1. 三部电梯的楼层都是20

2. 电梯运行一层时间3s，开关门时间6s

3. 程序开始/重置电梯处于1层

4. 电梯在运动过程中不能突然改变运动方法

5. 单部电梯要满足捎带需求

6. 三部电梯要满足运动量计算需求

7. ER指令不能调度，FR指令考虑调度

8. 可以捎带的前提是非同质请求

9. (FR,floor,UP/DOWN) (ER,#Elevator,floor)

10. 请求时间从系统中获得(系统时间的获取)

11. 第一条请求发生的时刻作为系统开始运行的时刻，参考这个时候的系统时间

12. 不管如何多线程电梯，可以首先将信息提取出来进行正则处理

13. 多线程电梯：首先选择能够捎带的（如果多部选择运动量最小的），如果没有捎带的但可以相应的（选择运动量最小的），如果没有相应的则等待一段时间

14. 当请求队列为空并且所有的请求都被处理之后，用户输入END结束程序（包括所有的线程）

生产者：不定时输入的请求(extends thread)

盘子：当前请求队列(public synchronized static void put/get()) (full属性：队列是否为空)

消费者：调度器使用（电梯（X3））(extends thread)

输入：

(FR, 1, UP)  
(FR, 3, UP)  
(FR, 20, DOWN)  
(ER, #1, 15)  
(ER, #2, 5); (FR, 10, UP); (FR, 15, DOWN); (ER, #3, 1); (ER, #2,  
20); (FR, 15, UP); (ER, #1, 16); (ER, #2, 13); (FR, 9, DOWN); (ER,  
#1, 7)  
(ER, #3, 7); (FR,7, DOWN)  
END

detail:

