Übungsblatt 9

Diese Aufgabe dient zur Wiederholung der Synchronisationskonstrukte. Sie demonstriert den Einsatz von Monitoren an einer typischen Problemstellung der Betriebssystemprogrammierung.

Eine typische Aufgabe von Betriebssystemen ist die Vergabe von Betriebsmitteln (Speicher, Drucker, Laufwerke usw.) Dabei kommt es darauf an, die Prozesse so zu synchronisieren, daß möglichst viele Prozesse gleichzeitig arbeiten können, ohne sich bei der Benutzung von Betriebsmitteln in die Quere zu kommen. (Man stelle sich das Ergebnis vor, wenn zwei Prozesse ohne Synchronisierung gleichzeitig den gleichen Drucker benutzen würden.)
Außerdem ist dafür zu sorgen, daß kein Prozeß unendlich lang auf ein angefordertes Betriebsmittel warten muß (das sog. Verhungern).

Diese klassische Aufgabenstellung wird durch das Philosophenproblem veranschaulicht: Um einen runden Tisch sitzen fünf Philosophen, jeweils vor einem Teller Spaghetti. Zwischen zwei Philosophen liegt je eine Gabel (also insgesamt ebensfalls fünf). Zum Essen benötigt jeder Philosoph die beiden Gabeln, die links und rechts von ihm liegen. [Es wurde bereits darauf hingewiesen, daß man Spaghetti auch mit einer Gabel essen kann. Man sollte daher vielleicht die Spaghetti durch Reis und die Gabeln durch Stäbchen ersetzen, was die Aufgabenstellung jedoch nur unwesentlich verändert.]

Der Tagesablauf eines Philosophen besteht nur aus Denken und Essen. Wird ein Philosoph beim Denken hungrig, so nimmt er nacheinander (!) die beiden Gabeln neben sich auf (sofern nicht bereits einer der Tischnachbarn diese besitzt) und ißt. Nach dem Essen legt er die Gabeln wieder ab und denkt, beginnt den Zyklus also von Neuem.

Wie sich unschwer erkennen läßt, spielen die Philosophen die Rolle der Prozesse, während die Gabeln für Betriebsmittel stehen. Das Denken entspricht den Phasen eines Prozesses, in denen er rechnet, ohne Betriebsmittel anzufordern. Das Essen symbolisiert die Phasen, in denen ein Prozeß zum Rechnen bestimmte Betriebsmittel (die Gabeln) benötigt.

Implementieren Sie eine Lösung des verallgemeinerten Philosophenproblems für n Philosophen und n Gabeln. Entwerfen Sie dazu die Klassen Philosoph, Gabel und Tisch. Ihre Lösung soll folgende Anforderungen erfüllen:

- Kein Philosoph wartet unendlich lang, bis er beide Gabeln erhält. Es kommt also nicht zum Verhungern (Verklemmungsfreiheit, Fairneß).
- Es soll maximale Parallelität ermöglicht werden, d.h. in der Regel ißt annähernd die Hälfte der Philosophen.
- Das Aufnehmen der Gabeln soll nacheinander geschehen, d.h. ein anderer Philosoph hätte die Chance, zwischen dem Aufnehmen der ersten und der zweiten Gabel, die letztere wegzuschnappen.
  Diese Forderung rührt daher, daß Prozesse zwischen dem Anfordern zweier Betriebsmittel vom Scheduler unterbrochen werden können und so ein anderer Prozeß die Chance erhält, sich das Betriebsmittel zu sichern.

Zur Implementierung der Phasen Denken und Essen können Sie die Methode `sleep` der Klasse `Thread` verwenden.

Begründen Sie ausführlich, weshalb Ihre Lösung die obigen Forderungen erfüllt.