

ThreadDemo

Threads

Diese Übung soll ihnen zeigen, welche Situationen **Threads** erfordern und wie diese eingesetzt werden können. Im ersten Schritt soll durch den Aufruf der Methode **sleep()** das Programm künstlich verlangsamt werden. Im zweiten Teil fügen wir einen unabhängigen **Thread** ein, auf diese Weise ist das Userinterface nicht mehr blockiert und kann weiterhin Befehle verarbeiten.

Aufgabe 1

Versuchen Sie mit einer leeren **for-Schleife** das Programm so zu verzögern, dass sich der Ball in einer sinnvollen Geschwindigkeit bewegt.

Die Geschwindigkeit des Balls ist jetzt im Wesentlichen von der Rechenleistung des Computers abhängig, was natürlich nicht im Sinne des Erfinders ist.

- Rufen Sie an Stelle der **for-Schleife** die Methode **sleep()** auf. Da diese eine **Exception** auslösen kann, muss sie in einem **try/catch** Block stehen:

```
try {  
    Thread.sleep(50);  
}  
catch (InterruptedException e){  
    System.err.println("Ausnahme bei sleep");  
}
```

Beim Aufruf dieser Methode wird das Programm für 50ms angehalten. Das heisst, dass die Rechenleistung für andere Prozesse zur Verfügung steht.

Aufgabe 2

- Ergänzen Sie ihr Programm nun so, das die Klasse Ball als unabhängiger **Thread** gestartet werden kann:
- Die Klasse Ball soll nun die Klasse **Thread** erweitern.
- Der Code innerhalb der Methode **run()** soll jetzt solange ausgeführt werden wie die boolsche Variable **weitermachen true** ist.
- Schreiben sie eine Methode **stoppen()**, welche die Variable weitermachen auf **false** setzt.

Aufgabe 3

Sorgen Sie nun dafür, dass bis zu 50 Bälle gestartet und wieder gestoppt werden können.