```python
from threading import Thread

from pybricks.tools import wait


# Class contains two forklifts
class DoubleForklift:
    def __init__(self, config, xlift, ylift):
        self.config = config
        self.xlift = xlift
        self.ylift = ylift

    def getPos(self):
        return [self.xlift.getPos(), self.ylift.getPos()]

    def stop(self):
        self.xlift.stop()
        self.ylift.stop()

    def initPos(self, xpos=0, ypos=None, Wait=3000):
        if self.config.state.getState() == 3:
            return

        if ypos == None:
            ypos = -self.ylift.RACKLENGTH/2
        Thread(target=self.xlift.initPos, args=(xpos)).start()  # type: ignore
        Thread(target=self.ylift.initPos, args=(ypos)).start()
        wait(50)
        while self.done() == False and Wait > 0:
            wait(50)

    def moveTo(self, x, y, x_speed=400, y_speed=400, Wait=10000):
        if self.config.state.getState() == 3:
            return

        Thread(target=self.xlift.moveTo, args=(
            x, x_speed), kwargs={"wait": Wait}).start()
        Thread(target=self.ylift.moveTo, args=(
            y, y_speed), kwargs={"wait": Wait}).start()
        while Wait and not self.done() and self.config.state.getState() != 3:
            wait(50)

    def move(self, delta_y, delta_x, x_speed=400, y_speed=400, Wait=10000):
        if self.config.state.getState() == 3:
            return

        self.xlift.move(delta_x, x_speed, wait=0)
        self.ylift.move(delta_y, y_speed, wait=Wait)
        while Wait and not self.done() and self.config.state.getState() != 3:
            wait(50)

    def done(self):
        return self.xlift.done() and self.ylift.done()

    def stalled(self):
        return self.xlift.stalled() and self.ylift.stalled()

    def printRange(self):
        print("xlift: min {}, max {}".format(self.xlift.getRange()[0],
                                             self.xlift.getRange()[1]))
        print("ylift: min {}, max {}".format(self.ylift.getRange()[0],
                                             self.ylift.getRange()[1]))

```