

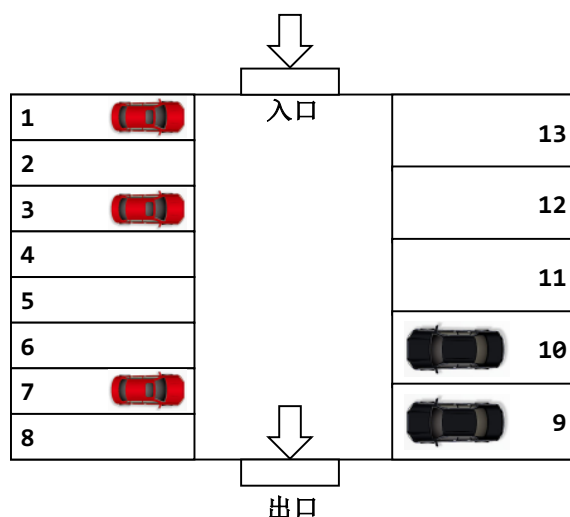
习题课讲解参考

1 给出一个 Java ADT 题目

某公司拟设计和开发一个停车场管理系统，其基本需求陈述如下：

- (1) 一个停车场有 n 个车位($n \geq 5$)，不同停车场包含的车位数目不同。
- (2) 一辆车进入停车场，如果该停车场有空车位且其宽度足以容纳车的宽度，则可以在此停车。支持自动分配车位和指定停车位两种分配方式。
- (3) 停在停车场里的车，随时可以驶离停车场。
- (4) 停车场管理员可以随时查看停车场的当前占用情况。
- (5) 计算出本次停车的费用（半小时 10 元，不足半小时按半小时计算）。

下图给出了一个包含 13 个停车位的小型停车场示例图，其中 1-8 号停车位较窄，9-13 号停车位较宽。在当前状态下，第 1、3、7、9、10 号车位被占用，其他车位空闲。



2 设计 ParkingField 接口，撰写 spec

- `void parking(String plateNo, int width, int lotNo) throws LotOccupiedException, NoSuchLotException, LotTooNarrowException, CarAlreadyParkingException;` 指定停车位
- `void parking(String plateNo, int width) throws ParkingFieldFullException, CarAlreadyParkingException;` 自动分配停车位
- `Record depart(String plateNo) throws CarNotInFieldException;` 驶离停车场
- `Map<Integer, String> status();` 获得停车场状态
- `Set<Record> records();` 获取停车记录
- `String getPFName();` 返回停车场的名字
- `int getNumberOfLots();` 获得停车场中总的车位数目

- `boolean isLotInParkingField(int lotNo,int width);` 查看停车场中是否存在指定车位号和宽度的车位
- `boolean isFull ();` 当前停车场是否已满
- `int getOneFreeLot(int width) throws ParkingFieldFullException;` 获得一个宽度大于指定宽度的停车位
- `Set<Integer> getLots();` 返回停车场停车位的编号列表
- `int getLotWidth(int lotNo)` 返回某车位宽度

3 为 ParkingField 接口增加静态工厂方法

创建停车场

- `static ParkingField create(String name, int[] lotnos, int[] width);`
此部分有代码

4 针对 ParkingField 接口设计测试用例

详见代码 ParkingFieldTest 类

5 编写辅助类 Car、Lot、Record

详见 Car Lot Record 三个类的代码实现

Car 和 Lot 是 Immutable 类，Record 是 mutable 类
设计 rep，是否有多种表示？

设计 RI，与 Rep 密切相关

实现 checkRep()

设计方法

override equals()、hashCode()、toString()

6 编写具体实现类 ConcreteParkingField

6.1 讲解 Rep、RI、AF 的设计

设计 rep

设计 AF、RI

实现 checkRep()

6.2 接口中各方法的实现

实现构造函数

实现各个方法

几个 ConcreteParkingField 中的辅助方法

6.3 equals()、hashCode()、toString()

ConcreteParkingField 是 mutable 的，所以不需要重写 equals()和 hashCode()

重写 toString ()

- (1) 停车场的名字和停车位数目
- (2) 停车场车位的占用比例
- (3) 每个停车位的编号、宽度、状态

例子：某 ConcreteParkingField 对象的名字是“A1”，它有 5 个停车位，状态如下表所示。

停车位编号	停车位宽度	当前所停车辆
1	200	车牌号：AB001，宽度 180
2	180	空闲
3	200	车牌号：CD002，宽度 180
4	170	车牌号：EF003，宽度 160
5	190	空闲

从上表看出，该停车场目前有 60%的停车位已停车 (=3/5)。针对该例子，你所写的 toString()的输出结果应为：

```
Parking field A1 has total number of lots: 5
Now 60% lots are occupied
Lot 1 (200):   Car AB001
Lot 2 (180):   Free
Lot 3 (200):   Car CD002
Lot 4 (170):   Car EF003
Lot 5 (190):   Free
```

为了实现 Lot 排序，实现了一个比较器

7 编写客户端程序

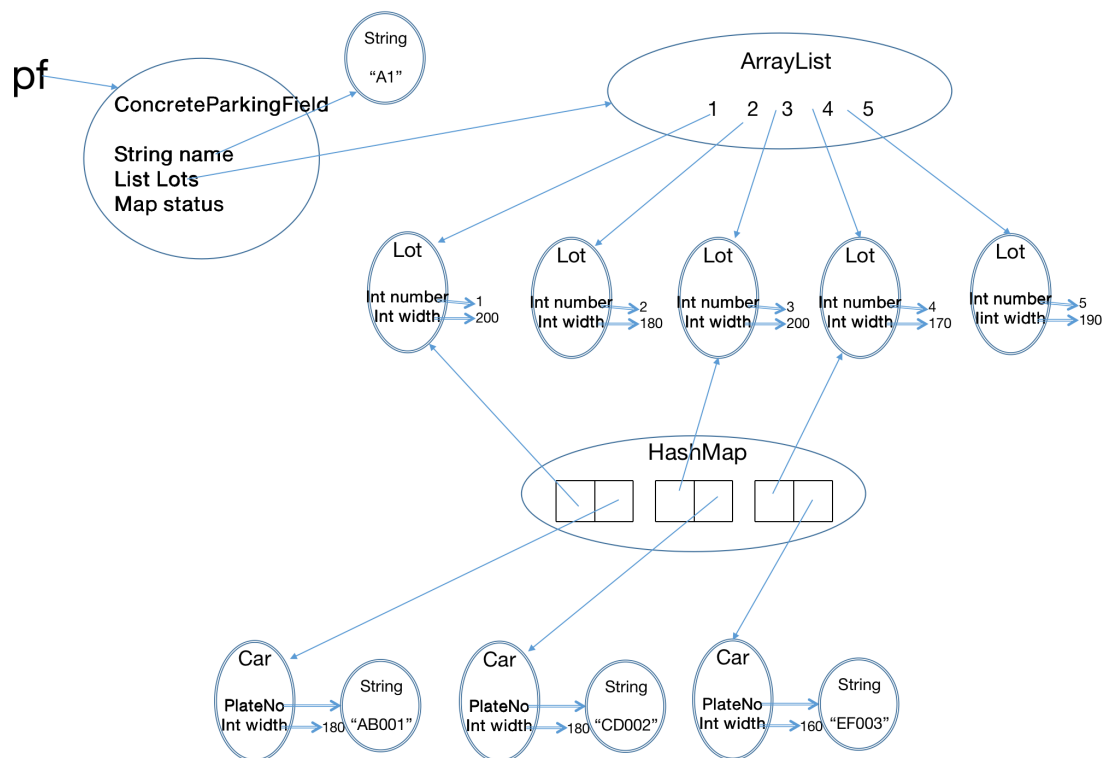
见 Client，运行后按照提示操作即可。

某个停车场管理系统，实现一系列功能

8 绘制 Snapshot Diagram

针对客户端程序运行到某个时刻，想象内存状态，绘制 snapshot

针对 6.3 中 toString 的例子，状态图如下（忽略了 Record）



9 表示泄露与安全性

考察 ParkingField、Car、Lot、Record 这些 ADT，考虑以下的客户端代码，判断这些 ADT 是否存在表示泄露？请逐项列出可能存在表示泄露的地方，分析其潜在风险，并给出其修改策略。

给出几个表示泄露的客户端例子，以及如何通过防御式拷贝策略修改 ADT 设计。

表示泄露的例子：

例子 1：

```
public Set<Integer> getLots(){ }
```

例子 2：

Status()方法因为返回的是 Map<Integer, String>，是新建的 Map 返回，所以不存在泄露(防御式拷贝)。

例子 3：

```
public Set<Record> records(){} 
```

这里存在 2 个问题 1.set 是可变的 2.Record 也是可变的(需要修改离开时间), 有 mutator, 需要创建副本返回。 我实现的方式是重写了 Record 的 clone()方法, 利用此方法, 采用防御式 copy 的方法, 返回 Record 的副本。

10 执行 JUnit 测试

以及查看测试覆盖度 Eclemma