

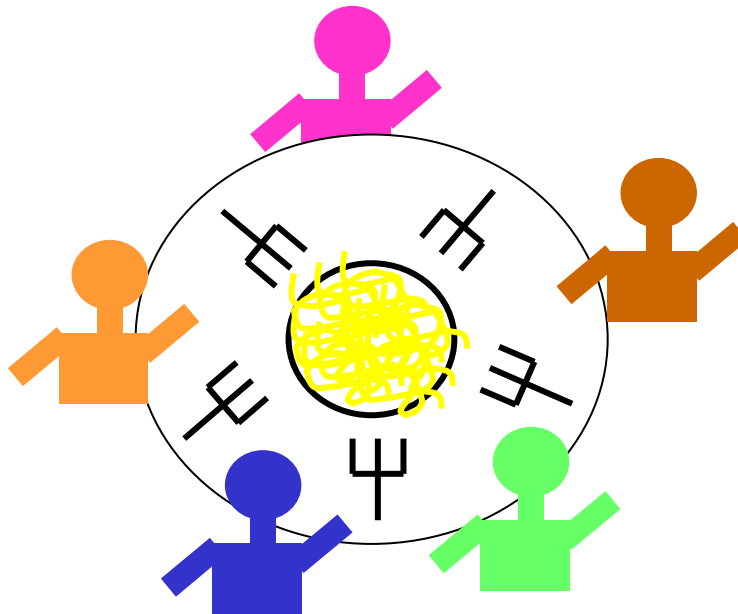
**UNIVERSITÄT KARLSRUHE (TH)**  
**Fakultät für Informatik**  
System Architecture Group  
**Frank Bellosa, Gerd Liefländer, Philipp Kupferschmied**  
**Dominik Bruhn, Atanas Dimitrov,**  
**Jonathan Dimond, Johannes Weiß**

**Basispraktikum Systemarchitektur - WS 2008/2009**

## **Dijkstra's Philosophenproblem**

### **1 Thematik**

Fünf (bzw.  $n > 1$ ) Philosophen sitzen an einem Tisch, in dessen Mitte eine nie versiegende Schüssel voll dampfender Spaghetti (al dente) steht. Zwischen je zwei Philosophen liegt zudem eine Gabel, allerdings benötigen die etwas ungeschickten Philosophen zwei Gabeln, wenn sie Spaghetti essen wollen. Die folgende Skizze verdeutlicht den Zusammenhang:



Das Leben dieser „armen Philosophen“ besteht nur aus folgendem Zyklus:

Life of a philosopher:

```
repeat forever
begin
  thinking
  getting hungry
  getting 2 forks
  eating
end
```

## 2 Grundlagen

Frage 1: Um welches prinzipielle Problem aus dem Betriebssystem handelt es sich dabei!

Frage 2: In welche Koordinationsprobleme könnte unsere 5 Philosophen, zumindest teilweise geraten, wenn wir nicht sorgfältig koordinieren?

### 2.1 Modellüberlegungen

Jeder Philosoph ist als Thread zu etablieren. Gestehen wir den vielleicht etwas weltfremden Philosophen gleichwohl etwas Alltagstauglichkeit zu, so sollten sie sich selbst so untereinander verständigen, dass jegliche Konflikte oder gar handgreifliche Auseinandersetzungen ausbleiben, sie also insbesondere keinen Oberschiedsrichter benötigen, damit sie ihr beschauliches Philosophenleben weiterhin unfallfrei genießen können.

Man beachte bei der Implementierung, dass Sie gezielt die von Ihnen erkannten Problemsituationen reproduzierbar herbeiführen können. Man beachte bei der Modellierung, dass die Denk- und Essensphasen jedes Philosophen im Lauf seines Lebens unterschiedlich lang sein können, außerdem mag es unter ihnen auch richtige „Vielesser“ und „Kurzdenker“ bzw. „Langdenker“ und „Wenigesser“ geben.

Sie können verschiedene Lösungsprotokolle für das Philosophenproblem überlegen und implementieren, wobei Sie auch auf die unterschiedlichsten Java-Koordinationsobjekte zurückgreifen dürfen.

### 2.2 Implementierung und Visualisierung

Man sollte durch entsprechende Schieberegler das Verhalten aller bzw. einzelner Philosophen verändern können. Sollte einer der Philosophen nicht rechtzeitig Spaghetti zu sich nehmen können, dann soll er verhungern.

Die Visualisierung sollte das Wesentliche hinsichtlich der ausgewiesenen Problematik hervorheben, dagegen unwesentliche Details weglassen.