

SEW3 UE14 2024/25

Streams for Beginners ;)

Die folgenden Beispiele sind hauptsächlich Übungs- oder Testangaben aus SEW für die erste Klasse an unserer HTL. Alle Beispiele sollen (mit ganz einfachen) Stream-Operationen gelöst werden.

Als kleine Hilfe hier eine (unsortierte) Aufzählung von Operationen, die zur Lösung hilfreich/notwendig sein könnten ... :

`Stream, IntStream, .filter() , .iterate(), .forEach() , .count() , .limit() , .range, .rangeClosed() , .ints() , .sum()`

1. Schreibe die Methode

```
void zahlenreihen() { ... },
```

die folgende Zahlenreihen in der Konsole ausgibt:

```
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
1 3 5 7 9 11 13 15 17 19 21 23 25
20 15 10 5 0 -5 -10 -15 -20
1 2 4 8 16 32 64
2 3 5 9 17 33 65 129
```

2. Schreibe die Methode

```
void printFolgeOhne3(int anz) { .. },
```

die die folgende Zahlenfolge ausgibt: 4 5 7 8 10 11 13 14 ...

- Die Folge enthält keine durch drei teilbaren Glieder.
- Die Folge startet immer bei „4“.
- Der Methode soll die Anzahl anz der auszugebenden Zahlen übergeben werden.

3. Schreibe die Methode

```
void wuerfelStatistik(int augenzahl) { .. },
```

die ermitteln und ausgeben soll, wie oft bei 1000 zufälligen Würfelwürfen die übergebene Augenzahl gewürfelt wurde.

4. Schreibe eine Methode

```
void alleTeiler(int z) {..},
```

die alle ganzzahligen Teiler der Zahl `z` in der Konsole ausgibt.

Ausgabebeispiel: `die Zahl 6 hat folgende Teiler: 1, 2, 3, 6`

5. Schreibe eine Methode

```
boolean istPrimzahl(int z) { .. }
```

mit der überprüft werden kann, ob eine Zahl eine Primzahl ist oder nicht.
(Eine Zahl ist dann eine Primzahl, wenn sie nur durch 1 und sich selbst teilbar ist)

6. Schreibe die Methode

```
int gaussSumme(int n) { .. },
```

die alle Zahlen von 1 bis n addiert und diese Summe zurückliefert.

Arbeiten mit String und char

Zeichen eines Strings:

```
String str = "Hallo Welt";  
IntStream charS = str.chars();
```

(es gibt keinen CharStream → IntStream)

7. Schreibe die Methode

```
void verticalString(String s) { ... }
```

die dafür sorgt, dass der übergebene String zeichenweise untereinander ausgegeben wird. Aus „hallo“ wird also:

```
h  
a  
l  
l  
o
```

8. Schreibe die Methode

```
void mitStern(String s) { ... }
```

die dafür sorgt, dass der übergebene String so ausgegeben wird, dass nach jedem Zeichen des Strings ein Stern angezeigt wird.

Aus „hallo“ wird also **h*a*l*l*o***

9. Schreibe eine Methode

```
void deleteBlanks(String s1) { ... } ,
```

die aus dem übergebenen String s1 alle Leerzeichen löscht und das Ergebnis ausgibt.

(Aus „Hallo Welt wie gehts“ wird „HalloWeltwiegehts“)

10. Schreibe eine Methode

```
boolean enthaeltZiffern(String s) { ... } ,
```

die genau dann **true** zurückliefert, wenn der übergebene String mindestens eine Ziffer ('0' .. '9') enthält.

```
enthaeltZiffern("Hallo 1AI") → true
```

```
enthaeltZiffern("Hallo Welt") → false
```