

Programmieraufgabe Teil 1:

Wir bauen einen Zug ... der im Folgenden beschriebene Aufbau und die dazugehörigen Funktionalitäten sollen in Java umgesetzt werden:

- Ein Zug besteht aus einer oder mehreren Lokomotiven und keinem bis mehreren Wagons
- Ein Zug gehört zu einer bestimmten Eisenbahngesellschaft.
- Jede Lokomotive und jeder Wagon hat ein fixes Leergewicht.
- Jede Lokomotive und jeder Wagon hat eine fixe Länge.
- Jede Lokomotive hat eine bestimmte Zugkraft, diese gibt an wieviel Gewicht die Lokomotive, zusätzlich zu ihrem eigenen Leergewicht, ziehen kann.
- Jeder Lokomotive kann eine bestimmte Anzahl an Passagieren (0 oder mehr) aufnehmen und hat ein maximales Zuladungsgewicht für Güter (0 oder mehr)
- Jeder Lokomotive hat eine Typenbezeichnung, einen Hersteller, ein Baujahr und eine eindeutige Seriennummer.
- Lokomotiven werden nach den Antriebsarten Diesel, Dampf, elektrisch unterschieden.
- Es gibt mehrere Arten von Wagons (Personenwagen, Schlafwagen, Speisewagen, Güterwagen, ...)
- Jeder Wagon kann eine bestimmte Anzahl an Passagieren (0 oder mehr) aufnehmen und hat ein maximales Zuladungsgewicht für Güter (0 oder mehr).
- Jeder Wagon hat eine Typenbezeichnung, einen Hersteller, ein Baujahr und eine eindeutige Seriennummer.
- Es kann das Leergewicht des gesamten Zuges abgefragt werden.
- Es kann die maximale Anzahl an Passagieren pro Zug abgefragt werden.
- Es kann das maximale Zuladungsgewicht für Güter pro Zug abgefragt werden.
- Es kann die maximale Zuladung eines Zuges abgefragt werden (= maximale Anzahl der Passagiere im Zug x 75kg + maximales Zuladungsgewicht für Güter)
- Es kann das maximale Gesamtgewicht des Zuges abgefragt werden.
- Es kann die Länge des Zuges abgefragt werden.
- Es soll geprüft werden können, ob der Zug fahrfähig ist, ob die im Zug vorhandenen Lokomotiven in der Lage sind, den Zug mit maximaler Zuladung (Passagiere und Güter) zu ziehen.
- Wenn ein Zug mehr als 0 maximale Passagiere hat, wird ein Schaffner auf diesem Zug benötigt.
- Pro 50 maximal beförderter Passagiere wird ein Schaffner benötigt.
- Es kann die Zahl der maximal benötigten Schaffner pro Zug abgefragt werden.
- Lokomotiven oder Wagons können immer nur einem Zug zugewiesen sein.
- Die Zusammenstellung von Zügen kann verändert werden.
- Weitere Prüfungen sind je nach Implementierung zu integrieren, zum Beispiel dürfen keine Zyklen innerhalb eines Zuges auftreten - Wagon A hängt an Wagon B und Wagon B hängt an Wagon A.
- Die einwandfreie Funktion der Implementierung ist mittels JUnit-Tests zu prüfen.

Programmieraufgabe Teil 2:

Wir lassen den in Teil 1 implementierten Zug fahren.

- Lokomotiven werden um eine durchschnittliche Geschwindigkeit in km/h ergänzt.
- Die Durchschnittsgeschwindigkeit eines Zuges entspricht der der langsamsten Lokomotive im Zug.
- Wir führen Bahnhöfe ein.
- Bahnhöfe haben Bahnsteige, diese geben an, wie viele Züge sich maximal gleichzeitig im Bahnhof befinden können.
- Bahnhöfe haben Verbindungen untereinander.
- Verbindungen sind bidirektional.
- Verbindungen zwischen den Bahnhöfen haben eine Länge (Angabe in Kilometer).
- Züge können entlang der Verbindungen zwischen Bahnhöfen hin- und herfahren.
- Die Fahrt entlang einer Verbindung nimmt Zeit in Anspruch - wir verwenden folgende Formel: Benötigte Zeit = Länge der Verbindung / Durchschnittsgeschwindigkeit des Zuges
- Wenn ein Bahnhof voll besetzt ist, muss ein Zug warten bis ein Bahnsteig frei wird.
- Bei Gleichzeitig ankommenden Zügen, wird der "Längere" bevorzugt behandelt.

Nach Erweiterung des Modells entsprechend obigen Angaben, bauen wir ein Testszenario/eine Simulation mit 3 Bahnhöfen und 3 Zügen:

Zug 1: Durchschnittsgeschwindigkeit 75 km/h

Zug 2: Durchschnittsgeschwindigkeit 50 km/h

Zug 3: Durchschnittsgeschwindigkeit 50 km/h

Bahnhof A: 2 Bahnsteige Verbunden mit Bahnhof B (150 km)

Zug 1 und 2 starten hier

Bahnhof B: 1 Bahnsteig

Bahnhof C: 2 Bahnsteige Verbunden mit Bahnhof C (100 km)

Zug 3 startet hier

Illustration: Anfangs- und Endposition der Züge

150 km 100km

A ===== B ===== C

Z1 Z3

Z2

Nun lassen wir die Züge 1 und 2 über B nach C fahren und wieder retour und Zug 3 über B nach A und retour. Die Züge halten dabei in jedem Bahnhof jeweils eine halbe Stunde.

Um das Ganze zu beschleunigen lassen wir es im Zeitraffer ablaufen. Fahrzeiten werden um den Faktor 3600 verkürzt (1 Stunde Halte- bzw. Fahrzeit entspricht also 1 Sekunde in der Simulation).