

Aufgabenzettel: Stream Operationen

Matthias Eurich – 2017-04-03



Table of Contents

Transformieren von Elementen: `map()`

Aufgabe: Cars

Aufgabe: Count

Mehrdimensionale Sammelstrukturen zusammenführen: `flatMap()`

Aufgabe: Mehrdimensionale-Liste

Aufgabe: Mehrdimensionales-Array Integer

Aufgabe: Mehrdimensionales-Array int

Filtern von Elementen: `filter()`

Aufgabe: Filter-Cars

Aufgabe: Filter-Predicate

Aufgabe: Eigenes Predicate

Transformieren von Elementen: `map()`

Aufgabe: Cars

Transformieren Sie die Elemente der Liste `cars` mit Verwendung der Methode `map()` so, dass sie in Großbuchstaben auf der Standardausgabe ausgegeben werden.

```
List<String> cars = Arrays.asList("BMW", "Audi", "Mercedes", "Porsche", "Opel",  
"Volkswagen");
```

Aufgabe: Count

Transformieren Sie die Elemente der Liste `cars` mit der Verwendung der Methode `map()` so, dass statt der Namen die Anzahl der Buchstaben ausgegeben wird.

Die Ausgabe soll für jedes Element folgendermaßen aussehen, wobei `<count>` nur ein Platzhalter für den tatsächlichen Wert ist:

```
Number of characters: <count>
```

Mehrdimensionale Sammelstrukturen zusammenführen: `flatMap()`

Aufgabe: Mehrdimensionale-Liste

Überführen Sie die nachfolgende, mehrdimensionale, Liste mit `flatMap()` in einen eindimensionalen Stream und geben Sie die Werte auf der Standardausgabe aus.

```
Arrays.asList(Arrays.asList(1, 2, 3, 4, 5), Arrays.asList(6, 7, 8, 9, 10));
```

Aufgabe: Mehrdimensionales-Array Integer

Überführen Sie das nachfolgende, mehrdimensionale, Array mit `flatMap()` in einen eindimensionalen Stream und geben Sie die Werte auf der Standardausgabe aus.

```
Integer[][] numbers = {new Integer[]{1, 2, 3, 4, 5}, new Integer[]{6, 7, 8, 9, 10}};
```

Aufgabe: Mehrdimensionales-Array int

Überführen Sie das nachfolgende, mehrdimensionale, Array mit `flatMap()` in einen eindimensionalen Stream und geben Sie die Werte auf der Standardausgabe aus.

```
int[][] intNumbers = {new int[]{1, 2, 3, 4, 5}, new int[]{6, 7, 8, 9, 10}};
```

Auf welches Problem stoßen Sie bei der Ausführung von `flatMap` ? Wie lässt sich das Problem lösen?

Filtern von Elementen: `filter()`

Aufgabe: Filter-Cars

Filtern Sie die Liste `cars` so, dass auf der Standardausgabe nur Marken ausgegeben werden, die mit dem Buchstaben `P` beginnen.

```
List<String> cars = Arrays.asList("BMW", "Audi", "Mercedes", "Porsche", "Opel", "Volkswagen");
```

Aufgabe: Filter-Predicate

Lagern Sie das `Predicate` aus der vorherigen Aufgabe in eine Variable aus, sodass der Aufruf von `filter()` folgendermaßen aussieht:

```
.filter(namesStartWithP)
```

Aufgabe: Eigenes Predicate

Implementieren Sie die Methode `namesStartWith`, um nicht für jeden Buchstaben des Alphabets ein eigenes `Predicate` händisch anlegen zu müssen. Die Methode `namesStartWith` nimmt als Parameter ein Suchwort `s` entgegen und gibt ein `Predicate` zurück.

```
public Predicate<String> startWith(String s) { ... }
```

Der Aufruf von `filter` soll folgendermaßen aussehen:

```
.filter(startWith("P"))
```

Last updated 2017-04-28 15:39:54 CEST