



Informatik Klasse 13, Foliensatz 10

Timer

Prof. G. Kemnitz

Institut für Informatik, Technische Universität Clausthal
21. Januar 2010



Timer

Eine von »Thread« abgeleitete Klasse. Ein Thread ist ein eigenständiger Kontrollfluss, der nebenläufig abgearbeitet wird.

Methoden von Timer:

- Konstruktor mit einer Wartezeit und einer auszuführenden Funktion als Parameter: erzeugt ein Timerobjekt
- start: startet den Thread
- cancel: Stoppt den Thread, falls der Timer noch im Wartezustand ist

```
from threading import Timer
def say_hallo():
    print "hallo"
T=Timer(3.0, say_hallo)
T.start()
```



Experiment: verkettete Timer-Treads

```
from threading import Timer
def say_hallo():
    print "hallo again "
    T=Timer(1.0, say_hallo)
    T.start()
T=Timer(3.0, say_hallo)
T.start()
print "hallo"
```

- Das aufgerufene Unterprogramm startet immer wieder einen neuen Timer; beenden mit ctrl-c



Experiment: Abbruch des aktuellen Timer-Treads

```
from threading import Timer
T = 0
def say_hallo():
    global T
    print "hallo again"
    T=Timer(1.0, say_hallo)
    T.start()
T=Timer(3.0, say_hallo)
T.start()
print "hallo"
raw_input('Press Enter to stop: ')
T.cancel()
```



Experiment: Messen der Zeitgenauigkeit

```
from threading import Timer
from time import time
T = 0
tm = 0
def say_hallo():
    global T, tm
    tmx = time()
    print "hallo again after" ,tmx-tm, 's'
    tm = tmx
    T=Timer(1.0, say_hallo)
    T.start()
T=Timer(3.0, say_hallo)
T.start()
tm = time()
print "hallo"
raw_input('Press Enter to stop: ')
T.cancel()
```