电梯设计：然后一半时间设计电梯，一层楼多个电梯，核心API是输入一个target楼层返回最近的电梯，楼主花了很多时间设计fields，然后最后正要写方法，三姐说不用了，直接拍照，说知道我会写，然后会记到feedback里，估计最后肯定没记。

Users: me, maintenance,

Elevator: go up, go down, idle, loading, un-conditioning – beep

Max floor, min floor

Building

<http://www.angelfire.com/trek/software/elevator.html>

<http://wdxtub.com/interview/14520604445920.html>

<https://hellohell.gitbooks.io/java-/content/dian_ti_she_ji.html>

Command pattern

<https://hellohell.gitbooks.io/java-/content/li_zi.html>



Email System

Book Recommendation

**Design a parking lot**:

Clarify the problem:

system design? Class hierarchy?

How you approach the problem:

how is the parking lot designed? Accessibility? Free? How many spots?

Multiple levels? Entrance, concurrency? Full-automatic parking lot? What’s the price policy? Premium parking lots?

Design the system in 4 sizes – small, medium, large, X-large

Abstract Vehicle

* License Plate: String
* Color: Enum

Car, Motor Cycle, Bus, Truck: Vehicle

M S XL L

Class Parking Lot

* Zip code: string
* Place Vehicle (vehicle: Vehicle) -> Spot

Class Spot

* Id: long
* Size: Enum

Coding part: what’s the goal?

4 stacks: place vehicle + put in HashMap O (1)

Spot: remove vehicle (vehicle: Vehicle)

Testing spot:

OOD其实就是你把要写的实体类的属性定义好，然后就是根据business logic写方法

飞机场调度设计

怎麼設計餐廳的點餐系統

餐馆预定座位，设计了customer，restaurant， table类，在面试官提示下补上了reservation类（最关键的类没第一时间说出），用的command pattern

OOD: 给你drawPoly(points), 设计Canvas class 可以画出rectangle and square

Polygon super class, Rect and Sq extend Polygon, in Canvas class has draw method, a loop to call drawPoly(points).

设计共享单车

design a set of cards我记得5是说 设计一套牌，可以发牌，参与人可以拿到牌，目前不清楚这副牌有什么功能，但是设计时要尽量为将来的使用考虑。

design warehouse

让设计一个测试工具 来模拟客户行为 测试 APP 在不同的 设备上行为。. 1point 3acres 璁哄潧  
     我就说可以用类似按键精灵的方法来测试，有一个基本的坐标行为，然后根据不同的设备，可以通过对应的坐标转换在不同的设备上进行测试。反正是画了一白板。