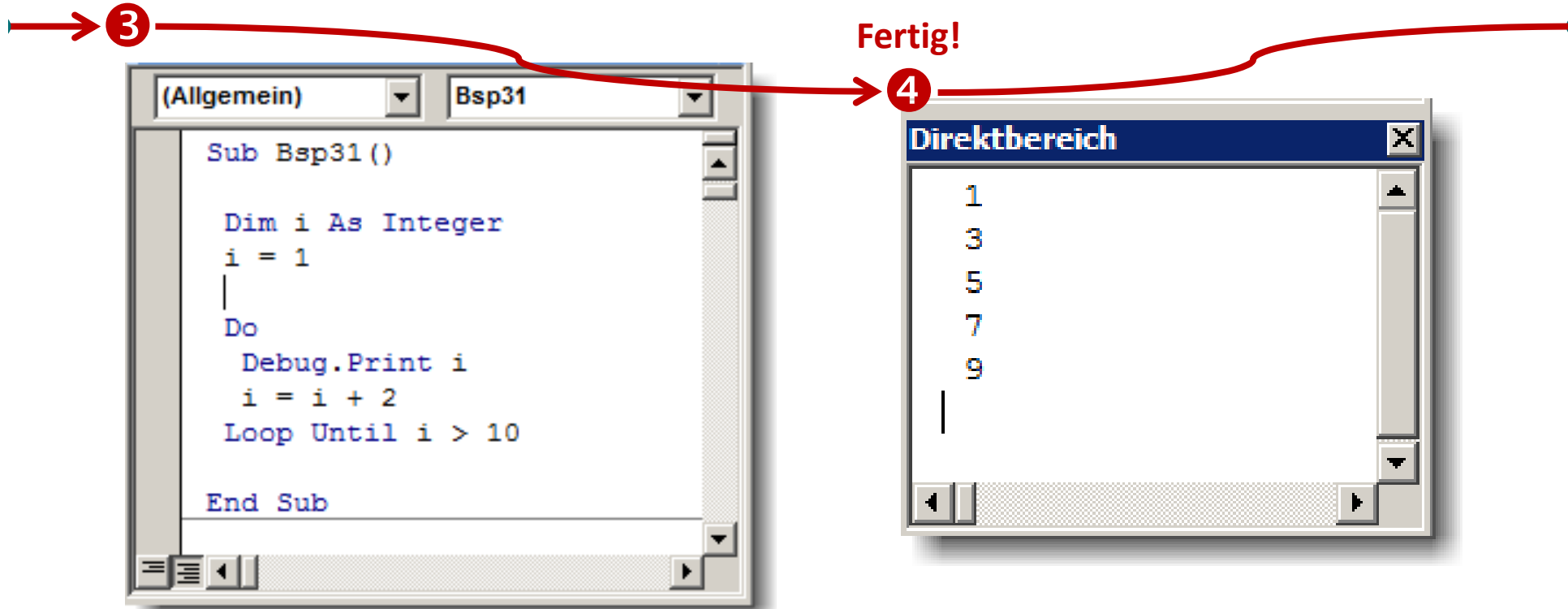
- **Überführen in Programmcode**

- Deklaration einer Variablen, die zum Zählen verwendet wird
- Schleife, die von 1 bis 10 in Zweierschritten zählt und eine Ausgabeanweisung enthält

- **Nutzung Do Until/Loop**

- **Was passiert beim Ausführen des Programmcodes mit der Variable i?**

Nachprüfende Schleife: Beispiel 05.04



- Was ist im Direktbereich sichtbar?
- Ausführen des Programmcodes liefert im Direktbereich schrittweise die Ausgabe der Zahlen 1, 3, 5, 7 und 9.

Nachprüfende Schleife: Beispiel 05.04



➤ 5 Ziel

- Weitere Möglichkeiten der nachprüfenden Schleife kennenlernen

Aufgabe

- Teil 1: Nachprüfende Schleife, die von 1 bis 5 die Zahlen im Direktbereich ausgibt
- Teil 2: Nachprüfende Schleife, die von 5 bis 1 die Zahlen im Direktbereich ausgibt
- Teil 3: Nachprüfende Schleife, die von 1 bis 10 die ungeraden Zahlen im Direktbereich ausgibt



Hinweise vor-/nachprüfende Schleife



Do While vs. Do Until bzw. Loop While vs. Loop Until

- While bedeutet, dass solange die Bedingung wahr ist die Schleife ausgeführt wird
- Until bedeutet, dass bis die Bedingung wahr wird, die Schleife ausgeführt wird
- Äquivalenz zwischen While und Until durch Einsatz der logischen Negation

```
Dim i As Byte
Let i = 0

Debug.Print "While:"
Do While i < 5
    Debug.Print i
    Let i = i + 1
Loop
```

```
Dim i As Byte
Let i = 0

Debug.Print "Until Not:"
Do Until Not (i < 5)
    Debug.Print i
    Let i = i + 1
Loop
```

Alternative
Negation:
 $i \geq 5$



Inhalt

Rückblick

Überblick

- Arten von Schleifen
- Unterschiede im Überblick

Implementierung von Schleifen mit VBA

- Vorprüfende Schleifen
- Nachprüfende Schleifen
- Zählerschleifen
- Weitere Möglichkeiten

Abschluss und Ausblick





Inhalt

Rückblick

Überblick

- Arten von Schleifen
- Unterschiede im Überblick

Implementierung von Schleifen mit VBA

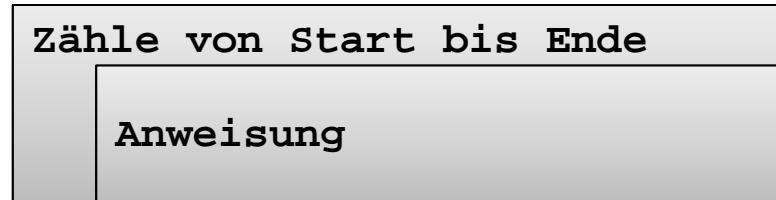
- Vorprüfende Schleifen
- Nachprüfende Schleifen
- Zählerschleifen
- Weitere Möglichkeiten

Abschluss und Ausblick

Zählerschleife

Schlüsselbegriffe

- Grundlegende Form
- Erweiterungen





Zählerschleife

Schlüsselbegriffe

- Grundlegende Form

For

<Var> = <Beginn>

To <Ende>

<Anweisung(en)>

Next

- Erweiterungen

- ...

Zähle von Start bis Ende

Anweisung



Zählerschleife

Schlüsselbegriffe

- Grundlegende Form

For

<Var> = <Beginn>

To <Ende>

<Anweisung(en)>

Next

Zähle von Start bis Ende

Anweisung

- Erweiterungen

- ...

Zählerschleife



Schlüsselbegriffe

- Grundlegende Form

For

<Var> = <Beginn>

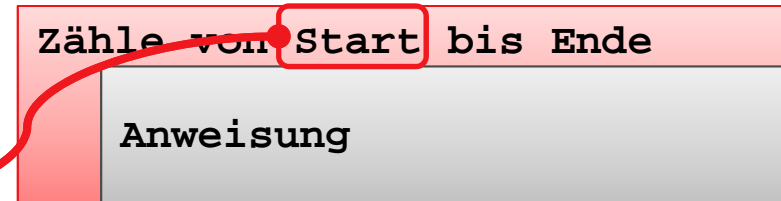
To <Ende>

<Anweisung(en)>

Next

- Erweiterungen

- ...



Zählerschleife



Schlüsselbegriffe

- Grundlegende Form

For

<Var> = <Beginn>

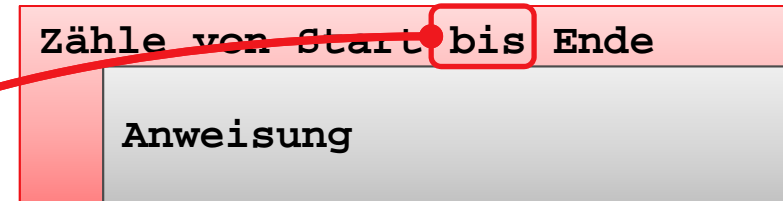
To *<Ende>*

<Anweisung(en)>

Next

- Erweiterungen

- ...





Zählerschleife

Schlüsselbegriffe

- Grundlegende Form

For

<Var> = <Beginn>

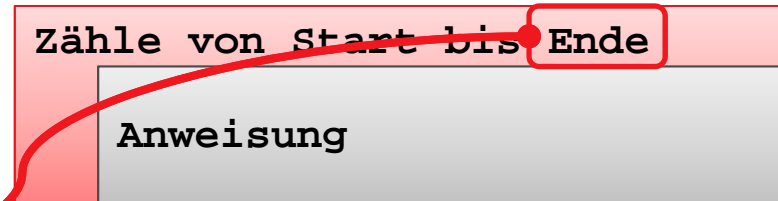
To *<Ende>*

<Anweisung(en)>

Next

- Erweiterungen

- ...





Zählerschleife

Schlüsselbegriffe

- Grundlegende Form

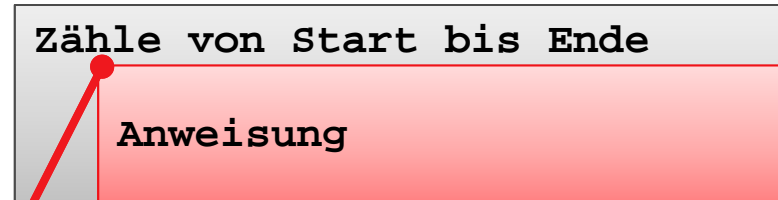
For

<Var> = <Beginn>

To <Ende>

<Anweisung(en)>

Next



- Erweiterungen

- ...



Zählerschleife

Schlüsselbegriffe

- Grundlegende Form

For

<Var> = <Beginn>

To <Ende>

<Anweisung(en)>

Next

Zähle von Start bis Ende

Anweisung

- Erweiterungen

- ...

Zählerschleife



Schlüsselbegriffe

- Grundlegende Form

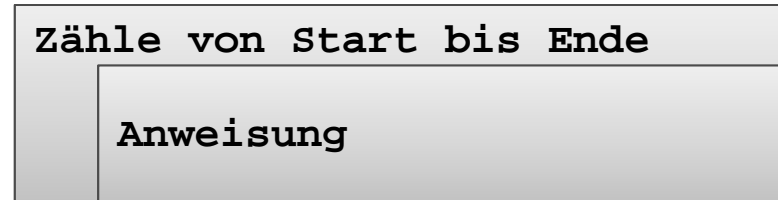
For

<Var> = <Beginn>

To <Ende>

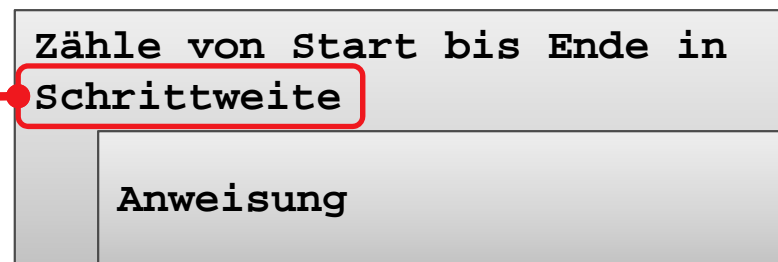
<Anweisung(en)>

Next



- Erweiterungen

- Schrittweite/-richtung beim Zählen **Step**



Zählerschleife



Grundlegende Form

- Generelle Syntax

```
For <Variable> = <Wert oder Ausdruck> To <Ausdruck>  
    <Anweisung(en)>  
Next
```

- Beispiele

```
Dim i As Integer  
For i = 1 To 5  
    Debug.Print i  
Next
```

```
1  
2  
3  
4  
5
```

Zählerschleife



Grundlegende Form

- Erweiterungen

```
For <Var> = <Wert/Ausdr> To <Ausdr> Step <Schritt>  
  <Anweisung(en)>  
Next
```

- Beispiele

```
Dim i As Integer  
For i = 2 To 8 Step 2  
  Debug.Print i  
Next
```

```
2  
4  
6  
8
```

```
Dim i As Integer  
For i = 9 To 1 Step -3  
  Debug.Print i  
Next
```

```
9  
6  
3
```

Zählerschleife: Beispiel 05.05



Aufgabe

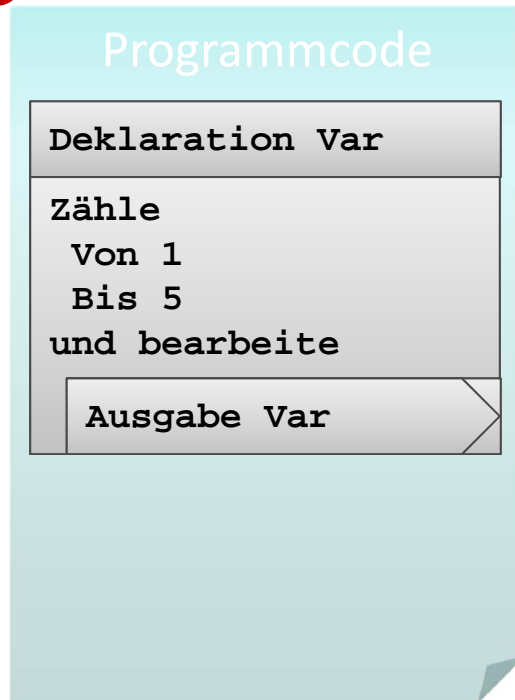
- Teil 1: Zählerschleife, die von 1 bis 5 die Zahlen im Direktbereich ausgibt
- Teil 2: Zählerschleife, die von 5 bis 1 die Zahlen im Direktbereich ausgibt
- Teil 3: Zählerschleife, die von 1 bis 10 die ungeraden Zahlen im Direktbereich ausgibt

0



Zählerschleife: Beispiel 05.05

1



2

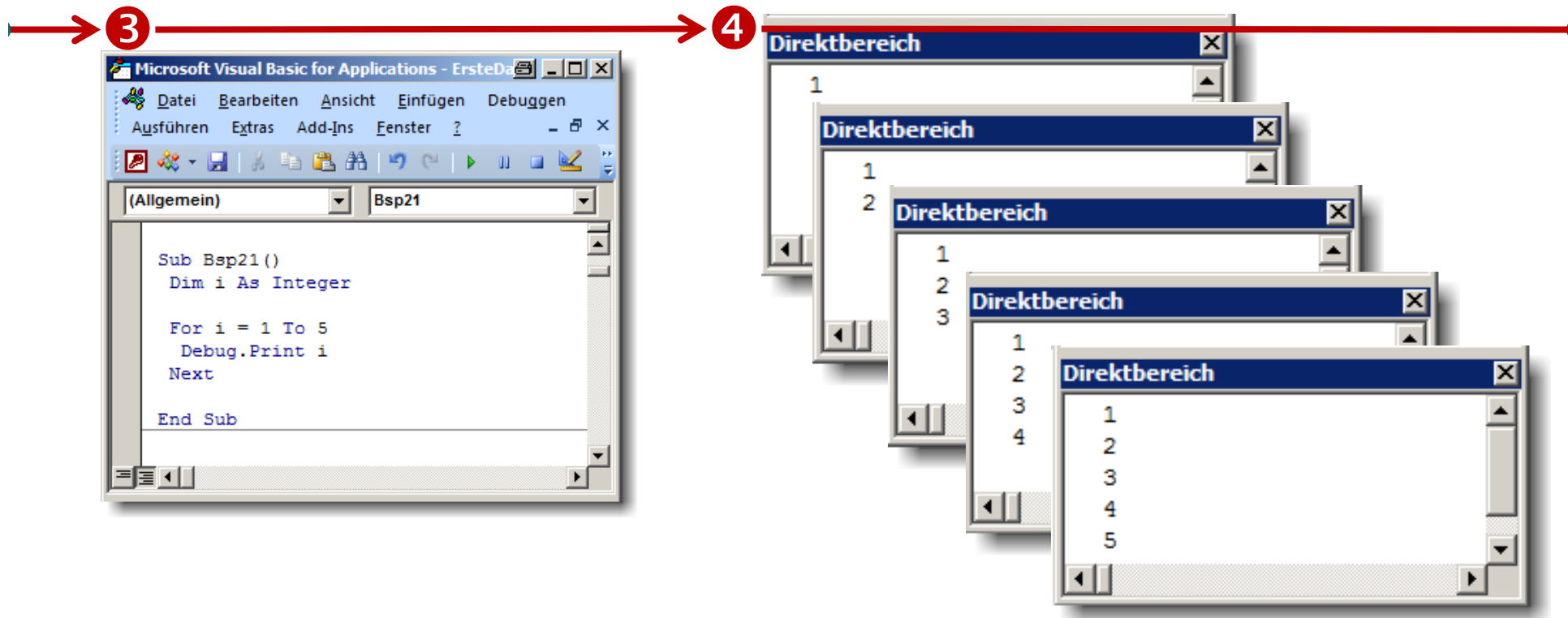
```
Sub Bsp21()  
  
    Dim i As  
    Integer  
  
    For i = 1 To 5  
        Debug.Print i  
    Next  
  
End Sub
```

- **Überführen in Programmcode**

- Deklaration einer Variablen, die zum Zählen verwendet wird
- Schleife, die von 1 bis 5 zählt und eine Ausgabeanweisung enthält

- **Was passiert beim Ausführen des Programmcodes mit der Variable i?**

Zählerschleife: Beispiel 05.05



- Was ist im Direktbereich sichtbar?
- Ausführen des Programmcodes liefert im Direktbereich schrittweise die Ausgabe der Zahlen von 1 bis 5

Zählerschleife: Beispiel 05.05



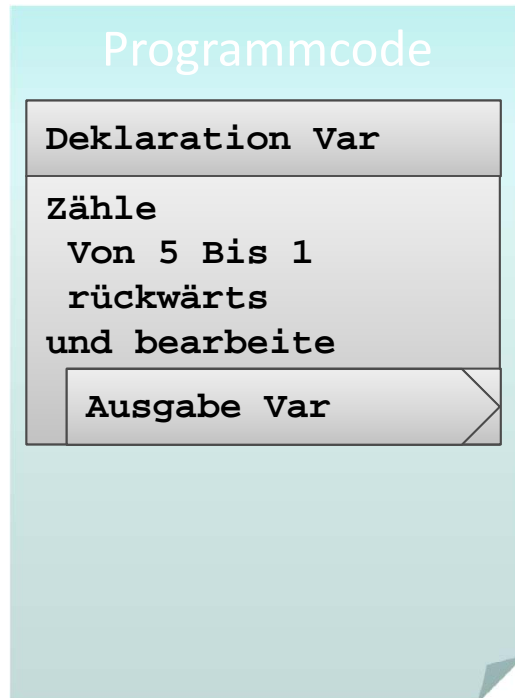
→ 5 Aufgabe

- Teil 1: Zählerschleife, die von 1 bis 5 die Zahlen im Direktbereich ausgibt
- Teil 2: Zählerschleife, die von 5 bis 1 die Zahlen im Direktbereich ausgibt
- Teil 3: Zählerschleife, die von 1 bis 10 die ungeraden Zahlen im Direktbereich ausgibt



Zählerschleife: Beispiel 05.05

1



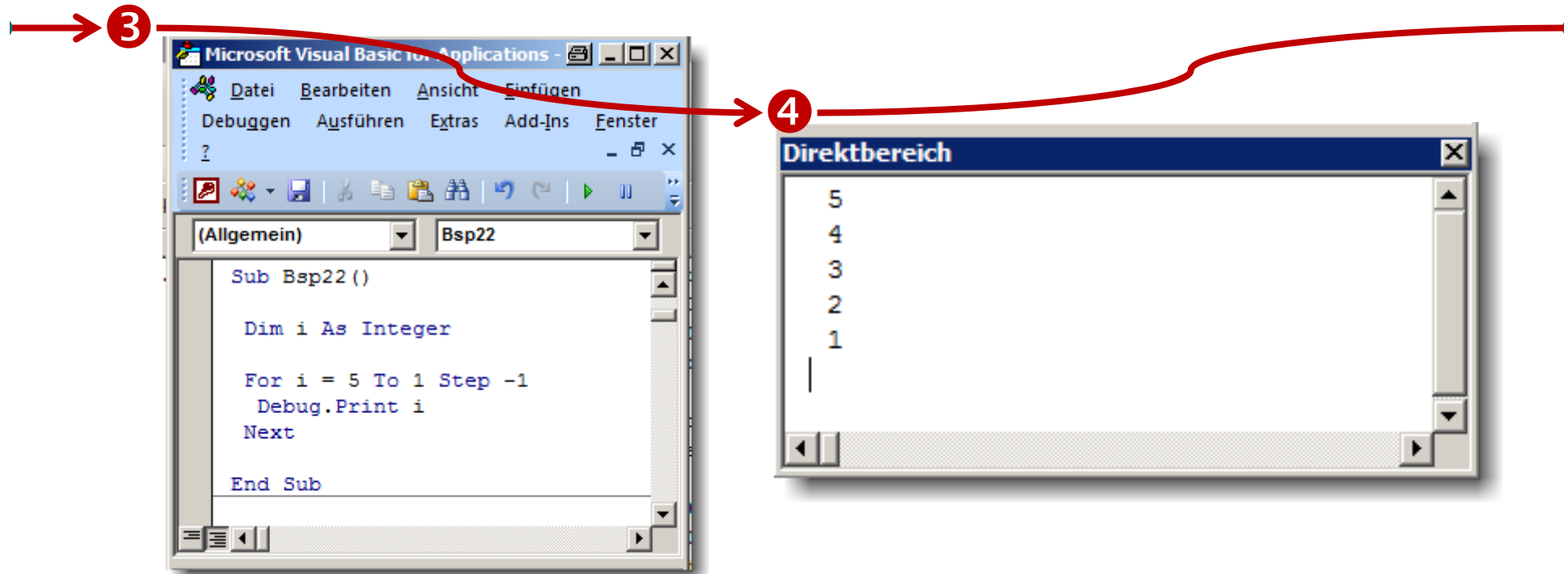
2

```
Sub Bsp22()  
  
    Dim i As Integer  
  
    For i = 5 To 1 Step -1  
        Debug.Print i  
    Next  
  
End Sub
```

- **Überführen in Programmcode**
 - Deklaration einer Variablen, die zum Zählen verwendet wird
 - Schleife, die von 5 bis 1 rückwärts zählt und eine Ausgabeanweisung enthält

- **Was passiert beim Ausführen des Programmcodes mit der Variable i?**

Zählerschleife: Beispiel 05.05



- Was ist im Direktbereich sichtbar?
- Ausführen des Programmcodes liefert im Direktbereich schrittweise die Ausgabe der Zahlen von 5 bis 1

Zählerschleife: Beispiel 05.05



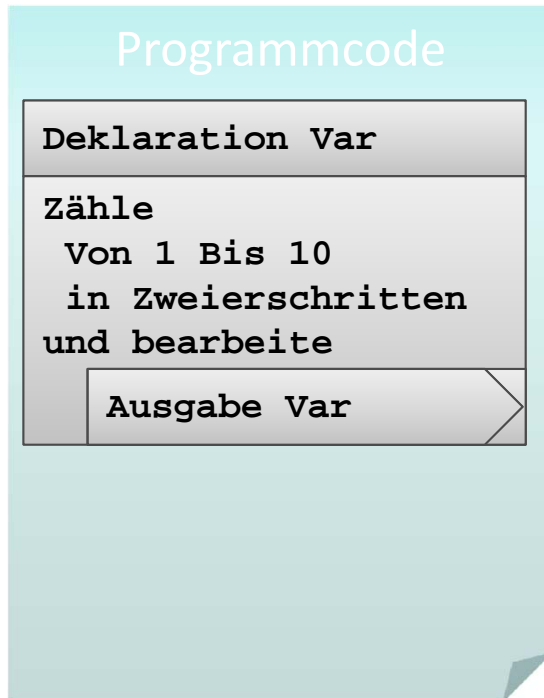
→ 5 Aufgabe

- Teil 1: Zählerschleife, die von 1 bis 5 die Zahlen im Direktbereich ausgibt
- Teil 2: Zählerschleife, die von 5 bis 1 die Zahlen im Direktbereich ausgibt
- Teil 3: Zählerschleife, die von 1 bis 10 die ungeraden Zahlen im Direktbereich ausgibt → 0



Zählerschleife: Beispiel 05.05

1



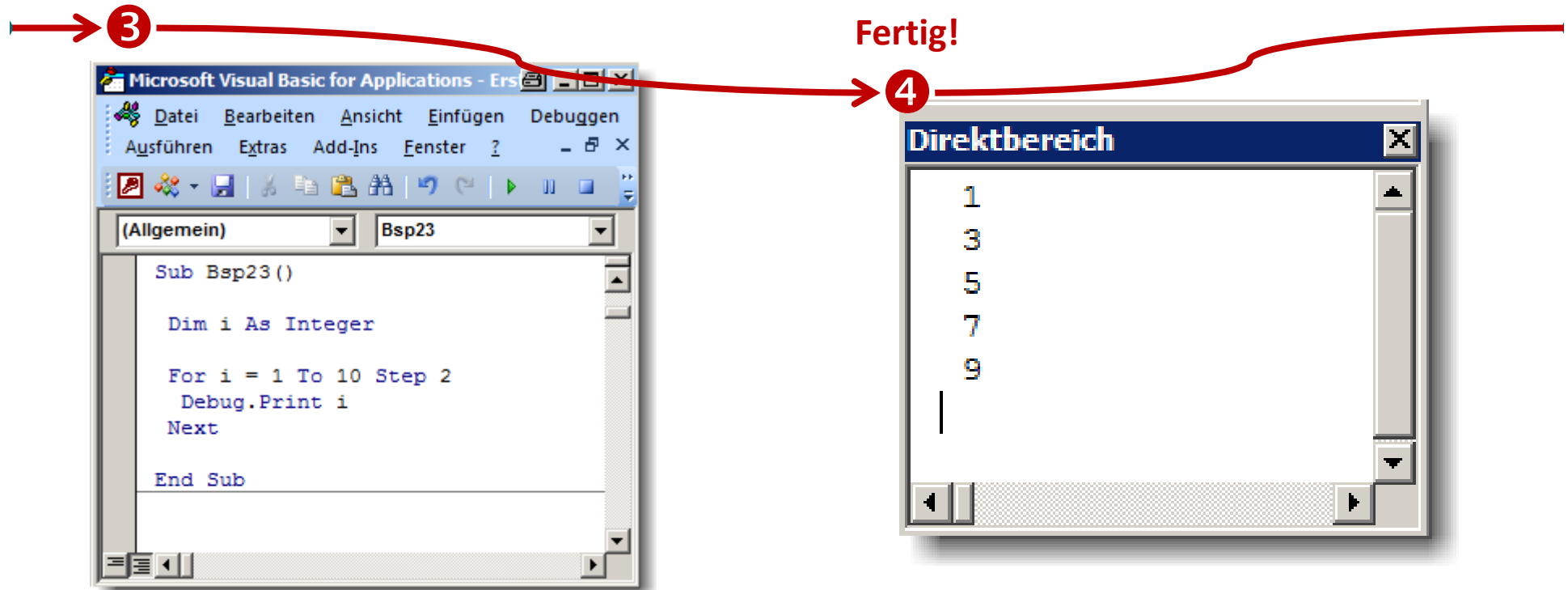
2

```
Sub Bsp23()  
  
    Dim i As Integer  
  
    For i = 1 To 10 Step 2  
        Debug.Print i  
    Next  
  
End Sub
```

- **Überführen in Programmcode**
 - Deklaration einer Variablen, die zum Zählen verwendet wird
 - Schleife, die von 1 bis 10 in Zweierschritten zählt und eine Ausgabeanweisung enthält

- **Was passiert beim Ausführen des Programmcodes mit der Variable i?**

Zählerschleife: Beispiel 05.05



- Was ist im Direktbereich sichtbar?
- Ausführen des Programmcodes liefert im Direktbereich schrittweise die Ausgabe der Zahlen 1, 3, 5, 7 und 9.

Zählerschleife: Beispiel 05.05



→ 5 Aufgabe

- Teil 1: Zählerschleife, die von 1 bis 5 die Zahlen im Direktbereich ausgibt
- Teil 2: Zählerschleife, die von 5 bis 1 die Zahlen im Direktbereich ausgibt
- Teil 3: Zählerschleife, die von 1 bis 10 die ungeraden Zahlen im Direktbereich ausgibt



Zählerschleife: Beispiel 05.06



Ziel

- Zählerschleife praktisch anwenden

Aufgabe

- Schreiben Sie ein Programm mit einer Zählerschleife
- Es soll die Summe aller Zahlen im Bereich von 0 bis 25 bilden
- Die jeweilige Zwischensumme und die zuletzt addierte Zahl sollen im Direktbereich ausgegeben werden
- Die Schleife soll bei 25 enden.
- Das Endergebnis ist hinter der Schleife im Direktbereich auszugeben.
- Berechnen Sie das Ergebnis zusätzlich mit der Gaußschen Summenformel und geben Sie es zum Vergleich aus:
 - $1 + 2 + \dots + n = n * (n+1)/2$





Zählerschleife Hinweise

Schleifenvariable

- für Zählerschleife ist eine Variable notwendig
- Variable muss vorher deklariert werden
- Initialisierung der Variable erfolgt im Schleifenkopf durch Zuweisung des Startwertes

Zählerschleife Hinweise



Welchen Wert hat die Schleifenvariable, nachdem die Schleife vollständig durchlaufen wurde?

– Beispiel

```
Dim i As Integer
For i = 2 To 8 Step 2
    Debug.Print "Innen: " & i
Next
Debug.Print "Dahinter: " & i
```

```
Innen: 2
Innen: 4
Innen: 6
Innen: 8
Dahinter: 10
```

– Antwort: Den Endwert der Schleife plus einen weiteren Schritt entsprechend der Schrittweite



Inhalt

Rückblick

Überblick

- Arten von Schleifen
- Unterschiede im Überblick

Implementierung von Schleifen mit VBA

- Vorprüfende Schleifen
- Nachprüfende Schleifen
- Zählerschleifen
- Weitere Möglichkeiten

Abschluss und Ausblick





Inhalt

Rückblick

Überblick

- Arten von Schleifen
- Unterschiede im Überblick

Implementierung von Schleifen mit VBA

- Vorprüfende Schleifen
- Nachprüfende Schleifen
- Zählerschleifen
- Weitere Möglichkeiten

Abschluss und Ausblick



Weitere Möglichkeiten

Risiko von Endlosschleifen

- Anweisungen innerhalb der Schleife müssen bei Vor- und nachprüfenden Schleifen Einfluss auf die zu prüfende Bedingung nehmen
- Zählerschleifen sollten die Zählervariable in den Anweisungen innerhalb der Schleife nur lesen, bei ändernden Zugriffen besteht die Gefahr der Endlosschleife
- Endlosschleifen in VBA abbrechen mit Tastenkombination
 - STRG + Pause
 - STRG + ALT + Pause (Laborrechner in B045L)
- Rechner ohne Pause-Taste (z.B. MacBook, Samsung Ativ)
 - Zusatzprogramm SharpKey installieren (<http://www.randyrants.com/sharpkeys/>) und damit eine Taste zur Pause-Taste (E0_46) machen



Weitere Möglichkeiten

Schachteln von Schleifen

- Äußere Schleife wird durchlaufen
- Innerhalb der Äußeren Schleife wird die innere Schleife durchlaufen
- mit allen Arten von Schleifen möglich
- Beispiel: Geschachtelte Zählerschleifen

```
Dim i As Byte, j As Byte

For i = 1 To 3
  For j = 1 To 3
    Debug.Print i & "*" & j & "=" & i * j
  Next
  Debug.Print
Next
```

Direktbereich		
1	*	1 = 1
1	*	2 = 2
1	*	3 = 3
2	*	1 = 2
2	*	2 = 4
2	*	3 = 6
3	*	1 = 3
3	*	2 = 6
3	*	3 = 9



Weitere Möglichkeiten

Vorzeitiges Verlassen von Schleifen

- Schleifen können verlassen werden, auch wenn Ende-Bedingung noch nicht erfüllt ist
- vor- und nachprüfende Schleifen mit Schlüsselwort: **Exit Do**
- Zählerschleifen mit Schlüsselwort: **Exit For**

Springen zum nächsten Schleifendurchlauf

- ist der aktuelle Schleifendurchlauf noch nicht beendet, soll aber keine weitere Anweisung innerhalb der Schleife verarbeitet werden, dann
- kann mit **Continue Do** bzw. **Continue For** zum jeweils nächsten Schleifendurchlauf "gesprungen" werden
- aber nicht in VBA ☹️

Alte Form der Schleife: While-Wend-Schleife wird nicht mehr benutzt

Nicht in VBA
Nicht mehr üblich



Inhalt

Rückblick

Überblick

- Arten von Schleifen
- Unterschiede im Überblick

Implementierung von Schleifen mit VBA

- Vorprüfende Schleifen
- Nachprüfende Schleifen
- Zählerschleifen
- Weitere Möglichkeiten

Abschluss und Ausblick





Inhalt

Rückblick

Überblick

- Arten von Schleifen
- Unterschiede im Überblick

Implementierung von Schleifen mit VBA

- Vorprüfende Schleifen
- Nachprüfende Schleifen
- Zählerschleifen
- Weitere Möglichkeiten

Abschluss und Ausblick



Inhalt

Rückblick

Überblick

- Arten von Schleifen
- Unterschiede im Überblick

Implementierung von Schleifen mit VBA

- Vorprüfende Schleifen
- Nachprüfende Schleifen
- Zählerschleifen
- Weitere Möglichkeiten

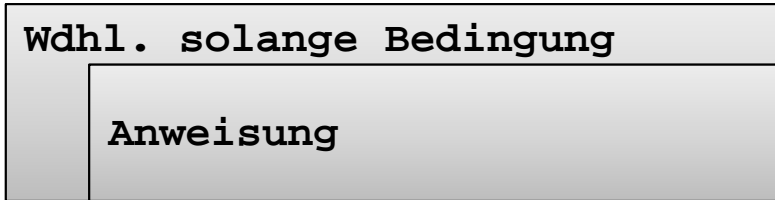
Abschluss und Ausblick

Abschluss



Vorprüfende/Kopfprüfende Schleife

- Einsatzzweck
 - erst Bedingung prüfen
 - dann ggf. Anweisung ausführen
 - anschließend Wiederholung der Prüfung usw.



- Generelle Syntax

```
Do While <Bedingung>  
  <Anweisung(en)>  
Loop
```

```
Do Until <Bedingung>  
  <Anweisung(en)>  
Loop
```

- Beispiel

```
Dim i As Integer  
Let i = 0  
Do While i < 5  
  Let i = i + 1  
  Debug.Print i  
Loop
```

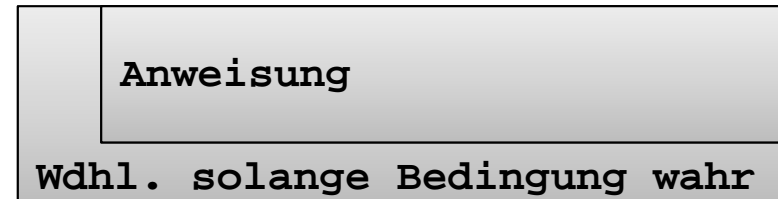
```
Dim j As Byte  
Let j = 0  
Do Until j > 4  
  Let j = j + 1  
  Debug.Print j  
Loop
```

Abschluss



Nachprüfende/Fußprüfende Schleife

- Einsatzzweck
 - erst Anweisung ausgeführt
 - dann Bedingung prüfen
 - anschließend Wiederholung der Anweisungsausführung usw.



– Generelle Syntax

```
Do  
  <Anweisung(en)>  
Loop While <Beding.>
```

```
Do  
  <Anweisung(en)>  
Loop Until <Beding.>
```

– Beispiel

```
Dim i As Integer  
Let i = 0  
Do  
  Let i = i + 1  
  Debug.Print i  
Loop While i < 5
```

```
Dim j As Byte  
Let j = 0  
Do  
  Let j = j + 1  
  Debug.Print j  
Loop Until j > 4
```

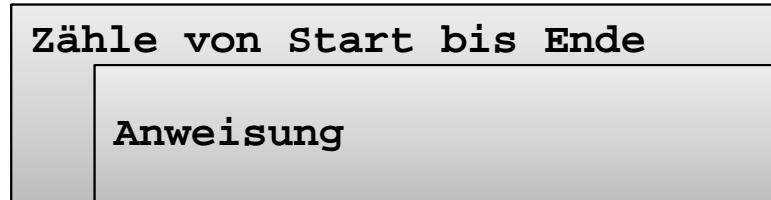
Abschluss



Zählerschleifen

– Einsatz

- Vorher bekannte Anzahl von Wiederholungen
- Anzahl gesteuert über Start und Ende
- Ausführung der Anweisung solange Anzahl Wiederholungen das Ende noch überschritten hat



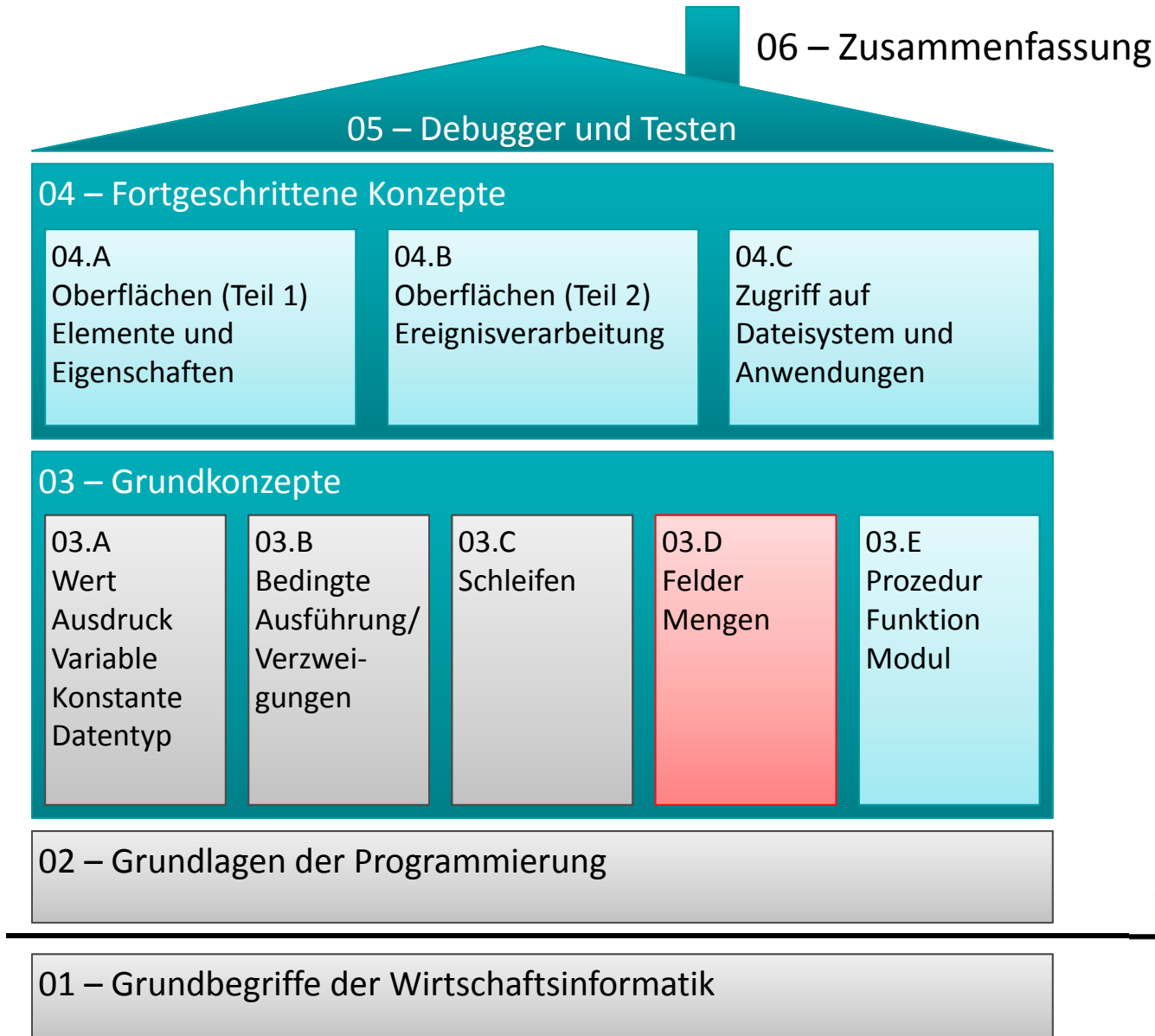
– Generelle Syntax

```
For <Var> = <Wert/Ausdr> To <Ausdr> Step <Schrittw>
  <Anweisung(en)>
Next
```

– Beispiel

```
Dim i As Integer
For i = 1 To 10 Step 2
  Debug.Print i
Next
```

Ausblick





BEUTH HOCHSCHULE FÜR TECHNIK BERLIN
University of Applied Sciences

Wirtschaftsinformatik 1

LE 05 – Schleifen

Prof. Dr. Thomas Off

<http://www.ThomasOff.de/lehre/beuth/wi1>