

JAVA

Tutoriumsblatt 3

Janosch Fuchs
Jürgen Dietel, Abdelrahman Elhabashy, Timon Römer, Daniel Weinholz
MATSE Gruppe
ITC - RWTH Aachen

WS24/25
06.09.2024
Abgabe: —

-
- Die Übungsblätter sollten alleine bearbeitet werden, aber Hilfe von Tutoren oder Studierenden ist gewünscht.
-

Tutoriumsaufgabe 3.1

Folgende Aufgaben sind schriftlich zu bearbeiten:

- (a) Wie würde die nachfolgende for-Schleife als while-Schleife aussehen?

```
int x = 5;
for (int i=0; i<x; i++){
    System.out.println(i);
}
```

- (b) Was bedeutet die Anweisung

```
for (; ;)
```

- (c) Welche Zahlen werden auf dem Bildschirm ausgegeben?

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

- (d) Wie oft wird die folgende Schleife mindestens durchlaufen?

```
do{
    System.out.println("Test");
}while (x<10);
```

- (e) Was geben die folgenden Methodenaufrufe zurück?

```
"abcdef".substring(1, 3);
"abcdef".substring(1);
"abcdef".indexOf("d");
"abcdef".indexOf("g");
```

Tutoriumsaufgabe 3.2

Schreiben Sie eine Funktion

```
public static int qdq(String z)
```

die die quadrierte Quersumme ermittelt und zurückgibt.

Beispiel: Das Quadrat der Quersumme von 47 ist $(4 + 7)^2 = 11^2 = 121$.

Tutoriumsaufgabe 3.3

Schreiben Sie eine Funktion

```
public static boolean hasDoubleChars(String s)
```

Die Funktion ermittelt, ob es Zeichen gibt, die im String s mehrfach vorhanden sind oder nicht.

Tutoriumsaufgabe 3.4

Schreiben Sie eine Funktion

```
public static char haeufigstesZeichen(String s)
```

Die Funktion gibt das Zeichen zurück, das in s am häufigsten vorkommt. Kommen mehrere Zeichen gleich häufig vor, wird eins davon zurückgegeben.

Tutoriumsaufgabe 3.5

Schreiben Sie eine Funktion

```
public static String komplettiere(String s)
```

Die Funktion soll alle Ziffern (0-9), die nicht im String vorkommen, an den String anhängen.

Tutoriumsaufgabe 3.6

Schreiben Sie eine Funktion

```
public static String loesche(String s)
```

Nach jeder Ziffer, die im String enthalten ist, werden so viele Zeichen gelöscht, wie der Wert der Ziffer angibt. Die Ziffer selbst ist das erste Zeichen, das gelöscht wird. Falls durch eine Ziffer ein zweite Ziffer gelöscht wird, wird die gelöschte Ziffer wie ein normales Zeichen behandelt.

Beispiele:

```
String r;  
r = loesche("abc2def"); // -> "abcef"  
r = loesche("abc1def"); // -> "abcdef"  
r = loesche("abc22def"); // -> "abcdef"
```

Tutoriumsaufgabe 3.7

Schreiben Sie eine Funktion

```
public static String dekomprimiere(String s)
```

s besteht dabei aus Ziffern, denen ein beliebiges Zeichen folgt (z.B. "1e3b2a"). Die Ziffer gibt die Anzahl Wiederholungen des nachfolgenden Zeichens an. Die Funktion soll den "unkomprimierten" String (im Beispiel "ebbaa") berechnen und zurückgeben.

Tutoriumsaufgabe 3.8

Schreiben Sie eine Funktion

```
public static int hoechsteZiffer(String s)
```

Die Funktion gibt die höchste Ziffer zurück, die im String s vorhanden ist. Gibt es in s keine Ziffer, wird -1 zurückgegeben.

Tutoriumsaufgabe 3.9

Schreiben Sie eine Funktion

```
public static int hoechstePosition(String s)
```

Die Funktion gibt die Position der höchsten Ziffer zurück, die im String s vorhanden ist. Ist die höchste Ziffer mehrmals vorhanden, wird die vorderste Position zurückgegeben. Gibt es in s keine Ziffer, wird -1 zurückgegeben.

Tutoriumsaufgabe 3.10

Schreiben Sie eine Funktion

```
public static void printTeilerZahlen(int start, int end, int n)
```

Die Methode gibt alle Zahlen zwischen $start$ und end (jeweils einschließlich) auf dem Bildschirm aus, die genau n Teiler haben. Die Zahl 1 und die Zahl selbst zählen dabei mit, sodass Primzahlen 2 Teiler haben.

Tutoriumsaufgabe 3.11

(a) Schreiben Sie eine Funktion

```
public static void ermittleAufgabe(int z)
```

Die Funktion erhält eine Zahl zwischen 1 und 100 und ermittelt alle Zahlenpaare (a,b), für die gilt:

- a und b sind ganze Zahlen zwischen 1 und 10 (jeweils einschließlich).
- a ist größer oder gleich b .
- Das Produkt $a \cdot b$ weicht maximal um 5 von z ab.

Diese Zahlenpaare werden auf dem Bildschirm ausgegeben (siehe Beispiel unten).

Beispiel: `ermittleAufgabe(65)` ergibt folgende Ausgabe.

```
8 x 8 = 64
9 x 7 = 63
10 x 6 = 60
10 x 7 = 70
```

(b) Schreiben Sie eine Funktion

```
public static void printZahlendreieck(int n)
```

n ist eine Zahl zwischen 1 und 9 (das muss nicht überprüft werden). Die Funktion gibt ein Zahlendreieck mit n Zeilen der folgenden Form auf dem Bildschirm aus:

```
1
22
333
4444
55555
usw.
```
