

Portfolio-Prüfung Programmierung II

Studiengang Wirtschaftsinformatik WI22A und WI22C

Dozent: Prof. Dr. Alexandros Nanopoulos

Organisatorisches

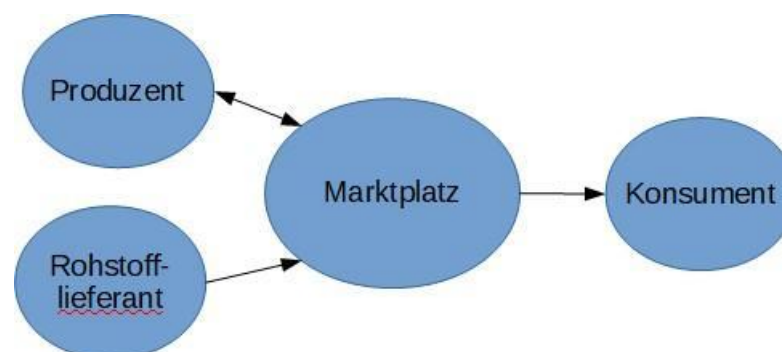
Die Portfolio-Prüfung findet unabhängig von der Klausur statt und wird auch unabhängig davon mit einer Note bewertet. Die Prüfung beinhaltet eine Programmieraufgabe, deren Lösung in Gruppen zu je 3 (maximal 4) Studierenden erstellt und am Ende der Vorlesungszeit kurz präsentiert wird (ein gemeinsam Termin pro Kurs am **26.05.2023** um 11:30 wird vereinbart). Es wird erwartet, dass alle Gruppenmitglieder gleichermaßen an der Entwicklung beteiligt werden und zu den entsprechenden Code-Passagen Auskunft geben können.

Aufgabenstellung

Es soll eine vereinfachte Marktsimulation geschrieben werden, wobei die Anzahl und Typen der Akteure eingeschränkt werden. Eine zentrale Rolle in dieser Simulation ist der Marktplatz, welcher die Anfragen der Produzenten und Konsumenten entgegennimmt und die Waren-Transaktion durchführt.

Insgesamt gibt es folgende Akteure (s.u. Abbildung):

- (mindestens drei) Produzenten,
- (mindestens drei) Konsumenten und
- (mindestens drei) Rohstofflieferanten



Die Rohstofflieferanten liefern pro Zeiteinheit nur ein bestimmtes Kontingent an Rohstoffen (z.B. zufallsbestimmt 2 aus 5 Rohstoffen in einer Menge von 5 bis 10 Einheiten).

Die Produzenten verfügen jeweils über zwei unterschiedliche Produktionsketten, um aus den Rohstoffen höherwertige Produkte (Konsumgüter) herstellen zu können. Jede Produktionskette soll aus mindestens zwei und höchstens drei Rohstoffen bestehen. Die Rohstoffe werden bei der Produktion verbraucht.

Die Konsumenten erwerben die Produkte, sobald sie auf dem Marktplatz angeboten werden. Dabei konkurrieren die Konsumenten um das jeweilige Produkt und erwerben pro Zeiteinheit 3 bis 5 Einheiten.

Die Rohstoffe wie auch die Produkte werden mit einem Preis angeboten, der sich durchaus der Zeit variieren kann. Die Erwerbung von Rohstoffen (bzw. Produkten) findet statt, nur wenn der Preis aus der Sicht von Produzenten (bzw. Konsumenten) als angemessen betrachtet wird. Für diese Simulation sollen alle Akteure über unbegrenzte Geldmittel verfügen.

Umsetzung

Der Marktplatz soll ein Singleton darstellen. Beachten Sie, dass es mehrere konkurrierende Anfragen an den Marktplatz geben kann (Stichwort: Synchronisation). Alle Akteure sind als unabhängige Threads zu realisieren.

Die Konfiguration über die Lieferung der Produzenten und der Rohstofflieferanten wird in den entsprechenden Konstruktoren festgelegt.

Der Code für die Produktionsketten (d.h. die Simulation der Herstellung der Produkte aus Rohstoffen) sollte während des Ablaufes nicht verändert werden.

Die Preise der Rohstoffe und Produkte können auch als Teil der Konfiguration in den entsprechenden Konstruktoren festgelegt werden. Die Höhe der Preise, die aus der Sicht der Produzenten und Konsumenten als angemessen betrachtet werden, kann auch analog festgelegt werden.

Die Variation in den angebotenen Preisen kann mithilfe von „zufälliger Irrfahrt“ (auf Englisch: *Random Walk*) simuliert werden: d.h. dass wenn der Preis eines Rohstoffs oder Produkts an Zeiteinheit t gleich P_t ist, dann wird der Preis in der nächsten Zeiteinheit gleich $P_{t+1} = P_t + R$; wobei R ist eine Zufallszahl, die gleichmäßig im Bereich $[-1\%P_t, +1\%P_t]$ verteilt ist. Effektiv also werden die Preise pro Zeiteinheit, um einen zufälligen Prozentwert zwischen -1% und +1% verändert. Die Anfangspreise P_0 sind, wie oben erwähnt, mithilfe von Konstruktoren definiert werden. Sie können selber entscheiden, ob die Höhe von angemessenen Preisen auch zeitlich angepasst wird oder nicht.

Lassen Sie die Simulation etwa 100 Zeiteinheiten simulieren. Pro Zeiteinheit sollen die verfügbaren Rohstoffe der drei Lieferanten, den Erwerb der Rohstoffe durch die Produzenten, die dadurch produzierten Produkte, die von den Konsumenten erworbenen Produkte und die entsprechenden Preise aufgelistet werden. Geben Sie also pro Zeiteinheit den Status aller Akteure (klar formatiert) auf der Kommandozeile aus. Diese Ausgabe kann aus/eingeschaltet werden.

Programmiertechnische Elemente

Der gesamte Quellcode soll mit JavaDoc kommentiert sein. Benutzen Sie eigene und aussagekräftige Exception-Klassen. Gruppieren Sie die zusammengehörigen Klassen in entsprechende Pakete.

Abgabe

Abzugeben wäre, per Moodle, **eine ZIP-Datei** pro Team mit folgenden Elementen:

- eine ausführbare JAR-Datei
- vollständiger und mit JavaDoc kommentierter Quellcode ein Bericht bzw. Dokumentation mit einer kurzen (max. 8 Seite) Beschreibung des Konzepts.

Abgabefrist: **25.05.2023** (EOB).