

Aufgabenblock 5



TECHNISCHE
UNIVERSITÄT
DARMSTADT

Softwarepraktikum, WS 17/18

Abgabetermin: 11.01.2018 23:59 Uhr

Bauphase, EnergyNetworkAnalyzer, Ausführungsphase

Hinweis

Wir messen der Einhaltung der Grundregeln der wissenschaftlichen Ethik größten Wert bei. Mit der Abgabe einer Lösung bestätigen Sie, dass Sie/Ihre Gruppe der alleinige Autor/die alleinigen Autoren des gesamten Materials sind. Falls Ihnen die Verwendung von Fremdmaterial gestattet war, so müssen Sie dessen Quellen deutlich zitiert haben. Weiterführende Informationen finden Sie unter <http://www.es.tu-darmstadt.de/lehre/plagiatshinweise.html>.

Weiterhin dürfen **keine Klassen unterhalb** von `org.sopra.internal.*` referenziert werden, dies führt zu einer Bewertung mit **0 Punkten** für die betreffende Aufgabe. Jede Abgabe muss alle notwendigen Dateien zum Kompilieren enthalten. Das in der jeweiligen Teilaufgabe geforderte Schema der Bezeichnung der Dateien ist einzuhalten.

Zusätzlich soll in jeder Klasse das Interface `org.sopra.api.exercises.ExerciseSubmission` implementiert werden und die in der Dokumentation beschriebene Funktionalität besitzen.

Lerninhalte

- Kennenlernen der Optimierungsmöglichkeiten einer Bauphase durch Implementieren einer vorgegebenen Bauphase.
- Umgang mit den in Aufgabenblock 3 und 4 implementierten Klassen durch Implementieren der Klasse `EnergyNetworkAnalyzer` üben.
- Kennenlernen der Optimierungsmöglichkeiten einer Ausführungsphase durch Implementieren einer vorgegebenen Ausführungsphase.

Aufgabe 5.1 - First Game (10 Punkte)

In dieser Aufgabe sollen Sie die Klasse `FirstGame` implementieren, die eine Bauphase repräsentiert. Die Klasse soll das Interface `Game` aus dem Paket `org.sopra.api` implementieren. Speichern Sie die Klasse im Paket `solutions.exercise5` ab.

Für diese Aufgabe sollen Sie nur die Methode `buildPhase(Scenario scenario)` implementieren, lassen Sie die Methode `executionPhase` leer.

In der Methode `buildPhase(Scenario scenario)` sollen Sie folgendes Verhalten implementieren:

- Bauen Sie zwei Windkraftwerke an möglichst geeigneten Orten. Es wird angenommen, dass ein Ort dann geeigneter als ein anderer ist, wenn die Baukosten für ein Windkraftwerk geringer ausfallen. Auswirkungen des Standorts auf das Produktionsniveau sollen nicht berücksichtigt werden. Verwenden Sie zur Ermittlung der Baukosten und zum Vergleich von Windkraftwerken die Implementierungen aus Aufgabenblock 2. Falls das übergebene Szenario nicht über zwei potenzielle Standorte für Kraftwerke verfügt, soll eine `RuntimeException` geworfen werden.

- Bauen Sie außerdem möglichst viele Leitungen auf HIGH_VOLTAGE aus. Welche Leitungen zum Ausbau ausgewählt werden, ist beliebig.
- Für den Bau der Kraftwerke und den Ausbau der Leitungen dürfen insgesamt maximal Kosten in Höhe von 20.000 entstehen.

Aufgabe 5.2 - EnergyNetworkAnalyzer implementieren (15 Punkte)

In dieser Aufgabe sollen Sie die Klasse `EnergyNetworkAnalyzerImpl` implementieren, die die für das Energienetz maximalen Auslastungen von Leitungen, Kraftwerken und Konsumenten berechnet. Die Klasse ist von `AbstractEnergyNetworkAnalyzer` im Paket `org.sopra.exercises.exercise5` abgeleitet. Speichern Sie die Klasse im Paket `solutions.exercise5` ab.

Zur Berechnung der maximalen Auslastung soll auf die Implementierungen aus den Aufgabenblöcken 3 und 4 zurückgegriffen werden.

- Implementieren Sie einen öffentlichen Konstruktor, der den Konstruktor der Superklasse aufruft.
- Implementieren Sie die Methode `createFlowGraph`. Die Methode wird im Konstruktor der Superklasse verwendet, um das Feld `flowGraph` zu initialisieren. Orientieren Sie sich zur Umwandlung eines Energienetzes in einen Flussgraphen an der Beschreibung der Javadocs und den Folien der Frontalveranstaltung.
- Implementieren Sie die Methode `calculateMaxFlow`, die für den Flussgraphen in `flowGraph` den maximalen Fluss berechnet und die Ergebnisse in den Feldern `producerLevels`, `consumerLevels` und `powerlineLevels` abspeichert. Verwenden Sie zur Bestimmung des maximalen Flusses die Implementierung des Ford-Fulkerson Algorithmus aus Aufgabenblock 4.
 - Beachten Sie zur Bestimmung der Auslastung von *Produzenten* folgenden Hinweis: Nach Berechnung des maximalen Flusses entspricht im Flussgraphen die Auslastung eines Produzenten dem Fluss vom `super source` Knoten zum jeweiligen Produzenten.
 - Beachten Sie zur Bestimmung der Auslastung von *Konsumenten* folgenden Hinweis: Nach Berechnung des maximalen Flusses entspricht im Flussgraphen die Auslastung eines Konsumenten dem Fluss vom jeweiligen Konsument zum `super sink` Knoten.
 - Beachten Sie zur Bestimmung der Auslastung von *Leitungen* folgenden Hinweis: Nach Berechnung des maximalen Flusses entspricht im Flussgraphen die Auslastung einer Leitung zwischen den Knoten i und j mit den Kanten e_{ij} und e_{ji} dem größeren der beiden Flüsse $f(e_{ij})$ und $f(e_{ji})$. Die Auslastung ist 0, falls beide Flüsse $f(e_{ij})$ und $f(e_{ji})$ gleich 0 sind.

Aufgabe 5.3 - SunEnergyBroker implementieren (15 Punkte)

In dieser Aufgabe sollen Sie die Klasse `SunEnergyBroker` implementieren, die von der Klasse `AbstractSunEnergyBroker` im Paket `org.sopra.exercises.exercise5` abgeleitet sein soll und eine Ausführungsphase repräsentiert. Speichern Sie die Klasse im Paket `solutions.exercise5` ab.

Der `SunEnergyBroker` soll für Szenarien, die ausschließlich aus `IndustrialParks` und `SolarPowerPlants` bestehen, die `IndustrialParks` so regeln, dass die angeforderte Leistung der `IndustrialParks` zu jedem Zeitpunkt durch `SolarPowerPlants` zur Verfügung gestellt werden kann. Außerdem sollen die `IndustrialParks` so geregelt werden, dass in jeder Runde mindestens 50 Prozent der bereitgestellten Leistung abgenommen wird. Bauen Sie keine zusätzlichen Kraftwerke und erhöhen Sie nicht die Kapazität von Leitungen.

Der `SunEnergyBroker` soll die genannten Vorgaben für Szenarien erfüllen, für die gilt:

-
- Es existieren ausschließlich Produzenten des Typs SolarPowerPlant, Konsumenten des Typs IndustrialPark sowie unbebaute Knoten des Typs TransformerStation.
 - Die Anzahl an Produzenten und Konsumenten ist so bemessen, dass die Zielvorgaben erfüllbar sind.
 - Es ist ausreichend Leitungskapazität verfügbar, um die Zielvorgaben zu erfüllen.

Eine Überprüfung, ob die Annahmen für das übergebene Szenario erfüllt sind, ist nicht erforderlich.

Kritik, Verbesserungsvorschläge und Bug-Report

Sollten Sie Kritik oder Verbesserungsvorschläge haben bzw. Bugs finden, dann nutzen Sie dafür bitte den Bug-Report Button im Moodle-Kurs.

