Prof. Dr. Th. Letschert

THM TECHNISCHE HOCHSCHULE MITTELHESSEN
MNI Mathematik, Naturwissenschaften und Informatik

NVP – Nebenläufige und Verteilte Programme

Aufgabenblatt 5

Aufgabe 1

1. Was ist ein Monitor?
2. Was ist ein *geschachtelter* und was ein *wiederbetretenden* Monitor?

Aufgabe 2

Seit Dijkstra im Jahre 1968 das Problem vorstellte[1] ist es eine wesentliche Aufgabe der Informatik (–Studenten) fünf nachdenkliche und gelegentlich hungrige Philosophen am Leben zu halten. Diese Philosophen sitzen an einem runden Tisch mit fünf Plätzen und tun nichts anderes als denken und essen. Zu Essen gibt es immer nur Spaghetti und ein Philosoph benötigt dazu einen Teller und zwei Gabeln. Die Spaghetti werden von dienstbaren Geistern stets nachgefüllt. Die Philosophen sind zu unpraktisch um Spaghetti – wie es sich gehört – mit einer Gabel und eventuell einem Löffel zu essen. Sie benötigen zwei Gabeln um essen zu können.

Die Teller sind unproblematisch: Auf dem Tisch stehen fünf Teller, jeweils einer vor jedem Philosoph, und jeder hat seinen eigenen Teller. Sie werden automatisch aufgefüllt.

Bei den Gabeln herrscht dagegen Knappheit: Ein Philosoph teilt sich die Gabel links von seinem Platz mit seinem linken Nachbarn und die Gabel rechts von seinem Platz mit seinem rechten Nachbarn.

Die speisenden Philosophen werden in der Regel als Problemstellung einer Allokation von Ressourcen betrachtet, bei der es zu Verklemmungen kommen kann. Greifen alle gleichzeitig nach ihrer linken Gabel, dann werden sie alle verhungern denn alle rechten Gabeln sind anschließend vergeben.

Es geht aber zunächst einmal, unabhängig vom Thema Verklemmung, um ein Synchronisationsproblem. Hier konzentrieren wir uns auf die Synchronisation, Verklemmungen spielen keine Rolle!

1. Informieren Sie sich (z.B. Wikipedia, Google, Unterlagen zu PIS, BS, etc.) über das Beispiel der der speisenden Philosophen und erläutern Sie, welche Synchronisationsprobleme gelöst werden müssen und in wie weit das Beispiel eine praktische Relevanz hat.
2. Modellieren Sie dann die Problemstellung nach dem Monitorprinzip: Philosophen sind die aktiven und Gabeln die passiven Objekte. Synchronisationsprobleme (gegenseitiger Ausschluss, Bedingungssynchronisation) werden in passiven Objekten gelöst: Welche, wie?
3. Definieren Sie geeignete Klassen. Testen Sie Ihre Lösung.

Aufgabe 3

Eine Brücke hat eine begrenzte Tragfähigkeit. Um eine Überlastung zu vermeiden muss sich jedes Fahrzeug vor der Auffahrt mit seinem Gewicht melden. Die Weiterfahrt wird nur erlaubt, wenn dadurch keine Überlast der Brücke entsteht.

Modellieren Sie die Problemstellung in Scala. Konzentrieren Sie sich dabei auf das Synchronisationsproblem. Autos sind aktive Objekte, die Brücke wird von einen Monitor bewacht. Ein– und ausfahrende Autos haben sich bei diesem zu

[1] E.W. Dijkstra *Cooperating Sequential Processes* in F. Genuys (Ed.): *Programming Languages*; Academic Press New York 1968

melden bevor sie die Brücke betreten, bzw. nachdem sie sie verlassen haben. (Es ist nicht notwendig die Brücke selbst zu modellieren ;-)