

Java Intro Workshop

© Michael Inden, 2021

7 Aufgaben zu Lambdas & Streams

Aufgabe 1: Lambda-Basics

15 min

Gegeben seien folgende Namen als Ausgangsdaten:

```
var sampleNames = List.of("TIM", "TOM", "MIKE", "JOHN", "MICHAEL",  
"STEFAN");
```

- Schreiben Sie einen Lambda, der nur die Namen mit gerader Länge als Ergebnis liefert. Schreiben Sie danach einen Lambda, der basierend darauf die Namen ungerade Länge ermittelt. Nutzen Sie beide in Kombination mit `filter()`. Zur Ausgabe verwenden Sie `forEach()` in Kombination mit einer Methodenreferenz.
- Schreiben Sie einen Lambda, der den Namen auf die Länge abbildet und dadurch bei der Verarbeitung eine Liste mit Längenangaben liefert.

Aufgabe 2: `filter()`, `map()` und Lambdas

10 min

Gegeben seien die ersten Primzahlen wie folgt:

```
var primes = List.of(2, 3, 5, 7, 11, 13, 17, 19, 23, 29);
```

Sie sollten für die Werte jeweils den kubischen Wert (x^3) bestimmen und dann eine Filterung auf alle Werte im Bereich größer als 20 und kleiner als 500 vornehmen.

Das Ergebnis sollte folgendermaßen aussehen:

```
[27, 125, 343]
```

Aufgabe 3: takewhile() / dropwhile() in Kombination 10 min

Gegeben sei folgende Folge von Werten, aus denen die Nutzinformationen herausextrahiert werden sollen:

```
var data = List.of("x", "y", "<START>", "FIRST", "SECOND", "<END>", "x", "y");
```

Um das zu ermöglichen, existieren mit <START> und <END> zwei spezielle Markierungen. Nutzen Sie dieses Wissen gewinnbringend.

Das Ergebnis sollte folgendermaßen aussehen:

```
[FIRST, SECOND]
```

Aufgabe 4: Häufigkeiten von Namen 15 min

Stellen Sie sich vor, es wäre eine Liste mit Namen gegeben:

```
jshell> var names = List.of("Tim", "Tom", "Mike", "Jim", "Tim",  
                           "Mike", "James", "Mike")  
names ==> [Tim, Tom, Mike, Jim, Tim, Mike, James, Mike]
```

Nun wollen Sie wissen, welcher Name am häufigsten vorkommt bzw. genauer, für alle Namen deren Anzahl ermitteln. Bereiten Sie das Ergebnis als Map etwa wie folgt auf:

```
{'Tim': 2, 'Tom': 1, 'Mike': 3, 'Jim': 1, 'James': 1}
```

Realisieren Sie dies einmal mit speziellen Methoden aus Map und einmal mithilfe des Stream-APIs.

Aufgabe 5: Lambdas & Strings**30 min**

Die Verarbeitung von Strings wurde in Java 11 mit einigen nützlichen Methoden erleichtert.

Aufgabe 5a

Nutze folgenden Stream als Eingabe

```
Stream.of(2,4,7,3,1,9,5)
```

Realisieren Sie eine Ausgabe, die die sieben Zahlen untereinander ausgibt, jeweils so oft wiederholt, wie die Ziffer, also verkürzt wie folgt:

```
22
4444
7777777
333
1
999999999
55555
```

Aufgabe 5b

Modifizieren Sie die Ausgabe so, dass die Zahlen rechtsbündig mit maximal 10 Zeichen ausgegeben werden:

```
'      22'
'     4444'
'    7777777'
'     333'
'      1'
' 999999999'
'    55555'
```

Modifizieren Sie die Ausgabe so, dass die grössten Zahlen zuletzt ausgegeben werden, etwa folgendermaßen:

```
'    55555'
'   7777777'
' 999999999'
```

Tipp: Nutzen Sie eine Hilfsmethode

```
private static String formatRightAligned(final int num,
                                          final int desiredLength)
{
    // TODO
}
```

Aufgabe 5c

Worin liegt der Unterschied in folgenden zwei Varianten? Finde eine Wertebelegung, die diese besser zeigt, als die hier vorgegebene!

```
Stream.of(2,4,7,3,1,9,5).sorted().map(mapper1)
Stream.of(2,4,7,3,1,9,5).map(mapper2).sorted()
```