JAVA 03: Schleifen in Java.

panitz

Zusammenfassung

Dieser Kurs enthält Aufgaben zur Programmiersprache Java bezüglich Iteration über Schleifenanweisungen mit while-Schleifen, for-Schleifen, foreach-Schleifen mit break und auch mit continue.
Von recht einfachen bis etwas kniffligeren Aufgaben. Fast alle Aufgaben sind parametrisiert und werden immer wieder mit neuen Daten vorgelegt.

Frage: while (1)

Wie häufig druckt das folgende Programm-Fragment »Welcome to Java«?

```
int count = ;
while (count < ){
  System.out.println("Welcome to Java");
  count += ;
}
```

Korrekte Antworten

- ▪

Erläuterung

Die Schleifenvariable `count` wird mit 1 initialisiert und am Ende des Schleifenrumpfs um 1 erhöht. Nach 13 Schleifendurchläufen wird die die Schleifenbedingung `count < 14` falsch.

Frage: while (2)

Was ist die Ausgabe des folgenden Programm-Fragments?

```
int x = ;
while (x < )
  x = x + ;
System.out.println("x ist "+ x);
```

(Eine Antwortmöglichkeit.)

- `x ist`
- `x ist`
- `x ist`
- Die Schleife wird nicht terminieren.
- `x ist`

Korrekte Antworten

- `x ist`

Erläuterung

Die Schleife wird 14 mal durchlaufen und jeweils die Variable `x` um 1 erhöht.

Frage: while (3)

Analysieren Sie den folgenden Code und beurteilen Sie den Wahrheitsgehalt der Antworten:

```
count =
while (count < ){
  // Point A
  System.out.println("Welcome to Java!");
  count += ;
  // Point B
}
// Point C
```

(Mehrere Antwortmöglichkeiten).

- `count <` ist an Punkt A immer true
- `count <` ist an Punkt B immer true.
- `count <` ist an Punkt B immer false.
- `count <` ist an Punkt C immer true.
- `count <` ist an Punkt C immer false.

Korrekte Antworten

- `count <` ist an Punkt A immer true
- `count <` ist an Punkt C immer false.

Erläuterung

Zu Beginn des Schleifenrumpfs ist die Variable `count` aufgrund der Schleifenbedingen immer echt kleiner 15. Nach Beendigung der Schleifen ist die Schleifenbedung nicht mehr erfüllt.

Frage: while (4)

Wie häufig druckt das folgende Programm-Fragment »Welcome to Java«?

```
int count = ;
while (count < )
  System.out.println("Welcome to Java");

count += ;
```

(Eine Antwortmöglichkeit.)

- mal
- mal
- mal
- mal
- unendlich oft (Endlosschleife!)

Korrekte Antworten

- unendlich oft (Endlosschleife!)

Erläuterung

Die Schleifenbedingung wird im Schleifenrumpf nicht verändert. `count` behält den Wert 1 und die Bedingung bleibt immer wahr. Es fehlen geschweifte Klammern, die den Rumpf der Schleife bilden, so dass das Erhöhen der Variable `count` im Schleifenrumpf passiert.

Frage: while (5)

Was wird ausgegeben, wenn der folgende Code ausgeführt wird?

```
int number = ;
while (number > 0){
  number -= ;
  System.out.print(number + ' ');
}
```

Korrekte Antworten

- ▪

Erläuterung

Es wird jeweils die Schleifenvariable `number` am Ende des Schleifenrumpfs ausgedruckt. Es werden keine Zeilenende ausgedruckt.

Frage: foreach (1)

Was druckt das folgende Codestückenchen aus?

```
for (char letter: "Java".toCharArray())
  System.out.print(letter+' ')
```

(Eine Antwortmöglichkeit.)

- J
  a
  v
  a
- J a v a
- Java
- Der Code kompiliert nicht, weil man nich über den Array `"Java".toCharArray()` iteriere kann.

Korrekte Antworten

- `J a v a`

Erläuterung

Die for-each Schleife iteriert über jeden Buchstaben des Wortes »Java«.

Frage: break - continue (1)

Wird das folgende Programmfragment terminieren?

```
int balance = start;
while (true){
  if (balance < ) break;
  balance = balance - ;
}
```

(Eine Antwortmöglichkeit.)

- Ja
- Nein
- Hängt vom Initialwert für `start` ab.

Korrekte Antworten

- Ja

Erläuterung

Tatsächlich irgendwann ja. Die Variable `balance` wird in jedem Schleifendurchlauf verringert, so dass sie irgendwann einen einen Wert kleiner 9 annimmt.

Frage: break - continue (2)

Welchen Wert hat die Variable `sum`, wenn die Schleife terminiert?

```
int sum = 0;
int item = 0;
while (item < ){
  item += 1;
  sum += item;
  if (sum > ) break;
}
```

Korrekte Antworten

-

Erläuterung

Die Schleifenbedingung lässt zwar 9 Durchläufe erwarten, aber die Schleife wird vorzeitig durch die `break`-Anweisung verlassen.

Frage: break - continue (3)

Welchen Wert hat die Variable `sigma`, wenn die Schleife terminiert?

```
int sigma = 0;
int item = 0;
while (item < ){
  item += 1;
  if (sigma >= ) continue;
  sigma += 1;
}
```

Korrekte Antworten

- ▪

Erläuterung

Im Schleifenrumpf wird jeweils zum nächsten Durchgang gesprungen, wenn die Variable `sigma` größer oder gleich 4 geworden ist. Damit wird sie nie ein weiteres Mal erhöht.

Frage: break - continue (4)

Wird das folgende Programmfragment terminieren?

```
int balance = start;
while (true){
  if (balance < ) continue;
  balance = balance - ;
}
```

(Eine Antwortmöglichkeit.)

- Ja
- Nein
- Hängt vom Initialwert für `start` ab.

Korrekte Antworten

- Nein

Erläuterung

Die Schleifenbedingung ist immer wahr und es gibt keine andere Anweisung, die zum Verlassen der Schleife führt.

Frage: Herausforderungen mit Float

Analysieren Sie das folgende Code-Segment.

```
float sum = 0;
float d = 0;
while (d != 10.0){
  d += 0.1;
  sum += sum + d;
  System.out.println(d+" "+sum);
}
```

(Eine Antwortmöglichkeit.)

- Das Programm kompiliert nicht, weil `sum` und `d` nicht korrekt initialisiert sind.
- Das Programm geht in eine Endlosschleife, weil `d` innerhalb des Schleifenrumpfes immer gleich 0.1 ist.
- Es kann sein, dass das Programm wegen numerischer Ungenauigkeit mit Floating-Point-Zahlen nicht an der vorgesehenen Stelle anhält.
- Nach der Schleife ist die sum$= 0 + 0.1 + 0.2 + 0.3 + \cdots + 9.9$

Korrekte Antworten

- Es kann sein, dass das Programm wegen numerischer Ungenauigkeit mit Floating-Point-Zahlen nicht an der vorgesehenen Stelle anhält.

Erläuterung

Als Konsequenz sollte man nie eine Bedingung zur Gleichheit auf Fließkommazahlen verwenden, sondern eine größer oder kleiner Relation verwenden.

Frage: Programmanalyse (1)

Was wird das folgende Programmfragment anzeigen?

```
int number = 25;
boolean isPrime = true;
int i = 2;
while (i < number && isPrime){
  if (number % i == 0)
    isPrime = false;
  i += 1;
}
System.out.println("i ist "+ i+ " isPrime ist "+isPrime);
```

(Eine Antwortmöglichkeit.)

- `i ist 5 isPrime ist true`
- `i ist 5 isPrime ist false`
- `i ist 6 isPrime ist true`
- `i ist 6 isPrime ist false`

Korrekte Antworten

- `i ist 6 isPrime ist false`

Erläuterung

Der erste Teiler größer 1 der Zahl 25 führt dazu, dass `isPrime` auf `false` gesetzt wird. Damit ist die verundete Schleifenbedingung nicht mehr wahr. Allerdings witd `i` noch ein weiteres Mal erhöht.

Frage: Programmanalyse (2)

Was wird das folgende Programmfragment anzeigen?

```
boolean isPrime = true;
int j = 5;
int number = 25;
int[] range = {2,3,4,5,6,7,8,9,10,11,12,13,14
               ,15,16,17,18,19,20,21,22,23,24,25};

for (int i :range){
  if (number % i == 0){
    isPrime = false;
    j = i;
    break;
  }
}

System.out.println("j ist "+ j+ " isPrime ist "+isPrime);
```

(Eine Antwortmöglichkeit.)

- `j ist 5 isPrime ist true`
- `j ist 5 isPrime ist false`
- `j ist 6 isPrime ist true`
- `j ist 6 isPrime ist false`

Korrekte Antworten

- `j ist 5 isPrime ist false`

Erläuterung

Es wird über alle Zahlen der Reihung iteriert, allerdings bei der ersten Zahl, die ein Teiler von 25 ist, die Schleife verlassen.

Frage: Programmanalyse (3)

Was wird aus folgendem Programm dann ausgedruckt?

```
int number = 9;
boolean isPrime = true;
int[] range = {2,3,4,5,6,7,8,9};
for (int i : range){
  if (number % i == 0) isPrime = false;
  System.out.println("i ist "+i);
  if (isPrime){
    System.out.println(number+ " ist eine Primzahl");
    break;
  }else{
    System.out.println(number+ " ist keine Primzahl");
  }}
```

(Eine Antwortmöglichkeit.)

- i ist 2
  9 ist eine Primzahl
- i ist 2
  9 ist keine Primzahl
- i ist 3
  9 ist eine Primzahl
- i ist 3
  9 ist keine Primzahl

Korrekte Antworten

- `i ist 2`
  `9 ist eine Primzahl`

Erläuterung

Die Variable `isPrime` bleibt im ersten Durchlauf wahr und somit wird die Schleife schon im ersten Durchlauf verlassen.

Frage: Verschachtelung (1)

Wie oft druckt das folgende Programm das Wort "javaäus?

```
for (int i1=;i1<;i1++)
  for (int i2=;i2<;i2++)^^I
    System.out.println("java");
```

Korrekte Antworten

-

Erläuterung

Für jeden Durchlauf der äußeren Scleife wird die innere Schleife ein weiteres Mal komplett gestartet. Damit multipliziert sich die Anzahl der Durchläufe beider Schleifen.

Frage: Verschachtelung (2)

Wie oft druckt das folgende Programm das Wort "javaäus?

```
for (int i1=;i1<;i1++){
  for (int i2=;i2<;i2++)^^I
    System.out.println("java");
  if (i1 > 6) break;
}
```

Korrekte Antworten

- ▪

Erläuterung

Für jeden Durchlauf der äußeren Scleife wird die innere Schleife ein weiteres Mal komplett gestartet. Damit multipliziert sich die Anzahl der Durchläufe beider Schleifen. Die äußere Schleife wird allerdings beendet, wenn die Laufvariable i1 den Wert 6 überschritten hat.

Frage: Verschachtelung (3)

Wie oft druckt das folgende Programm das Wort "javaäus?

```
for (int i1=;i1<;i1++)
  for (int i2=;i2<;i2++){^^I
    System.out.println("java");
    if (i2 > 5) break;
  }
```

Korrekte Antworten

- ▪

Erläuterung

Für jeden Durchlauf der äußeren Scleife wird die innere Schleife ein weiteres Mal komplett gestartet. Damit multipliziert sich die Anzahl der Durchläufe beider Schleifen. Die innere Schleife wird allerdings beendet, wenn die Laufvariable i2 den Wert 5 überschritten hat.

Frage: Verschachtelung (4)

Wie oft druckt das folgende Programm das Wort »java« aus?

```
s1:for (int i1=;i1<;i1++)
  for (int i2=;i2<;i2++){^^I
    System.out.println("java");
    if (i2 >= 5) break s1;
  }
```

Korrekte Antworten

-

Erläuterung

Die äußere Schleife trägt das Label s1. In der inneren Schleife wird der break-Befehl für die äußere Schleife aufgerufen, da ihm das Label s1 angehängt ist. Damit wird die äußere Schleife bereits beim ersten Durchlauf durch das break der inneren Schleife beendet.

Frage: do-while

Welchen Wert hat die Variable x, wenn die Schleife terminiert?

```
var x = ;
do{
  x +=
}while(x<)
```

Korrekte Antworten

-

Erläuterung

Es handelt sich um eine vorgeprüfte Schleife, die mindestens einmal durchlaufen wird und so oft wiederholt wird, bis die Schleifenbedingung (`11<4`) nicht mehr zu `true` auswertet.

Frage: switch-case

Welchen Wert hat die Variable x, nach Durchlaufen dieses Code-Fragments?

```
var x = ;
switch (x){
  case : x+=1;
  case : x+=2;
  case : x+=3;
  case : x+=4;
  case : x+=5;
}
```

Korrekte Antworten

-

Erläuterung

Man beachte, dass das switch nicht eine Auswahl zu einer Alternative ist, sondern eine Art Sprung zu einer position, von der an alle folgenden Fälle abgearbeitet werden. Möchten man nur eine Fall ausführen, so muss dieser mit dem Schlüsselwort `break` enden.