

SEW3 UE15 2024/25

Streams II

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

1) Lottoziehung

Schreibe die Methode

```
Set<Integer> lottoziehung() { .. } ,
```

die ein passendes Set mit 6 Lottozahlen aus dem Bereich 1 bis 45 zurückliefert, die jeweils eindeutig sein müssen. Die Zahlen im Set sollen aufsteigend sortiert sein.

2) Pi berechnen

Es gibt eine Vielzahl von Formeln, mit deren Hilfe man näherungsweise die Zahl PI berechnen kann. Eine davon ist folgende:

$$\frac{\pi}{4} = 1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} - \dots$$

(Beachte: der erste Summand kann als 1/1 interpretiert werden)

Schreibe eine Methode

```
... berechnePI(int anzGlieder) { .. } ,
```

die mit Hilfe der obigen Formel einen Näherungswert für die Zahl PI **berechnet und zurückliefert**.
Der Parameter `anzGlieder` gibt an, wie viele Bruchglieder (1/1, 1/3, 1/5, ...) für die Berechnung verwendet werden sollen.

!!Beachte die wechselnden Vorzeichen und dass die Formel **PI/4** berechnet!!

Wird ein ungültiger Wert für `anzGlieder` übergeben, soll die Methode -1.0 zurückliefern.

Teste (durch Ausgabe des Ergebnisses) mit mindestens 3 unterschiedlichen Werten für `anzGlieder`

Beispiele:

```
berechnePi(-5)      : -1.0
berechnePi(1)       : 4.0
berechnePi(2)       : 2.666666..
berechnePi(3)       : 3.466666..
berechnePi(1001)    : 3.1425915..
```

3) zähle Zeichen aus Vorrat: Schreibe die Methode

```
int zaehleZeichenAusVorrat(String s, String vorrat) { ... } ,
```

die zählt, wie viele Zeichen von **s** aus dem Vorratsstring **vorrat** sind.

```
zaehleZeichenAusVorrat ("Hallo 1AI" , „aeiou“) → 2  
zaehleZeichenAusVorrat ("*code*123#" , „?%!#"$“) → 3
```

4) erzeuge Zufallsstring aus Vorrat: Schreibe die Methode

```
String createRandomString(String vorrat, int len) { .. } ,
```

die einen String mit der Länge **len** erzeugt, der nur aus zufällig gewählten Zeichen aus dem String **vorrat** besteht.

Hinweis: Verwende die Methode `.collect(Collectors.joining())` .. die mehrere Strings aus einem Stream zu einem einzelnen String verbindet

```
createRandomString ("1234", 5) → (z.B.: 24211)
```

5) (anspruchsvoll – in der 1. Klasse ;)) Doppelte löschen: Schreibe die Methode

```
int[] deleteEquals(int[] ia) { .. } ,
```

die aus dem Array **ia** alle doppelten Zahlen herauslöscht und das (wahrscheinlich kürzere) Array zurückgibt.

Beispiel:

Das Array {1, 3, 3, 1, 2, 1, 5} nach dem Löschen der doppelten Zahlen: {1, 3, 2, 5}.

6) Zeichenstatistik: Schreibe die Methode

```
Map<Character, Integer> zeichenStatistik(String s) { .. } ,
```

die eine Map zurückliefert, deren Keys die Zeichen aus dem String **s** sind und als Value jeweils gespeichert ist, wie oft das jeweilige Zeichen vorkommt.

```
zeichenStatistik("Hallo Welt wie geht es heute") ->  
{ =5, a=1, e=6, g=1, h=2, H=1, i=1, l=3, o=1, s=1, t=3, u=1, w=1, W=1}
```

7) UE03 (2022/23) – Klasse Birthday-Datamining

Überarbeite die einzelnen Funktionen in der Klasse **BirthdayDatamining** aus UE03 so, dass möglichst viele Methoden mit Hilfe von Streams umgesetzt werden (dies betrifft vor allem die ersten 5 Teilaufgaben ..)