

SEW3 UE13 2024/25

Für die folgenden Beispiel sollen vorgegebene Funktionen aus Collection (List, ...) mit Hilfe von Lambda-Ausdrücken implementiert werden.

Folgende Methoden werden u.A. benötigt:

- `.forEach(Consumer)`
- `.replaceAll(UnaryOperator)`
- `.removeIf(Predicate)`
- `.sort(Comparator)`

Aufgaben zu Lambdas

Verwende für die folgenden Aufgaben stets Lambdas, keine (inneren) Klassen.

1) Schreibe die Methode

```
void print(Collection<?> collection) { ... }
```

die jedes Element der `collection` auf der Konsole ausgibt (`println`).

2) Schreibe die Methode

```
Collection<String> hauptworte(Collection<String> worte) { ... }
```

die eine Collection liefert, die nur diejenigen `worte` enthält, die Hauptworte sind.

- Hauptworte bestehen nur aus Buchstaben und beginnen mit einem Großbuchstaben.
- Verwende dazu die Collection-Methode `removeIf()`.
- Einzelne `worte` könnten auch Leerstrings oder `null` sein.

Beispiel:

Für die Collection: `>null, "Hugo", "HTL", "HTL3R", "", "Haus-Boot", "SEW", "Softwareentwicklung"<`
wird die Collection `>"Hugo", "HTL", "SEW", "Softwareentwicklung"<` zurückgegeben.

3) Schreibe die Methode

```
List<Double> mult(List<Double> zahlen, double faktor) { ... }
```

Die Methode soll eine `List<Double>` liefern, die alle `zahlen` enthält, jedoch multipliziert mit dem `faktor`.
Verwende dazu die List-Methode `replaceAll()`.

4) Schreibe die Methode

```
List<Double> func(List<Double> zahlen, UnaryOperator<Double> op) {...}
```

Die Methode soll eine `List<Double>` liefern, die alle `zahlen` enthält, wobei auf jede `zahl` der Operator `op` angewendet wurde. Verwende dazu die List-Methode `replaceAll()`.

5) Schreibe die Methode

```
List<String> numerisch(String ... elements) { ... }
```

die eine neue List liefert. Alle Elemente, **die wie ganze Zahlen aussehen**, sollen aufsteigend nach dem Zahlenwert sortiert werden. Alle anderen Elemente (Strings) sollen in alphabetisch sortiert werden.

Beispiel:

Die Elemente `"12", "Wappler", "8", "-5", "8", "Hugo", "-9", "-10", "Hugo", "-10", "7"`
werden folgendermaßen sortiert: `[-10, -10, -9, -5, 7, 8, 8, 12, Hugo, Hugo, Wappler]`