doubleForklift.py

```
1
     from threading import Thread
 2
 3
     from pybricks.tools import wait
 4
 5
 6
     # Class contains two forklifts
 7
     class DoubleForklift:
 8
             init (self, config, xlift, ylift):
9
             self.config = config
10
             self.xlift = xlift
11
             self.ylift = ylift
12
13
         def getPos(self):
14
             return [self.xlift.getPos(), self.ylift.getPos()]
15
16
         def stop(self):
17
             self.xlift.stop()
18
             self.ylift.stop()
19
20
         def initPos(self, xpos=0, ypos=None, Wait=3000):
21
             if self.config.state.getState() == 3:
22
                 return
23
24
             if ypos == None:
25
                 ypos = -self.ylift.RACKLENGTH/2
26
             Thread(target=self.xlift.initPos, args=(xpos)).start() # type: ignore
27
             Thread(target=self.ylift.initPos, args=(ypos)).start()
28
             wait (50)
29
             while self.done() == False and Wait > 0:
30
                 wait (50)
31
32
         def moveTo(self, x, y, x speed=400, y speed=400, Wait=10000):
33
             if self.config.state.getState() == 3:
34
                 return
35
36
             Thread(target=self.xlift.moveTo, args=(
37
                 x, x speed), kwargs={"wait": Wait}).start()
38
             Thread(target=self.ylift.moveTo, args=(
39
                 y, y speed), kwargs={"wait": Wait}).start()
40
             while Wait and not self.done() and self.config.state.getState() != 3:
41
                 wait (50)
42
43
         def move(self, delta y, delta x, x speed=400, y speed=400, Wait=10000):
44
             if self.config.state.getState() == 3:
45
                 return
46
47
             self.xlift.move(delta x, x speed, wait=0)
             self.ylift.move(delta y, y speed, wait=Wait)
48
             while Wait and not self.done() and self.config.state.getState() != 3:
49
50
                 wait (50)
51
52
         def done(self):
53
             return self.xlift.done() and self.ylift.done()
54
55
         def stalled(self):
56
             return self.xlift.stalled() and self.ylift.stalled()
57
58
         def printRange(self):
59
             print("xlift: min {}, max {}".format(self.xlift.getRange()[0],
60
                                                    self.xlift.getRange()[1]))
61
             print("ylift: min {}, max {}".format(self.ylift.getRange()[0],
62
                                                    self.ylift.getRange()[1]))
63
```