

```
1  import threading          # For thread events
2  from threads import thrd  # For other threads
3  import sound
4  import time
5
6  class Control:
7
8      def __init__(self, takeFun, checkTakeFun, giveFun, checkGiveFun, takeArgs=None, check
        TakeArgs=None, giveArgs=None, checkGiveArgs=None):
9          self.takenFlag=threading.Event()    # Create event flag to allow input from
        check() thread
10         self.takeFun=takeFun                # Function executed to take control
11         self.checkTakeFun=checkTakeFun      # Condition checked for control taken (Bool
        ean function)
12         self.giveFun=giveFun                # Function executed to give control
13         self.checkGiveFun=checkGiveFun      # Condition checked for control give (Boole
        an function)
14
15         if takeArgs!=None: self.takeArgs=takeArgs
16         if checkTakeArgs!=None: self.checkTakeArgs=checkTakeArgs
17         if giveArgs!=None: self.giveArgs=giveArgs
18         if checkGiveArgs!=None: self.checkGiveArgs=checkGiveArgs
19
20
21     def take(self):
22
23         def takeThrd():
24             while not self.takenFlag.isSet():
25                 try:
26                     self.takeArgs
27                 except:
28                     self.takeFun
29                 else:
30                     self.takeFun(self.takeArgs)
31             self.takenFlag.wait(5) # Lock thread until released via self.taken
        Flag.set() in self.check() method or timed out
32
33         takeClass=thrd(takeThrd)
34         takeClass.name="takeClass"
35         takeClass.start()
36
37
38     def checkTake(self):
39
40         def checkTakeThrd():
41             while not self.takenFlag.isSet():
42                 try:    # Call function either with or without arguments
43                     self.checkTakeArgs
44                 except:
45                     if self.checkTakeFun():
46                         time.sleep(0.1)
47                         sound.beep(1000,1000)
48                         self.takenFlag.set()
49                 else:
50                     if self.checkTakeFun(self.checkTakeArgs):
51                         time.sleep(0.1)
52                         sound.beep(1000,1000)
53                         self.takenFlag.set()
54
55         checkTakeClass=thrd(checkTakeThrd)
56         checkTakeClass.name="checkTakeClass"
57         checkTakeClass.start()
58
```

```
59
60     def give(self):
61
62         def giveThrd():
63             while self.takenFlag.isSet():
64                 try:
65                     self.giveArgs
66                 except:
67                     self.giveFun
68                 else:
69                     self.giveFun(self.giveArgs)
70             self.takenFlag.wait(0.02) # Lock thread until released via self.takenFlag.set() by self.check() method or timed out
71
72         giveClass=thrd(giveThrd)
73         giveClass.name="giveClass"
74         giveClass.start()
75
76
77     def checkGive(self):
78
79         def checkGiveThrd():
80             while self.takenFlag.isSet():
81                 try: # Call function either with or without arguments
82                     self.checkGiveArgs
83                 except:
84                     if self.checkGiveFun():
85                         self.takenFlag.clear()
86                 else:
87                     if self.checkGiveFun(self.checkGiveArgs):
88                         self.takenFlag.clear()
89
90         checkGiveClass=thrd(checkGiveThrd)
91         checkGiveClass.name="checkGiveClass"
92         checkGiveClass.start()
93
```