对于Python中threading.Timer的研究

我发现了一个问题:当计时器已经没有功效的时候,就是不再执行计时功能的时候.计时器的线程还活着吗?因此我编写了一个函数来证实这个问题发现:

|  |
| --- |
| **import** threading,time **def p**():  **print** type(t1) t1 = threading.Timer(5,p) t1.start() **print** t1.getName() **print** t1.isAlive() time.sleep(4) t1.cancel()  **print** t1.getName() **print** t1.isAlive() |

结果是

Thread-1

True

Thread-1

True

是的,这个线程还活着.那么就不能单纯的使用cancel来结束这个线程,这个方法只是取消计时.为了解决这个问题,不要使用cancel来结束计时器而是用exit()于是我想杀死线程,可是python不推荐kill掉线程.那么还是放弃杀掉线程.

时间窗口观察者职责:

每一个自助考勤对象维护一个时间窗口类对象,时间窗口类有一个timer可以初始化时为None,一旦不是None时,就说明这个考勤对象开启了计时,那么需要观察这个考勤对象 ,当他从队列中出去的时候,一定要保证能停止这个考勤对象的计时.如果这个考勤对象压根进入过队首,也没有过计时,那就好办,pass即可.

可能我的代码会出现一些没必要的方法和结论,不过这都是简单工厂模式的原则,各个类各司其职,必要的时候相互帮忙,体现了内聚性的特点.