Hochschule **RheinMain**
University of Applied Sciences
Wiesbaden Rüsselsheim Geisenheim

Fachbereich Design Informatik Medien
Studiengang Bachelor Allgemeine Informatik

Programmieren 3 AI – Praktikum

Übungsblatt 2

1,5 Punkte

Sebastian Flothow

2011-10-24

Abnahme: 2011-10-31
Zur Abnahme ist die persönliche Anwesenheit erforderlich. Bearbeiten Sie die Aufgaben so, dass Sie nicht nur funktionierenden Code vorzuweisen haben, sondern diesen auch erläutern und Fragen dazu beantworten können. Zusätzlich sind die Lösungen am Tag der Abnahme per Email an `sebastian@flothow.de` einzusenden; dies dient u.a. der Prüfung auf eventuelle Plagiate.

Die speisenden Wilden

Mehrere Wilde sitzen um einen Topf, aus dem sie gemeinsam essen. Solange der Topf nicht leer ist, können alle Wilden unabhängig voneinander etwas aus dem Topf herausnehmen und verspeisen. Wenn der Topf leer ist muss der Koch geweckt werden, der daraufhin neues Essen kocht. Der Koch bereitet immer eine feste Anzahl von Portionen auf einmal zu und darf nur geweckt werden wenn der Topf leer ist. Wilde, die Essen aus dem Topf nehmen wollen, müssen dann warten bis der Koch fertig ist.

Zu diesem Blatt existiert vorgegebener Quelltext, der die Grundstruktur dieses Ablaufs umsetzt, jedoch noch nicht die nötigen Synchronisationsmechanismen enthält. Laden Sie sich von der Veranstaltungswebsite die Datei `DiningSavages.zip` herunter. Dieses Archiv wurde mit der Export-Funktion von Eclipse erzeugt, lässt sich also direkt als Projekt in Eclipse importieren, kann aber natürlich auch mit beliebigen anderen Programmen entpackt werden.

Öffnen Sie die Datei `doc/index.html` in einem Webbrowser Ihrer Wahl und lesen Sie die Dokumentation der vier Klassen. Lesen Sie dann auch die Quelltexte der Klassen. Lassen Sie das Programm laufen und beobachten Sie die Ausgaben; nach ca. 10 s müssten

sich alle Threads infolge einer Exception beendet haben. Wiederholen Sie die Schritte in diesem Absatz solange, bis Sie verstanden haben warum die einzelnen Exceptions auftreten.

Lesen Sie außerdem in der Dokumentation der Java-Klassenbibliothek auf der Seite zu `java.lang.Object` die Beschreibung der Methoden `wait`, `notify` und `notifyAll`, insbesondere die Hinweise zur Verwendung dieser Methoden in Verbindung mit `synchronized`-Blöcken und dem Umgang mit *spurious wakeups*.

Aufgabe 1

Erweitern Sie den vorgegebenen Quelltext so, dass die beschriebenen Synchronisationsbedingungen eingehalten werden. Ihre Lösung muss folgenden Anforderungen genügen:

- Es dürfen keine Exceptions auftreten
- Es dürfen keine Verklemmungen möglich sein
- Beliebig viele Wilde müssen gleichzeitig essen (also die Methode `eat` ausführen) können
- Wenn auf das Eintreten einer Bedingung gewartet wird muss dies blockierend geschehen; sogenanntes *busy waiting*, bei dem fortlaufend Rechenzeit verbraucht wird, ist also nicht zulässig. Wenn ein Thread eine potentiell blockierende Methode aufruft soll er vorher eine entsprechende Ausgabe machen, um den Programmablauf besser nachvollziehbar zu machen.