

# Übung 3



TECHNISCHE  
UNIVERSITÄT  
DARMSTADT

## Aufgabe 1 (Wissensfrage)

Beantworten Sie die folgenden Fragen:

- a) Welche Typen von **while-Schleifen** gibt es in Java? Worin unterscheiden sich diese?
- b) Ist die **if-Anweisung** eine **Schleife**? Erläutern Sie Ihre Antwort.
- c) Welchen Wert hat der Ausdruck `zahlen.length` nach Ausführung des folgenden Codes?  

```
int[] zahlen = {33, 44, 55};
```
- d) Welchen Wert hat der Ausdruck `matrix.length` nach Ausführung des folgenden Codes?  

```
int[][] matrix = { {10, 20, 5}, {1, 2, 3, 4}};
```
- e) Welchen Wert hat der Ausdruck `matrix[1].length` nach Ausführung des Codes aus Aufgabe 1d)?

a) while-Schleifen, do while-Schleifen.

Kopfgesteuerte Schleife, Fußgesteuerte Schleife.

Beurteilung, Ausführung vs. Ausführung, Beurteilung.

b) If-Anweisung ist keine Schleife. Es gibt keine Wiederholung der Anweisung.

c) `length = 3`

d) `length = 2`. (reflektiert die Dimensionen des Arrays).

e) `length = 4`

---

## Aufgabe 2 (Schleifen: UML und Programmieren)

---

In der folgenden Aufgabe sollen vier verschiedene Algorithmen implementiert werden. Erstellen Sie für die Aufgabenteile a) - c) ein UML-Aktivitätsdiagramm.

Implementieren Sie danach für alle Aufgabenteile den Algorithmus in Java. Prüfen Sie Ihre Lösungen auch mit Hilfe der Ausgaben auf der Konsole von Eclipse. Erstellen Sie zuerst für jede Aufgabe eine eigene Java Klasse (z.B. „Aufgabe2a“, „Aufgabe2b“).

- a) Erstellen Sie einen Algorithmus, der mit Hilfe einer Schleife die Zahlen von 1 bis 1000 in aufsteigender Reihenfolge auf der Konsole ausgibt. Benutzen Sie beim Implementieren des Algorithmus in Eclipse eine for-Schleife.
- b) Erstellen und implementieren Sie einen Algorithmus, der mit Hilfe einer Schleife von 1 bis 1000 aufwärts zählt und für jede dieser Zahlen die jeweilige Quadratzahl ausgibt. Benutzen Sie beim Implementieren des Algorithmus in Eclipse eine for-Schleife. Die Ausgabe soll wie folgt aussehen:

```
1. Durchlauf: 1
2. Durchlauf: 4
3. Durchlauf: 9
4. Durchlauf: 16
...
```

**Hinweis:** Orientieren Sie sich für die Ausgabe in Java an folgendem Code:

```
System.out.println ("Durchlauf: " + a); // wobei a eine Integer Variable ist
```

- c) Implementieren Sie nun einen Algorithmus, der mit Hilfe einer Schleife von 1 bis 1000 zählt, für jede Zahl die Quadratzahl berechnet und die Quadratzahl nur dann ausgibt, wenn die Quadratzahl durch 5 teilbar ist. Auch hier soll eine for-Schleife für die Implementierung in Java verwendet werden. Die Ausgabe könnte so aussehen:

```
25
100
225
400
625
...
```

- d) Schreiben Sie schließlich ein Java Programm, in dem ein Array mit zehn ganzzahligen Werten angelegt wird (überlegen Sie sich hier selbst beliebige Werte). Danach soll für jede der Zahlen die Quadratzahl berechnet und ausgegeben werden.  
(Hinweis: Hier ist kein UML-Diagramm notwendig)

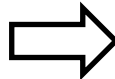
### Aufgabe 3 (Codeverständnis)

Analysieren Sie die nachfolgenden sieben Schleifen a) - g). Notieren Sie für jede Schleife alle auftretenden Bildschirmausgaben, die auf der Konsole in jedem Durchlauf ausgegeben werden. Falls eine Schleife kein einziges Mal durchlaufen wird, erklären Sie hierfür den Grund.

Verwenden Sie hierzu erst einmal kein Eclipse.

Ein Beispiel mit Lösung:

```
1 int i = 42;
2 while (i < 45) {
3     System.out.println(i);
4     i = i+1;
5 }
```

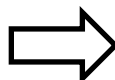


Durchlauf	Ausgabe
1	42
2	43
3	44
4	

Aufgabenteile:

a)

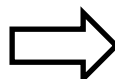
```
1 int a = 0;
2 while (a < 5) {
3     a = a + 1;
4     System.out.println(a);
5 }
```



Durchlauf	Ausgabe
1	1
2	2
3	3
4	4
5	5
6	

b)

```
1 int a = 0;
2 int b = 42;
3 while (b < 42) {
4     a = a * a;
5     System.out.println(a);
6     System.out.println(b);
7     b = b + 1;
8 }
```



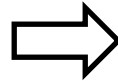
Durchlauf	Ausgabe
1	
2	

c)

```

1  int a = 0;
2  do {
3      a = a + 1;
4      System.out.println(a);
5  } while (a < 0);

```



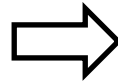
Durchlauf	Ausgabe
1	1
2	

d)

```

1  int a = 3;
2  int b = a;
3  do {
4      a = a - 2;
5      System.out.println(a);
6      System.out.println(b);
7  } while (a > 0 && b > 0);

```



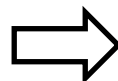
Durchlauf	Ausgabe
1	1 3
2	-1 3

e)

```

1  int a = 1;
2  while (a < 0) {
3      a++;
4      System.out.println(a);
5  }

```



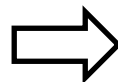
Durchlauf	Ausgabe
1	
2	

f)

```

1  int a = 1;
2  int b = 0;
3  while (a < 5) {
4      b = b + 2 * a;
5      a = a + 1;
6      System.out.println(a);
7      System.out.println(b);
8  }

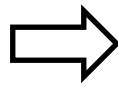
```



Durchlauf	Ausgabe
1	2 2
2	3 6
3	4 12
4	5 20
5	

g)

```
01 int a1 = 1;
02 int a2 = 9;
03 int a3 = 10;
04 while (a3 < 50) {
05     a2 = a2 * a1;
06     a1 = 2 * a1;
07     a3 = a3 + a1;
08     if (a3 < a2) {
09         a3 = a2;
10         a3 = a3 + a1;
11     } else {
12         a3 = a1;
13         a3 = a3 - a2;
14     }
15     System.out.println(a1);
16     System.out.println(a2);
17     System.out.println(a3);
18 }
```



Durchlauf	Ausgabe
1	2 9 -7
2	4 18 22
3	8 72 80
4	

---

## Aufgabe 4 (Fehler erkennen)

---

Im Folgenden wird erläutert, nach welchen Regeln sich bestimmt, ob ein bestimmtes Jahr ein Schaltjahr ist oder nicht. Ein Schaltjahr ist ein Jahr mit einem extra Tag, dem 29. Februar. Ein Schaltjahr hat damit 366 Tage. Z.B. war das Jahr 2016 ein Schaltjahr. Mit den folgenden Regeln kann man bestimmen, ob ein bestimmtes Jahr ein Schaltjahr ist:

- In der Regel gilt:  
Jahreszahlen, die durch 4 teilbar sind, sind Schaltjahre.
- Erste Ausnahme:  
Jahreszahlen, die ebenso durch 100 teilbar sind, sind keine Schaltjahre.
- Zweite Ausnahme:  
Jahreszahlen, die hingegen durch 400 teilbar sind, sind TROTZDEM Schaltjahre.

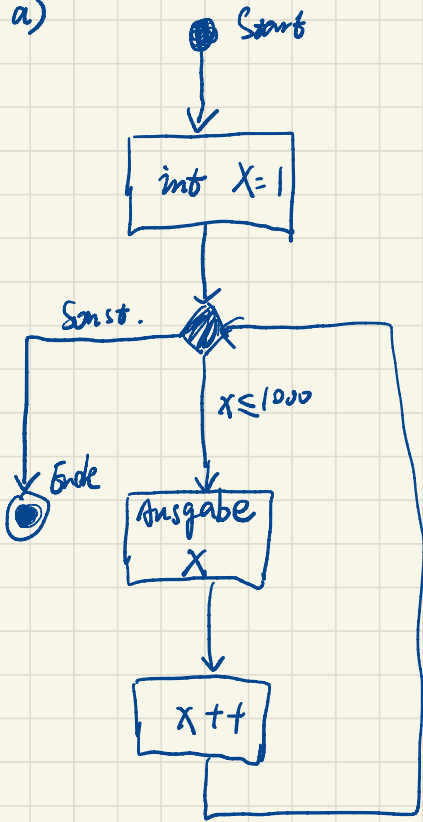
Das folgende Java-Programm soll diese Regeln implementieren. Die Jahreszahlen, welche im Array `int[] jahre` gespeichert werden, sollen anhand der oben genannten Regeln darauf geprüft werden, ob sie ein Schaltjahr sind oder nicht. Leider haben sich hierbei ein paar Fehler eingeschlichen, welche Sie nun finden sollen.

```
01  int[] jahre = 1800, 1900, 1950, 1968, 1969, 1990, 2000, 2001, 2002,  
    2010, 2011, 2012, 2013, 2014;  
02  
03  for (int i = 0; i < jahre.length; i+) {  
04      int j = jahre[i];  
05  
06      if (j % 400 == 0 && (j % 4 == 0 && j % 100 != 0)) {  
07          System.out.println("Das Jahr " + j + " ist ein  
            Schaltjahr.");  
08      } else {  
09          System.out.println("Das Jahr " + j + " ist KEIN  
            Schaltjahr.");  
10      }  
11  }
```

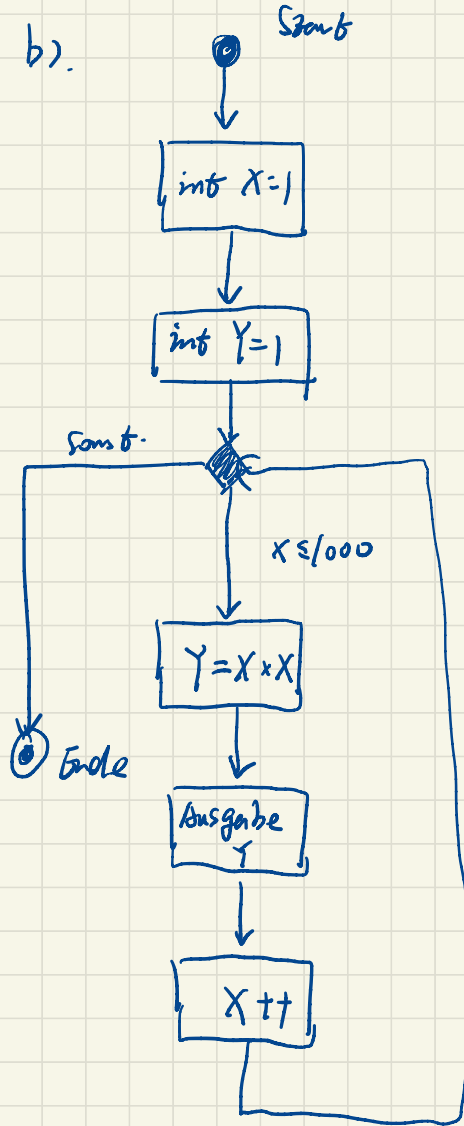
- a) Kennzeichnen Sie:  
**drei Syntax-Fehler** (die bereits von Eclipse erkannt werden)  
**einen Semantik-Fehler** (der die Regeln falsch implementiert)
- b) Korrigieren Sie das Programm anschließend in Eclipse und führen Sie es aus. Legen Sie hierzu eine Klasse „Aufgabe4“ an. Den Code können Sie beispielsweise in der Mainmethode ausführen.

# Aufgabe 2:

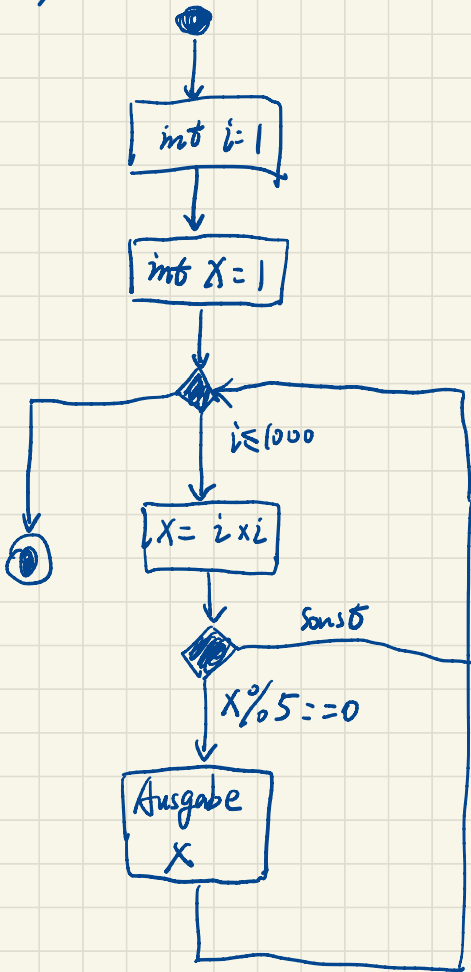
a)



b)



c)





## Aufgabe 4:

a) Syntax-Fehler:

① Initialisierung der Array:

`int jahre[] = {1800, 1900, ...};`

② For-Schleife:

`j++`

③ Logische Operatoren:

`!=`

Semantik-Fehler:

`if( j%4==0 && (j%100↑!=0 || j%400↑==0) )`