

Programmierkurs Anwendungsentwicklung

Praktikum 07

Aufgabe 1

- In dem unten angegebenen Beispiel dauert die Abarbeitung der Methode `dauertLange ()` etwa 5 bis 10 Sekunden. Aufgabe ist es, diese Zeit zu überbrücken, indem jede halbe Sekunde ein Punkt nach Standard-Out geschrieben wird.

```
public class Sanduhr
{
    public static void main (String[] arg)
    {
        Sanduhr su = new Sanduhr ();
        su.dauertLange ();
    }
}
```

```
public void dauertLange ()
{
    System.out.println ("Achtung! Das dauert jetzt ein bisschen.");
    try
    {
        Thread.sleep ((long)(Math.random ()*5000 + 5000));
    }
    catch (InterruptedException e) {}
    System.out.println ("Endlich! Es ist vorbei.");
}
}
```

Aufgabe 2

- Ein berühmtes Synchronisations-Problem ist das Problem der n Philosophen, die an einem runden Tisch sitzen und zyklisch immer wieder denken und speisen. Für jeden Philosophen ist eine Gabel ausgelegt. Um aber speisen zu können, benötigt ein Philosoph auch die Gabel seines rechten Nachbarn. Das bedeutet, dass nicht alle Philosophen gleichzeitig speisen können. (Die Philosophen wechseln nicht ihre Plätze.) Es gilt nun zu verhindern, dass zwei Philosophen dieselbe Gabel greifen. Natürlich darf auch keine Verklemmung auftreten.
- ➔ Lösen Sie dieses Problem mit dem Monitor-Konzept von Java.

Aufgabe 3

- In einer Stadt von n Personen erfindet eine Person einen Witz und erzählt ihn anderen Personen, die ihn nach folgenden Regeln weitererzählen:
 - Die Personen treffen zufällig aufeinander.
 - Wird der Witz einer Person erzählt, die ihn noch nicht kennt, dann erzählt auch sie den Witz weiter.
 - Wird der Witz einer Person erzählt, die ihn schon kennt, dann hören beide auf, den Witz weiterzuerzählen.
- Wieviel Personen erfahren von dem Witz? Lösen Sie das Problem durch Simulation, indem Sie für jede Person ein Objekt anlegen, das als Witz-verbreitender Thread laufen kann.