

Praktikum Programmiermethodik 2 (Technische Informatik)

WS 2016/2017, Hochschule für Angewandte Wissenschaften (HAW), Hamburg
Prof. Dr. Philipp Jenke, Kasperczyk-Borgmann



Für dieses Aufgabenblatt gelten die folgenden Regeln:

- Der Java Code Style (siehe EMIL) ist einzuhalten. Es werden keine Abgaben abgenommen, die diese Anforderungen nicht erfüllen.
- Pro Team muss ein gemeinsames Git-Repository für das Projekt vorhanden sein (z.B. auf GitHub, Bitbucket oder dem HAW-Informatik-Home-Verzeichnis).
- Der Code muss vollständig getestet sein.

Aufgabenblatt 2: Lambdas, Streams, Threads

Aufgabe 2.1: Lambdas

Lernziele: Definition und Verwenden funktionaler Interfaces (SAM), Definition und Anwenden von Lambda-Ausdrücken.

Aufgabe:

- Schreiben Sie eine Klasse Rechner, die die vier Grundrechenarten beherrscht. Dazu bietet Rechner die Methode `double berechne(Operation, double, double)`.
- `Operation` ist ein Enum mit Konstanten für die vier Grundrechenarten.
- Intern verwaltet Rechner die vier Rechenoperationen in einer Map, wobei die Konstanten aus dem Enum die Schlüssel sind.
- Die Werte der Map sind jeweils Lambdas vom Typ `BinaryOperator`.
- In `berechne()` wird über die `Operation` der richtige Lambda-Ausdruck gewählt, auf die Argumente angewendet und das Ergebnis zurückgeliefert.

Beispielaufruf: `rechner.berechne(Operation.ADDITION, 12, 11);`

- Definieren Sie ein eigenes funktionales Interfaces (SAM) `DoubleDoubleZuDouble` mit der Methode `werteAus()`. Die Methode soll zwei Doubles als Argumente akzeptieren und als Ergebnis einen Double-Wert zurückliefern.
- Testen Sie das Interface mit mindesten zwei Lambda-Ausdrücken (z.B. Multiplikation + Nullstelle der Gleichung $y=ax+b$ berechnen)

Aufgabe 2.2: Streams

Lernziele: Streams erzeugen, verarbeiten und terminieren

Aufgaben:

- 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.
- Beispiel:

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

Aufgabe 2.3: Threads

Lernziele: Threads erstellen, starten und einfache Synchronisation verwenden

Aufgaben: In dieser Aufgabe erstellen Sie die Simulation eines Flughafens. Diese besteht aus einem Flughafen und beliebig vielen Flugzeugen. Beide werden durch je eine Klasse repräsentiert, und beide sind Threads.

Ein Flugzeug durchläuft folgenden Zyklus:

- Erzeugung
- Flug (solange Zeit seit Start < flugDauer)
- Landung

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 500ms 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 Landen ist eine Methode des Flughafens. Diese Methode darf nur von einem Flugzeug gleichzeitig durchlaufen werden. Das Landen dauert 1500ms.

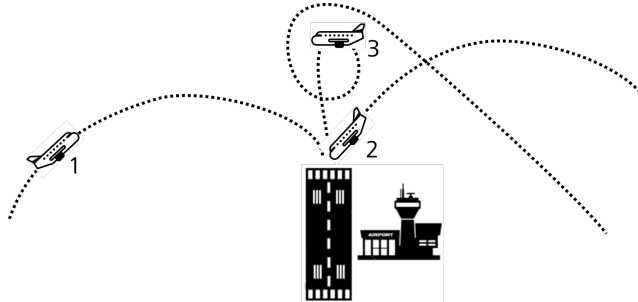
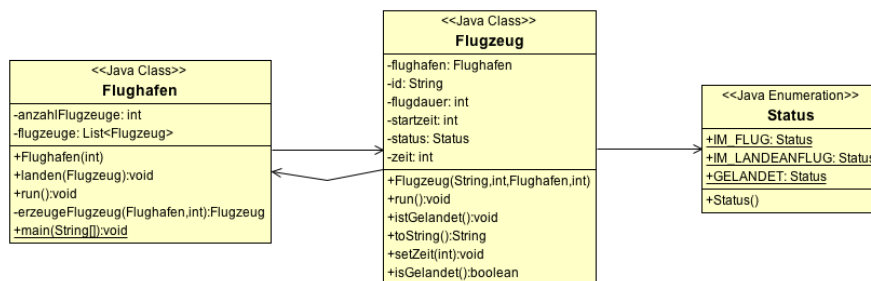


Abbildung 1: Flugzeug 1 ist im Flug, Flugzeug 2 landet und Flugzeug 3 muss vor der Landung warten, weil die Landebahn belegt ist.

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



Ein Ausschnitt aus einer Simulationsausgabe kann so aussehen:

[...]

Zeit: 8

Flugzeug Lufthansa 6072 (IM_LANDEANFLUG, Zeit bis Ziel: 0)

Flugzeug Lufthansa 2297 (IM_FLUG, Zeit bis Ziel: 1)

Flugzeug Lufthansa 9295 (IM_FLUG, Zeit bis Ziel: 2)

-> Flugzeug gelandet: Flugzeug Lufthansa 6072 (GELANDET, Zeit bis Ziel: 0)

Zeit: 9

-> Neues Flugzeug erzeugt: Flugzeug Air Berlin 7167 (IM_FLUG, Zeit bis Ziel: 16)

Flugzeug Lufthansa 2297 (IM_FLUG, Zeit bis Ziel: 0)

Flugzeug Lufthansa 9295 (IM_FLUG, Zeit bis Ziel: 1)

Flugzeug Air Berlin 7167 (IM_FLUG, Zeit bis Ziel: 7)

[...]