

## Aufgabenblatt 03: $\lambda$ 's, Streams, Datum, Zeitraum etc.

### 1 Datum und Erzeugen von Streams

Dem Aufgabenzeitraum angepasst, ranken sich die betrachteten Datumsangaben um den 1. Mai.

1. Der erste Mai wurde auf dem Gründungskongress der Zweiten Internationale 1889 als „Kampftag der Arbeiterklasse“ ausgerufen. Ermitteln Sie bitte mit Java-Code ab diesem Jahre diejenigen Jahre, in denen der 1. Mai auf einen Dienstag fiel bzw. fallen wird (bis 2100).

**Hinweis:** Erzeugen Sie einen *Stream* mittels *generate* oder *iterate*. Denken Sie bitte an das begrenzen!

2. Ermitteln Sie bitte die Abstände zwischen zwei 1. Mai's , die auf einen Dienstag fallen als Objekte einer geeigneten Klasse. Geben Sie diese Zeitraum in geeigneten Zeiteinheiten formatiert aus!
3. Ermitteln Sie bitte zu einem Datum (z. B. . 01. Mai 2018) folgende weitere Daten:
  - 3.1. Den nächsten Sonntag.
  - 3.2. Das nächste Schaltjahr.
  - 3.3. Das Datum 42 Tage später.

### 2 Funktionale Interfaces

Hier geht es um generische funktionale Interfaces, variabel lange Parameterlisten und Methoden einiger funktionaler Interface.

Die Methode *filter* erhält als Parameter ein *Predicate*. Wollen Sie mehrere *Predicates* verwenden, so müssen Sie sie mit logischen Methoden verknüpfen oder *filter* mehrfach aufrufen. Hierfür sollen Sie Hilfsmethoden schreiben. Das Interface *Predicate<T>* aus dem Paket *java.util.function* hat *default*-Methoden *and* und *or*.

1. Schreiben Sie bitte eine Methode *Predicate*<*T*> *andAll*(*Predicate*<? *super T*> [ / *predicates*, die eine variable Anzahl von *Predicates* übergeben bekommt und ein einzelnes *Predicate* zurückgibt, dass alle *Predicates* mit logischem und verknüpft prüft.
2. Schreiben Sie bitte eine Methode *Predicate*<*T*> *orAny*(*Predicate*<? *super T*> [ / *predicates*, die eine variable Anzahl von *Predicates* übergeben bekommt und ein einzelnes *Predicate* zurückgibt, dass alle *Predicates* mit logischem oder verknüpft prüft. (mindestens) eines der *Predicates* true liefert.
3. Vergessen Sie bitte das Testen nicht!

**Hinweise:** Stören Sie sich nicht am Typ des Parameters *Predicate*<? *super T*>, nehmen Sie das bitte einfach hin. Ich werde Ihnen diesen Teil von Generics in aller Ruhe in der Vorlesung vorstellen. Außerdem finden Sie viel in Kap. 18 des Skripts. Es ist nicht verboten, sich den Source-Code z. B. des Interfaces *Predicate* anzusehen.

### 3 Verarbeitung von Streams

Im Lernraum finden Sie Dateien *sgb-words.txt* (die 5757 englischen Worte mit fünf Buchstaben) und *scrabble.txt* (die zulässigen Worte für das Spiel „Word With Friends“, angeblich auch von vielen Scrabble Clubs akzeptiert) in einer Datei *worte.zip*. Wer eine entsprechende Datei für Scrabble Worte findet, die im Deutschen zulässig sind, kann gerne die verwenden. Alle Worte sind in Kleinbuchstaben, so dass Sie auf Groß- bzw. Kleinschreibung nicht achten müssen. Auf diesen Dateien können Sie nun „filtern“ (verwenden Sie *Files.lines* um aus der Datei einen *Stream* zu machen).

1. Verwenden Sie bitte einige Worte aus dem Anfang der Dateien, um die Methoden an Hand einiger *Predicates* zu testen!
2. Finden Sie bitte alle Palindrome mit sechs Buchstaben in *scrabble.txt*!
3. Finden Sie bitte alle Worte in *scrabble.txt*, die an der 2., 4. und 6. Stelle ein „e“ haben.
4. Finden Sie bitte alle Worte in *scrabble.txt*, die Paare von Vokalen enthalten (aa, ee, ... )!
5. Gibt es Worte, die in beiden Dateien vorkommen? Welche sind es?

Abgabetermin:

Donnerstag, 03.05.2018, 8:00