



# Anweisungen wiederholt ausführen



## Wiederholungen

```
Sub quartalertraege()
```

```
Dim ertrag As Integer, quartalertrag As Integer
```

```
ertrag = 0 ' Diese Zeile könnte auch entfallen
```

```
quartalertrag = Val(InputBox("Bitte Ertrag des 1.  
Quartals eingeben"))
```

```
ertrag = ertrag + quartalertrag ... u.s.w. bis
```

```
quartalertrag = Val(InputBox("Bitte Ertrag des 4.  
Quartals eingeben"))
```

```
ertrag = ertrag + quartalertrag
```

```
Debug.Print (ertrag)
```

```
End Sub
```



Im Beispiel musste der selbe Befehl für die Quartale 1, 2, 3 und 4 nacheinander ausgeführt werden.

Technisch gesprochen: Für die Variablenbelegungen `quartal = 1`, `quartal = 2`, `quartal = 3` und `quartal = 4`

Zähle [Variable] von [Start] bis [Ende], Schrittweite [s]

Anweisungen



## For-Schleifen

For Zählervariable = Anfangswert To Endwert

    Anweisungen

Next Zählervariable

For i = 1 To 10

    Debug.print(i)

Next i



## Schrittweite

```
For i = 1 To 10 Step 2  
    Debug.print(i)  
Next i
```

Ausgabe im Direktfenster:

1  
3  
5  
7  
9



## Eingabe der Quartalerträge mit for-Schleife

```
Sub quartalertraege3()
```

```
Dim ertrag As Integer, quartalertrag As Integer,  
    quartal As Integer
```

```
ertrag = 0    ' Diese Zeile könnte auch entfallen
```

```
For quartal = 1 To 4
```

```
    quartalertrag = Val(TextBox("Bitte Ertrag des"  
    & quartal & ". Quartals eingeben"))
```

```
    ertrag = ertrag + quartalertrag
```

```
Next quartal
```

```
Debug.Print (ertrag)
```

```
End Sub
```

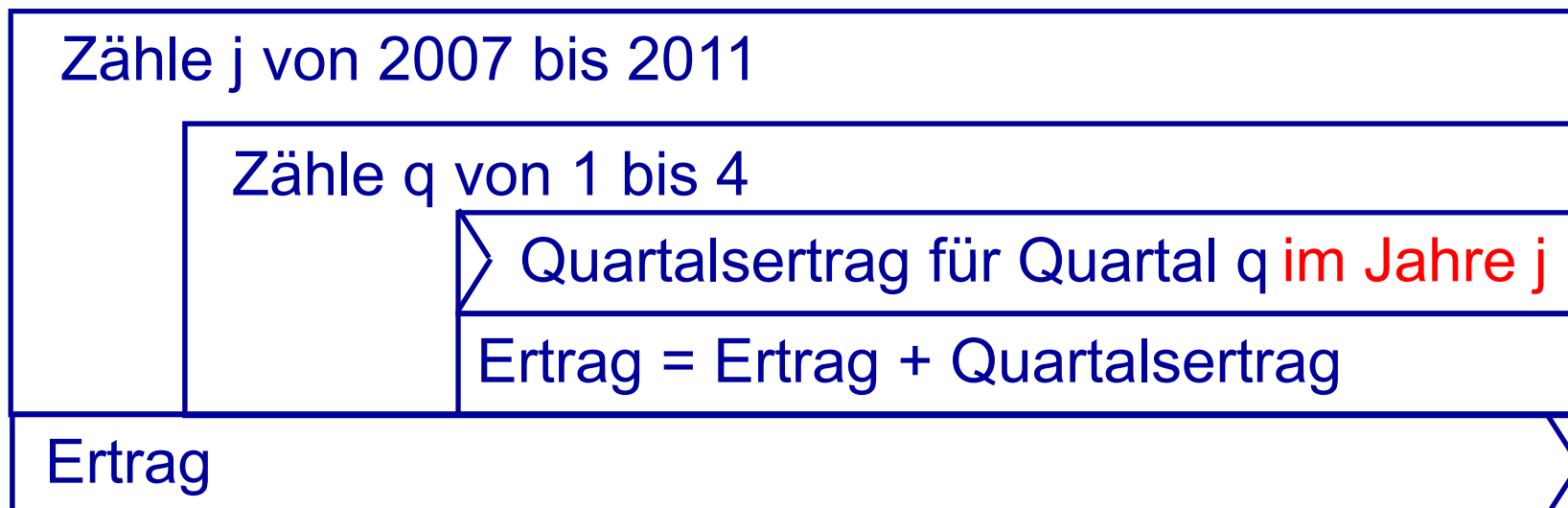


## Aufgabe:

Frage für die Jahre 2011 bis 2020 jeweils die vier Quartalerträge einzeln ab und gib danach die Gesamtsumme dieser Quartalerträge aus



# Verschachtelte for-Schleifen





## Wir merken uns:

Eine For-Schleife ist immer dann nützlich, wenn man dieselben Anweisungen mehrfach ausführt und die Zahl der Ausführungen vorher kennt.



## Aufgabe:

Bestimme die Quersumme einer eingegebenen Zahl  $x$ .



## Analyse

- Quersumme = Summe der Dezimalziffern

$$2351 \rightarrow 2+3+5+1 = 11$$

Wie kommen wir an die einzelnen Ziffern?



## 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       $125^{(1/3)}$

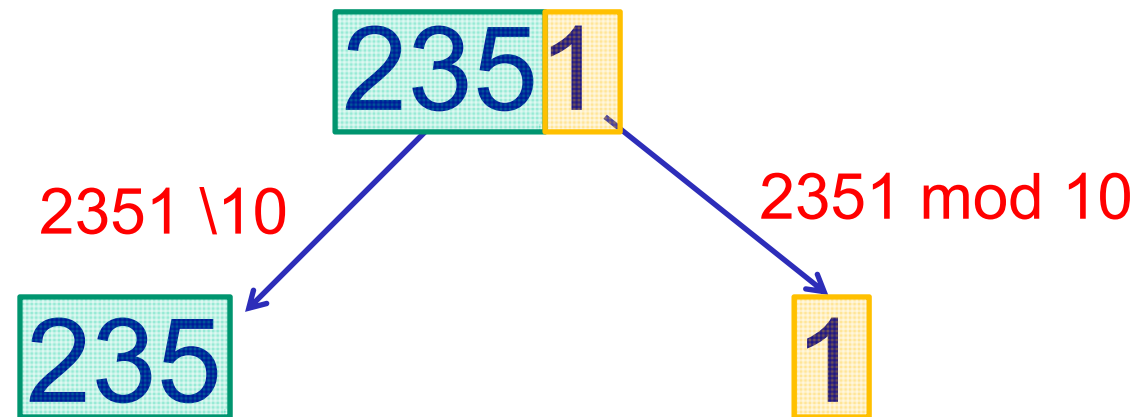


## Analyse

- Quersumme = Summe der Dezimalziffern

$$2351 \rightarrow 2+3+5+1 = 11$$

Wie kommen wir an die einzelnen Ziffern?





➤ Zahl
Quersumme = 0
Letzte = Letzte Ziffer von Zahl
Quersumme = Quersumme + Letzte
Zahl = vorderer Teil der Zahl

2351

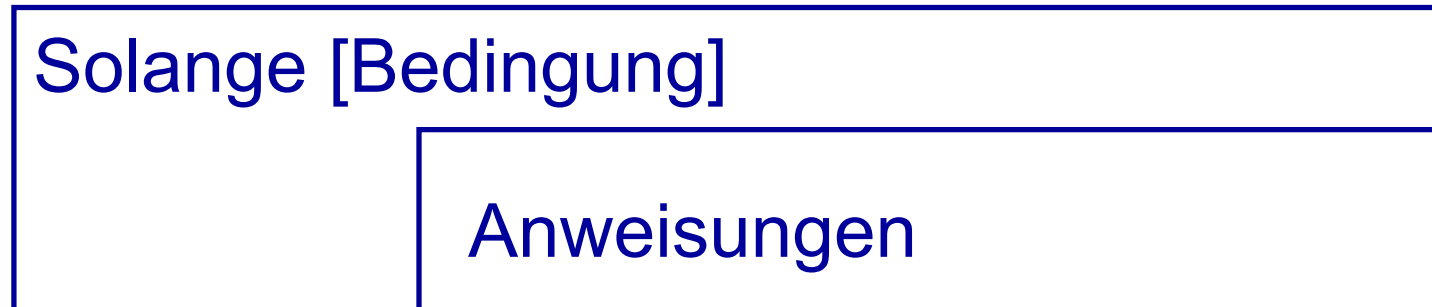
1

235

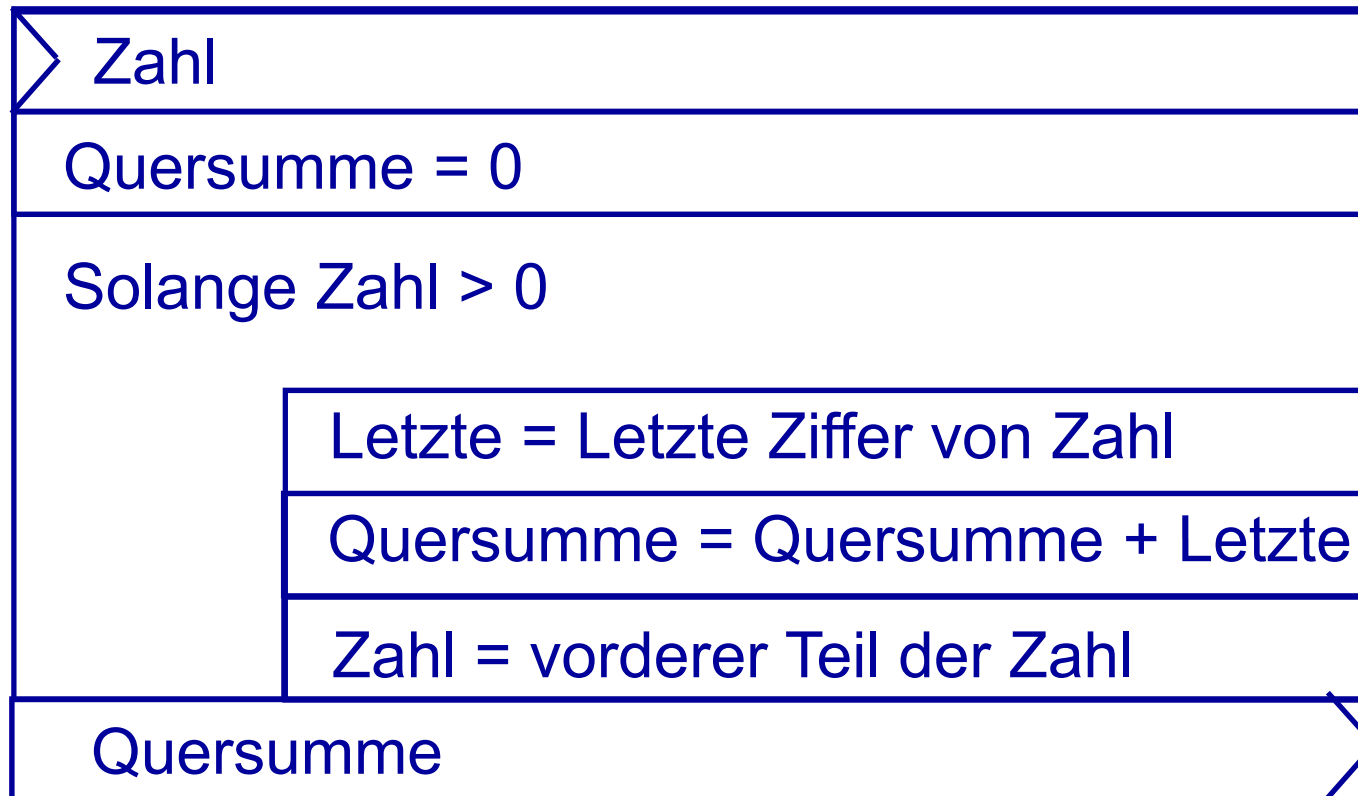
MUSS WIEDERHOLT WERDEN, SOLANGE NOCH  
"ZIFFERN DA SIND"



## Schleife "solange Bedingung gilt, führe aus"

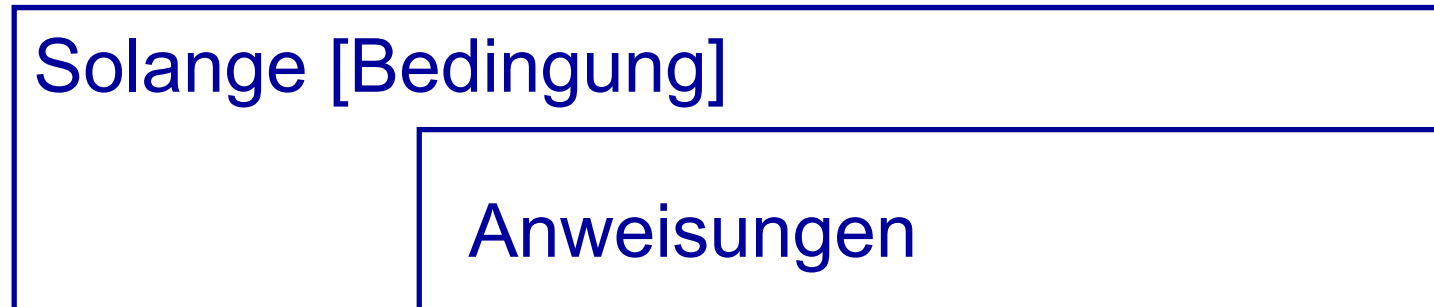


Die Anweisungen werden (wiederholt) ausgeführt, so lange die Bedingung wahr ist.





## Schleife "solange Bedingung gilt, führe aus"



**Do While Bedingung**

**Anweisungen**

**Loop**



```
Sub quersumme()
```

```
Dim zahl As Integer, letzte As Integer, quersumme  
    As Integer
```

```
zahl = Val(InputBox("Zahl eingeben:"))
```

```
quersumme = 0
```

```
Do While zahl > 0
```

```
    letzte = zahl Mod 10
```

```
    quersumme = quersumme + letzte
```

```
    zahl = zahl \ 10
```

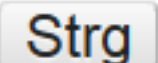
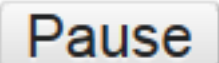
```
Loop
```

```
MsgBox ("Die Quersumme ist" & quersumme)
```

```
End Sub
```



## Wir merken uns:

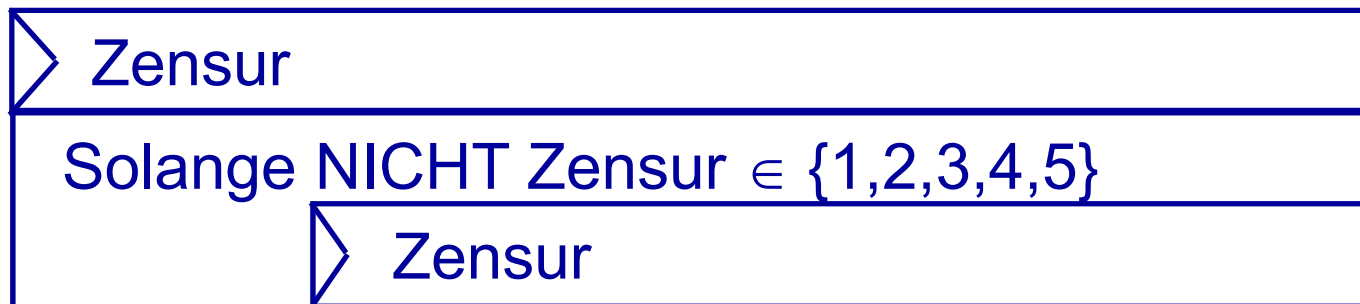
- Eine Schleife der Form Do While ... Loop benutzen wir, wenn Anweisungen wiederholt auszuführen sind, solange eine bestimmte Bedingung gilt.
- Die Bedingung wird getestet, bevor die Anweisungen ausgeführt werden. Ist die Bedingung gleich zu Anfang falsch, werden diese Anweisungen gar nicht ausgeführt.
- Der Programmierer muss dafür sorgen, dass die Bedingung irgendwann falsch wird, sonst läuft das Programm in einer Endlosschleife.
- Endlosschleife abbrechen:  



## Aufgabe:

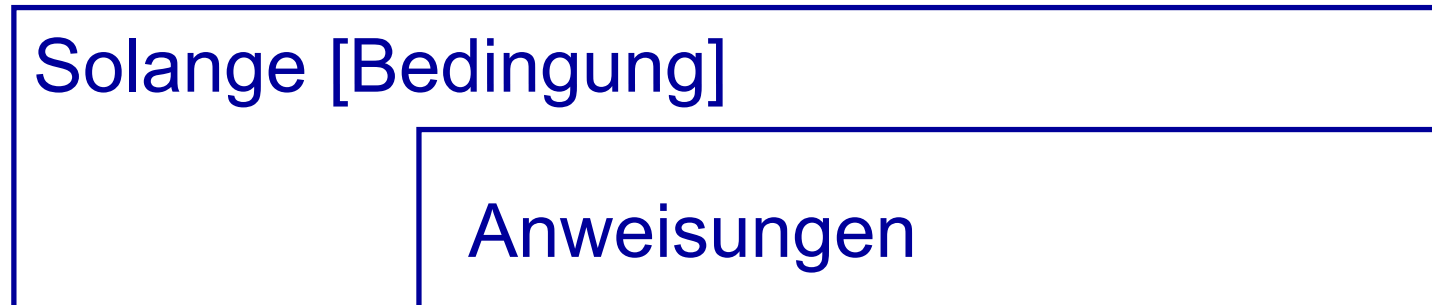
Das Programm fordert zur Eingabe einer Zensur (einer ganzen Zahl von 1 bis 5) auf.

Wird eine ganze Zahl eingegeben, die nicht in diesen Bereich fällt, muss die Eingabe wiederholt werden, und zwar so lange, bis eine korrekte Zensur eingegeben wurde.

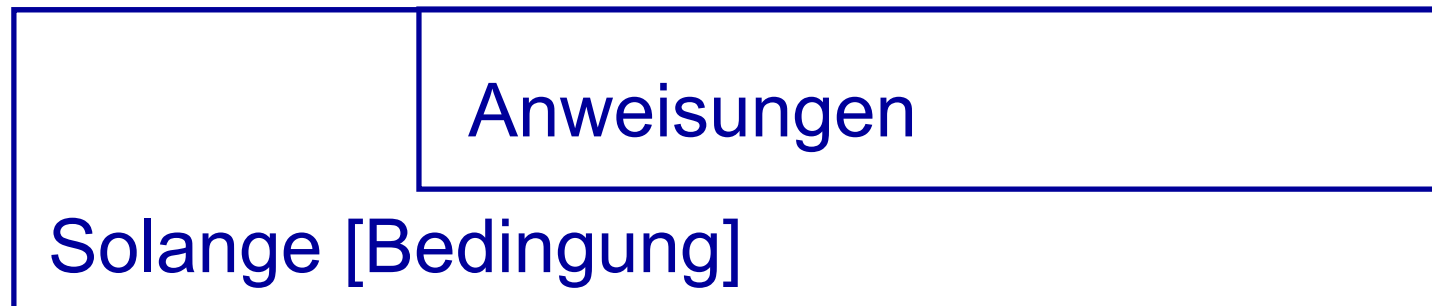




## Kopf- und fußgesteuerte Schleife



erst Prüfung, dann ggf. Anweisungen ausführen, dann das Ganze von vorne.  
Es kann passieren, dass die Anweisungen nie ausgeführt werden.



erst Anweisungen ausführen, dann prüfen (ob die Anweisungen noch einmal ausgeführt werden sollen).  
Die Anweisungen werden immer mindestens einmal ausgeführt.



▷ Zensur

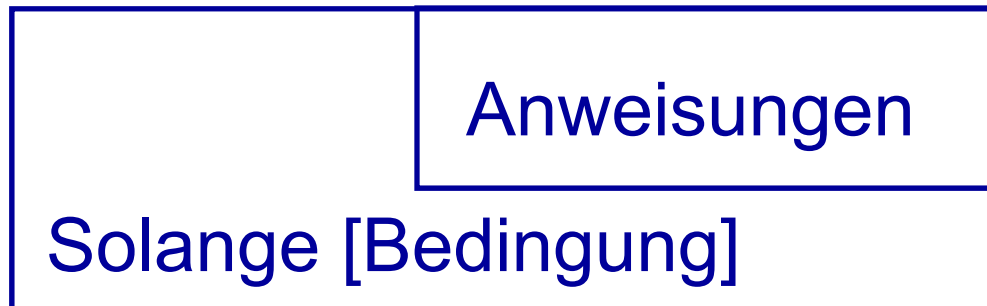
Solange NICHT Zensur  $\in \{1,2,3,4,5\}$



## Kopf- und fußgesteuerte Schleife



Do While Bedingung  
Anweisungen  
Loop



Do  
Anweisungen  
Loop While Bedingung



```
Sub zensureneingabe()
```

```
Dim zensur As Integer
```

```
Do
```

```
    zensur = Val(InputBox("Zensur eingeben:"))
```

```
Loop While zensur < 1 Or zensur > 5
```

```
MsgBox ("Eingegeben wurde: " & Str(zensur))
```

```
End Sub
```



## Zusammenfassung: drei Schleifen

Zähle [Variable] von [Start] bis [Ende], Schrittweite [s]

Anweisungen

For Zählervariable = Anfangswert To Endwert Step s

Anweisungen

Next Zählervariable



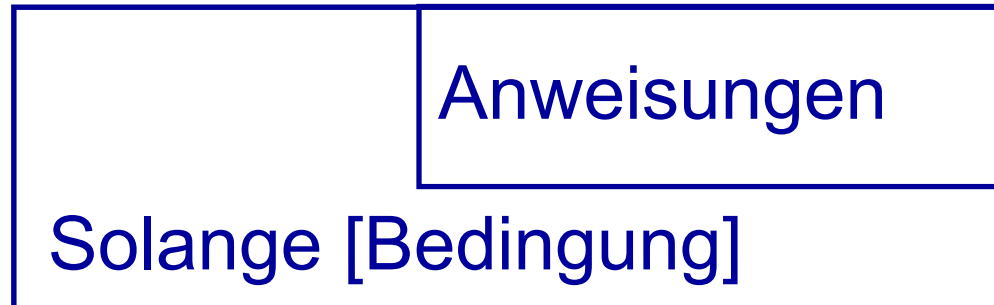
## Zusammenfassung: drei Schleifen



Do While Bedingung  
Anweisungen  
Loop



## Zusammenfassung: drei Schleifen



Do

Anweisungen

Loop While Bedingung



## Merksätze: drei Schleifen

- Die For-Schleife ist zu bevorzugen, wenn die Zahl der Durchläufe feststeht.
- Soll es möglich sein, dass die Schleife nie durchlaufen wird, muss eine kopfgesteuerte Schleife genutzt werden.
- Vermeidung von Endlosschleifen: In einer Do-Loop-Schleife muss etwas geschehen, so dass die Bedingung falsch werden kann.

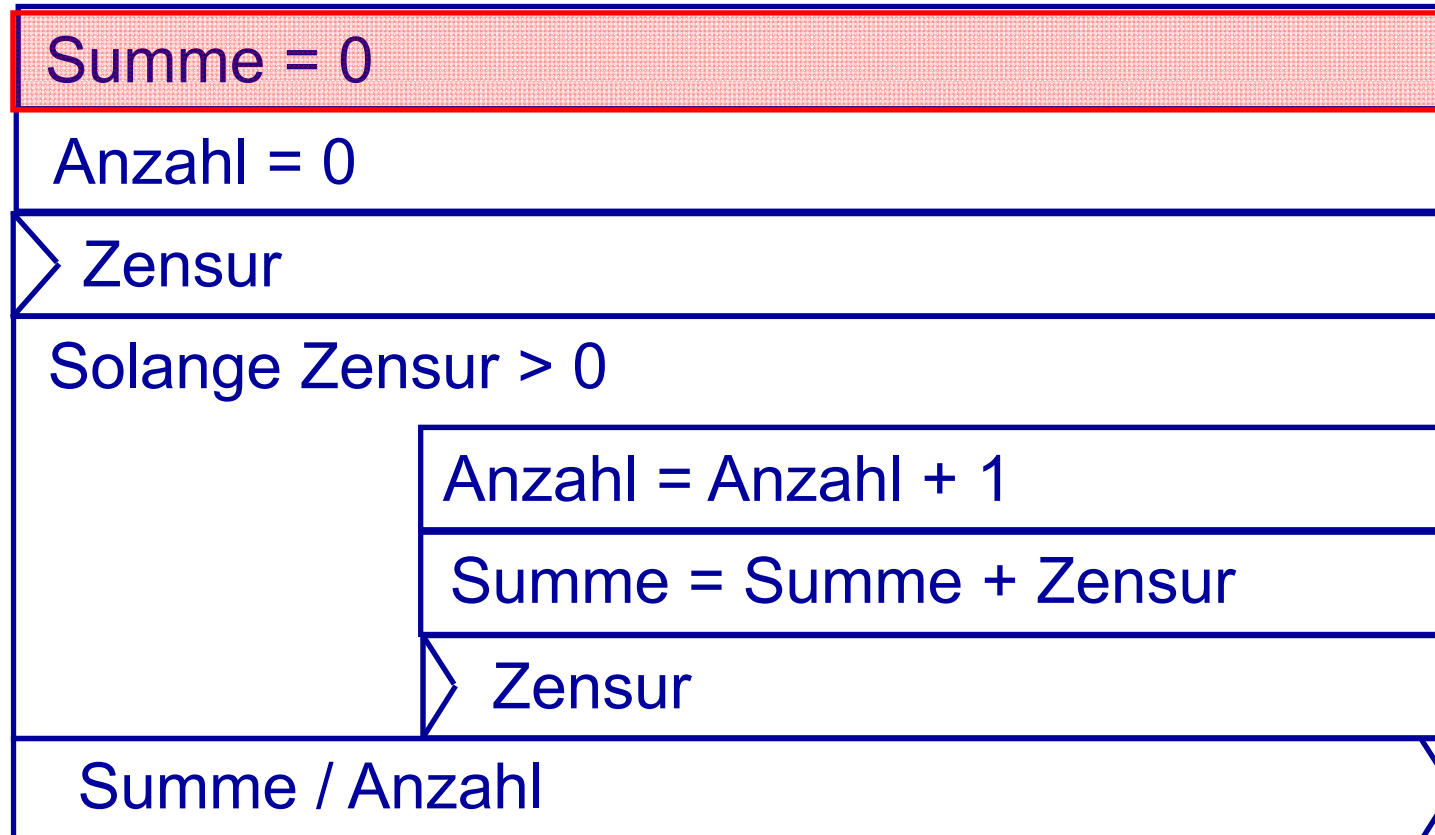


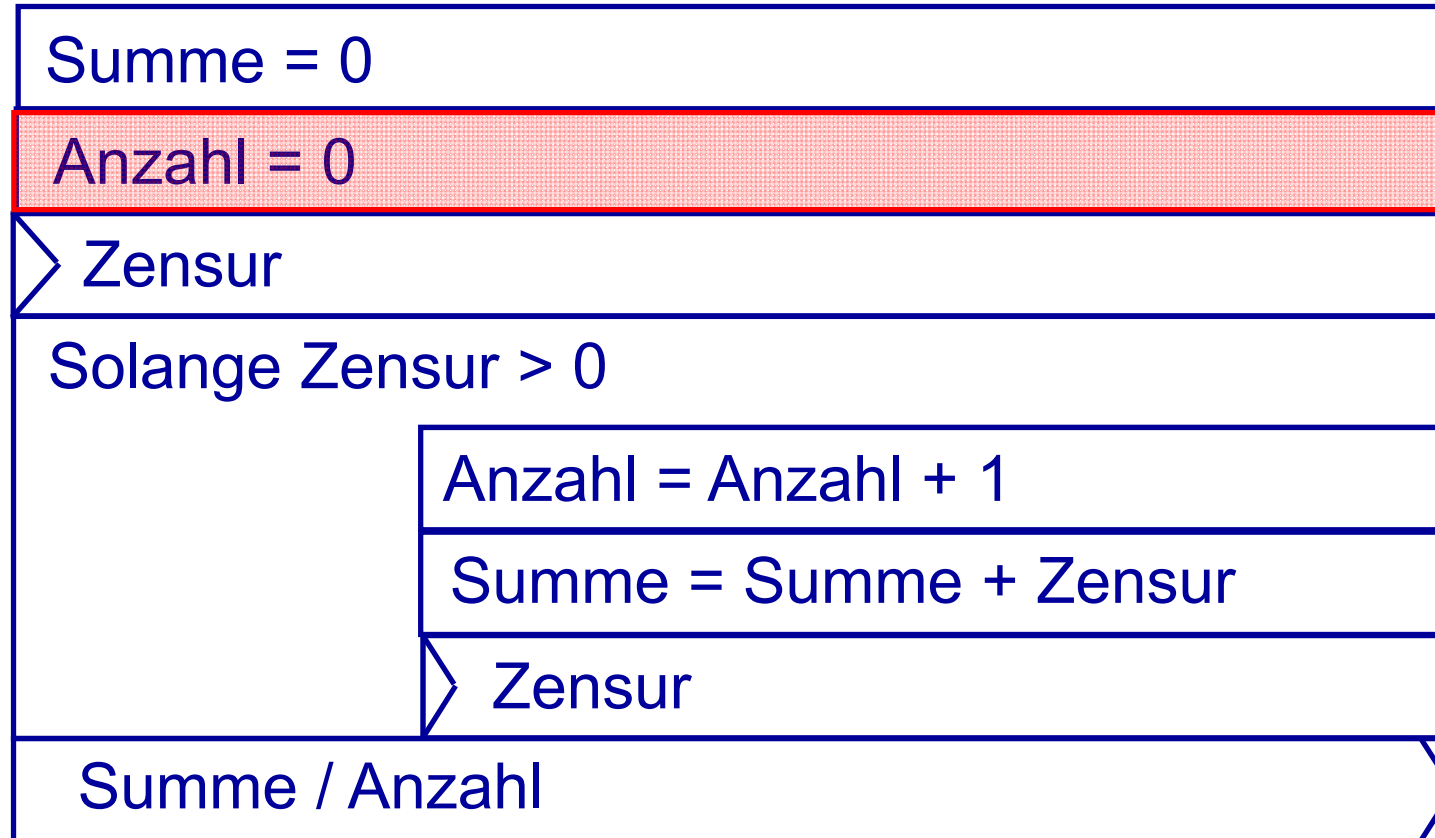
## Aufgabe:

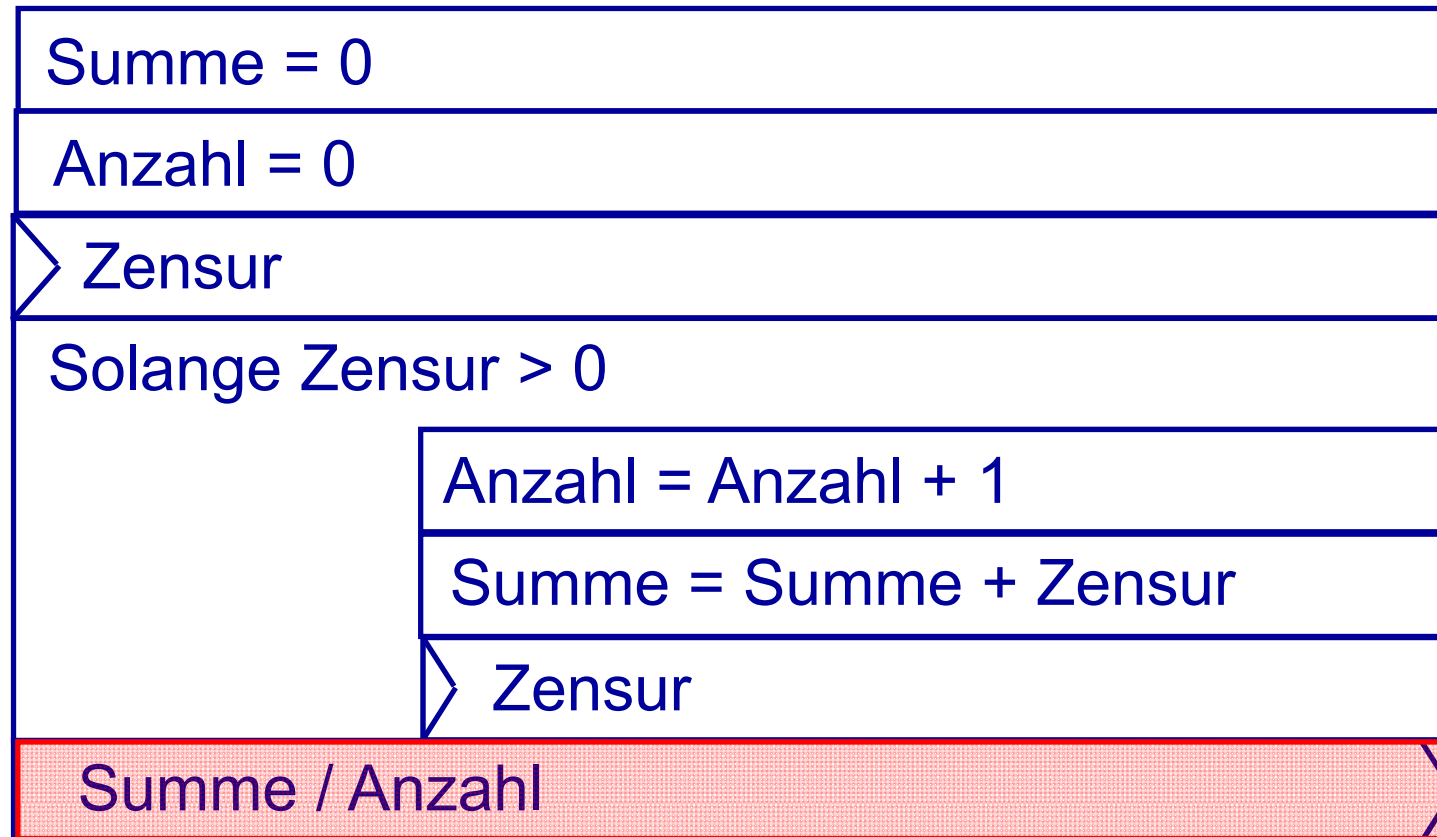
Es sollen die in einer Prüfung erzielten Schulnoten (von 1 bis 5) eingegeben werden.

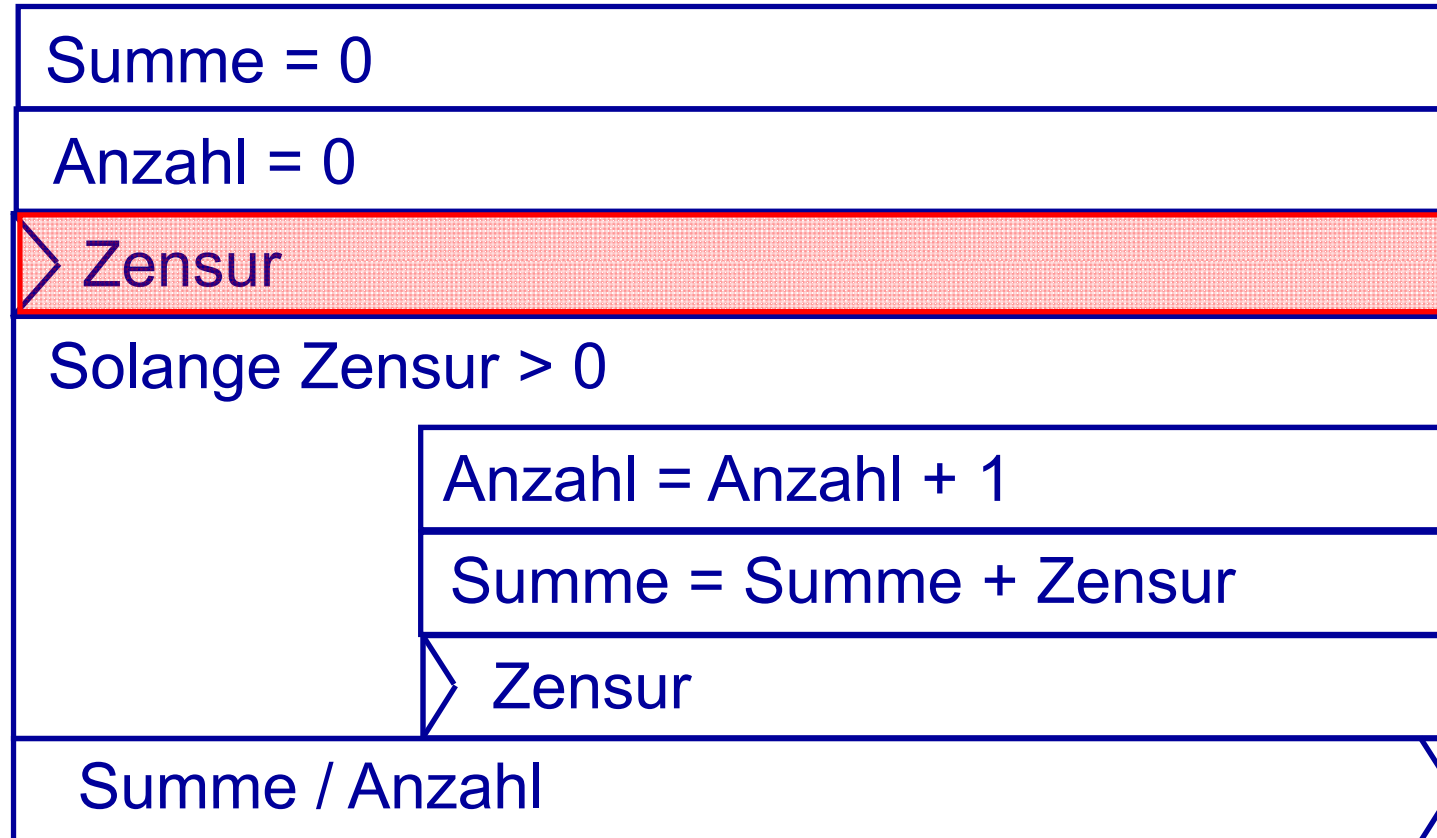
Durch die Eingabe der Zahl 0 teilt der Lehrer dem Programm mit, dass keine weiteren Eingaben folgen.

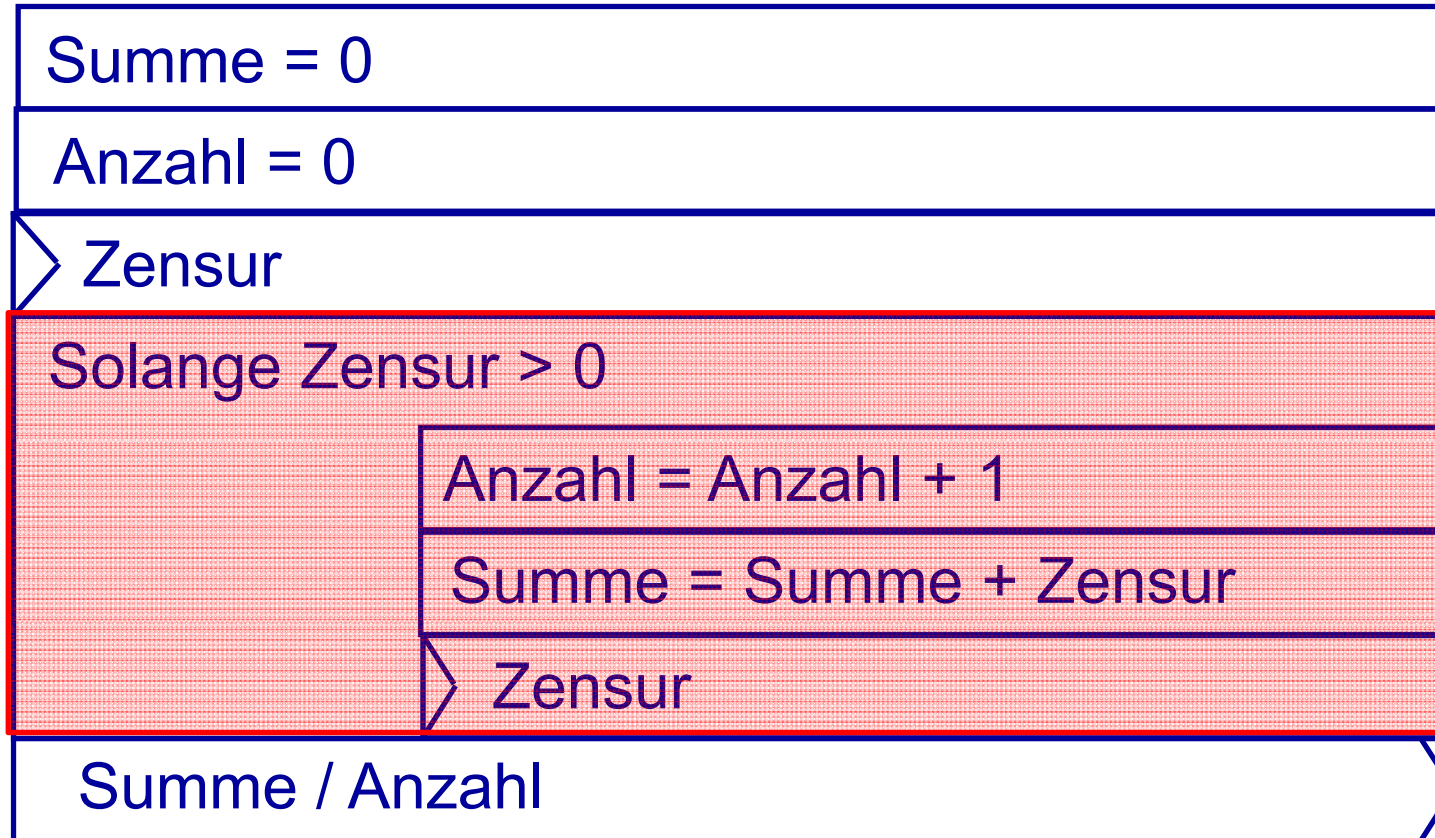
Das Programm soll dann die Durchschnittsnote, gerundet auf eine Stelle nach dem Komma ausgeben.

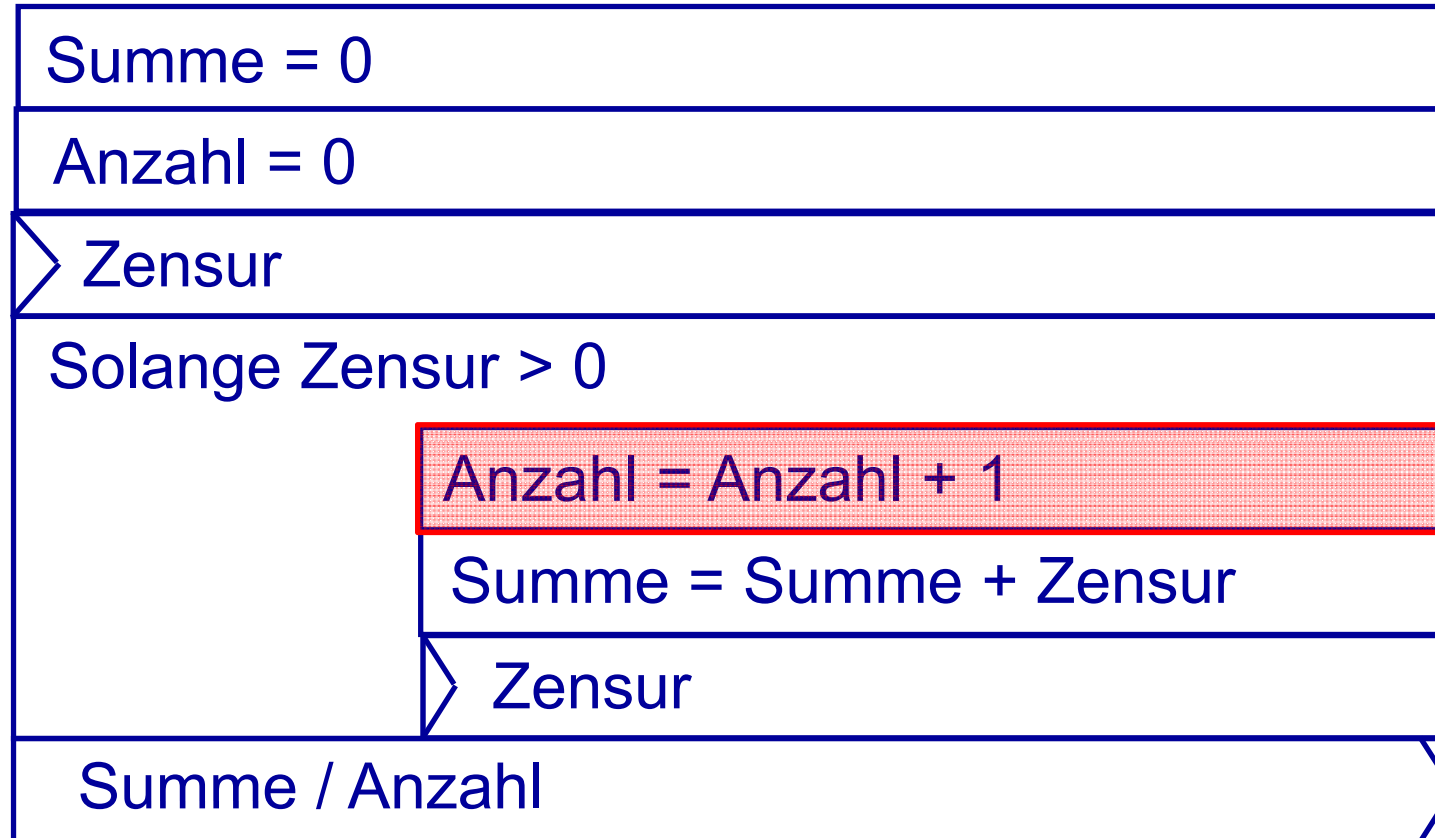


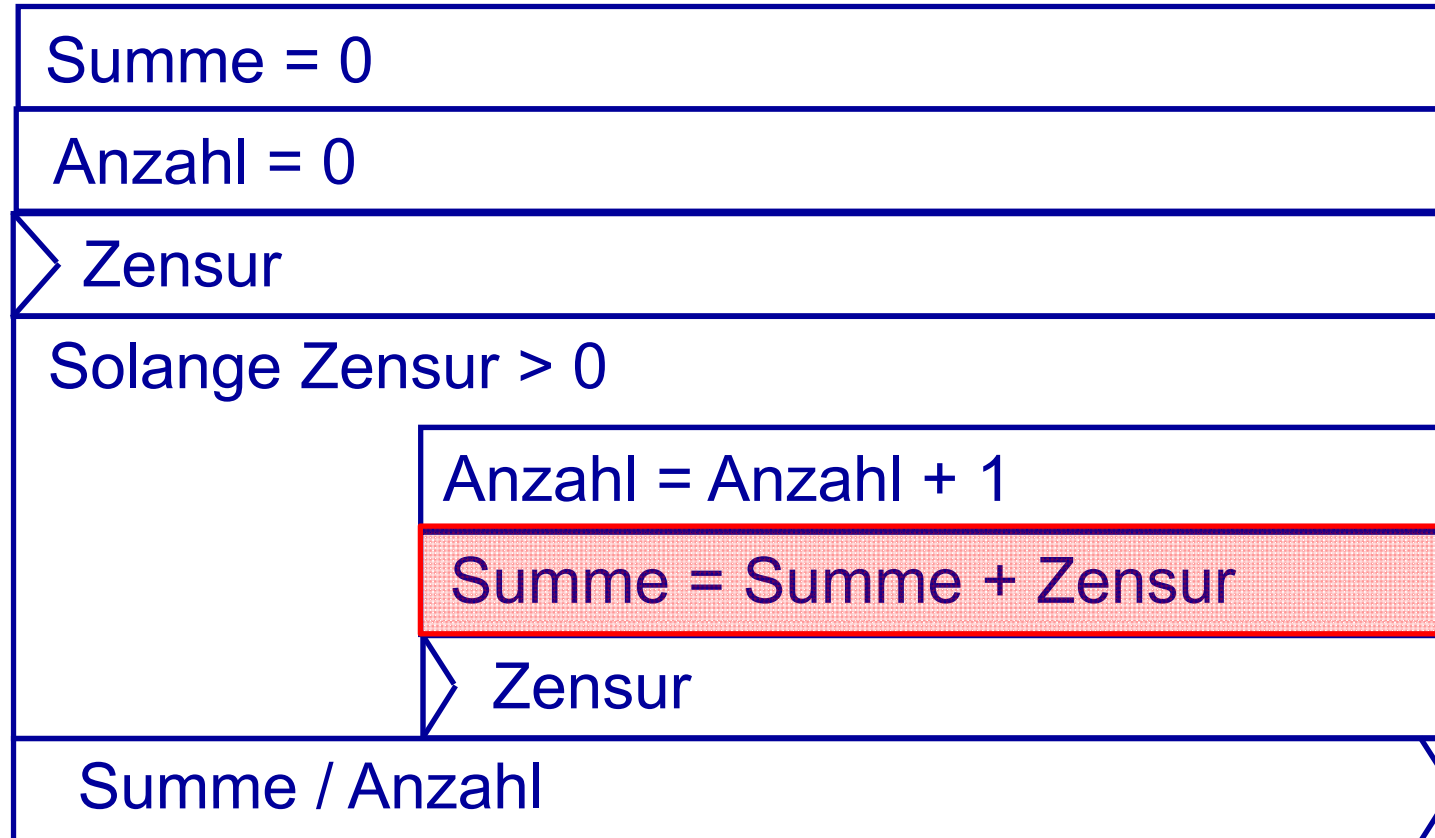


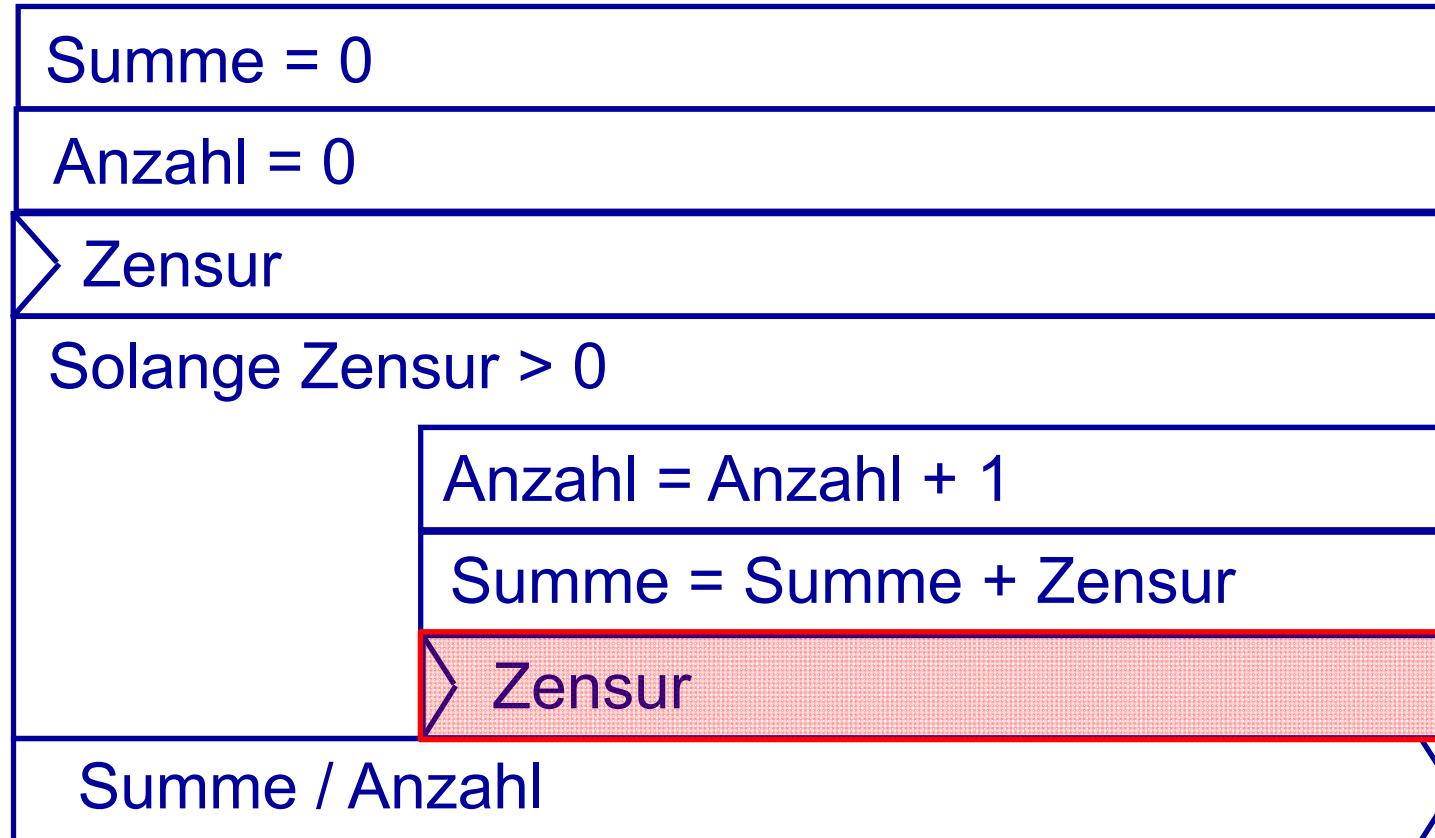














Summe = 0

Anzahl = 0

▷ Zensur

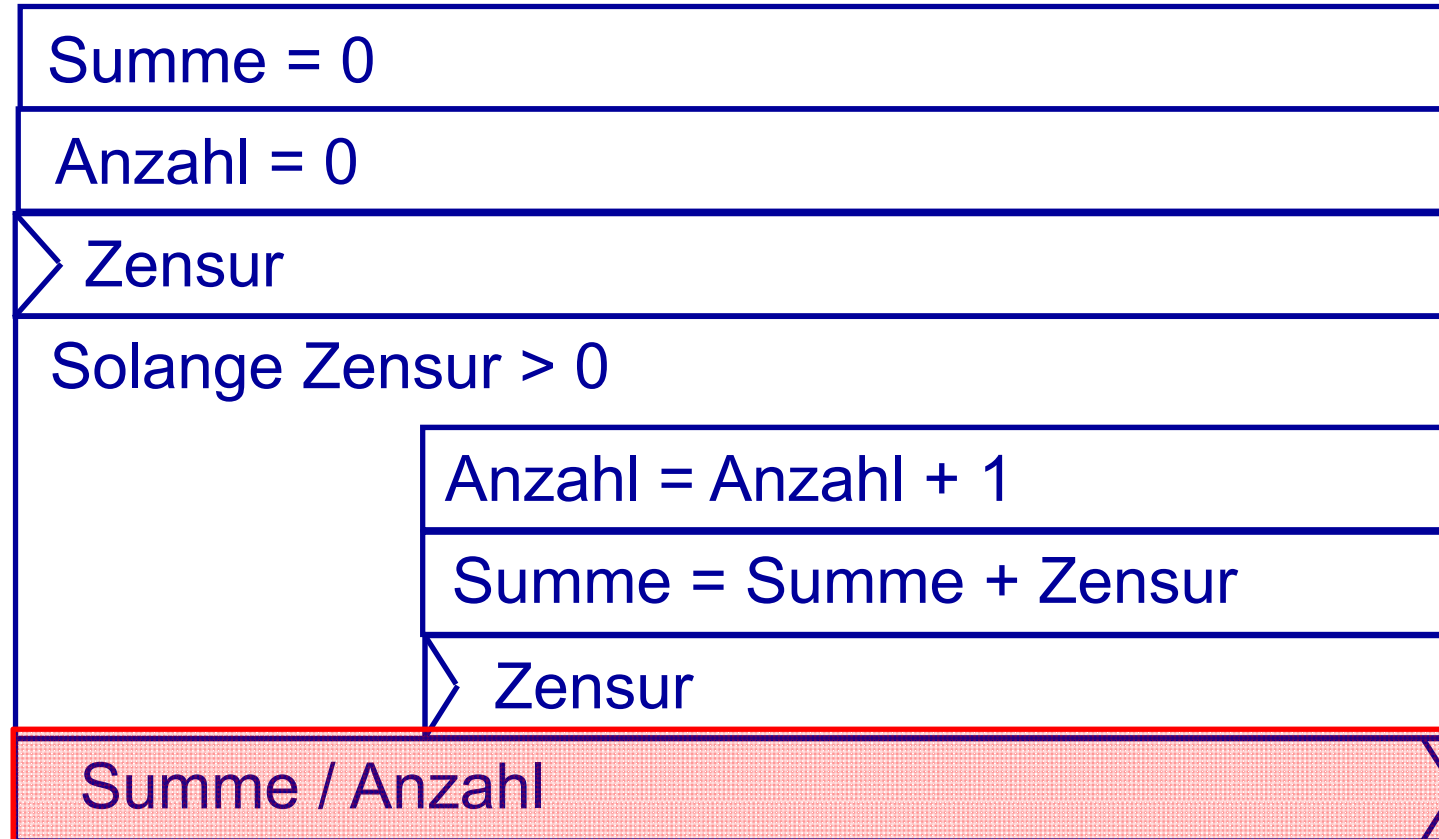
Solange Zensur > 0

Anzahl = Anzahl + 1

Summe = Summe + Zensur

▷ Zensur

Summe / Anzahl





```
Sub zensuren()
```

```
Dim summe As Integer, zensur As Integer, anzahl As Integer
```

```
zensur = Val(InputBox("Zensur eingeben:"))
```

```
Do While zensur > 0
```

```
    anzahl = anzahl + 1
```

```
    summe = summe + zensur
```

```
    zensur = Val(InputBox("Zensur eingeben:"))
```

```
Loop
```

```
MsgBox ("Zensurendurchschnitt: " & (summe / anzahl))
```

```
End Sub
```

Anweisung muss doppelt geschrieben werden.



Summe = 0

Anzahl = 0

Zensur

Anzahl = Anzahl + 1

Summe = Summe + Zensur

Solange Zensur > 0

Summe / (Anzahl-1)



```
Sub zensuren2()
```

```
Dim summe As Integer, zensur As Integer, anzahl  
As Integer
```

```
Do
```

```
    zensur = Val(InputBox("Zensur eingeben:"))
```

```
    anzahl = anzahl + 1
```

```
    summe = summe + zensur
```

```
Loop While zensur > 0
```

```
MsgBox ("Zensurendurchschnitt: " & (summe /  
    (anzahl - 1)))
```

```
End Sub
```



## Häufiger Anwendungsfall: Prüfung auf Korrektheit der Eingaben



Im Beispiel:

Do

```
zensur = Val(TextBox("Zensur eingeben:"))
```

```
Loop While zensur < 1 Or zensur > 5
```

oder:

Do

```
zensur = Val(TextBox("Zensur eingeben:"))
```

```
Loop While Not (zensur >=1 And zensur <=5)
```

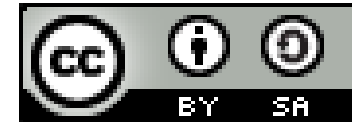


## Klausuraufgabe 3

```
eingabe = InputBox("Ist der Gast  
Klubmitglied?")
```

```
If eingabe = "ja" Or eingabe ="j" Then  
    mitglied = true
```

- Es soll sichergestellt werden, dass die Antwort auf jede Frage nur "ja" (bzw. "j") oder "nein" (bzw. "n") lauten darf. Gibt der Benutzer etwas anderes ein, muss er die Eingabe wiederholen. Wie müssen Sie das Programm ändern, um diese Forderung umzusetzen?
- Notieren Sie den notwendigen VBA-Code am Beispiel der ersten Frage "Ist der Gast Klubmitglied?"



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/>