

Schrittweise Ausführung

Möglichkeiten einer
schrittweisen Ausführung
von Anwendungen
zu Demo-Zwecken

Schrittweise Ausführung

Problem

- Enthält die Testanwendung mehrere aufeinander folgende Anweisungen, werden diese alle erst ausgeführt, bevor die Zeichenfläche angezeigt wird.
- Das Bild zeigt also nur den Endzustand.

Schrittweise Ausführung

- Lösung 1: In der OnInit-Methode ein Timerobjekt erzeugen, an die App binden und über eine Methode steuern

```
self.timer = wx.Timer(self)
```

```
self.Bind(wx.EVT_TIMER, self.Drehe, self.timer)
```

```
self.SchrittweiseDrehen()
```

- Die Methoden ...

Schrittweise Ausführung

- Die Methoden sind beispielsweise:

```
def Drehe(self, evt):  
    stuhl.Drehe(10)  
    self.anzahl -= 1  
    if self.anzahl < 1: self.timer.Stop()
```

```
def SchrittweiseDrehen(self):  
    self.anzahl = 36  
    self.timer.Start(100)
```

Schrittweise Ausführung

- Lösung 2: Eine externe Klasse Schritte verwenden (*Kopf und Konstruktor*)

```
class Schritte(wx.Timer) :
    def __init__(self,
                 owner,
                 anzahlSchritte,
                 schrittProzedur) :
        wx.Timer.__init__(self, owner)
        self.anzahl = anzahlSchritte
        self.schrittProzedur=schrittProzedur
```

Schrittweise Ausführung

- Lösung 2: Eine externe Klasse Schritte verwenden

(diese jeweils aufgerufene Methode muss auch dafür sorgen, dass irgendwann abgebrochen wird)

```
def Notify(self):  
    self.schrittProzedur()  
    self.anzahl -= 1  
    if self.anzahl < 0:  
        self.Stop()
```

Schrittweise Ausführung

- Lösung 3: Threads verwenden (ähnliche Konstruktion, geht aber anders)

```
import time
```

```
import _thread
```

- In Testanwendung

```
_thread.start_new_thread(self.Action,  
                          (0.05, 50))
```

Schrittweise Ausführung

- Lösung 3: Threads verwenden
- die aufgerufene Methode beispielsweise

```
def Action(self, sec, anzahl):  
    for i in xrange(anzahl):  
        self.sessel.Drehe(2)  
        time.sleep(sec)
```