



Aufgabenblatt 2: Lambdas + Streams, Threads

Aufgabe 2.1: Streams und Lambdas

- Gegeben ist ein String-Array mit Benutzereingaben, die korrigiert werden sollen. Schreiben Sie dazu eine Verarbeitung mit einer Streams-Kette, die folgende Funktionalität bietet:
 - Entfernen von null-Eingaben
 - Abschneiden der Leerzeichen am Anfang und Ende
 - Konvertierung von Klein- zu Großbuchstaben
 - Ersetzen Ä→AE, Ö→OE, Ü→UE, ß→SS
 - Kürzen der Strings auf maximal 8 Zeichen
- Im Ergebnis sollen die Strings in einer List<T> vorliegen.
- Verwenden Sie wann immer möglich Lambda-Ausdrücke. Binden Sie die Lambda-Ausdrücke jeweils zunächst an eine Variable. Setzen Sie dann aus den Lambda-Ausdruck-Variablen die Stream-Kette zusammen.
- Beispiel:

`{"Eingabe ", "Äußeres ", null, "Strassen-Feger", " ein Haus" } →
[EINGABE, AEUSSERE, STRASSEN, EIN HAUS]`

Sichern Sie Ihre Lösung durch Tests ab.

Aufgabe 2.2: Threads

In dieser Aufgabe erstellen Sie die Simulation eines Flughafens. Diese besteht aus einem Flughafen und beliebig vielen Flugzeugen. Flughafen und Flugzeug werden durch je eine Klasse repräsentiert, und beide sind Threads. Ein Flug dauert zwischen 1 und 10 Sekunden (Zufallswert).

Ein Flugzeug durchläuft folgenden Zyklus:

- Erzeugung
- Flug (solange Zeit seit Start < flugDauer)
- Landung (1,5s)
- Gelandet

Der Flughafen repräsentiert die eigentliche Simulation. Dazu hat der Flughafen eine Liste der Flugzeuge. In seiner run()-Methode läuft eine Dauerschleife, die in jedem Durchlauf 0,5s wartet. Außerdem wird dort

- die Zeit (Ganzzahl) verwaltet und an die Flugzeuge gemeldet
- ein neues Flugzeug generiert und gestartet, wenn ein anderes gelandet ist

Das Flugzeug befinden sich immer in einem der drei Stati IM-FLUG, IM_LANDEANFLUG, GELANDET. In seiner run()-Methode wird der Zustand des Flugzeug abhängig von der bisherigen Flugdauer aktualisiert.

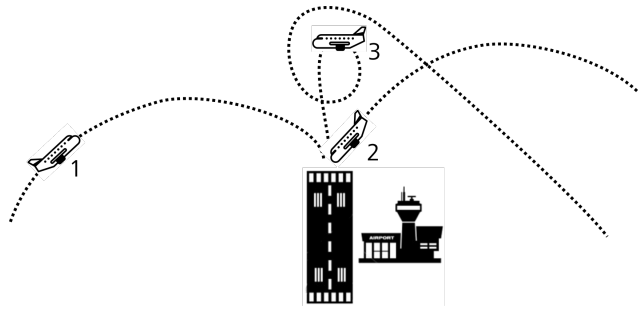
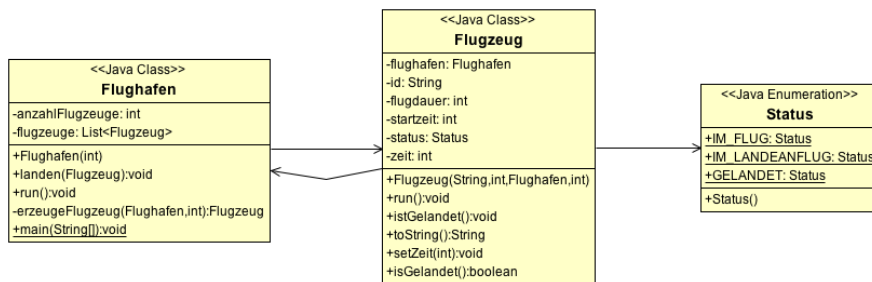


Abbildung 1: Simulationsszene mit drei Flugzeugen

Dieses Klassendiagramm kann als Hilfe für die Umsetzung dienen. In Details dürfen Sie aber von dieser Vorgabe auch abweichen.



Der Ablauf der Simulation soll auf der Konsole nachvollziehbar sein.