设计一个大楼的停车场系统，在大楼的入口处有一个自动售票以及显示等待时间的仪表（Meter）。该仪表能够显示下一个空车位的最少等待时间。

例如，一个停车场有三个停车位分别为A，B，C。初始状态为空车位（Available）。

假设场景

9：00 AM

一号汽车驶入，输入预计离开时间 11：00 AM. 系统查询车位，发现三个车位有空位，随机分发A号车位给一号汽车。一号汽车停入A号车位。

9:30 AM

二号汽车驶入，输入预计离开时间12：00PM.系统查询车位，发现有两个车位有空位，随机分发B号车位给二号汽车。二号汽车停入B号车位。

10：00AM

三号汽车驶入，输入预计离开时间1：00PM。系统查询车位，发现仅剩一个空车位C号，随分配C号车位给三号汽车，三号汽车停入C号车位。

10：30AM

四号汽车驶入，输入预计离开时间2：00PM。系统查询车位，发现所有车位已满。同时发现A号车位的一号汽车即将于11：00AM离开，系统随机返回给四号汽车，您的最少等待时间为30分钟（11：00 – 10:30 = 30）.

可自定义语言，设计一个停车系统模拟上述情景。

假设：

1. 可认为每个汽车按照输入的离开时间按时离开。不存在提前开走，或者延后的情况。
2. 如何有空位，可认为等待时间为0.

要求：

1. 系统数据结构清晰，流程合理。
2. 自定义每个需要的接口。
3. 对每个方法解析时间复杂度。
4. 写出main函数，描述如何调用该系统。

附加问题：

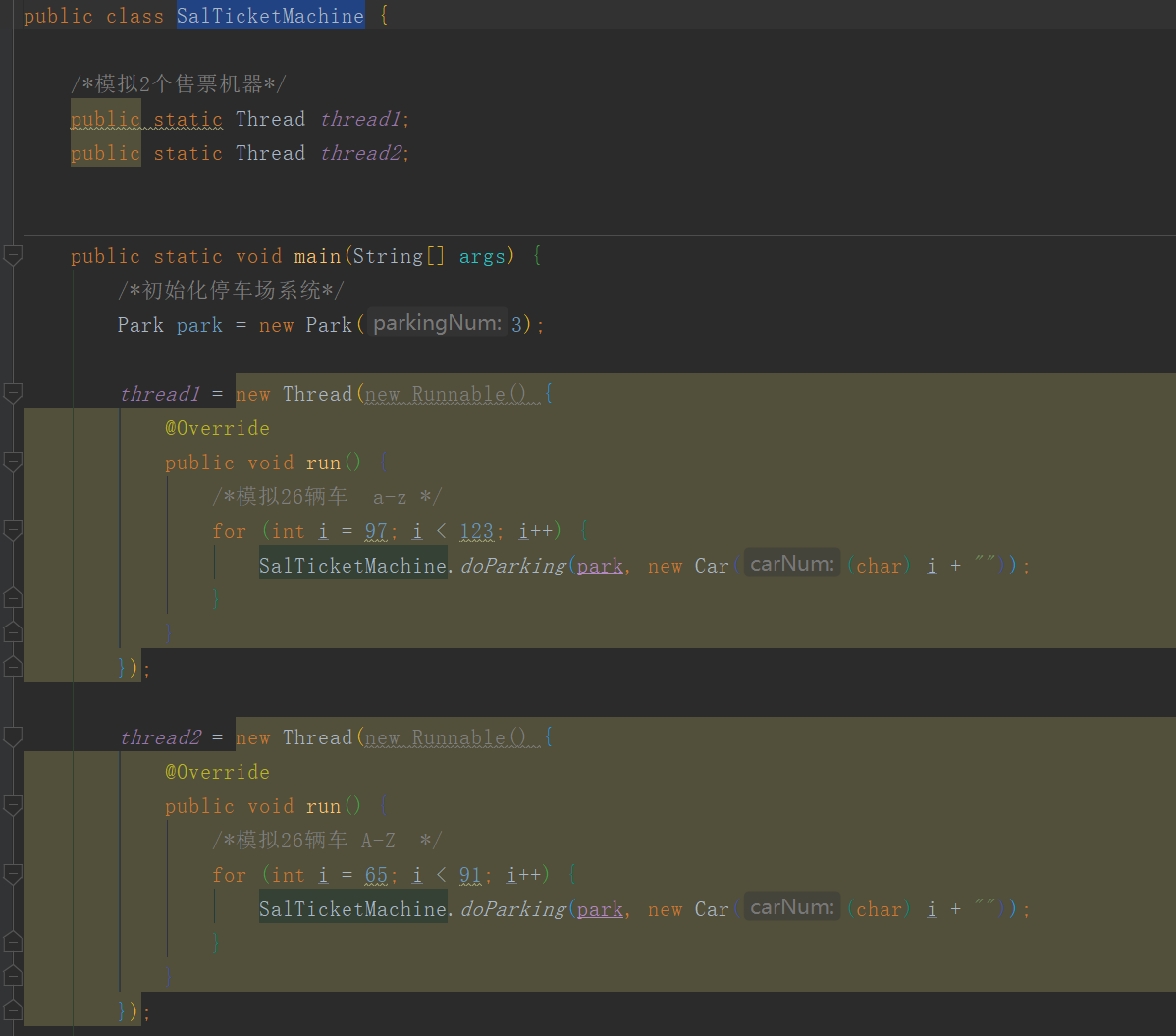
如果存在汽车提前驶离的情况（但不存在延后的情形），该如何调整你现有的系统？

# 1、类说明

* Car:车辆类
* Park:停车场主系统，提供停车所需服务；
* ParkingSpace:车位信息
* SalTicketMachine:模拟售票系统及车辆实际进入情况

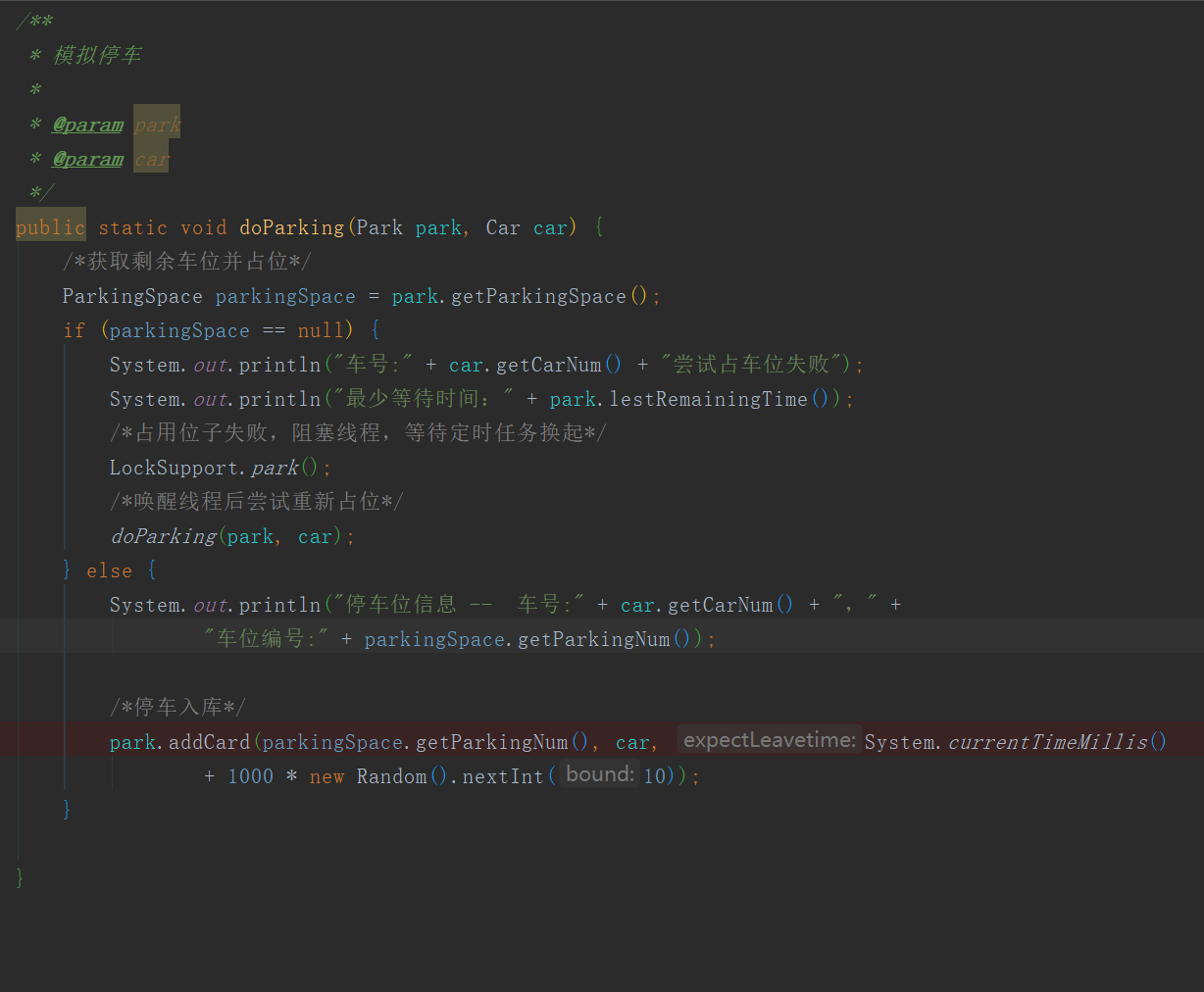


# 2、模拟说明



**说明**：

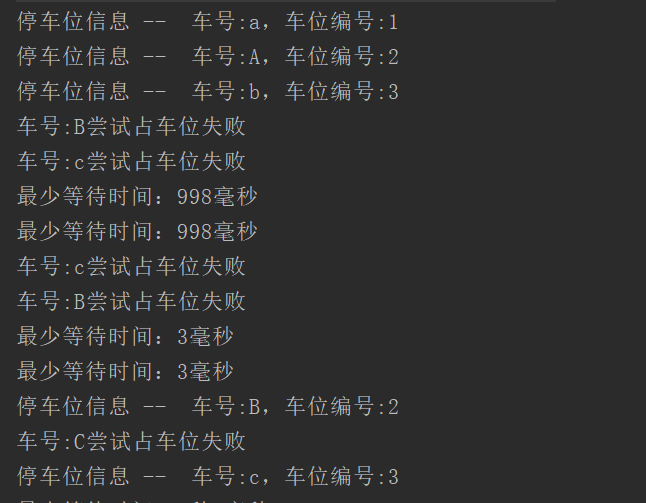
* + thread1和thread2模拟2个自动停车售票口。
    - 售票口1（Thread1）: 车牌号为 “a-z” 一共26两车入库；
    - 售票口2（Thread2）: 车牌号为 “A-Z” 一共26两车入库；
  + 2个自动售票机公用一个停车库，初始化的同时设置车库有3个位子

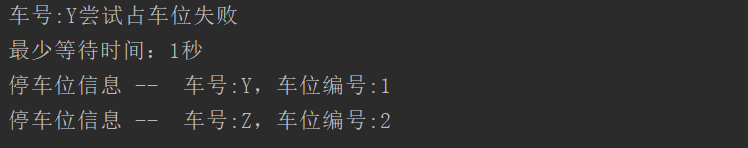


**说明**：

* + 2个售票口尝试占位，这里占用座位接口使用同步锁
    - 占位成功，则将车辆入库
    - 占位失败，则阻塞当前线程（模拟车辆在自动售票扣等待的过程）
    - 定时器每5秒执行一次将到达预定离开时间的车辆移除（模拟不存在延期情况），并将车位空出。同时定时器每次执行都会唤醒上一步占位失败线程，尝试再次占座。
  + 停车入库的时候随机0-10秒的预期驶离时间。

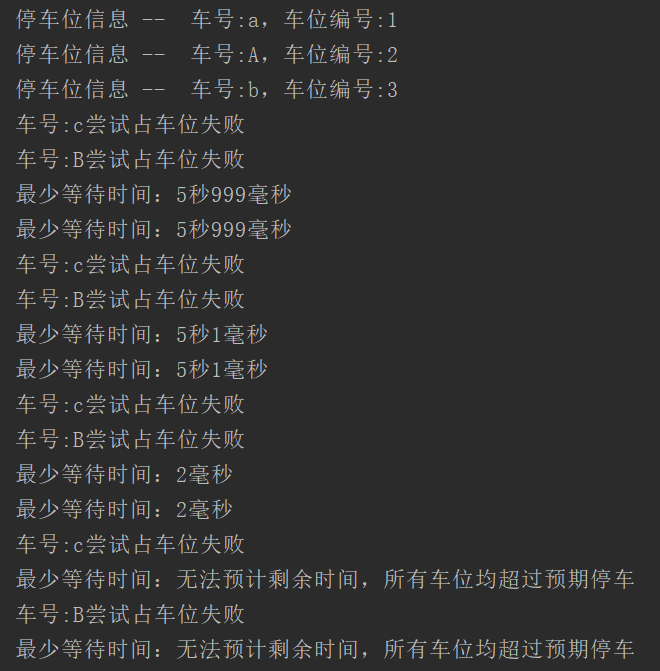
# 3、不存在延后的情形





* **结果**：模拟2个售票处，车牌a-z的汽车队列 和 车牌 A-Z队列，所有车辆均有序排队停车

# 4、存在延后的情形结果



* **结果**：模拟2个售票处，由于车辆延期，无法预估已停车辆离开时间，所以后续车辆均无法停车。如果允许延期车辆驶离，则后续可提供驶离接口，更新对应车位占用为空，则可继续完成新车入库停车。