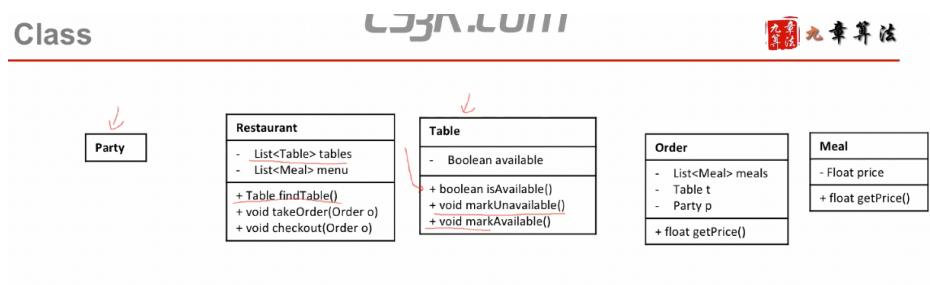


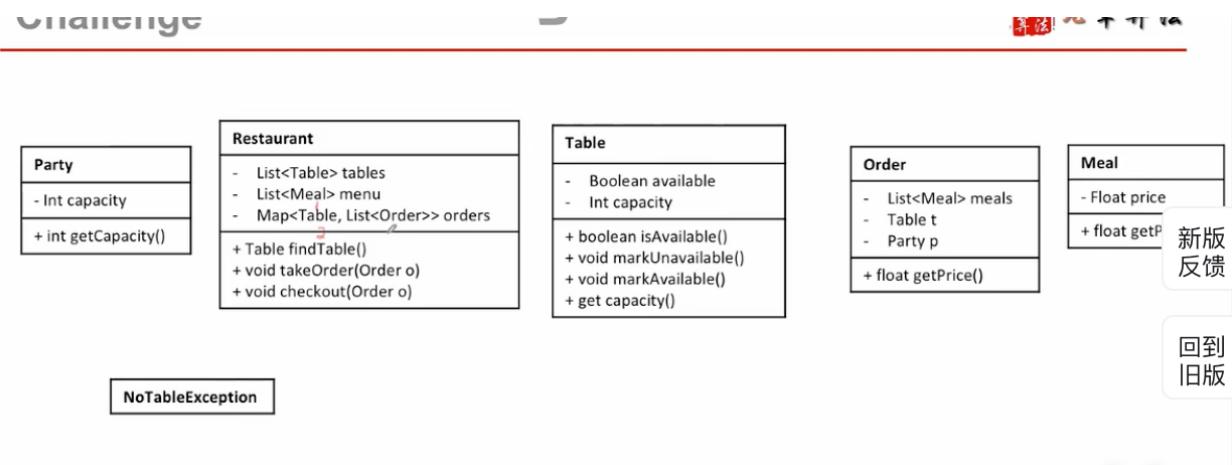
# Lecture 3 - 预定类: Hotel & Airline Ticket

## 餐馆 OOD

### 初步设计



如何修改这个可以让他支持拼桌



用 map 来对应 table 和 list of orders, map 可以找到每个 table 对应有哪些 order (这个暂时不用, 因为复杂一些)

或者直接在 table 里加 capacity 这个变量，每次当有人坐就减去相应的人，如果减完到 0 了那就把 available 变成 false

## Clarify

- What
  - 管理: Party -> Restaurant -> Table
- How
  - 能否预约
  - 能否外送
- Who - Optional
  - 思考模式 1: 过于复杂
    - Party 进入餐馆 -> Host 指引到空桌 (find table) -> 一个 waiter 负责这桌客人 (assign waiter) -> 客人点菜 (take order) -> Chief 拿到 order, 按顺序做菜 (cook by order) -> Order 做好后, waiter 上菜 (serve order) -> 客人吃完后付钱 (check out)
  - 思考模式 2
    - 客人进入餐馆, 餐馆返回一个 Table
    - 客人点菜, 餐馆返回一桌菜
    - 客人付账, 餐馆清空 Table

## Use Case & Classes

- Restaruant
  - Find Table (serve)
  - Take Order (serve)
  - Checkout (checkout)

## 预定类 OOD

- 题库
  - Restaurant reservation system
  - Hotel reservation system
  - Flight/Bus/Train reservation system
- 解题思路
  - What: 考虑预定的东西
  - Use Case = Reserve
    - Search
 

```
Criteria -> search() -> List<Result> -> select() -> Receipt
```

      - Reservation findTableForReservation(Time slot) throws NoTableForReservationException
      - void confirmReservation(Reservation reservation)
    - select
    - cancel

关于 `search critia` 可以问的问题

- Search criteria

Make a reservation

- 人数: 无拼桌, 每张桌子大小相同, 不会有超过桌子大小的人数
- 日期: 是否允许预定多日以后的? - 允许
- 时间: 是否所有时间都允许预定? - 24/7
- Design: `FindTableForReservation(Timeslot t)`
- `Timeslot contains Date and time`

- 做法一:

Make a reservation

Make a reservation

- List<Result>

Result == Timeslot

- Design: List<Timeslot> findTableForReservation(Timeslot t)
- Possible Challenge: 跟面试官讨论如何获得这个List

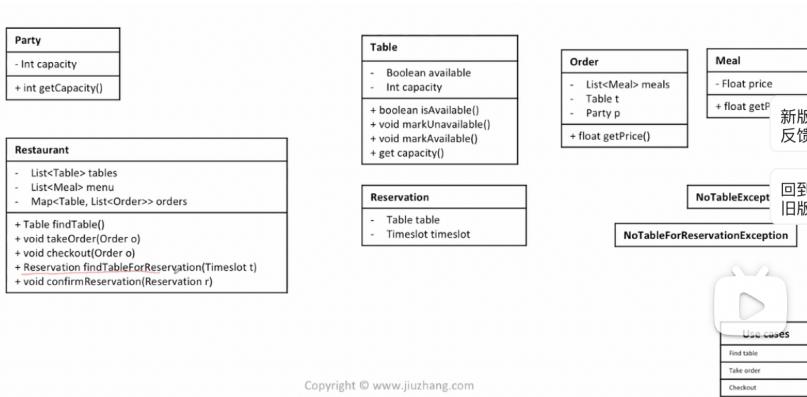
## Reservation

- 做法二：
- 可以预定：直接进入Confirm阶段
- 不能预定：Throw exception / Show message

- Design: Pair<Table, Timeslot> findTableForReservation(Time slot)  
throws NoTableForReservationException
- Design: void confirmReservation(Pair<Table, Timeslot> reservation)

- 为什么我们可以跳过List<Result>这个步骤？

因为Table是一样的，用户不用选择也不会知道是订1号桌还是2号桌



还需要的改进？如何判断 table 是否可预定

Any problems?

CS&C

九章算法

- How to know if a table is open for reservation for a timeslot?

```
if (Timeslot <= MAX_DINETIME > CURRENT_TIME)
{
    if no reservation during Timeslot - MAX_DINETIME to Timeslot + MAX_DINETIME
        return true;
    else
        return false;
}
else
{
    return table.isAvailable();
}
```

Copyright © www.jiuzhang.com

Visit www.cs&c.com to get the latest videos

判断它的前后时间段是否有人预定

- 如何知道对于一张Table:

Any reservation during Timeslot - MAX\_DINETIME to Timeslot + MAX\_DINETIME ?

Solution 1:

每张Table按时间排序保留一个List, 对于每一个进来query的timeslot, 都检查是否有符合上述区间的timeslot已经被预定了

新  
旧



Solution 2:

保存一个Centralized的Map, 有被confirm过的timeslot就插入这个Map.

Map<Timeslot, Set<Table>>

当query一个新的timeslot的时候, 检查这个区间内已经被预定过的Table, 排除之后剩下的就是可选的桌子, 随意选一张即可



- 分析各自的优缺点:

**Solution 1: Reservation**的信息集中在Table里, 不方便统一管理

e.g. 清理今日的所有reservation

**Solution 2:** 如果Timeslot可以分得很细 (每五分钟一个timeslot), MAX\_DINETIME假如为2小时, 那么需要查24个entry

Party
- Int capacity
+ int getCapacity()

Table
- Boolean available
- Int capacity
+ boolean isAvailable()
+ void markUnavailable()
+ void markAvailable()
+ getCapacity()

Order
- List<Meal> meals
- Table t
- Party p
+ float getPrice()

Meal
- Float price
+ float getPrice()

新版  
反馈

NoTableExcept  
回到  
旧版

NoTableForReservationException



Use cases
Find table
Take order
Checkout

Copyright © www.jiuzhang.com

Hotel reservation system

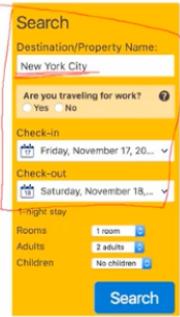
## Hotel Reservation

- Can you design a hotel reservation system
- What
  - 是为一间酒店设计预定房间系统，还是先选择酒店的系统？
  - 搜索条件区别：人数+时间 VS 人数+时间+地址
  - Search criteria -> Search() -> List -> Select() -> Receipt

Hotel reservation system 

---

- Can you design a hotel reservation system?



Welcome to New York City, the Big Apple  
433 properties found – including 112 value deals!

3 Reasons to Visit: Central Park, Metropolitan Museum, Art, Fine Art Museums

Our Top Picks: Lowest Price First, Value for Money

Recommended for You:

MODERN Seek design, contemporary comfort.

NoMo SoHo   
SoHo, New York City, NY – Show on map  
Subway Access  
45 people are looking right now  
Booked 41 times in the last 24 hours  
Great Value Today    
Double Room    
In high demand!  
Risk Free: You can cancel later, so lock in

VS.



漫趣乐园-如家上海浦东机场店   
★★★★★ 5.0 分 28 点评

 上海市浦东新区川沙路4518号 

 入住10月10日 / 离店 10月11日 

小魔仙大床房A  

会员价 > 398元起 

超级飞侠大床房A  

会员价 > 398元起 

Copyright © www.jiuzhang.com

左边是给出不同 hotel，右边是一个 specific hotel

- What

是为一间酒店设计预定房间系统，还是先选择酒店的系统？

- 针对本题：

先设计一间酒店，再设计选择酒店的系统

- What

Search criteria -> Search() -> **List<Result>** -> Select() -> **Receipt**

除了考虑题目中的名词之外，还需要从上述的三处考虑，**What**类型的提问  
主要针对**List<Result>**

新  
反

回  
旧

如何设计房间类？

Room
- int capacity
- float price

**List<Result>** -> **List<Room>**

Room_1	Room_2	Room_3	Room_4	Room_5	Room_6
Capacity : 2	Capacity : 2	Capacity : 2	Capacity : 2	Capacity : 1	Capacity : 128
Price: 198	Price: 128				

Room Type	Single	Double
Single Room	 This room includes 1 single bed, a desk, a chair, a TV and a work area, a private bathroom, free WiFi, a sofa, a free television, a telephone, and a closet. <b>Price: 198</b>	 This room includes 2 single beds, a refrigerator, and a double bed. <b>Price: 300</b>
Double Room	 This room includes 1 single bed, a desk, a chair, a TV and a work area, a private bathroom, free WiFi, a sofa, a free television, a telephone, and a closet. <b>Price: 198</b>	 This room includes 2 single beds, a refrigerator, and a double bed. <b>Price: 300</b>
Single Room	 This room includes 1 single bed, a desk, a chair, a TV and a work area, a private bathroom, free WiFi, a sofa, a free television, a telephone, and a closet. <b>Price: 198</b>	 This room includes 2 single beds, a refrigerator, and a double bed. <b>Price: 300</b>
Double Room	 This room includes 1 single bed, a desk, a chair, a TV and a work area, a private bathroom, free WiFi, a sofa, a free television, a telephone, and a closet. <b>Price: 198</b>	 This room includes 2 single beds, a refrigerator, and a double bed. <b>Price: 300</b>

实际上我们应该返回的是每个 roomtype，然后给你选择每个 roomtype 你要定几间

How many room types do we have for this hotel?

针对本题：两种（Single room 和 Double room）

## Challenge

Restaurant VS. Hotel

什么情况下需要考虑Result Type (Room Type)?

价格不同时

单人间 VS. 双人间

头等舱 VS. 经济舱

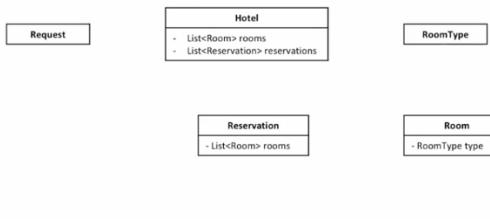
大桌 VS. 小桌



## Core object

Restaurant.LUUU

九章算



## Use cases

CS3K.

Hotel:

- Search for available rooms
- Make reservation
- Cancel reservation

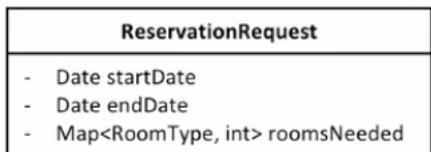
## Search for available rooms

- 1: Based on search criteria
- 2: Go through rooms to check availability
- 3: list available room types and available count

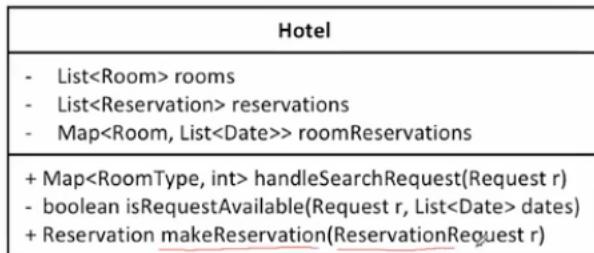


## Make reservation

- 1: Add RoomType and number of rooms in a request
- 2: Send request to Hotel
- 3: If there is enough room left, confirm the reservation
- 4: If there isn't enough room left, throw exception



1.



2.

3.

### NotEnoughRoomForReservationException

4.

#### Search for available rooms

- 1: Based on search criteria
- 2: Go through rooms to check availability
- 3: list available room types and room count

#### Make reservation

- 1: Add RoomType and number of rooms in a request
- 2: Send request to Hotel
- 3: If there is enough room left, confirm the reservation
- 4: If there isn't enough room left, throw exception

```
Map<RoomType, List<Room>> map = new HashMap<>();

for(Entry<Room, List<Date>> entry : roomReservations.entrySet())
{
    Room r = entry.getKey();
    List<Date> roomBooked = entry.getValue();

    if(isRequestAvailable(roomBooked))
    {
        if(map.containsKey(r.getRoomType()))
        {
            List<Room> roomList = map.get(r.getRoomType());
            roomList.add(r);
            map.put(r.getRoomType(), roomList);
        }
        else
        {
            List<Room> roomList = new ArrayList<>();
            roomList.add(r);
            map.put(r.getRoomType(), roomList);
        }
    }
}
```

## Challenge

L3K.COM

- Map<RoomType, List<Room>> map
  - Go through rooms to check availability
  - If there is enough room left, confirm the reservation
  - If there isn't enough room left, throw exception

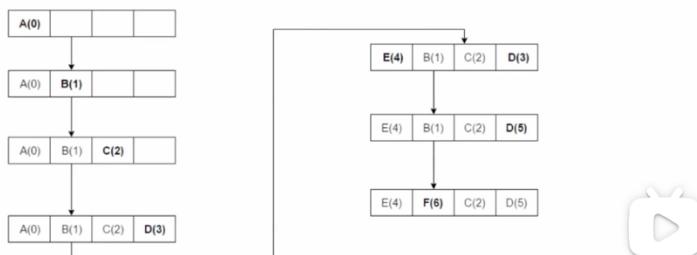
## Challenge

L3K.LU!!!

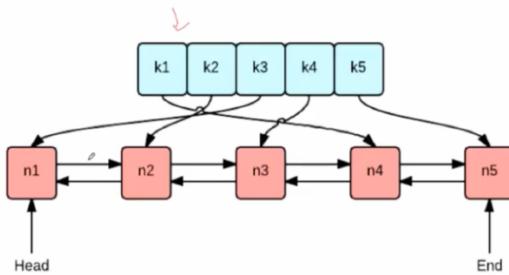
九章算法

- LRU Cache

The access sequence for the below example is A B C D E D F.



- LRU Cache

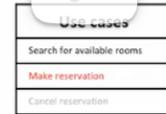
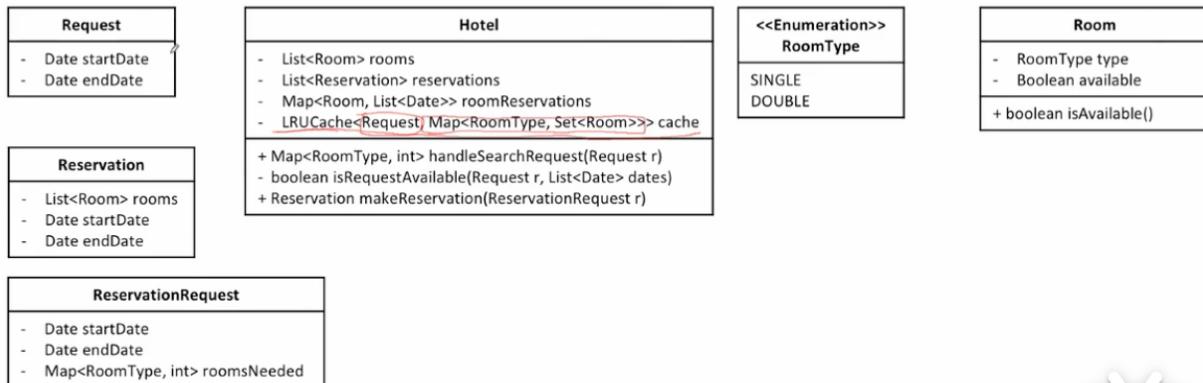


或者直接用 java 提供的 LinkedHashMap 来实现，只需要提供他的 capacity 就行

```
class LRUCache extends LinkedHashMap<Request, Map<RoomType, List<Room>>>
{
    private int capacity;
    public LRUCache(int capacity)
    {
        super(capacity);
        this.capacity = capacity;
    }
    @Override
    protected boolean removeEldestEntry(Map.Entry<Request, Map<RoomType, List<Room>>> eldest){
        return size() > this.capacity;
    }
}
```



# Classes



Copyright © www.jiuzhang.com