

Nebenläufigkeit: die Klasse Thread

Möchte man mehrere Prozesse gleichzeitig laufen lassen, so müssen diese parallel laufen. Dazu müssen alle Klassen, die in eigenen Programmfäden (Threads) nebenläufig agieren sollen, von der Klasse ‚Thread‘ erben.

Gestartet wird ein Thread mit der Methode ‚start()‘. ‚start()‘ ruft dann automatisch die Methode ‚run()‘ der entsprechenden Klasse auf.

Beispiel:

```
class XY(frame, Thread):           #erbt von frame und Thread
    .
    .
    .
    def run(self):
        .
        .
        .
if __name__ == "__main__":
    objektVonXY = XY()
    objektVonXY.start()
```

In der Klasse XY muss es die Methode run() geben.

In der Methode __main__ des Programms wird run() durch start() aufgerufen.

Referenzattribute:

Werden Attribute oder Methoden einer anderen Klasse benötigt, so kann diese als Referenzattribut deklariert werden.

Beispiel: (auch die Klasse XY muss als Thread laufen)

```
class Zeit(Thread):
    def __init__(self,objXY):
        Thread.__init__(self)
        self.view = objXY

    def run(self):
        .
        .
        .
objXY = XY( ...)
z = Zeit(objXY)
```

[siehe Beispielprogramm 'Beispiel_zu_Threads.py'](#)