Technische Hochschule Ingolstadt
Fakultät EI
Prof. Dr. Christian Facchi

Übung Verteilte Systeme
WS 2018/19
Übungsblatt 5

1) Aufgabe (reader/writer in JAVA):

Implementieren Sie eine Lösung des reader/writer-Problems (Blatt 3, Aufgabe 1) mit Java.

Verwenden Sie zwei Semaphore (Datentyp `Semaphor` aus `java.util.concurrent`) mit den zugehörigen Operation `acquire` und `release`. Die Anzahl der gerade lesenden Prozesse koennen Sie mit einer statischen globalen Variable realisieren, statt des Semaphors aus Blatt 3.

Gehen Sie hierbei wie folgt schrittweise vor:

a) Implementieren Sie die Klassen `DBReader` und `DBWriter`, die jeweils einen Lese- bzw. Schreibprozess implementieren, als threads und die Klasse Threads erweitern (`extends`). Verwenden Sie hierbei als ganzzahliges Attribut `myNumber`, das die Instanz des jeweiligen Prozesses beschreibt (z.B. bezeichnet 3 den dritten Leseprozess). Realisieren Sie auch einen dementsprechenden Konstruktor.

b) Implementieren Sie eine umgebende Klasse `ReaderWriter`, in der die globalen Daten deklariert und initialisiert werden. Dann starten Sie jeweils 3 Lese und 4 Schreibprozesse.

Testen Sie die Klassen ausreichend. Überprüfen Sie hierbei insbesondere die Programmausgaben.

2) Aufgabe (Dining Philosophers in JAVA):

Realisieren Sie nun den aus Blatt 2 bekannten Philosophenalgorithmus. Da sie in JAVA keine Semaphorgruppen behandeln können, verwenden Sie die Lösungsidee aus Blatt2 / Aufgabe 2.

Implementieren Sie hierzu eine Klasse `Philosopher`, die die Aktionen einer Instanz eines Philosophen in einer Endlosschleife beschreibt.

Testen Sie die Klassen ausreichend. Überprüfen Sie hierbei insbesondere die Programmausgaben, die Sie einführen.