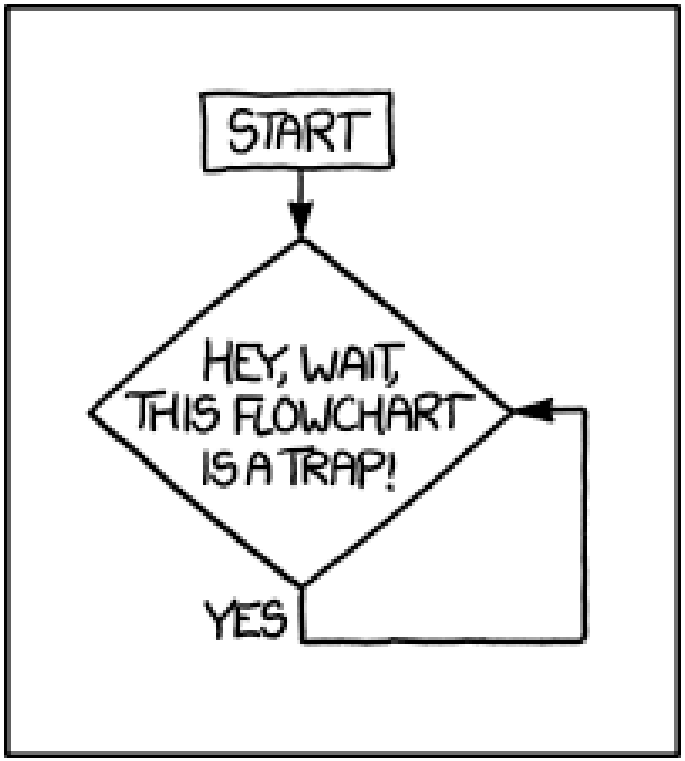
Schleifen, Diagramme

Vorkurs Informatik

Universität Stuttgart
MINT-Kolleg Baden-Württemberg

Universität Stuttgart
Karlsruher Institut für Technologie
F. Schweiner
MINT-KOLLEG BADEN-WÜRTTEMBERG

Getting started ...



©Randall Munroe, http://xkcd.com/1195/

Outline

(nach Folien von L. Vettin, V. Weidler, W. Kessler)


Creative Commons Attribution-NonCommercial-ShareAlike 4.0 Licence

Outline

Aufgabe: Summenberechnung

Berechnen Sie die Summe der natürlichen Zahlen von 1 bis n:

$$\sum_{i=1}^{n} i$$

Summe von 1 bis 1: 1
Summe von 1 bis 2: $1 + 2 = 3$

Aufgabe: Summenberechnung

Berechnen Sie die Summe der natürlichen Zahlen von 1 bis n:

$$\sum_{i=1}^{n} i$$

Summe von 1 bis 1: 1
Summe von 1 bis 2: $1 + 2 = 3$
Summe von 1 bis 3: $1 + 2 + 3 = 6$

Aufgabe: Summenberechnung

Berechnen Sie die Summe der natürlichen Zahlen von 1 bis n:

$$\sum_{i=1}^{n} i$$

Summe von 1 bis 1: 1
Summe von 1 bis 2: $1 + 2 = 3$
Summe von 1 bis 3: $1 + 2 + 3 = 6$
Summe von 1 bis 4: $1 + 2 + 3 + 4 = 10$

Aufgabe: Summenberechnung

Berechnen Sie die Summe der natürlichen Zahlen von 1 bis n:

$$\sum_{i=1}^{n} i$$

Summe von 1 bis 1: 1
Summe von 1 bis 2: $1 + 2 = 3$
Summe von 1 bis 3: $1 + 2 + 3 = 6$
Summe von 1 bis 4: $1 + 2 + 3 + 4 = 10$
Summe von 1 bis 5: $1 + 2 + 3 + 4 + 5 = 15$

Aufgabe: Summenberechnung

Berechnen Sie die Summe der natürlichen Zahlen von 1 bis n:

$$\sum_{i=1}^{n} i$$

Summe von 1 bis 1: 1
Summe von 1 bis 2: $1 + 2 = 3$
Summe von 1 bis 3: $1 + 2 + 3 = 6$
Summe von 1 bis 4: $1 + 2 + 3 + 4 = 10$
Summe von 1 bis 5: $1 + 2 + 3 + 4 + 5 = 15$
Summe von 1 bis 6: $1 + 2 + 3 + 4 + 5 + 6 = 21$
...

MINT-KOLLEG BADEN-WÜRTTEMBERG

Programm für Summenberechnung (1. Versuch)

```
int sum2 = 1+2;
System.out.println("The sum from 1 to 2 is: " + sum2);
```

Programm für Summenberechnung (1. Versuch)

```
int sum2 = 1+2;
System.out.println("The sum from 1 to 2 is: " + sum2);

int sum3 = sum2+3;
System.out.println("The sum from 1 to 3 is: " + sum3);
```

Programm für Summenberechnung (1. Versuch)

```
int sum2 = 1+2;
System.out.println("The sum from 1 to 2 is: " + sum2);

int sum3 = sum2+3;
System.out.println("The sum from 1 to 3 is: " + sum3);

int sum4 = sum3+4;
System.out.println("The sum from 1 to 4 is: " + sum4);
```

Programm für Summenberechnung (1. Versuch)

```
int sum2 = 1+2;
System.out.println("The sum from 1 to 2 is: " + sum2);

int sum3 = sum2+3;
System.out.println("The sum from 1 to 3 is: " + sum3);

int sum4 = sum3+4;
System.out.println("The sum from 1 to 4 is: " + sum4);

int sum5 = sum4+5;
System.out.println("The sum from 1 to 5 is: " + sum5);
```


mint-KOLLEG BADEN-WÜRTTEMBERG

Programm für Summenberechnung (1. Versuch)

```
int sum2 = 1+2;
System.out.println("The sum from 1 to 2 is: " + sum2);

int sum3 = sum2+3;
System.out.println("The sum from 1 to 3 is: " + sum3);

int sum4 = sum3+4;
System.out.println("The sum from 1 to 4 is: " + sum4);

int sum5 = sum4+5;
System.out.println("The sum from 1 to 5 is: " + sum5);
```

Extrem umständlich für größere Werte!!

Schleifen als Kontrollstrukturen

Schleifen

Eine Schleife ist eine Anweisung, die einen Codeabschnitt mehrere Male hintereinander ausführen kann.

Schleifen als Kontrollstrukturen

Schleifen

Eine Schleife ist eine Anweisung, die einen Codeabschnitt mehrere Male hintereinander ausführen kann.

- Schleifen erlauben es, einen Codeblock mit beliebigen Anweisungen mehrfach zu durchlaufen.
- Es muss nicht unbedingt in jedem Durchlauf exakt das gleiche getan werden (Stichwort: `if`).
- Die Anzahl der Durchläufe wird erst zur Laufzeit festgelegt.
- Es gibt verschiedene Arten von Schleifen: `for`, `while`.

Outline

for-Schleife

for-Schleife

Eine for-Schleife verwendet man, wenn ein Codeblock eine bestimmte Anzahl oft durchlaufen werden soll.

```
for ( int i=0; i<5; i=i+1 ) {
   System.out.println("loop in action, i = " + i );
}
```

for-Schleife

for-Schleife

Eine `for`-Schleife verwendet man, wenn ein Codeblock eine bestimmte Anzahl oft durchlaufen werden soll.

```
for ( int i=0; i<5; i=i+1 ) {
   System.out.println("loop in action, i = " + i );
}
```

- `int i=0` wird vor dem ersten Durchlauf der Schleife ausgeführt.

mint-KOLLEG BADEN-WÜRTTEMBERG

`for`-Schleife

`for`-Schleife

Eine `for`-Schleife verwendet man, wenn ein Codeblock eine bestimmte Anzahl oft durchlaufen werden soll.

```
for ( int i=0; i<5; i=i+1 ) {
   System.out.println("loop in action, i = " + i );
}
```

- `int i=0` wird vor dem ersten Durchlauf der Schleife ausgeführt.
- `i<5` wird vor jedem Durchlauf geprüft. Die Schleife wird durchlaufen, wenn die Bedingung `true` ist.

mint-KOLLEG BADEN-WÜRTTEMBERG

`for`-Schleife

`for`-Schleife

Eine `for`-Schleife verwendet man, wenn ein Codeblock eine bestimmte Anzahl oft durchlaufen werden soll.

```
for ( int i=0; i<5; i=i+1 ) {
   System.out.println("loop in action, i = " + i );
}
```

- `int i=0` wird vor dem ersten Durchlauf der Schleife ausgeführt.
- `i<5` wird vor jedem Durchlauf geprüft. Die Schleife wird durchlaufen, wenn die Bedingung `true` ist.
- `i=i+1` wird vor dem nächsten Durchlauf ausgeführt.

mint-KOLLEG BADEN-WÜRTTEMBERG

for-Schleife – Ausgabe

Code:

```
for (int i=0; i<5; i=i+1) {
   System.out.println("loop in action, i = " + i );
}
```

Ausgabe:

```
loop in action, i = 0
```

for-Schleife – Ausgabe

Code:

```
for (int i=0; i<5; i=i+1) {
   System.out.println("loop in action, i = " + i );
}
```

Ausgabe:

```
loop in action, i = 0
loop in action, i = 1
```

`for`-Schleife – Ausgabe

Code:

```
for (int i=0; i<5; i=i+1) {
   System.out.println("loop in action, i = " + i );
}
```

Ausgabe:

```
loop in action, i = 0
loop in action, i = 1
loop in action, i = 2
```

`for`-Schleife – Ausgabe

Code:

```
for (int i=0; i<5; i=i+1) {
   System.out.println("loop in action, i = " + i );
}
```

Ausgabe:

```
loop in action, i = 0
loop in action, i = 1
loop in action, i = 2
loop in action, i = 3
```

for-Schleife – Ausgabe

Code:

```
for (int i=0; i<5; i=i+1) {
   System.out.println("loop in action, i = " + i );
}
```

Ausgabe:

```
loop in action, i = 0
loop in action, i = 1
loop in action, i = 2
loop in action, i = 3
loop in action, i = 4
```

Syntax `for`-Schleife

```
for ( <Initialisierung>; <Bedingung>; <Inkrement> ) {
   <Anweisung>;
   <Anweisung>;
   ...
}
```

- Die Initialisierung deklariert die Schleifenvariable (normalerweise vom Typ `int`) und gibt ihr einen Anfangswert.
- Die Schleifenbedingung ist vom Typ `boolean`.
- Das Inkrement ist eine Anweisung zur Veränderung des Werts der Schleifenvariablen nach jedem Durchlauf der Schleife.
- Guter Stil ist es, die Anweisungen in der Schleife einzurücken.

MINT-KOLLEG BADEN-WÜRTTEMBERG

Schleifenzähler

```
for ( int i=0; i<=10; i++ ) {
   System.out.println("loop in action, i = " + i );
}
System.out.println(i); // geht nicht!
```

- `i` heißt Schleifenvariable, Schleifenzähler oder Laufvariable.
- `i` wird innerhalb der Schleife deklariert und ist auch nur innerhalb der Schleife zugreifbar.
- Das Inkrement kann jeder beliebige Ausdruck sein, der `i` verändert, z. B. `i = i - 1`, `i = 2 * i`.
  Es ist guter Stil, nur `i++` zu verwenden!
- `i++` ist die Kurzschreibweise für `i = i + 1`. Analog entspricht `i--` dem Ausdruck `i = i - 1`.

Quiz: `for`-Schleifen

```
int result = 0;
for ( int i=5; i>0; i-- ) {
   result = result + (5-i);
}
System.out.println("result: " + result);
```

Welche Aussagen sind wahr?

- Am Anfang wird `i` auf 0 gesetzt.
- Bei jedem Durchlauf wird `i` um eins erhöht.
- Die Schleife wird genau 5 Mal durchlaufen.
- Am Ende hat `result` den Wert 10.
- Die letzte Zeile ist ein Fehler, da auf `result` nicht außerhalb der Schleife zugegriffen werden kann.

Quiz: `for`-Schleifen

```
int result = 0;
for ( int i=5; i>0; i-- ) {
   result = result + (5-i);
}
System.out.println("result: " + result);
```

Welche Aussagen sind wahr?

- Am Anfang wird `i` auf 0 gesetzt.
- Bei jedem Durchlauf wird `i` um eins erhöht.
- Die Schleife wird genau 5 Mal durchlaufen.
- Am Ende hat `result` den Wert 10.
- Die letzte Zeile ist ein Fehler, da auf `result` nicht außerhalb der Schleife zugegriffen werden kann.

Lösung: Die dritte und vierte Aussage sind wahr.

MINT-KOLLEG BADEN-WÜRTTEMBERG

Besseres Programm für Summenberechnung

```
int sum = 0;
for ( int i=1; i<5; i++ ) {
   sum = sum + i;
   System.out.println("The sum from 1 to " + i + " is " + sum );
}
```


mINT-KOLLEG BADEN-WÜRTTEMBERG

Besseres Programm für Summenberechnung

```
int sum = 0;
for ( int i=1; i<5; i++ ) {
   sum = sum + i;
   System.out.println("The sum from 1 to " + i + " is " + sum );
}
```

Ausgabe:

```
The sum from 1 to 1 is 1
```

Besseres Programm für Summenberechnung

```
int sum = 0;
for ( int i=1; i<5; i++ ) {
   sum = sum + i;
   System.out.println("The sum from 1 to " + i + " is " + sum );
}
```

Ausgabe:

```
The sum from 1 to 1 is 1
The sum from 1 to 2 is 3
```

Besseres Programm für Summenberechnung

```
int sum = 0;
for ( int i=1; i<5; i++ ) {
   sum = sum + i;
   System.out.println("The sum from 1 to " + i + " is " + sum );
}
```

Ausgabe:

```
The sum from 1 to 1 is 1
The sum from 1 to 2 is 3
The sum from 1 to 3 is 6
```


MINT-KOLLEG BADEN-WÜRTTEMBERG

Besseres Programm für Summenberechnung

```
int sum = 0;
for ( int i=1; i<5; i++ ) {
   sum = sum + i;
   System.out.println("The sum from 1 to " + i + " is " + sum );
}
```

Ausgabe:

```
The sum from 1 to 1 is 1
The sum from 1 to 2 is 3
The sum from 1 to 3 is 6
The sum from 1 to 4 is 10
```


MINT-KOLLEG BADEN-WÜRTTEMBERG

Besseres Programm für Summenberechnung

```
int sum = 0;
for ( int i=1; i<5; i++ ) {
   sum = sum + i;
}
System.out.println("The sum from 1 to 4 is " + sum );
```

Besseres Programm für Summenberechnung

```
int sum = 0;
for ( int i=1; i<5; i++ ) {
   sum = sum + i;
}
System.out.println("The sum from 1 to 4 is " + sum );
```

Ausgabe:

```
The sum from 1 to 4 is 10
```

Outline

while-Schleife

while-Schleife

Eine while-Schleife verwendet man, wenn ein Codeblock so lange durchlaufen werden soll, bis eine Bedingung nicht mehr zutrifft.

```
int i = 1; // Initialisiere i
while ( i<5 ) {
   System.out.println("loop in action, i = " + i );
   i = i + 1; // Erhöhe i
}
```


mint-KOLLEG BADEN-WÜRTTEMBERG

`while`-Schleife

`while`-Schleife

Eine `while`-Schleife verwendet man, wenn ein Codeblock so lange durchlaufen werden soll, bis eine Bedingung nicht mehr zutrifft.

```
int i = 1; // Initialisiere i
while ( i<5 ) {
   System.out.println("loop in action, i = " + i );
   i = i + 1; // Erhöhe i
}
```

- `i<5` wird vor jedem Durchlauf geprüft. Die Schleife wird durchlaufen, wenn die Bedingung `true` ist.

MINT-KOLLEG BADEN-WÜRTTEMBERG

while-Schleife

while-Schleife

Eine `while`-Schleife verwendet man, wenn ein Codeblock so lange durchlaufen werden soll, bis eine Bedingung nicht mehr zutrifft.

```
int i = 1; // Initialisiere i
while ( i<5 ) {
   System.out.println("loop in action, i = " + i );
   i = i + 1; // Erhöhe i
}
```

- `i<5` wird vor jedem Durchlauf geprüft. Die Schleife wird durchlaufen, wenn die Bedingung `true` ist.
- Initialisieren und Verändern der Laufvariable von Hand.

MINT-KOLLEG BADEN-WÜRTTEMBERG

Syntax `while`-Schleife

```
while ( <Bedingung> ) {
   <Anweisung>;
   <Anweisung>;
   ...
}
```

- Die Schleifenbedingung ist vom Typ `boolean`.
- Guter Stil ist es, die Anweisungen in der Schleife einzurücken.

MINT-KOLLEG BADEN-WÜRTTEMBERG

Summenberechnung mit `while`-Schleife

```
int i = 1; // Initialisiere i
int sum = 0;

while ( i<5 ) {
   sum = sum + i;
   System.out.println("The sum from 1 to " + i + " is " + sum );
   i = i+1; // Erhöhe i
}
```

- Selbes Verfahren wie bei der `for`-Schleife.
- Zusätzlich Verwaltung der Schleifenvariable `i` von Hand.

Warum `while`-Schleifen?

Nützlich, wenn man nicht genau weiß, wie oft man die Schleife durchlaufen muss, um das gewünsche Ergebnis zu bekommen:

Wann wird die Summe größer als 100?

Warum `while`-Schleifen?

Nützlich, wenn man nicht genau weiß, wie oft man die Schleife durchlaufen muss, um das gewünsche Ergebnis zu bekommen:

Wann wird die Summe größer als 100?

```
int i = 1; // Initialisiere i
int sum = 0;

while ( sum < 100 ) {
   sum = sum + i;
   i = i+1; // Erhoehe i
}
System.out.println("The sum exceeds 100 for i=" + i );
```


mint-KOLLEG BADEN-WÜRTTEMBERG

`for` vs. `while`

- Jede `for`-Schleife kann durch eine `while`-Schleife ersetzt werden, aber nicht jede `while`-Schleife kann durch eine `for`-Schleife ersetzt werden.
- Bei einer `for`-Schleife steht die Anzahl der Wiederholungen schon vor Ausführung der Schleife fest, bei `while`-Schleifen nicht.

Negativbeispiele für `while`-Schleifen

```
while ( true ) { // Hört nie auf zu laufen
   System.out.println("... runs ... and runs ...");
}
```


MINT-KOLLEG BADEN-WÜRTTEMBERG

Negativbeispiele für `while`-Schleifen

```
while ( true ) { // Hört nie auf zu laufen
   System.out.println("... runs ... and runs ...");
}
```

```
int i = 0;
while ( i > 0 ) { // Wird nie durchlaufen
   System.out.println(i);
   i = i + 1;
}
```

Negativbeispiele für `while`-Schleifen

```
while ( true ) { // Hört nie auf zu laufen
   System.out.println("... runs ... and runs ...");
}
```

```
int i = 0;
while ( i > 0 ) { // Wird nie durchlaufen
   System.out.println(i);
   i = i + 1;
}
```

```
int i = 10;
while ( i > 0 ) { // Hört nie auf zu laufen
   System.out.println(i);
   i = i + 1;
}
```


MINT-KOLLEG BADEN-WÜRTTEMBERG

Quiz: `while`-Schleifen

```
int counter = 30;
while ( counter >= 10 ) {
   if ( counter % 5 == 0 ) {
      System.out.println(counter + " is divisible by 5");
   }
   counter = counter - 1;
}
```

Welche Aussagen sind wahr?

- Die Schleife läuft unendlich lang.
- Die Schleife wird nie betreten.
- Die Schleife wird genau 5 Mal durchlaufen.
- Es wird genau 5 Mal der Text ausgegeben.

Quiz: `while`-Schleifen

```
int counter = 30;
while ( counter >= 10 ) {
   if ( counter % 5 == 0 ) {
      System.out.println(counter + " is divisible by 5");
   }
   counter = counter - 1;
}
```

Welche Aussagen sind wahr?

- Die Schleife läuft unendlich lang.
- Die Schleife wird nie betreten.
- Die Schleife wird genau 5 Mal durchlaufen.
- Es wird genau 5 Mal der Text ausgegeben.

Lösung: Die letzte Aussage ist wahr.

Outline

Schleifen in Schleifen

```
for ( int i=0; i<2; i++ ) {
   for ( int j=0; j<3; j++ ) {
      System.out.println("i = " + i + "; j = " + j + ";");
   }
}
```

Ausgabe:

```
i = 0; j = 0;
```

Schleifen in Schleifen

```
for ( int i=0; i<2; i++ ) {
   for ( int j=0; j<3; j++ ) {
      System.out.println("i = " + i + "; j = " + j + ";");
   }
}
```

Ausgabe:

```
i = 0; j = 0;
i = 0; j = 1;
```

Schleifen in Schleifen

```
for ( int i=0; i<2; i++ ) {
   for ( int j=0; j<3; j++ ) {
      System.out.println("i = " + i + "; j = " + j + ";");
   }
}
```

Ausgabe:

```
i = 0; j = 0;
i = 0; j = 1;
i = 0; j = 2;
```

Schleifen in Schleifen

```
for ( int i=0; i<2; i++ ) {
   for ( int j=0; j<3; j++ ) {
      System.out.println("i = " + i + "; j = " + j + ";");
   }
}
```

Ausgabe:

```
i = 0; j = 0;
i = 0; j = 1;
i = 0; j = 2;
i = 1; j = 0;
```

Schleifen in Schleifen

```
for ( int i=0; i<2; i++ ) {
   for ( int j=0; j<3; j++ ) {
      System.out.println("i = " + i + "; j = " + j + ";");
   }
}
```

Ausgabe:

```
i = 0; j = 0;
i = 0; j = 1;
i = 0; j = 2;
i = 1; j = 0;
i = 1; j = 1;
```


MINT-KOLLEG BADEN-WÜRTTEMBERG

Schleifen in Schleifen

```
for ( int i=0; i<2; i++ ) {
   for ( int j=0; j<3; j++ ) {
      System.out.println("i = " + i + "; j = " + j + ";");
   }
}
```

Ausgabe:

```
i = 0; j = 0;
i = 0; j = 1;
i = 0; j = 2;
i = 1; j = 0;
i = 1; j = 1;
i = 1; j = 2;
```


MINT-KOLLEG BADEN-WÜRTTEMBERG

Schleifen in Schleifen

```
for ( int i=0; i<2; i++ ) {
   for ( int j=0; j<3; j++ ) {
      System.out.println("i = " + i + "; j = " + j + ";");
   }
}
```

Ausgabe:

```
i = 0; j = 0;
i = 0; j = 1;
i = 0; j = 2;
i = 1; j = 0;
i = 1; j = 1;
i = 1; j = 2;
```


mINT-KOLLEG BADEN-WÜRTTEMBERG

Quiz: Schleifen in Schleifen (1)

```
for ( int i=0; i<10; i++ ) {
   System.out.println("Hello");
   for ( int j=10; j>0; j=j-2 ) {
      System.out.println("World!");
   }
}
```

- Wie oft wird “Hello” ausgegeben?
- Wie oft wird “World!” ausgegeben?

Quiz: Schleifen in Schleifen (1)

```
for ( int i=0; i<10; i++ ) {
   System.out.println("Hello");
   for ( int j=10; j>0; j=j-2 ) {
      System.out.println("World!");
   }
}
```

- Wie oft wird “Hello” ausgegeben? (10 mal)
- Wie oft wird “World!” ausgegeben? (10*5= 50 mal)

Quiz: Schleifen in Schleifen (2)

```
for ( int i=0; i<5; i++ ) {
   System.out.println("Hello");
   for ( int j=i; j<5; j++ ) {
      System.out.println("World!");
   }
}
```

- Wie oft wird “Hello” ausgegeben?
- Wie oft wird “World” ausgegeben?

Quiz: Schleifen in Schleifen (2)

```
for ( int i=0; i<5; i++ ) {
   System.out.println("Hello");
   for ( int j=i; j<5; j++ ) {
      System.out.println("World!");
   }
}
```

- Wie oft wird “Hello” ausgegeben? (5 mal)
- Wie oft wird “World” ausgegeben? (5+4+3+2+1= 15 mal)

Outline

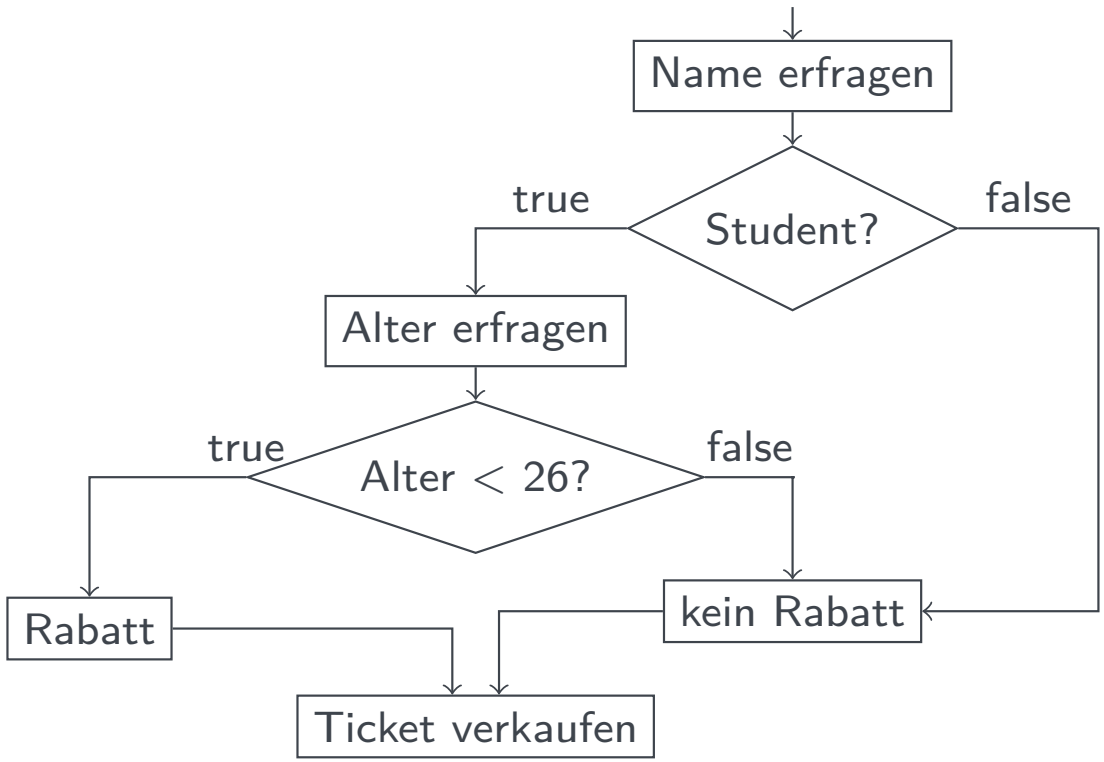
Ablaufdiagramme

- Diagramme veranschaulichen den Ablauf eines Programms.
- Der Inhalt ist programmiersprachenunabhängig und auf das Wesentliche konzentriert.
- Es gibt mehrere Arten von Diagrammen.

Ablaufdiagramme

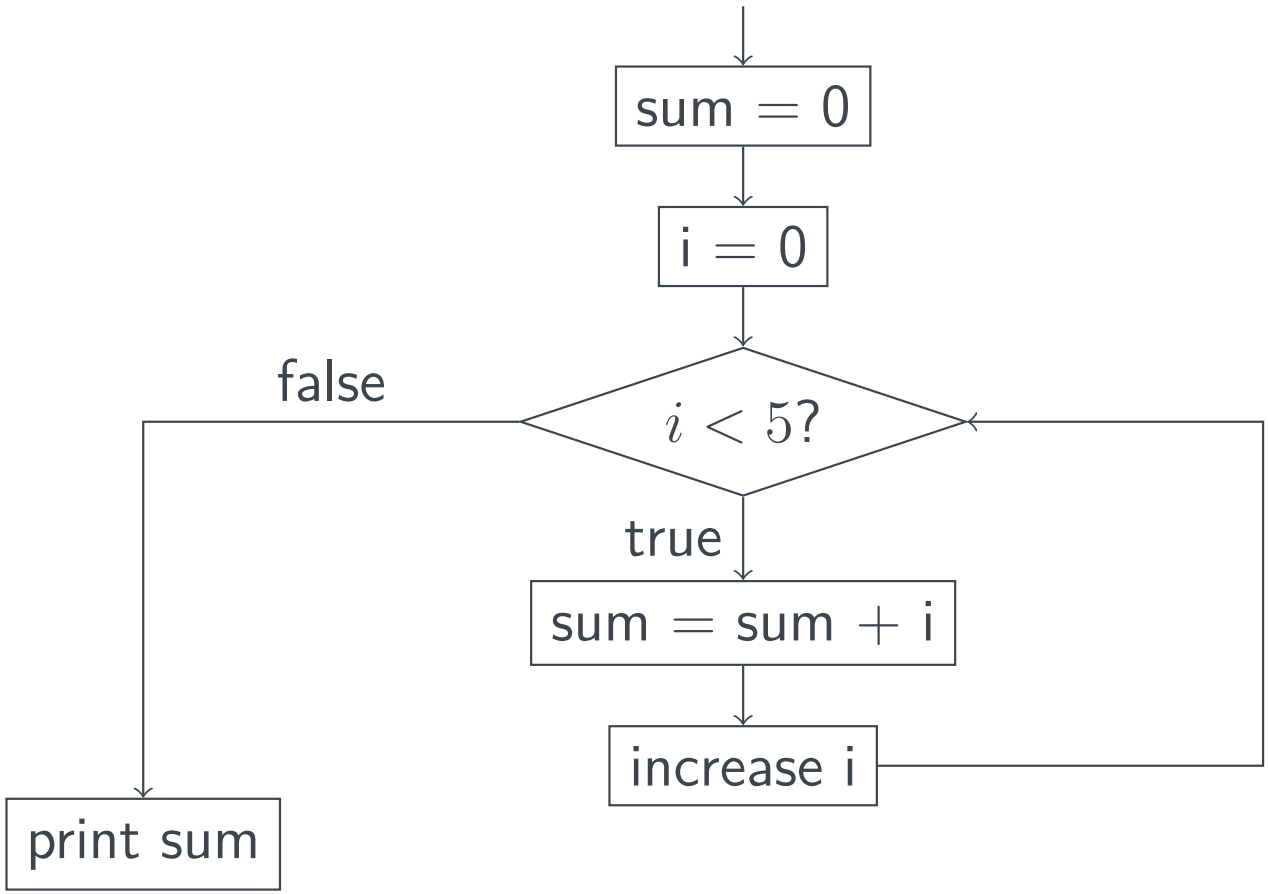
Ablaufdiagramm, Flussdiagramm

Ein Ablaufdiagramm veranschaulicht eine Abfolge von Befehlen in der Reihenfolge, in der sie durchlaufen werden.



MINT-KOLLEG BADEN-WÜRTTEMBERG

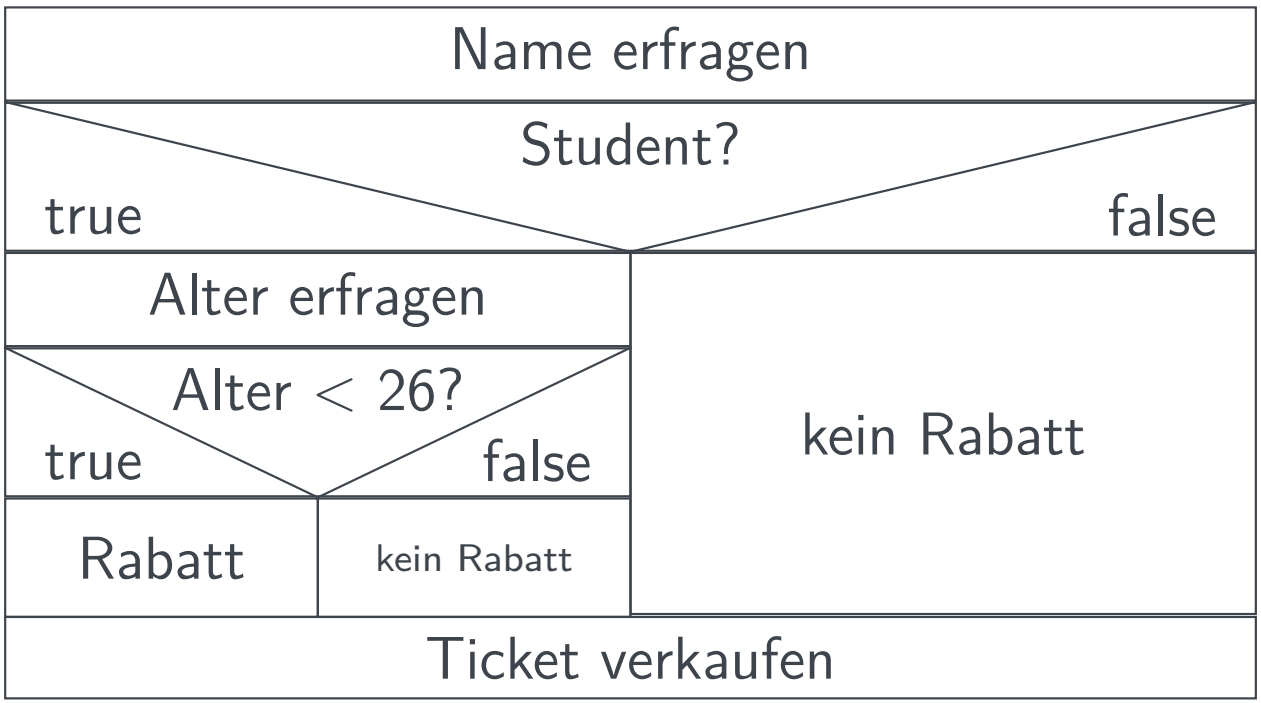
Schleife im Ablaufdiagramm



Struktogramme

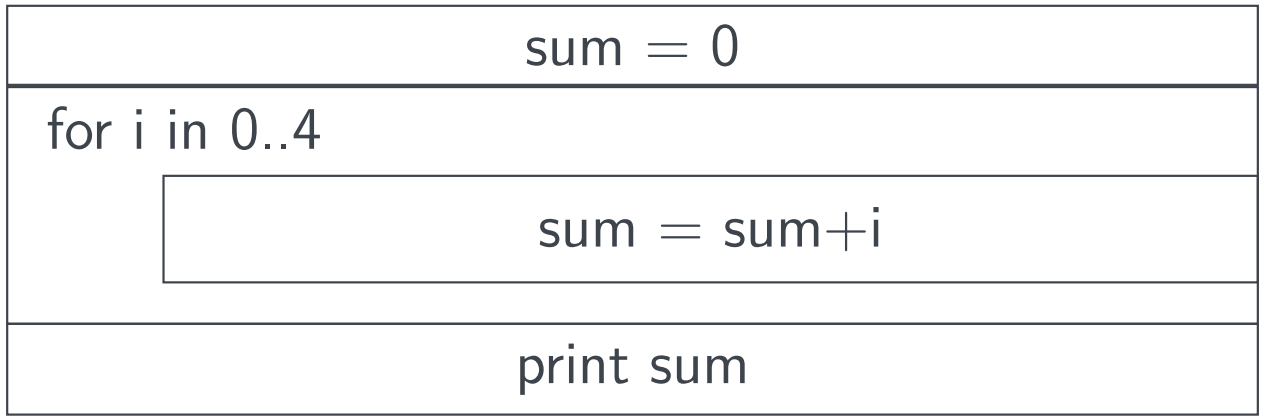
Struktogramm, Nassi-Shneiderman-Diagramm

Ein Struktogramm veranschaulicht Befehlssequenzen und Kontrollstrukturen in geschachtelten rechteckigen Strukturblöcken.



MINT-KOLLEG BADEN-WÜRTTEMBERG

Schleife im Struktogramm



mINT-KOLLEG BADEN-WÜRTTEMBERG

Outline

Zusammenfassung: Schleifen

- Schleifen erlauben es, einen Codeblock **mehrfach** zu durchlaufen.
- Eine **for-Schleife** kann verwendet werden, um einen Programmteil eine festgelegte Anzahl oft zu durchlaufen.
- Eine **while-Schleife** kann verwendet werden, um einen Programmteil so lange zu durchlaufen, bis eine Bedingung nicht mehr zutrifft (das kann auch 0 Mal oder unendlich oft sein).
- Schleifen können ineinander **geschachtelt** werden.

Universität Stuttgart
MINT-Kolleg Baden-Württemberg

Vielen Dank!

Frank Schweiner

E-Mail frank.schweiner@mint-kolleg.de
Telefon +49 (0) 711 685-84326
Fax —

Universität Stuttgart
MINT-Kolleg Baden-Württemberg

Azenbergstr. 12
70174 Stuttgart