

Lösung Übungsblatt 4

Christoph van Heteren-Frese (Matr.-Nr.: 4465677),

Sven Wildermann (Matr.-Nr.: 4567553)

Tutor: Alexander Steen, eingereicht am 9. Mai 2013

Aufgabe 1

Aufgabe 2

Aufgabe 3

Aufgabe 4

a)

Das in [1] erläuterte Prozessmodell unterscheidet fünf Zustände: *nicht existent*, *bereit*, *aktiv*, *blockiert* und *beendet*. Eine Semaphore kann in Java wie folgt implementiert werden:

```
1  public class Semaphore {
2      private int zaehler = 0;
3      private Thread warteschlange = null;
4      public void init(int i)
5      {
6          zaehler = i;
7          warteschlange = null;
8      }
9      public void P(Thread thread)
10     {
11         zaehler--;
12         if(zaehler<0)
13         {
14             warteschlange = thread;
15             try
16             {
17                 warteschlange.suspend();
18             }
19             catch(Exception e)
20             {
21             }
22         }
23     }
24     public void V()
25     {
26         zaehler++;
27         // if(warteschlange!=null)
28         if(zaehler<=0)
29             warteschlange.resume();
30     }
```

Listing 1: Implementierung eines Semaphor in Java nach [2]. (Die Methoden `suspend()` und `resume()` sind veraltet und sollten in der Praxis nicht mehr verwendet werden.)

b)

c)

Literatur

- [1] Christian Maurer. *Nichtsequentielle Programmierung mit Go 1 Kompakt*. Springer Vieweg, 2012. ISBN 978-3642299681.
- [2] Guido Kramann. Implementierung und Anwendung von Semaphoren. URL http://www.kramann.info/67/_Echtzeitsysteme/05/_Semaphor/03/_Semaphor/.