Übung 3, Lösungsvorschlag

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)?

Lösung Aufgabe 1

a) In der Vorlesung haben wir zwei Typen von while-Schleifen kennengelernt:
Die *while-Schleife*, und die *do-while-Schleife*.
   a. Bei einer *while-Schleife* wird vor der (eventuellen) Ausführung Anweisung eine Bedingung geprüft. Nur wenn dieses Bedingung wahr (`true`) ist, wird die Anweisung anschließend erstmals (oder erneut) ausgeführt.
   b. Bei einer *do-while Schleife* wird erst nach dem ersten Durchlauf der Anweisung eine Bedingung überprüft, ob die Anweisung erneut ausgeführt werden soll. Die Anweisung wird also in jedem Fall ein Mal ausgeführt.

b) Nein, **die if-Anweisung ist keine Schleife**! Es ist keine Schleife, weil Schleifen Anweisungen wiederholt ausführen. Das ist bei einer if-Anweisung nicht der Fall. Sie wird nur ein Mal ausgeführt.
c) Das Array `zahlen` wurde mit drei Werten statisch erzeugt. Somit hat es eine Länge von Drei. Der Befehl ergibt somit den Wert 3.
d) Das zweidimensionale Array `matrix` wird mit zwei Elementen statisch erzeugt. Bei beiden Elementen handelt es sich wiederum um Arrays: `{10, 20, 5}` sowie `{1, 2, 3, 4}`. Damit hat das Array `matrix` zwei Einträge und das Ergebnis des Ausdrucks ist 2. Die Länge der geschachtelten Arrays spielt hier also keine Rolle.
e) Im Vergleich zu 1d) wird nun auf den Eintrag aus dem Array `matrix` mit dem Index `[1]` zugegriffen. Das Element an dieser Stelle ist: `{1, 2, 3, 4}`. Da dieses Element ein Array mit vier Einträgen ist, ergibt der Aufruf 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:

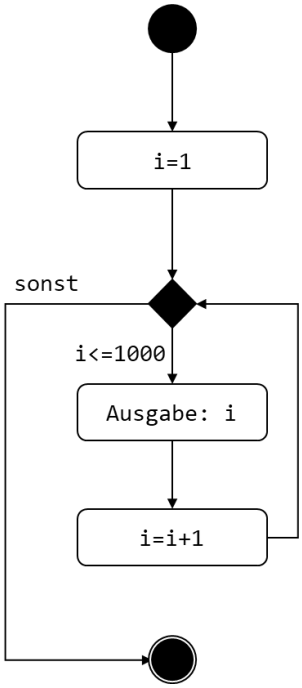
```
25
100
225
400
625
…
```

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)

Lösung Aufgabe 2

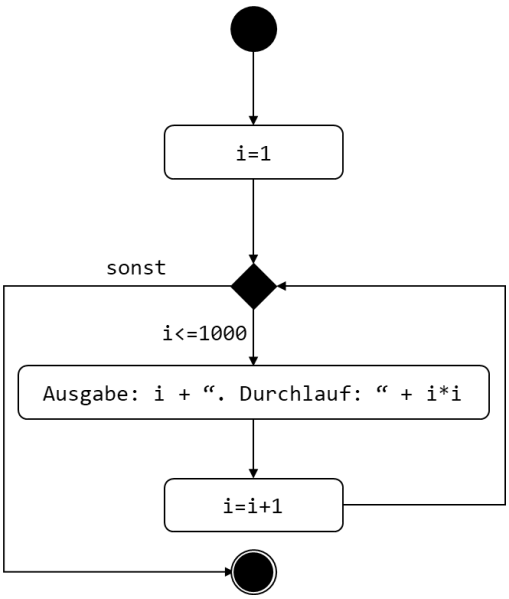
a)

```
1  public class Aufg2a {
2    public static void main(String[]
   args) {
3      for (int i = 1; i <= 1000; i++) {
4        System.out.println(i);
5      }
6    }
7  }
```



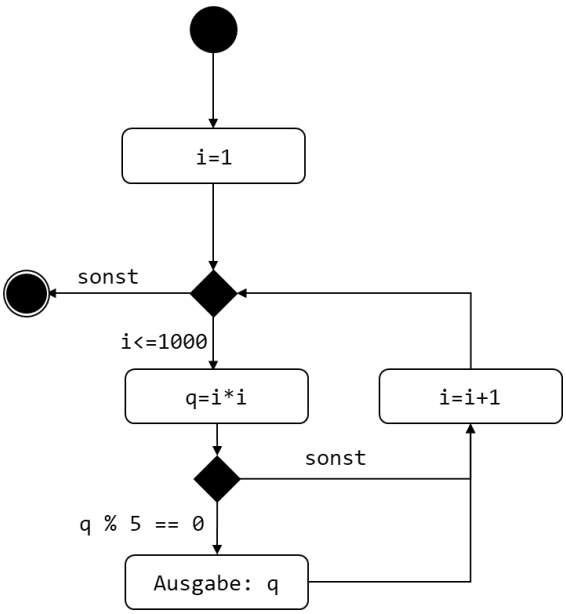
b)

```
1  public class Aufg2b {
2    public static void main(String[] args) {
3      for (int i = 1; i <= 1000; i++) {
4        System.out.println(i + ". Durchlauf: " + i
   * i);
5      }
6    }
7  }
```



c)

```
1   public class Aufg2c {
2     public static void main(String[]
    args) {
3       int q;
4       for (int i = 1; i <= 1000; i++) {
5         q = i * i;
6         if (q % 5 == 0) {
7           System.out.println(q);
8         }
9       }
10    }
11  }
```



d)

```
1   public class Aufg2d {
2     public static void main(String[] args) {
3       int[] zahlen = {22, 34, 16, 3, 679, 392, 37, 98, 76, 54 };
4       for (int i = 0; i < zahlen.length; i++) {
5           System.out.println(zahlen[i] * zahlen[i]);
6       }
7     }
8   }
```

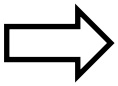
Aufgabe 3 (Codeverständnis)

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  }
```

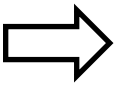


Durchlauf	Ausgabe
1	42
2	43
3	44
4	

Aufgabenteile:

a)

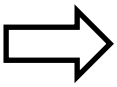
```
1  int a = 0;
2  while (a < 5) {
3    a = a + 1;
4    System.out.println(a);
5  }
```



Durchlauf	Ausgabe
1	
2	
3	
4	
5	
6	

b)

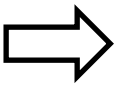
```
1  int a = 0;
2  int b = 42;
3  while (b < 42) {
4    a = a * a;
5    System.out.println(a);
6    System.out.println(b);
7    b = b + 1;
8  }
```



Durchlauf	Ausgabe
1	
2	

c)

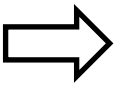
```
1  int a = 0;
2  do {
3    a = a + 1;
4    System.out.println(a);
5  } while (a < 0);
```



Durchlauf	Ausgabe
1	
2	

d)

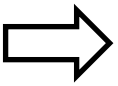
```
1  int a = 3;
2  int b = a;
3  do {
4    a = a - 2;
5    System.out.println(a);
6    System.out.println(b);
7  } while (a > 0 && b > 0);
```



Durchlauf	Ausgabe
1	
2	

e)

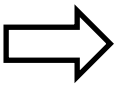
```
1  int a = 1;
2  while (a < 0) {
3    a++;
4    System.out.println(a);
5  }
```



Durchlauf	Ausgabe
1	
2	

f)

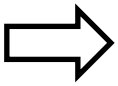
```
1  int a = 1;
2  int b = 0;
3  while (a < 5) {
4    b = b + 2 * a;
5    a = a + 1;
6    System.out.println(a);
7    System.out.println(b);
8  }
```



Durchlauf	Ausgabe
1	
2	
3	
4	
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	
3	
4	

Lösung Aufgabe 3

a)

Durchlauf		1	2	3	4	5
Ausgabe	(int a)	1	2	3	4	5

b) Die Schleife wird nicht durchlaufen. Beim ersten Prüfen der Bedingung b<42 wird bereits im Schleifenkopf festgestellt, dass diese nicht erfüllt ist. Damit wird der Anweisungsblock der Schleife nicht ausgeführt. Da beide Ausgabebefehle des Anweisungsblocks der Schleife stehen, werden diese nicht erreicht und es wird nichts ausgegeben. (Die Werte der Variablen nach der Schleife sind `a=0` und `b=42`)

c)

Durchlauf		1
Ausgabe	(int a)	1

d)

Durchlauf		1	2
Ausgabe	(int a)	1	-1
	(int b)	3	3

e)

Auch in diesem Fall ist die Bedingung der while-Schleife schon zu Beginn nicht erfüllt. Deshalb wird auch hier der Anweisungsblock nicht ausgeführt. Die Variable int a wird nicht erhöht und es erfolgt keine Ausgabe.

f)

Durchlauf		1	2	3	4
Ausgabe	(int a)	2	3	4	5
	(int b)	2	6	12	20

g)

Durchlauf		1	2	3
Ausgabe	(int a1)	2	4	8
	(int a2)	9	18	72
	(int a3)	-7	22	80

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.

Lösung Aufgabe 4

a) **Syntax-Fehler**:
1. Bei der Initialisierung des Arrays `jahre` fehlen die geschweiften Klammern.
2. Bei der Inkrementierung von `i` in der `for`-Schleife fehlt ein `+` (`i++` anstelle von `i+`)
3. In der if-Anweisung muss es `!= 0` anstelle von `!!= 0` heißen.

Semantik-Fehler:

In der `if`-Anweisung muss die erste logische Verknüpfung ein ODER `||` sein.

b) Der korrigierte Code lautet wie folgt:

```
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  }
```