预定类面向对象设计

文泰来 老师

扫描二维码关注微信/微博
获取最新面试题及权威解答

微信: ninechapter
知乎专栏： http://zhuanlan.zhihu.com/jiuzhang
微博: http://www.weibo.com/ninechapter
官网: www.jiuzhang.com

Copyright © www.jiuzhang.com

Restaurant

- Can you design a restaurant?

Copyright © www.jiuzhang.com

Clarify

- What
- How
- Who

Copyright © www.jiuzhang.com

Clarify

- What

关键字：Restaurant

Copyright © www.jiuzhang.com

Clarify

- What

关键字：Restaurant

管理：？

Copyright © www.jiuzhang.com

Clarify

- What

关键字：Restaurant

管理：？

Party / Table

Copyright © www.jiuzhang.com

Clarify

- What

关键字：Restaurant

管理：

Party / Table

Copyright © www.jiuzhang.com

Clarify

- What

关键字：Restaurant

管理：

Party / Table

Copyright © www.jiuzhang.com

Clarify

- What

关键字：Restaurant, Party, Table

Copyright © www.jiuzhang.com

What

关键字：Restaurant

属性：？

Copyright © www.jiuzhang.com

What

关键字：Restaurant

Copyright © www.jiuzhang.com

What

关键字：Restaurant

- 吧台

- 大堂

- 包间

Copyright © www.jiuzhang.com

What

关键字：Restaurant

- 桌子的规格不一样，能坐的人数区别
- 吧台里，party的区分
- 收费？

Copyright © www.jiuzhang.com

Clarify

- 针对本题

大堂，所有桌子都一样

Copyright © www.jiuzhang.com

Clarify

- How

规则？

Copyright © www.jiuzhang.com

Clarify

- How

- 是否能够预约？

Copyright © www.jiuzhang.com

Clarify

- How

- 是否能够送外卖？

Copyright © www.jiuzhang.com

Clarify

- How

- 每个Order需要区分是Dine-in还是Dine-out

Copyright © www.jiuzhang.com

Clarify

- 针对本题

没有Reservation

没有Dine-out

Copyright © www.jiuzhang.com

Clarify

- Who

Optional, 通过思考题目当中是否有人的出现，来帮助确定解题范围

*一般可以考虑人的角色以及人的属性，看是否题目需要

Copyright © www.jiuzhang.com

Clarify

- Who

人物越多越复杂

Copyright © www.jiuzhang.com

Clarify

- Who

List<Guest> ?

Party ?

Copyright © www.jiuzhang.com

Clarify

- Who

思考模式1：

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)

Copyright © www.jiuzhang.com

Clarify

- Who

Input → BLACK BOX → Output

思考模式2：

1. 客人进入餐馆，餐馆返回一个Table
2. 客人点菜，餐馆返回一桌菜
3. 客人付账，餐馆清空Table

Copyright © www.jiuzhang.com

Clarify

- Who

思考模式1：

Copyright © www.jiuzhang.com

Clarify

- Who

思考模式2：

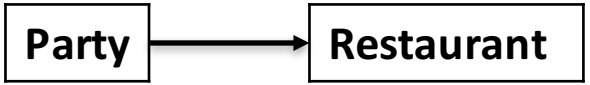
Copyright © www.jiuzhang.com

Core Object

Restaurant

Copyright © www.jiuzhang.com

Core Object

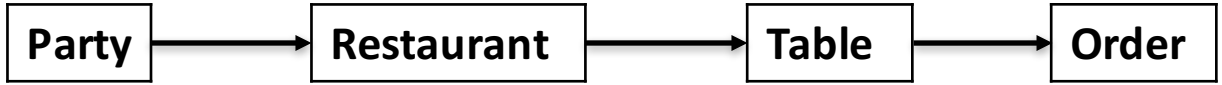


Copyright © www.jiuzhang.com

Core Object

Copyright © www.jiuzhang.com

Core Object



Copyright © www.jiuzhang.com

Core Object

Copyright © www.jiuzhang.com

Core Object

Party

Restaurant
- List<Table> tables

Table

Order

Meal

Copyright © www.jiuzhang.com

Core Object

Party

Restaurant
- List<Table> tables - List<Meal> menu

Table

Order

Meal

Copyright © www.jiuzhang.com

Core Object

Party

Restaurant
- List<Table> tables - List<Meal> menu

Table
- Party p

Order

Meal

Copyright © www.jiuzhang.com

Core Object

Party

Restaurant
- List<Table> tables - List<Meal> menu

Table

Order

Meal

Copyright © www.jiuzhang.com

Core Object

Party

Restaurant
- List<Table> tables - List<Meal> menu

Table

Order
- List<Meal> meals

Meal

Copyright © www.jiuzhang.com

Cases

- Party

Copyright © www.jiuzhang.com

Cases

- Party

- Make order ?

Copyright © www.jiuzhang.com

Cases

- Restaurant

Copyright © www.jiuzhang.com

Cases

- Restaurant

- Find table

Copyright © www.jiuzhang.com

Cases

- Restaurant

- Find table
- Take Order

Copyright © www.jiuzhang.com

Cases

- Restaurant

- Find table
- Take Order
- Checkout

Copyright © www.jiuzhang.com

Cases

- Table

- N/A

Copyright © www.jiuzhang.com

Cases

- Order

- N/A

Copyright © www.jiuzhang.com

Cases

- Meal

- N/A

Copyright © www.jiuzhang.com

Use case summary

- Mangement 类常见use case

Reserve:

Serve:

Checkout:

Copyright © www.jiuzhang.com

Use case summary

- Mangement 类常见use case

Reserve: 暂不考虑

Copyright © www.jiuzhang.com

Use case summary

- Mangement 类常见use case

Reserve: 暂不考虑

Serve: Find table

Copyright © www.jiuzhang.com

Use case summary

- Mangement 类常见use case

Reserve: 暂不考虑

Serve: Find table

Checkout: checkout

Copyright © www.jiuzhang.com

Use case summary

- Mangement 类常见use case

Reserve: 暂不考虑

Serve: Find table, take order

Checkout: checkout

Copyright © www.jiuzhang.com

Class

Party

Restaurant

- List<Table> tables
- List<Meal> menu

Table

Order

- List<Meal> meals

Meal

Use cases

Find table

Take order

Checkout

Copyright © www.jiuzhang.com

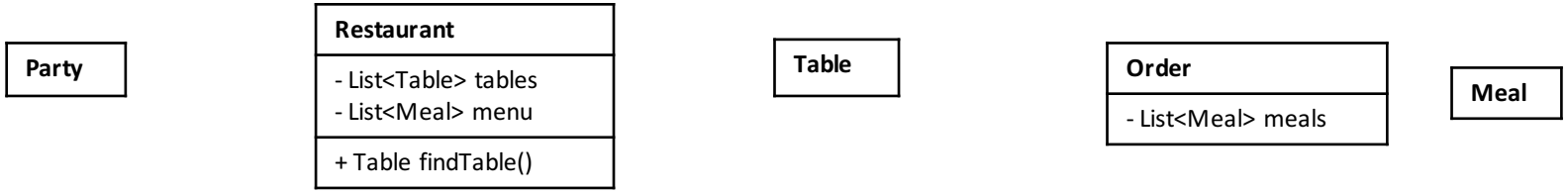
Class

- Use case: Find table

- Restaurant find an available table, an change the table to be unavailable

Copyright © www.jiuzhang.com

Class



Use cases
Find table
Take order
Checkout

Copyright © www.jiuzhang.com

Class

Party

Restaurant

- List<Table> tables
- List<Meal> menu

+ Table findTable()

Table

- Boolean available

Order

- List<Meal> meals

Meal

Use cases

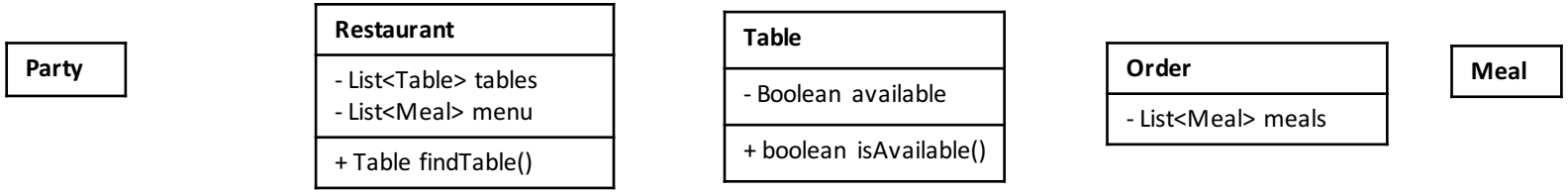
Find table

Take order

Checkout

Copyright © www.jiuzhang.com

Class



Use cases
Find table
Take order
Checkout

Copyright © www.jiuzhang.com

Class

九章算法

Party

Restaurant

- List<Table> tables
- List<Meal> menu

+ Table findTable()

Table

- Boolean available

+ boolean isAvailable()
+ void changeStatus()

Order

- List<Meal> meals

Meal

Use cases

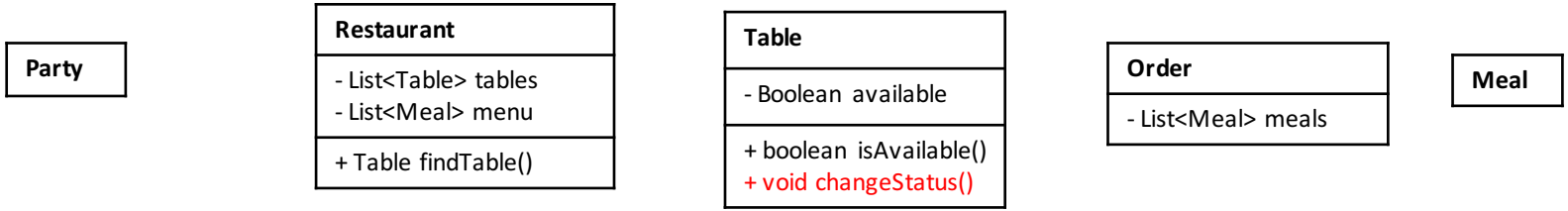
Find table

Take order

Checkout

Copyright © www.jiuzhang.com

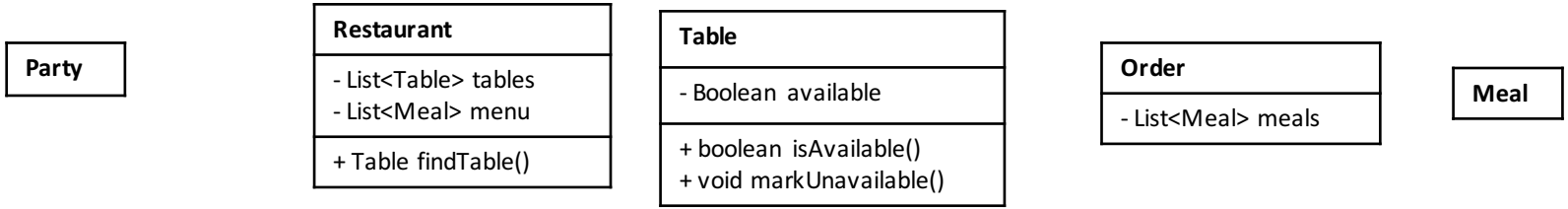
Best practice



Use cases
Find table
Take order
Checkout

Copyright © www.jiuzhang.com

Best practice



Use cases
Find table
Take order
Checkout

Copyright © www.jiuzhang.com

Class

- Use case: Take order

- Restaurant takes an order

Copyright © www.jiuzhang.com

Class

九章算法

Party

Restaurant

- List<Table> tables
- List<Meal> menu

+ Table findTable()
+ void takeOrder(Order o)

Table

- Boolean available

+ boolean isAvailable()
+ void markUnavailable()

Order

- List<Meal> meals

Meal

Use cases

Find table

Take order

Checkout

Copyright © www.jiuzhang.com

Class

- Use case: Check out

- Restaurant checks out a table/order, mark the table available again
- Calculate order price

Copyright © www.jiuzhang.com

Class

Party

Restaurant

- List<Table> tables
- List<Meal> menu

+ Table findTable()
+ void takeOrder(Order o)

Table

- Boolean available

+ boolean isAvailable()
+ void markUnavailable()

Order

- List<Meal> meals

Meal

Use cases
Find table
Take order
Checkout

Copyright © www.jiuzhang.com

Class

- Option 1: Order 里面记录Table信息
- Option 2: Table 里面记录Order信息

Copyright © www.jiuzhang.com

Option 1

九章算法

Party

Restaurant

- List<Table> tables
- List<Meal> menu

+ Table findTable()
+ void takeOrder(Order o)

Table

- Boolean available
- Order order

+ boolean isAvailable()
+ void markUnavailable()

Order

- List<Meal> meals

Meal

Use cases
Find table
Take order
Checkout

Copyright © www.jiuzhang.com

Option 1

Party

Restaurant

- List<Table> tables
- List<Meal> menu

+ Table findTable()
+ void takeOrder(Order o)
+ void checkout(Table t)

Table

- Boolean available
- Order order

+ boolean isAvailable()
+ void markUnavailable()

Order

- List<Meal> meals

Meal

Use cases
Find table
Take order
Checkout

Copyright © www.jiuzhang.com

Option 1

九章算法

Party

Restaurant

- List<Table> tables
- List<Meal> menu

+ Table findTable()
+ void takeOrder(Order o)
+ void checkout(Table t)

Table

- Boolean available
- Order order

+ boolean isAvailable()
+ void markUnavailable()
+ void markAvailable()

Order

- List<Meal> meals

Meal

Use cases
Find table
Take order
Checkout

Copyright © www.jiuzhang.com

Option 1

- Single responsibility principle

Copyright © www.jiuzhang.com

Option 1

- Single responsibility principle
- Force Table – Order mapping

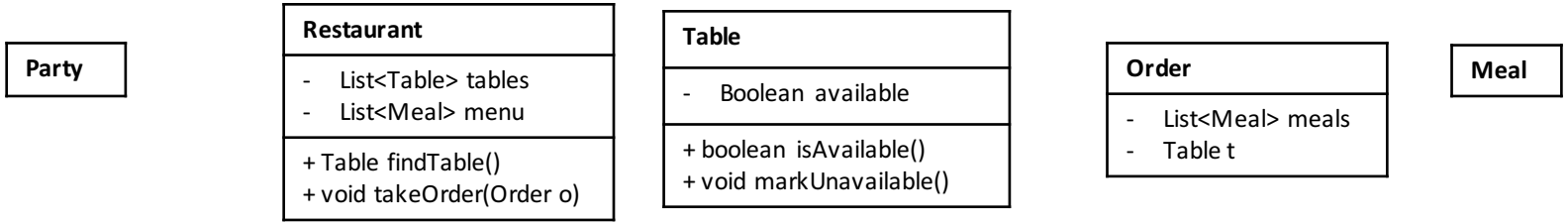
Copyright © www.jiuzhang.com

Option 1

- Single responsibility principle
- Force Table – Order mapping
- Share table use case

Copyright © www.jiuzhang.com

Option 2



Use cases
Find table
Take order
Checkout

Copyright © www.jiuzhang.com

Option 2

九章算法

Party

Restaurant

- List<Table> tables
- List<Meal> menu

+ Table findTable()
+ void takeOrder(Order o)
+ void checkout(Order o)

Table

- Boolean available

+ boolean isAvailable()
+ void markUnavailable()

Order

- List<Meal> meals
- Table t

Meal

Use cases
Find table
Take order
Checkout

Copyright © www.jiuzhang.com

Option 2

九章算法

Party

Restaurant

- List<Table> tables
- List<Meal> menu

+ Table findTable()
+ void takeOrder(Order o)
+ void checkout(Order o)

Table

- Boolean available

+ boolean isAvailable()
+ void markUnavailable()
+ void markAvailable()

Order

- List<Meal> meals
- Table t

Meal

Use cases
Find table
Take order
Checkout

Copyright © www.jiuzhang.com

Option 2

Party

Restaurant

- List<Table> tables
- List<Meal> menu

+ Table findTable()
+ void takeOrder(Order o)
+ void checkout(Order o)

Table

- Boolean available

+ boolean isAvailable()
+ void markUnavailable()
+ void markAvailable()

Order

- List<Meal> meals
- Table t

Meal

- Float price

+ float getPrice()

Use cases
Find table
Take order
Checkout

Copyright © www.jiuzhang.com

Option 2

九章算法

Party

Restaurant

- List<Table> tables
- List<Meal> menu

+ Table findTable()
+ void takeOrder(Order o)
+ void checkout(Order o)

Table

- Boolean available

+ boolean isAvailable()
+ void markUnavailable()
+ void markAvailable()

Order

- List<Meal> meals
- Table t

+ float getPrice()

Meal

- Float price

+ float getPrice()

Use cases

Find table

Take order

Checkout

Copyright © www.jiuzhang.com

Option 2

- Single responsibility principle
- Force Table – Order mapping
- Share table use case

Copyright © www.jiuzhang.com

Option 2

Party

Restaurant

- List<Table> tables
- List<Meal> menu

+ Table findTable()
+ void takeOrder(Order o)
+ void checkout(Order o)

Table

- Boolean available

+ boolean isAvailable()
+ void markUnavailable()
+ void markAvailable()

Order

- List<Meal> meals
- Table t
- Party p

+ float getPrice()

Meal

- Float price

+ float getPrice()

Use cases

Find table

Take order

Checkout

Copyright © www.jiuzhang.com

Class

Party

Restaurant

- List<Table> tables
- List<Meal> menu

+ Table findTable()
+ void takeOrder(Order o)
+ void checkout(Order o)

Table

- Boolean available

+ boolean isAvailable()
+ void markUnavailable()
+ void markAvailable()

Order

- List<Meal> meals
- Table t
- Party p

+ float getPrice()

Meal

- Float price

+ float getPrice()

Use cases

Find table

Take order

Checkout

Copyright © www.jiuzhang.com

Challenge

- How to mark a table available?

Copyright © www.jiuzhang.com

Class

Party

Restaurant

- List<Table> tables
- List<Meal> menu
- Map<Table, List<Order>> orders

+ Table findTable()
+ void takeOrder(Order o)
+ void checkout(Order o)

Table

- Boolean available

+ boolean isAvailable()
+ void markUnavailable()
+ void markAvailable()

Order

- List<Meal> meals
- Table t
- Party p

+ float getPrice()

Meal

- Float price

+ float getPrice()

Use cases

Find table

Take order

Checkout

Copyright © www.jiuzhang.com

Correctness

Party

Restaurant

- List<Table> tables
- List<Meal> menu
- Map<Table, List<Order>> orders

+ Table findTable()
+ void takeOrder(Order o)
+ void checkout(Order o)

Table

- Boolean available

+ boolean isAvailable()
+ void markUnavailable()
+ void markAvailable()

Order

- List<Meal> meals
- Table t
- Party p

+ float getPrice()

Meal

- Float price

+ float getPrice()

Use cases

Find table

Take order

Checkout

Copyright © www.jiuzhang.com

Correctness

Party

Restaurant

- List<Table> tables
- List<Meal> menu
- Map<Table, List<Order>> orders

+ Table findTable()
+ void takeOrder(Order o)
+ void checkout(Order o)

Table

- Boolean available

+ boolean isAvailable()
+ void markUnavailable()
+ void markAvailable()

Order

- List<Meal> meals
- Table t
- Party p

+ float getPrice()

Meal

- Float price

+ float getPrice()

NoTableException

Use cases

Find table

Take order

Checkout

Copyright © www.jiuzhang.com

Challenge

- 桌子大小不同？

Copyright © www.jiuzhang.com

Challenge

- 桌子大小不同？

Party && Table -> Capacity

Find table based on capacity

Copyright © www.jiuzhang.com

Challenge

九章算法

Party

- Int capacity

+ int getCapacity()

Restaurant

- List<Table> tables
- List<Meal> menu
- Map<Table, List<Order>> orders

+ Table findTable()
+ void takeOrder(Order o)
+ void checkout(Order o)

Table

- Boolean available

+ boolean isAvailable()
+ void markUnavailable()
+ void markAvailable()

Order

- List<Meal> meals
- Table t
- Party p

+ float getPrice()

Meal

- Float price

+ float getPrice()

NoTableException

Use cases

Find table

Take order

Checkout

Copyright © www.jiuzhang.com

Challenge

九章算法

Party

- Int capacity

+ int getCapacity()

Restaurant

- List<Table> tables
- List<Meal> menu
- Map<Table, List<Order>> orders

+ Table findTable()
+ void takeOrder(Order o)
+ void checkout(Order o)

Table

- Boolean available
- Int capacity

+ boolean isAvailable()
+ void markUnavailable()
+ void markAvailable()
+ get capacity()

Order

- List<Meal> meals
- Table t
- Party p

+ float getPrice()

Meal

- Float price

+ float getPrice()

NoTableException

Use cases

Find table

Take order

Checkout

Copyright © www.jiuzhang.com

Challenge

- How can you change your design to support reservation in your restaurant?

Copyright © www.jiuzhang.com

课程大纲

- 预定类OOD题型
- 预定类OOD解题思路
- Hotel reservation system
- Booking.com

Copyright © www.jiuzhang.com

预定类题库

九章算法

- Restaurant reservation system

Copyright © www.jiuzhang.com

预定类题库

- Restaurant reservation system
- Hotel reservation system

Copyright © www.jiuzhang.com

预定类题库

九章算法

- Restaurant reservation system
- Hotel reservation system
- Flight/Bus/Train reservation system

Copyright © www.jiuzhang.com

预定类题目特点

- 频率：中

Copyright © www.jiuzhang.com

预定类题目特点

- 频率：中
- 难度：高

Copyright © www.jiuzhang.com

预定类题目解题思路

九章算法

- What

- 考虑预定的东西

Copyright © www.jiuzhang.com

预定类题目解题思路

九章算法

- What

- 考虑预定的东西

例子：机票

Copyright © www.jiuzhang.com

预定类题目解题思路

九章算法

- What

- 考虑预定的东西

例子：机票

机舱/座位号/…

Copyright © www.jiuzhang.com

预定类题目解题思路

九章算法

- Use case

- Search
- Select

- Cancel

Copyright © www.jiuzhang.com

预定类题目解题思路

九章算法

- Use case

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

Copyright © www.jiuzhang.com

Reservation

- 有哪些需要和面试官统一的contract?

Copyright © www.jiuzhang.com

Reservation

- 有哪些需要和面试官统一的contract?

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

Copyright © www.jiuzhang.com

Reservation

- Search criteria

Copyright © www.jiuzhang.com

Reservation

- Search criteria

Make a reservation

2 people | Oct 10, 2017 | 7:00 PM | Find a Table

Copyright © www.jiuzhang.com

Reservation

- Search criteria

Make a reservation

2 people	Oct 10, 2017	7:00 PM	Find a Table

- 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

Copyright © www.jiuzhang.com

Reservation

- Search criteria

Make a reservation

2 people | Oct 10, 2017 | 7:00 PM | Find a Table

- 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？

Copyright © www.jiuzhang.com

Reservation

- Search criteria

Make a reservation

2 people | Oct 10, 2017 | 7:00 PM | Find a Table

- 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？
- 时间： 是否所有时间都允许预定？

Copyright © www.jiuzhang.com

Reservation

- Search criteria

Make a reservation

2 people	Oct 10, 2017	7:00 PM	Find a Table

- 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

Copyright © www.jiuzhang.com

Reservation

- Search criteria

Make a reservation

2 people	Oct 10, 2017	7:00 PM	Find a Table

- 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)

Copyright © www.jiuzhang.com

Reservation

- Search criteria

Make a reservation

2 people	Oct 10, 2017	7:00 PM	Find a Table

- 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

Copyright © www.jiuzhang.com

Reservation

- List<Result>

Copyright © www.jiuzhang.com

Reservation

- List<Result>

-当选择的时间段可以/不行时，系统应该给出什么反馈？

Copyright © www.jiuzhang.com

Reservation

- 做法一：

Make a reservation

2 people | Nov 2, 2017 | 7:30 PM | Find a Table

Booked 22 times today

5:30 PM | 6:00 PM | 7:30 PM | 8:00 PM | 8:30 PM

See affiliated restaurants ›

Make a reservation

2 people | Oct 10, 2017 | 7:00 PM | Find a Table

Booked 22 times today

You're in luck! We still have 1 timeslot left

9:30 PM

See affiliated restaurants ›

Copyright © www.jiuzhang.com

Reservation

- List<Result>

Result == Timeslot

Copyright © www.jiuzhang.com

Reservation

- List<Result>

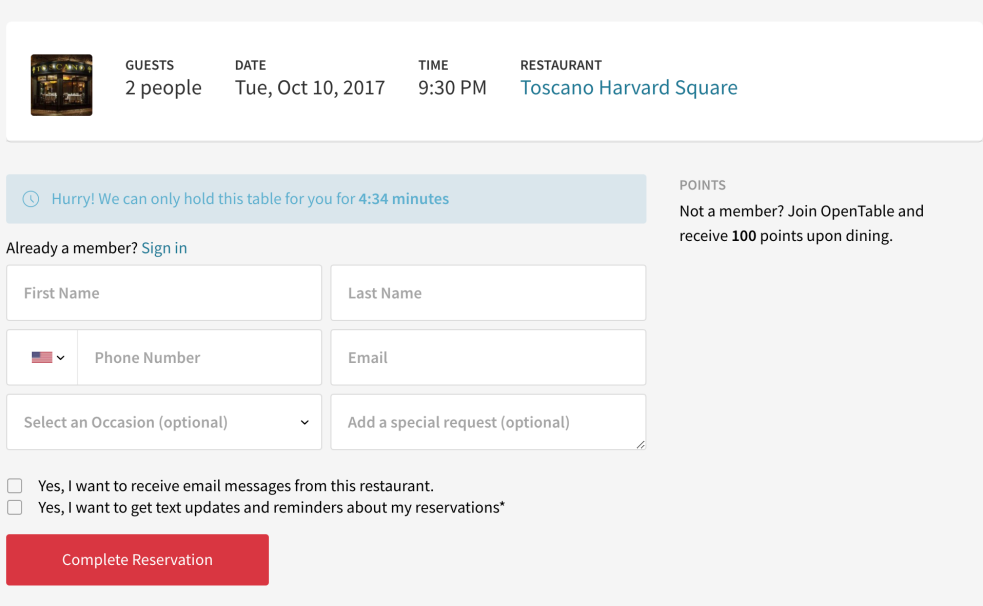
Result == Timeslot

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

Copyright © www.jiuzhang.com

Reservation

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



Copyright © www.jiuzhang.com

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

Copyright © www.jiuzhang.com

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)

Copyright © www.jiuzhang.com

Reservation

- 为什么我们可以跳过List<Result>这个步骤？

Copyright © www.jiuzhang.com

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

Copyright © www.jiuzhang.com

Restaurant

九章算法

Party

- Int capacity

+ int getCapacity()

Restaurant

- List<Table> tables
- List<Meal> menu
- Map<Table, List<Order>> orders

+ Table findTable()
+ void takeOrder(Order o)
+ void checkout(Order o)

Table

- Boolean available
- Int capacity

+ boolean isAvailable()
+ void markUnavailable()
+ void markAvailable()
+ get capacity()

Order

- List<Meal> meals
- Table t
- Party p

+ float getPrice()

Meal

- Float price

+ float getPrice()

NoTableException

Use cases
Find table
Take order
Checkout

Copyright © www.jiuzhang.com

Restaurant

九章算法

Party

- Int capacity

+ int getCapacity()

Restaurant

- List<Table> tables
- List<Meal> menu
- Map<Table, List<Order>> orders

+ Table findTable()
+ void takeOrder(Order o)
+ void checkout(Order o)

Table

- Boolean available
- Int capacity

+ boolean isAvailable()
+ void markUnavailable()
+ void markAvailable()
+ get capacity()

Order

- List<Meal> meals
- Table t
- Party p

+ float getPrice()

Meal

- Float price

+ float getPrice()

Reservation

- Table table
- Timeslot timeslot

NoTableException

Use cases

Find table

Take order

Checkout

Copyright © www.jiuzhang.com

Restaurant

Party

- Int capacity

+ int getCapacity()

Table

- Boolean available
- Int capacity

+ boolean isAvailable()
+ void markUnavailable()
+ void markAvailable()
+ get capacity()

Order

- List<Meal> meals
- Table t
- Party p

+ float getPrice()

Meal

- Float price

+ float getPrice()

Restaurant

- List<Table> tables
- List<Meal> menu
- Map<Table, List<Order>> orders

+ Table findTable()
+ void takeOrder(Order o)
+ void checkout(Order o)
+ Reservation findTableForReservation(Timeslot t)

Reservation

- Table table
- Timeslot timeslot

NoTableException

Use cases

Find table

Take order

Checkout

Copyright © www.jiuzhang.com

Restaurant

Party

- Int capacity

+ int getCapacity()

Table

- Boolean available
- Int capacity

+ boolean isAvailable()
+ void markUnavailable()
+ void markAvailable()
+ get capacity()

Order

- List<Meal> meals
- Table t
- Party p

+ float getPrice()

Meal

- Float price

+ float getPrice()

Restaurant

- List<Table> tables
- List<Meal> menu
- Map<Table, List<Order>> orders

+ Table findTable()
+ void takeOrder(Order o)
+ void checkout(Order o)
+ Reservation findTableForReservation(Timeslot t)

Reservation

- Table table
- Timeslot timeslot

NoTableException

NoTableForReservationException

Use cases

Find table

Take order

Checkout

Copyright © www.jiuzhang.com

Restaurant

Party

- Int capacity

+ int getCapacity()

Table

- Boolean available
- Int capacity

+ boolean isAvailable()
+ void markUnavailable()
+ void markAvailable()
+ get capacity()

Order

- List<Meal> meals
- Table t
- Party p

+ float getPrice()

Meal

- Float price

+ float getPrice()

Restaurant

- List<Table> tables
- List<Meal> menu
- Map<Table, List<Order>> orders

+ Table findTable()
+ void takeOrder(Order o)
+ void checkout(Order o)
+ Reservation findTableForReservation(Timeslot t)
+ void confirmReservation(Reservation r)

Reservation

- Table table
- Timeslot timeslot

NoTableException

NoTableForReservationException

Use cases

Find table

Take order

Checkout

Copyright © www.jiuzhang.com

Any problems?

Copyright © www.jiuzhang.com

Any problems?

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

Copyright © www.jiuzhang.com

Any problems?

- 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

Any problems?

- 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

Any problems?

- 如何知道对于一张Table:

Any reservation during Timeslot – MAX_DINETIME to Timeslot + MAX_DINETIME ?

Copyright © www.jiuzhang.com

Any problems?

- 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

Copyright © www.jiuzhang.com

Any problems?

- 如何知道对于一张Table:

Any reservation during Timeslot – MAX_DINETIME to Timeslot + MAX_DINETIME ?

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

Copyright © www.jiuzhang.com

Any problems?

- 如何知道对于一张Table:

Any reservation during Timeslot – MAX_DINETIME to Timeslot + MAX_DINETIME ?

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

Copyright © www.jiuzhang.com

Solution 1 vs. Solution 2

Copyright © www.jiuzhang.com

Solution 1 vs. Solution 2

九章算法

- 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

Copyright © www.jiuzhang.com

Solution 1 vs. Solution 2

- 分析各自的优缺点：

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

Copyright © www.jiuzhang.com

Restaurant

Party

- Int capacity

+ int getCapacity()

Table

- Boolean available
- Int capacity

+ boolean isAvailable()
+ void markUnavailable()
+ void markAvailable()
+ get capacity()

Order

- List<Meal> meals
- Table t
- Party p

+ float getPrice()

Meal

- Float price

+ float getPrice()

Restaurant

- List<Table> tables
- List<Meal> menu
- Map<Table, List<Order>> orders

+ Table findTable()
+ void takeOrder(Order o)
+ void checkout(Order o)
+ Reservation findTableForReservation(Timeslot t)
+ void confirmReservation(Reservation r)

Reservation

- Table table
- Timeslot timeslot

NoTableException

NoTableForReservationException

Use cases

Find table

Take order

Checkout

Copyright © www.jiuzhang.com

Restaurant

Party

- Int capacity

+ int getCapacity()

Table

- Boolean available
- Int capacity
- List<Timeslot> reservations

+ boolean isAvailable()
+ void markUnavailable()
+ void markAvailable()
+ get capacity()

Order

- List<Meal> meals
- Table t
- Party p

+ float getPrice()

Meal

- Float price

+ float getPrice()

Restaurant

- List<Table> tables
- List<Meal> menu
- Map<Table, List<Order>> orders

+ Table findTable()
+ void takeOrder(Order o)
+ void checkout(Order o)
+ Reservation findTableForReservation(Timeslot t)
+ void confirmReservation(Reservation r)

Reservation

- Table table
- Timeslot timeslot

NoTableException

NoTableForReservationException

Use cases

Find table

Take order

Checkout

Copyright © www.jiuzhang.com

Restaurant

Party

- Int capacity

+ int getCapacity()

Table

- Boolean available
- Int capacity

+ boolean isAvailable()
+ void markUnavailable()
+ void markAvailable()
+ get capacity()

Order

- List<Meal> meals
- Table t
- Party p

+ float getPrice()

Meal

- Float price

+ float getPrice()

Restaurant

- List<Table> tables
- List<Meal> menu
- Map<Table, List<Order>> orders
- Map<Table, List<Timeslot>> reservation

+ Table findTable()
+ void takeOrder(Order o)
+ void checkout(Order o)
+ Reservation findTableForReservation(Timeslot t)
+ void confirmReservation(Reservation r)

Reservation

- Table table
- Timeslot timeslot

NoTableException

NoTableForReservationException

Use cases

Find table

Take order

Checkout

Copyright © www.jiuzhang.com

Cancel reservation

- Restaurant takes a cancel reservation request, and ask the table to free that timeslot.

Copyright © www.jiuzhang.com

Restaurant

Party

- Int capacity

+ int getCapacity()

Table

- Boolean available
- Int capacity

+ boolean isAvailable()
+ void markUnavailable()
+ void markAvailable()
+ get capacity()

Order

- List<Meal> meals
- Table t
- Party p

+ float getPrice()

Meal

- Float price

+ float getPrice()

Restaurant

- List<Table> tables
- List<Meal> menu
- Map<Table, List<Order>> orders
- Map<Table, List<Timeslot>> reservation

+ Table findTable()
+ void takeOrder(Order o)
+ void checkout(Order o)
+ Reservation findTableForReservation(Timeslot t)
+ void confirmReservation(Reservation r)

Reservation

- Table table
- Timeslot timeslot

NoTableException

NoTableForReservationException

Use cases

Find table

Take order

Checkout

Copyright © www.jiuzhang.com

Restaurant

Party

- Int capacity

+ int getCapacity()

Table

- Boolean available
- Int capacity

+ boolean isAvailable()
+ void markUnavailable()
+ void markAvailable()
+ get capacity()

Order

- List<Meal> meals
- Table t
- Party p

+ float getPrice()

Meal

- Float price

+ float getPrice()

Restaurant

- List<Table> tables
- List<Meal> menu
- Map<Table, List<Order>> orders
- Map<Table, List<Timeslot>> reservation

+ Table findTable()
+ void takeOrder(Order o)
+ void checkout(Order o)
+ Reservation findTableForReservation(Timeslot t)
+ void confirmReservation(Reservation r)
+ void cancelReservation(Reservation r)

Reservation

- Table table
- Timeslot timeslot

NoTableException

NoTableForReservationException

Use cases

Find table

Take order

Checkout

Copyright © www.jiuzhang.com

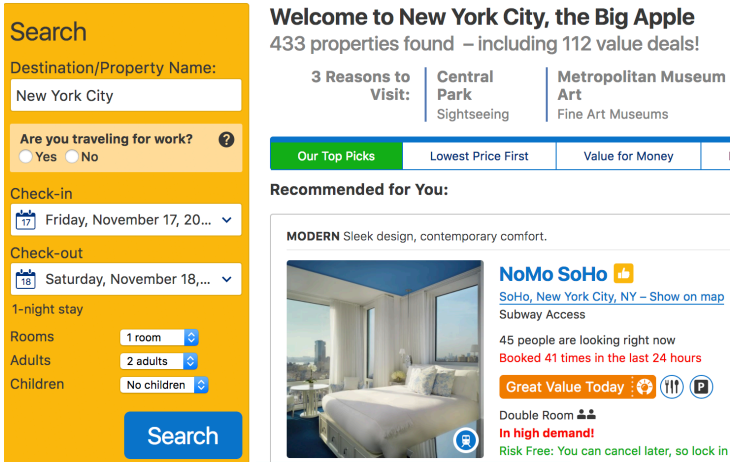
Hotel reservation system

- Can you design a hotel reservation system?

Copyright © www.jiuzhang.com

Hotel reservation system

- Can you design a hotel reservation system?



VS.

Copyright © www.jiuzhang.com

Clarify

- 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？

Design上会有哪些区别？

Copyright © www.jiuzhang.com

Clarify

- 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

Copyright © www.jiuzhang.com

Clarify

- 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？

- 返回结果区别

Rooms VS. Hotels

Copyright © www.jiuzhang.com

Clarify

九章算法

- 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

Copyright © www.jiuzhang.com

Clarify

- What

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

Copyright © www.jiuzhang.com

Clarify

- 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>

Copyright © www.jiuzhang.com

Clarify

- What

关键字：Room

Copyright © www.jiuzhang.com

Clarify

- What

关键字：Room

Room Type	Sleeps	Today's Price	Your Choices	Select Rooms	Confirm Your Reservation
Classic Room with King Bed In high demand! 1 king bed This room includes 1 extra large double bed, cable TV and a work area. Private bathroom Free WiFi • Iron • Free toiletries • Bathroom • Telephone • Towels • Linens		$259 Excludes: Taxes and US$ 21.78 Property service charge per night There's an even lower price available! Sign in to see it	• Low rate – no money back • Service charge	0	I'll reserve Confirmation is immediate 101 other people looking now
		$344 Excludes: Taxes and US$ 21.78 Property service charge per night	✔ FREE cancellation before Oct 19, 2017 ✔ No advance payment • Service charge	0	
		$354 Excludes: Taxes and US$ 21.78 Property service charge per night There's an even lower price available! Sign in to see it	Breakfast included ✔ FREE cancellation before Oct 19, 2017 ✔ No advance payment • Service charge	0	
Penn 5000 Room with Double Bed Only 1 room left! 1 full bed This room includes a refrigerator and a flat-screen TV. Private bathroom Free WiFi More		$269 Excludes: Taxes and US$ 21.78 Property service charge per night There's an even lower price available! Sign in to see it	• Low rate – no money back • Service charge	0	
		$354 Excludes: Taxes and US$ 21.78 Property service charge per night There's an even lower price available! Sign in to see it	✔ FREE cancellation before Oct 19, 2017 ✔ No advance payment • Service charge	0	
		$364 Excludes: Taxes and US$ 21.78 Property service charge per night	Breakfast included ✔ FREE cancellation before Oct 19, 2017 ✔ No advance payment • Service charge	0	

Copyright © www.jiuzhang.com

Clarify

- What

针对本题：房间的人数和价格可能会不同

Copyright © www.jiuzhang.com

Challenge

针对本题：房间的人数和价格可能会不同。

如何设计房间类？

Copyright © www.jiuzhang.com

Challenge

如何设计房间类？

Room
- int capacity - float price

Copyright © www.jiuzhang.com

Challenge

如何设计房间类？

Room
- int capacity - float price

List<Result> -> List<Room>

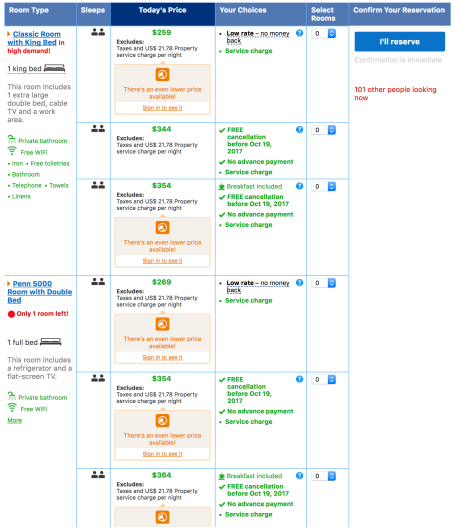
Room_1	Room_2	Room_3	Room_4	Room_5	Room_6
Capacity : 2 Price: 198	Capacity : 2 Price: 198	Capacity : 2 Price: 198	Capacity : 2 Price: 198	Capacity : 1 Price: 128	Capacity : 1 Price: 128

Copyright © www.jiuzhang.com

Challenge

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 Price: 198	Capacity : 2 Price: 198	Capacity : 2 Price: 198	Capacity : 2 Price: 198	Capacity : 1 Price: 128	Capacity : 1 Price: 128

Copyright © www.jiuzhang.com

Challenge

How many room types do we have for this hotel?

Copyright © www.jiuzhang.com

Challenge

How many room types do we have for this hotel?

针对本题：两种 （Single room 和 Double room）

Copyright © www.jiuzhang.com

Challenge

Restaurant VS. Hotel

什么情况下需要考虑Result Type (Room Type)?

Copyright © www.jiuzhang.com

Challenge

Restaurant VS. Hotel

什么情况下需要考虑Result Type (Room Type)?

价格不同时

Copyright © www.jiuzhang.com

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

Copyright © www.jiuzhang.com

Clarify

- How

规则？

Copyright © www.jiuzhang.com

Clarify

- 预定类规则：

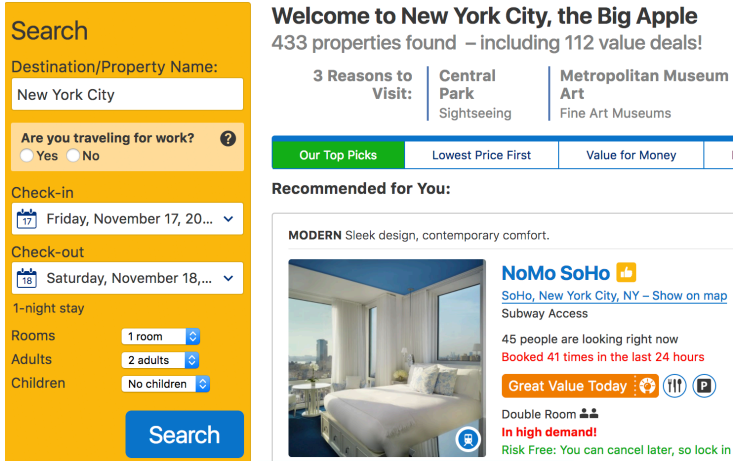
Search criteria

Copyright © www.jiuzhang.com

Clarify

- 预定类规则：

Search criteria



Copyright © www.jiuzhang.com

Clarify

- 当人数超过房间大小怎么办？

Copyright © www.jiuzhang.com

Clarify

- 当人数超过房间大小怎么办？

1. 列出所有组合的选择

e.g. 3人入住：

DoubleRoom	DoubleRoom	SingleRoom
DoubleRoom	SingleRoom	SingleRoom
		SingleRoom

Copyright © www.jiuzhang.com

Clarify

- 当人数超过房间大小怎么办？

1. 列出所有组合的选择
2. 由大到小自动分配

e.g. 3人入住：

DoubleRoom

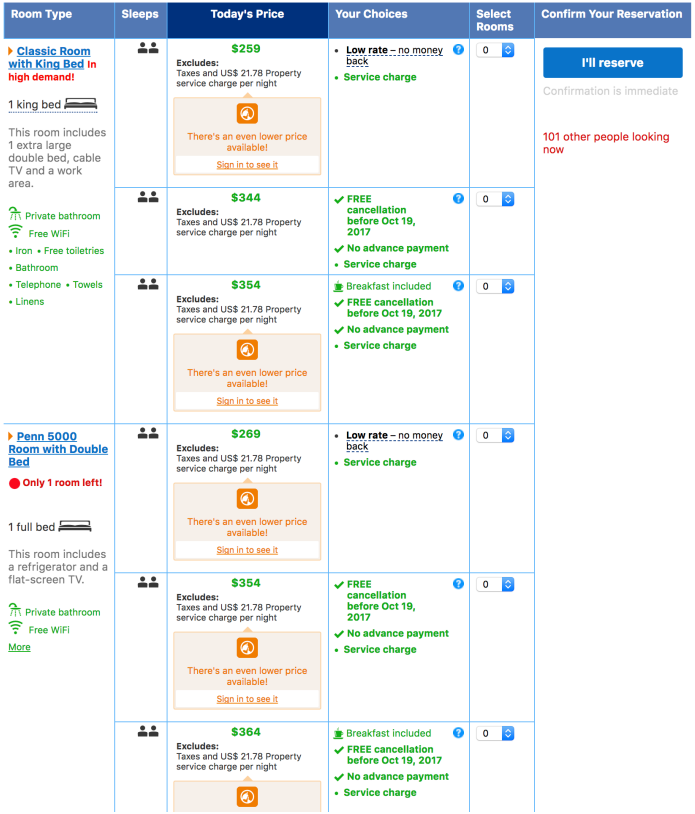
SingleRoom

Copyright © www.jiuzhang.com

Clarify

- 当人数超过房间大小怎么办？

1. 列出所有组合的选择
2. 由大到小自动分配
3. 用户自主选择房间数量



Copyright © www.jiuzhang.com

Clarify

Who && When

N/A

Copyright © www.jiuzhang.com

Core object

Copyright © www.jiuzhang.com

Core object

Hotel

Copyright © www.jiuzhang.com

Core object

Request

Hotel

Copyright © www.jiuzhang.com

Core object

Request

Hotel

RoomType

Copyright © www.jiuzhang.com

Core object

Request

Hotel

RoomType

Room

Copyright © www.jiuzhang.com

Core object

Request

Hotel

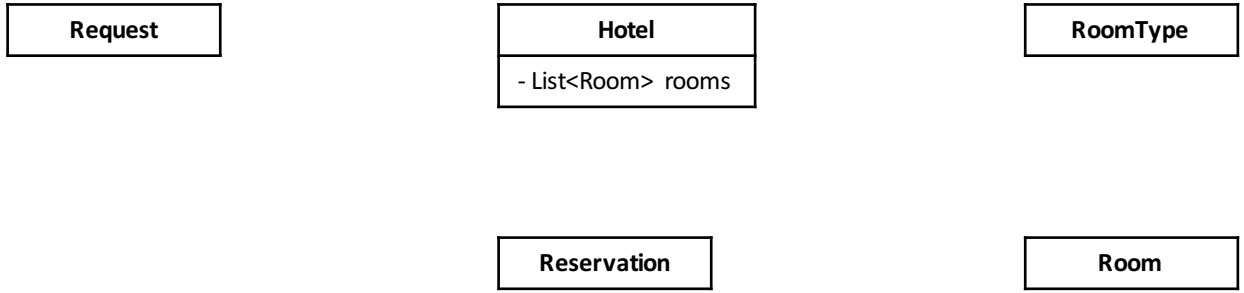
RoomType

Reservation

Room

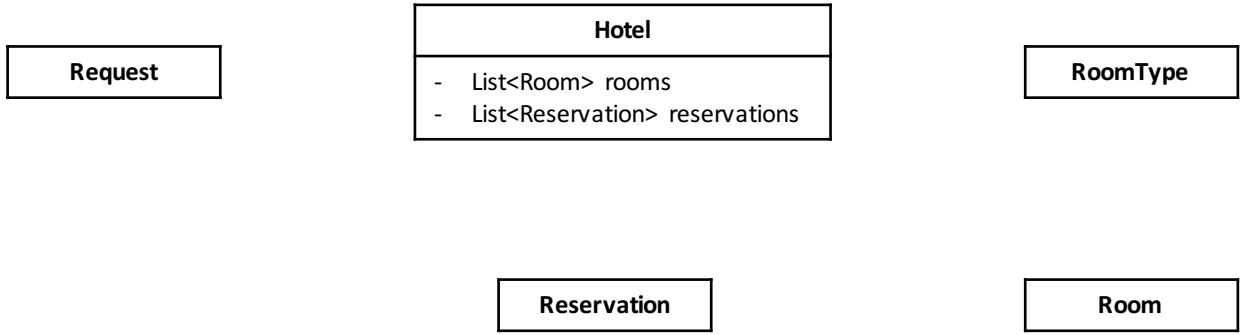
Copyright © www.jiuzhang.com

Core object



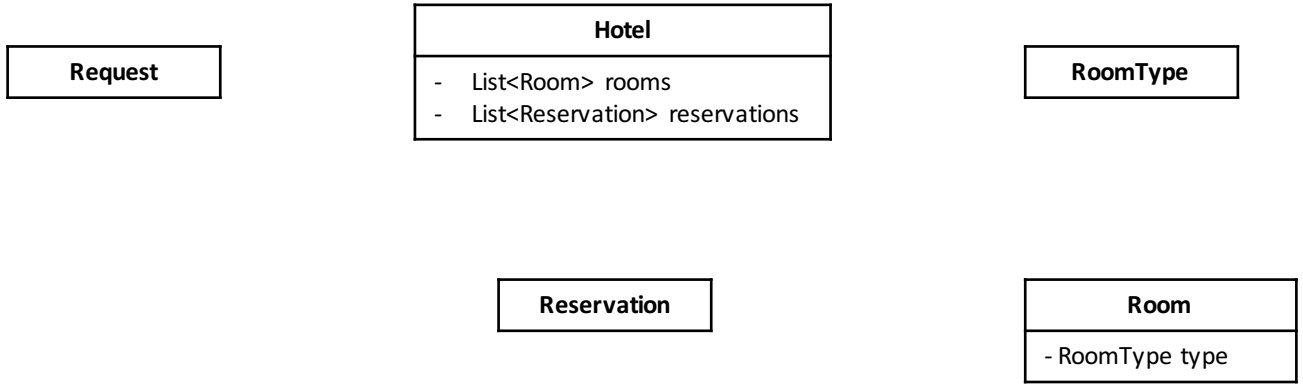
Copyright © www.jiuzhang.com

Core object



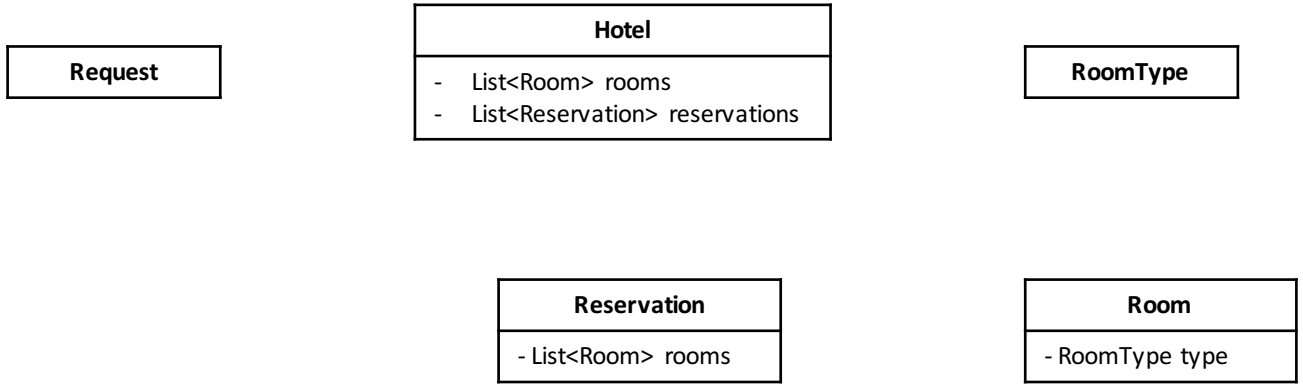
Copyright © www.jiuzhang.com

Core object



Copyright © www.jiuzhang.com

Core object



Copyright © www.jiuzhang.com

Use cases

Hotel

Copyright © www.jiuzhang.com

Use cases

Hotel:

- Search for available rooms

Copyright © www.jiuzhang.com

Use cases

Hotel:

- Search for available rooms
- Make reservation

Copyright © www.jiuzhang.com

Use cases

Hotel:

- Search for available rooms
- Make reservation
- Cancel reservation

Copyright © www.jiuzhang.com

Classes

Request

Hotel

- List<Room> rooms
- List<Reservation> reservations

RoomType

Room

- RoomType type

Reservation

- List<Room> rooms

Use cases

Search for available rooms

Make reservation

Cancel reservation

Copyright © www.jiuzhang.com

Use cases

Search for available rooms

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

Copyright © www.jiuzhang.com

Use cases

Search for available rooms

1: Based on search criteria

Copyright © www.jiuzhang.com

Classes

Request
- Date startDate

Hotel
- List<Room> rooms - List<Reservation> reservations

RoomType

Room
- RoomType type

Reservation
- List<Room> rooms

Use cases
Search for available rooms
Make reservation
Cancel reservation

Copyright © www.jiuzhang.com

Classes

九章算法

Request
- Date startDate - Date endDate

Hotel
- List<Room> rooms - List<Reservation> reservations

RoomType

Room
- RoomType type

Reservation
- List<Room> rooms

Use cases
Search for available rooms
Make reservation
Cancel reservation

Copyright © www.jiuzhang.com

Classes

九章算法

Request
- Date startDate - Date endDate

Hotel
- List<Room> rooms - List<Reservation> reservations
+ Map<RoomType, int> handleSearchRequest(Request r)

RoomType

Room
- RoomType type

Reservation
- List<Room> rooms

Use cases
Search for available rooms
Make reservation
Cancel reservation

Copyright © www.jiuzhang.com

Use cases

Search for available rooms

2: Go through rooms to check availability

Copyright © www.jiuzhang.com

Classes

Request
- Date startDate - Date endDate

Hotel
- List<Room> rooms - List<Reservation> reservations
+ Map<RoomType, int> handleSearchRequest(Request r)

RoomType

Room
- RoomType type - List<Date> reservations

Reservation
- List<Room> rooms

Use cases
Search for available rooms
Make reservation
Cancel reservation

Copyright © www.jiuzhang.com

Classes

九章算法

Request
- Date startDate - Date endDate

Hotel
- List<Room> rooms - List<Reservation> reservations - Map<Room, List<Date>> roomReservations
+ Map<RoomType, int> handleSearchRequest(Request r)

RoomType

Room
- RoomType type

Reservation
- List<Room> rooms

Use cases
Search for available rooms
Make reservation
Cancel reservation

Copyright © www.jiuzhang.com

Classes

九章算法

Request
- Date startDate - Date endDate

Hotel
- List<Room> rooms - List<Reservation> reservations - Map<Room, List<Date>> roomReservations
+ Map<RoomType, int> handleSearchRequest(Request r) - boolean isRequestAvailable(Request r, List<Date> dates)

RoomType

Room
- RoomType type

Reservation
- List<Room> rooms

Use cases
Search for available rooms
Make reservation
Cancel reservation

Copyright © www.jiuzhang.com

Classes

九章算法

Request
- Date startDate - Date endDate

Hotel
- List<Room> rooms - List<Reservation> reservations - Map<Room, List<Date>> roomReservations
+ Map<RoomType, int> handleSearchRequest(Request r) - boolean isRequestAvailable(Request r, List<Date> dates)

RoomType

Room
- RoomType type - Boolean available

Reservation
- List<Room> rooms

Use cases
Search for available rooms
Make reservation
Cancel reservation

Copyright © www.jiuzhang.com

Classes

九章算法

Request

- Date startDate
- Date endDate

Hotel

- List<Room> rooms
- List<Reservation> reservations
- Map<Room, List<Date>> roomReservations

+ Map<RoomType, int> handleSearchRequest(Request r)
- boolean isRequestAvailable(Request r, List<Date> dates)

RoomType

Room

- RoomType type
- Boolean available

+ boolean isAvailable()

Reservation

- List<Room> rooms

Use cases

Search for available rooms

Make reservation

Cancel reservation

Copyright © www.jiuzhang.com

Use cases

Search for available rooms

3: list available room types and counts

Copyright © www.jiuzhang.com

Classes

九章算法

Request
- Date startDate - Date endDate

Hotel
- List<Room> rooms - List<Reservation> reservations - Map<Room, List<Date>> roomReservations
+ Map<RoomType, int> handleSearchRequest(Request r) - boolean isRequestAvailable(Request r, List<Date> dates)

RoomType
SINGLE DOUBLE

Room
- RoomType type - Boolean available
+ boolean isAvailable()

Reservation
- List<Room> rooms

Use cases
Search for available rooms
Make reservation
Cancel reservation

Copyright © www.jiuzhang.com

Classes

九章算法

Request
- Date startDate - Date endDate

Hotel
- List<Room> rooms - List<Reservation> reservations - Map<Room, List<Date>> roomReservations
+ Map<RoomType, int> handleSearchRequest(Request r) - boolean isRequestAvailable(Request r, List<Date> dates)

<<Enumeration>> RoomType
SINGLE DOUBLE

Room
- RoomType type - Boolean available
+ boolean isAvailable()

Reservation
- List<Room> rooms

Use cases
Search for available rooms
Make reservation
Cancel reservation

Copyright © www.jiuzhang.com

Use cases

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

Copyright © www.jiuzhang.com

Use cases

Make reservation

1: Add RoomType and number of rooms in a request

Copyright © www.jiuzhang.com

Classes

Request
- Date startDate - Date endDate

Hotel
- List<Room> rooms - List<Reservation> reservations - Map<Room, List<Date>> roomReservations
+ Map<RoomType, int> handleSearchRequest(Request r) - boolean isRequestAvailable(Request r, List<Date> dates)

<<Enumeration>> RoomType
SINGLE DOUBLE

Room
- RoomType type - Boolean available
+ boolean isAvailable()

Reservation
- List<Room> rooms

Use cases
Search for available rooms
Make reservation
Cancel reservation

Copyright © www.jiuzhang.com

Classes

九章算法

Request

- Date startDate
- Date endDate

Hotel

- List<Room> rooms
- List<Reservation> reservations
- Map<Room, List<Date>> roomReservations

+ Map<RoomType, int> handleSearchRequest(Request r)
- boolean isRequestAvailable(Request r, List<Date> dates)

<<Enumeration>>
RoomType

SINGLE
DOUBLE

Room

- RoomType type
- Boolean available

+ boolean isAvailable()

Reservation

- List<Room> rooms

ReservationRequest

Use cases

Search for available rooms
Make reservation
Cancel reservation

Copyright © www.jiuzhang.com

Classes

九章算法

Request

- Date startDate
- Date endDate

Hotel

- List<Room> rooms
- List<Reservation> reservations
- Map<Room, List<Date>> roomReservations

+ Map<RoomType, int> handleSearchRequest(Request r)
- boolean isRequestAvailable(Request r, List<Date> dates)

<<Enumeration>>
RoomType

SINGLE
DOUBLE

Room

- RoomType type
- Boolean available

+ boolean isAvailable()

Reservation

- List<Room> rooms

ReservationRequest

- Date startDate
- Date endDate

Use cases

Search for available rooms
Make reservation
Cancel reservation

Copyright © www.jiuzhang.com

Classes

九章算法

Request

- Date startDate
- Date endDate

Hotel

- List<Room> rooms
- List<Reservation> reservations
- Map<Room, List<Date>> roomReservations

+ Map<RoomType, int> handleSearchRequest(Request r)
- boolean isRequestAvailable(Request r, List<Date> dates)

<<Enumeration>>
RoomType

SINGLE
DOUBLE

Room

- RoomType type
- Boolean available

+ boolean isAvailable()

Reservation

- List<Room> rooms

ReservationRequest

- Date startDate
- Date endDate
- Map<RoomType, int> roomsNeeded

Use cases

Search for available rooms
Make reservation
Cancel reservation

Copyright © www.jiuzhang.com

Use cases

Make reservation

2: Send request to Hotel

Copyright © www.jiuzhang.com

Classes

九章算法

Request

- Date startDate
- Date endDate

Hotel

- List<Room> rooms
- List<Reservation> reservations
- Map<Room, List<Date>> roomReservations

+ Map<RoomType, int> handleSearchRequest(Request r)
- boolean isRequestAvailable(Request r, List<Date> dates)

<<Enumeration>>
RoomType

SINGLE
DOUBLE

Room

- RoomType type
- Boolean available

+ boolean isAvailable()

Reservation

- List<Room> rooms

ReservationRequest

- Date startDate
- Date endDate
- Map<RoomType, int> roomsNeeded

Use cases

Search for available rooms
Make reservation
Cancel reservation

Copyright © www.jiuzhang.com

Classes

九章算法

Request

- Date startDate
- Date endDate

Hotel

- List<Room> rooms
- List<Reservation> reservations
- Map<Room, List<Date>> roomReservations

+ Map<RoomType, int> handleSearchRequest(Request r)
- boolean isRequestAvailable(Request r, List<Date> dates)
+ Reservation makeReservation(ReservationRequest r)

<<Enumeration>> RoomType

SINGLE
DOUBLE

Room

- RoomType type
- Boolean available

+ boolean isAvailable()

Reservation

- List<Room> rooms

ReservationRequest

- Date startDate
- Date endDate
- Map<RoomType, int> roomsNeeded

Use cases

Search for available rooms
Make reservation
Cancel reservation

Copyright © www.jiuzhang.com

Use cases

Make reservation

3: If there is enough room left, confirm the reservation

Copyright © www.jiuzhang.com

Classes

Request

- Date startDate
- Date endDate

Hotel

- List<Room> rooms
- List<Reservation> reservations
- Map<Room, List<Date>> roomReservations

+ Map<RoomType, int> handleSearchRequest(Request r)
- boolean isRequestAvailable(Request r, List<Date> dates)
+ Reservation makeReservation(ReservationRequest r)

<<Enumeration>> RoomType

SINGLE
DOUBLE

Room

- RoomType type
- Boolean available

+ boolean isAvailable()

Reservation

- List<Room> rooms
- Date startDate
- Date endDate

ReservationRequest

- Date startDate
- Date endDate
- Map<RoomType, int> roomsNeeded

Use cases

Search for available rooms
Make reservation
Cancel reservation

Copyright © www.jiuzhang.com

Use cases

九章算法

Make reservation

4: If there isn’t enough room left, throw exception

Copyright © www.jiuzhang.com

Classes

九章算法

Request

- Date startDate
- Date endDate

Hotel

- List<Room> rooms
- List<Reservation> reservations
- Map<Room, List<Date>> roomReservations

+ Map<RoomType, int> handleSearchRequest(Request r)
- boolean isRequestAvailable(Request r, List<Date> dates)
+ Reservation makeReservation(ReservationRequest r)

<<Enumeration>>
RoomType

SINGLE
DOUBLE

Room

- RoomType type
- Boolean available

+ boolean isAvailable()

Reservation

- List<Room> rooms
- Date startDate
- Date endDate

ReservationRequest

- Date startDate
- Date endDate
- Map<RoomType, int> roomsNeeded

NotEnoughRoomForReservationException

Use cases

Search for available rooms
Make reservation
Cancel reservation

Copyright © www.jiuzhang.com

Challenge

九章算法

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

Copyright © www.jiuzhang.com

Challenge

九章算法

```java
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);
        }
    }
}
```

Copyright © www.jiuzhang.com

Challenge

- Map<RoomType, List<Room>> map

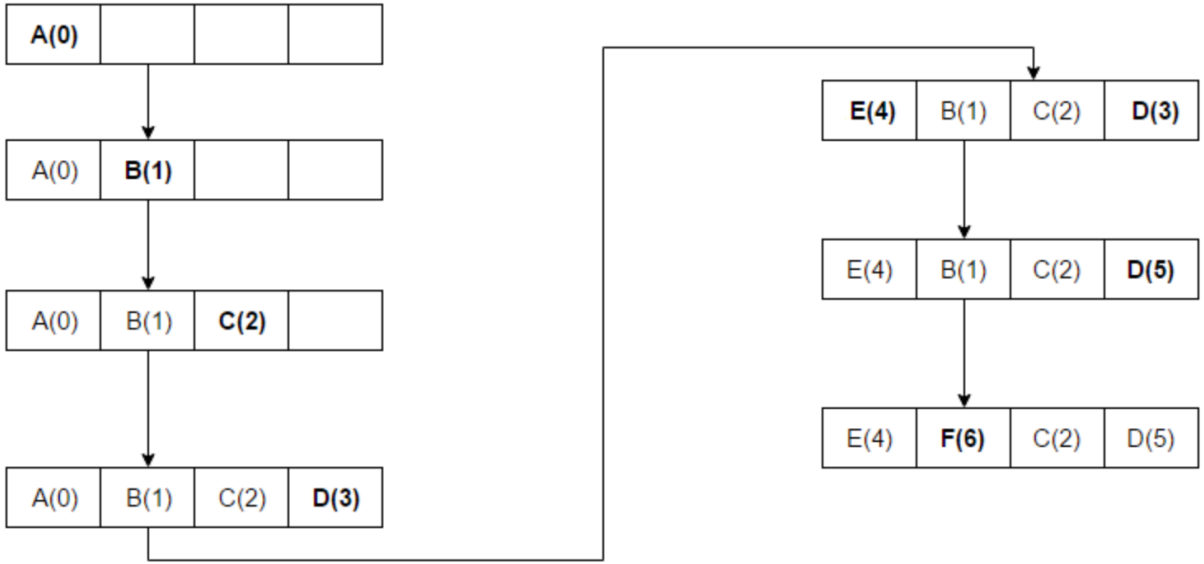
- Go through rooms to check availability
- If there is enough room left, confirm the reservation
- If there isn’t enough room left, throw exception

Copyright © www.jiuzhang.com

Challenge

- LRU Cache

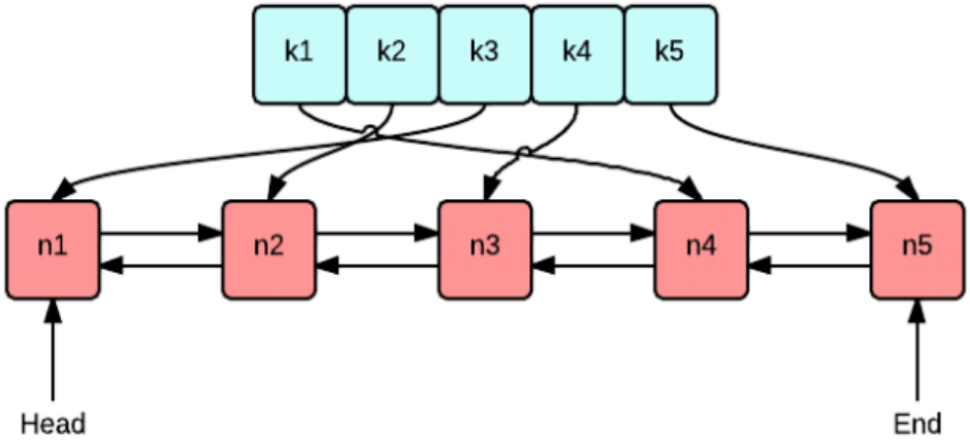
The access sequence for the below example is A B C D E D F.



Copyright © www.jiuzhang.com

Challenge

- LRU Cache



Copyright © www.jiuzhang.com

Challenge

九章算法

```java
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;
   }
}
```

Copyright © www.jiuzhang.com

Classes

Request

- Date startDate
- Date endDate

Hotel

- List<Room> rooms
- List<Reservation> reservations
- Map<Room, List<Date>> roomReservations
- LRUCache<Request, Map<RoomType, Set<Room>>> cache

+ Map<RoomType, int> handleSearchRequest(Request r)
- boolean isRequestAvailable(Request r, List<Date> dates)
+ Reservation makeReservation(ReservationRequest r)

<<Enumeration>>
RoomType

SINGLE
DOUBLE

Room

- RoomType type
- Boolean available

+ boolean isAvailable()

Reservation

- List<Room> rooms
- Date startDate
- Date endDate

ReservationRequest

- Date startDate
- Date endDate
- Map<RoomType, int> roomsNeeded

NotEnoughRoomForReservationException

Use cases

Search for available rooms
Make reservation
Cancel reservation

Copyright © www.jiuzhang.com

Use cases

Cancel reservation

1: Hotel system takes a reservation, and cancel it.

Copyright © www.jiuzhang.com

Classes

Request

- Date startDate
- Date endDate

Hotel

- List<Room> rooms
- List<Reservation> reservations
- Map<Room, List<Date>> roomReservations
- LRUCache<Request, Map<RoomType, Set<Room>>> cache

+ Map<RoomType, int> handleSearchRequest(Request r)
- boolean isRequestAvailable(Request r, List<Date> dates)
+ Reservation makeReservation(ReservationRequest r)
+ void cancelReservation(Reservation r)

<<Enumeration>>
RoomType

SINGLE
DOUBLE

Room

- RoomType type
- Boolean available

+ boolean isAvailable()

Reservation

- List<Room> rooms
- Date startDate
- Date endDate

ReservationRequest

- Date startDate
- Date endDate
- Map<RoomType, int> roomsNeeded

NotEnoughRoomForReservationException

Use cases

Search for available rooms
Make reservation
Cancel reservation

Copyright © www.jiuzhang.com

Classes – Final view

Request

- Date startDate
- Date endDate

Hotel

- List<Room> rooms
- List<Reservation> reservations
- Map<Room, List<Date>> roomReservations
- LRUCache<Request, Map<RoomType, Set<Room>>> cache

+ Map<RoomType, int> handleSearchRequest(Request r)
- boolean isRequestAvailable(Request r, List<Date> dates)
+ Reservation makeReservation(ReservationRequest r)
+ void cancelReservation(Reservation r)

<<Enumeration>>
RoomType

SINGLE
DOUBLE

Room

- RoomType type
- Boolean available

+ boolean isAvailable()

Reservation

- List<Room> rooms
- Date startDate
- Date endDate

ReservationRequest

- Date startDate
- Date endDate
- Map<RoomType, int> roomsNeeded

NotEnoughRoomForReservationException

Use cases

Search for available rooms
Make reservation
Cancel reservation

Copyright © www.jiuzhang.com

Challenge

面试官：

Can you extend your design to a booking.com like system?

Copyright © www.jiuzhang.com

Challenge

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

Copyright © www.jiuzhang.com

Challenge

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

Search criteria A -> Search() -> List<Result_A> -> Select() -> Search criteria B -> Search() -> List<Result_B> -> Select() -> Receipt

Copyright © www.jiuzhang.com

Challenge

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

Search criteria A -> Search() -> List<Result_A> -> Select() -> Search criteria B -> Search() -> List<Result_B> -> Select() -> Receipt

Search criteria A : 10/30 -11/5 Boston

Copyright © www.jiuzhang.com

Challenge

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

Search criteria A -> Search() -> List<Result_A> -> Select() -> Search criteria B -> Search() -> List<Result_B> -> Select() -> Receipt

Search criteria A : 10/30 -11/5 Boston

BookingSystem.Search()

Copyright © www.jiuzhang.com

Challenge

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

Search criteria A -> Search() -> List<Result_A> -> Select() -> Search criteria B -> Search() -> List<Result_B> -> Select() -> Receipt

Search criteria A : 10/30 -11/5 Boston

BookingSystem.Search()

List<Result_A>: Hotel_1, Hotel_2,…

Copyright © www.jiuzhang.com

Challenge

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

Search criteria A -> Search() -> List<Result_A> -> Select() -> Search criteria B -> Search() -> List<Result_B> -> Select() -> Receipt

Search criteria A : 10/30 -11/5 Boston

BookingSystem.Search()

List<Result_A>: Hotel_1, Hotel_2,…

BookingSystem.Select()

Copyright © www.jiuzhang.com

Challenge

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

Search criteria A -> Search() -> List<Result_A> -> Select() -> Search criteria B -> Search() -> List<Result_B> -> Select() -> Receipt

Search criteria A : 10/30 -11/5 Boston

Search criteria B : 10/30 -11/5

BookingSystem.Search()

List<Result_A>: Hotel_1, Hotel_2,…

BookingSystem.Select()

Copyright © www.jiuzhang.com

Challenge

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

Search criteria A -> Search() -> List<Result_A> -> Select() -> Search criteria B -> Search() -> List<Result_B> -> Select() -> Receipt

Search criteria A : 10/30 -11/5 Boston

BookingSystem.Search()

List<Result_A>: Hotel_1, Hotel_2,…

BookingSystem.Select()

Search criteria B : 10/30 -11/5

Hotel_1.Search()

Copyright © www.jiuzhang.com

Challenge

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

Search criteria A -> Search() -> List<Result_A> -> Select() -> Search criteria B -> Search() -> List<Result_B> -> Select() -> Receipt

Search criteria A : 10/30 -11/5 Boston

BookingSystem.Search()

List<Result_A>: Hotel_1, Hotel_2,…

BookingSystem.Select()

Search criteria B : 10/30 -11/5

Hotel_1.Search()

List<Result_A>: RoomType_1, …

Copyright © www.jiuzhang.com

Challenge

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

Search criteria A -> Search() -> List<Result_A> -> Select() -> Search criteria B -> Search() -> List<Result_B> -> Select() -> Receipt

Search criteria A : 10/30 -11/5 Boston

BookingSystem.Search()

List<Result_A>: Hotel_1, Hotel_2,…

BookingSystem.Select()

Search criteria B : 10/30 -11/5

Hotel_1.Search()

List<Result_A>: RoomType_1, …

BookingSystem.Select()

Copyright © www.jiuzhang.com

Clarify

What

Copyright © www.jiuzhang.com

Clarify

What

Search criteria A

List<Result_A>

Search criteria B

List<Result_B>

Receipt

Copyright © www.jiuzhang.com

Clarify

What

Search criteria A

List<Result_A>

Search criteria B

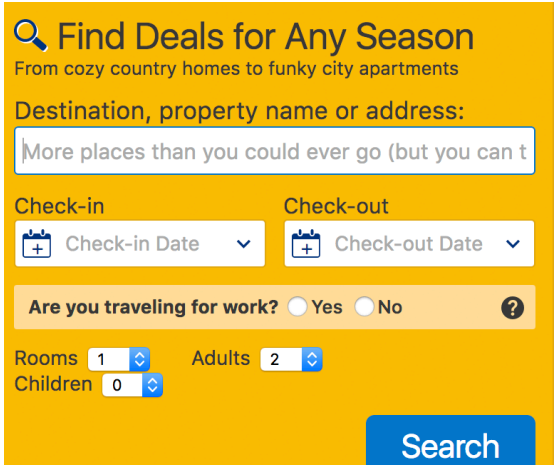
List<Result_B>

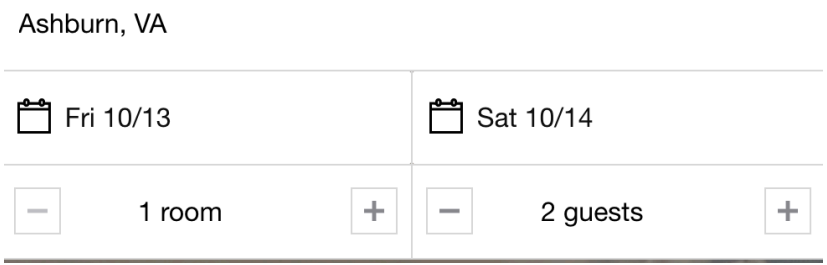
Receipt

Copyright © www.jiuzhang.com

Clarify

What





What are the search criteria for booking system ?

Copyright © www.jiuzhang.com

Clarify

针对本题：

用Start Date, End Date, Group size, City

Copyright © www.jiuzhang.com

Clarify

How

Search()

Select()

Search()

Select()

Copyright © www.jiuzhang.com

Clarify

How

Search()

Select()

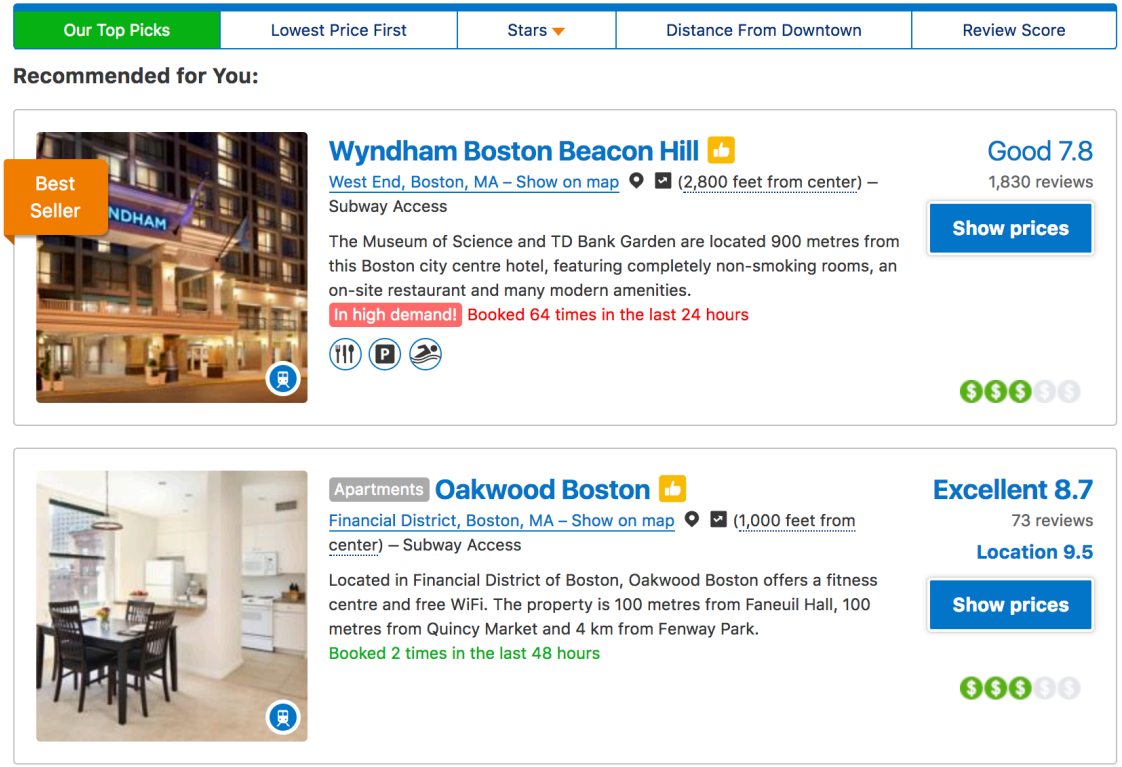
Search()

Select()

Copyright © www.jiuzhang.com

Clarify

How



- 哪些Hotel可以被放进List里？
- 按照什么顺序排序？

Copyright © www.jiuzhang.com

Clarify

九章算法

How

针对本题：

- 在同样的City就能放进List里
- Hotel在List里的顺序不重要

Copyright © www.jiuzhang.com

Core Object

九章算法

How

针对本题：

- 在同样的City就能放进List里
- Hotel在List里的顺序不重要

Copyright © www.jiuzhang.com

Core object

Request

- Date startDate
- Date endDate

Hotel

- List<Room> rooms
- List<Reservation> reservations
- Map<Room, List<Date>> roomReservations
- LRUCache<Request, Map<RoomType, Set<Room>>> cache

+ Map<RoomType, int> handleSearchRequest(Request r)
- boolean isRequestAvailable(Request r, List<Date> dates)
+ Reservation makeReservation(ReservationRequest r)
+ void cancelReservation(Reservation r)

<<Enumeration>>
RoomType

SINGLE
DOUBLE

Room

- RoomType type
- Boolean available

+ boolean isAvailable()

Reservation

- List<Room> rooms
- Date startDate
- Date endDate

ReservationRequest

- Date startDate
- Date endDate
- Map<RoomType, int> roomsNeeded

NotEnoughRoomForReservationException

Use cases

Search for available rooms
Make reservation
Cancel reservation

Copyright © www.jiuzhang.com

Core object

Request

- Date startDate
- Date endDate

Hotel

- List<Room> rooms
- List<Reservation> reservations
- Map<Room, List<Date>> roomReservations
- LRUCache<Request, Map<RoomType, Set<Room>>> cache

+ Map<RoomType, int> handleSearchRequest(Request r)
- boolean isRequestAvailable(Request r, List<Date> dates)
+ Reservation makeReservation(ReservationRequest r)
+ void cancelReservation(Reservation r)

<<Enumeration>> RoomType

SINGLE
DOUBLE

Room

- RoomType type
- Boolean available

+ boolean isAvailable()

Reservation

- List<Room> rooms
- Date startDate
- Date endDate

BookingSystem

ReservationRequest

- Date startDate
- Date endDate
- Map<RoomType, int> roomsNeeded

NotEnoughRoomForReservationException

Use cases

Search for available rooms
Make reservation
Cancel reservation

Copyright © www.jiuzhang.com

Core object

九章算法

Request

- Date startDate
- Date endDate

Hotel

- List<Room> rooms
- List<Reservation> reservations
- Map<Room, List<Date>> roomReservations
- LRUCache<Request, Map<RoomType, Set<Room>>> cache

+ Map<RoomType, int> handleSearchRequest(Request r)
- boolean isRequestAvailable(Request r, List<Date> dates)
+ Reservation makeReservation(ReservationRequest r)
+ void cancelReservation(Reservation r)

<<Enumeration>>
RoomType

SINGLE
DOUBLE

Room

- RoomType type
- Boolean available

+ boolean isAvailable()

Reservation

- List<Room> rooms
- Date startDate
- Date endDate

BookingSystem

ReservationRequest

- Date startDate
- Date endDate
- Map<RoomType, int> roomsNeeded

SearchHotelRequest

NotEnoughRoomForReservationException

Use cases

Search for available rooms
Make reservation
Cancel reservation

Copyright © www.jiuzhang.com

Core object

Request

- Date startDate
- Date endDate

Hotel

- List<Room> rooms
- List<Reservation> reservations
- Map<Room, List<Date>> roomReservations
- LRUCache<Request, Map<RoomType, Set<Room>>> cache

+ Map<RoomType, int> handleSearchRequest(Request r)
- boolean isRequestAvailable(Request r, List<Date> dates)
+ Reservation makeReservation(ReservationRequest r)
+ void cancelReservation(Reservation r)

<<Enumeration>> RoomType

SINGLE
DOUBLE

Room

- RoomType type
- Boolean available

+ boolean isAvailable()

Reservation

- List<Room> rooms
- Date startDate
- Date endDate

BookingSystem

- List<Hotel> hotels

ReservationRequest

- Date startDate
- Date endDate
- Map<RoomType, int> roomsNeeded

SearchHotelRequest

NotEnoughRoomForReservationException

Use cases

Search for available rooms
Make reservation
Cancel reservation

Copyright © www.jiuzhang.com

Use cases

- BookingSystem

Copyright © www.jiuzhang.com

Use cases

- BookingSystem

- Search for hotels

Copyright © www.jiuzhang.com

Use cases

- BookingSystem

- Search for hotels
- Make reservation

Copyright © www.jiuzhang.com

Use cases

- BookingSystem

- Search for hotels
- Make reservation
- Cancel reservation

Copyright © www.jiuzhang.com

Class

Request

- Date startDate
- Date endDate

Hotel

- List<Room> rooms
- List<Reservation> reservations
- Map<Room, List<Date>> roomReservations
- LRUCache<Request, Map<RoomType, Set<Room>>> cache

+ Map<RoomType, int> handleSearchRequest(Request r)
- boolean isRequestAvailable(Request r, List<Date> dates)
+ Reservation makeReservation(ReservationRequest r)
+ void cancelReservation(Reservation r)

<<Enumeration>>
RoomType

SINGLE
DOUBLE

Room

- RoomType type
- Boolean available

+ boolean isAvailable()

Reservation

- List<Room> rooms
- Date startDate
- Date endDate

BookingSystem

- List<Hotel> hotels

ReservationRequest

- Date startDate
- Date endDate
- Map<RoomType, int> roomsNeeded

SearchHotelRequest

NotEnoughRoomForReservationException

Use cases

- Search for available rooms
- Make reservation
- Cancel reservation
- Search for hotels
- Make reservation (Booking System)
- Cancel reservation (Booking System)

Copyright © www.jiuzhang.com

Classes

- Search for hotels

- Based on request (start date + end date + city + group size)
- Check for all hotels in this city
- For such hotels, if this is enough capacity for group size during dates
- List all hotels that satisfy above criteria

Copyright © www.jiuzhang.com

Classes

- Search for hotels

- Based on request (start date + end date + city + group size)

Copyright © www.jiuzhang.com

Class

Request

- Date startDate
- Date endDate

Hotel

- List<Room> rooms
- List<Reservation> reservations
- Map<Room, List<Date>> roomReservations
- LRUCache<Request, Map<RoomType, Set<Room>>> cache

+ Map<RoomType, int> handleSearchRequest(Request r)
- boolean isRequestAvailable(Request r, List<Date> dates)
+ Reservation makeReservation(ReservationRequest r)
+ void cancelReservation(Reservation r)

<<Enumeration>> RoomType

SINGLE
DOUBLE

Room

- RoomType type
- Boolean available

+ boolean isAvailable()

Reservation

- List<Room> rooms
- Date startDate
- Date endDate

BookingSystem

- List<Hotel> hotels

ReservationRequest

- Date startDate
- Date endDate
- Map<RoomType, int> roomsNeeded

SearchHotelRequest

- Date startDate
- Date endDate
- String city
- Int groupSize

NotEnoughRoomForReservationException

Use cases

- Search for available rooms
- Make reservation
- Cancel reservation
- Search for hotels
- Make reservation (Booking System)
- Cancel reservation (Booking System)

Copyright © www.jiuzhang.com

Classes

- Search for hotels

- Based on request (start date + end date + city + group size)
- Check for all hotels in this city

Copyright © www.jiuzhang.com

Class

九章算法

Request

- Date startDate
- Date endDate

Hotel

- List<Room> rooms
- List<Reservation> reservations
- Map<Room, List<Date>> roomReservations
- LRUCache<Request, Map<RoomType, Set<Room>>> cache
- String city

+ Map<RoomType, int> handleSearchRequest(Request r)
- boolean isRequestAvailable(Request r, List<Date> dates)
+ Reservation makeReservation(ReservationRequest r)
+ void cancelReservation(Reservation r)

<<Enumeration>> RoomType

SINGLE
DOUBLE

Room

- RoomType type
- Boolean available

+ boolean isAvailable()

Reservation

- List<Room> rooms
- Date startDate
- Date endDate

BookingSystem

- List<Hotel> hotels

ReservationRequest

- Date startDate
- Date endDate
- Map<RoomType, int> roomsNeeded

SearchHotelRequest

- Date startDate
- Date endDate
- String city
- Int groupSize

NotEnoughRoomForReservationException

Use cases

- Search for available rooms
- Make reservation
- Cancel reservation
- Search for hotels
- Make reservation (Booking System)
- Cancel reservation (Booking System)

Copyright © www.jiuzhang.com

Class

九章算法

Request

- Date startDate
- Date endDate

Hotel

- List<Room> rooms
- List<Reservation> reservations
- Map<Room, List<Date>> roomReservations
- LRUCache<Request, Map<RoomType, Set<Room>>> cache
- String city

+ Map<RoomType, int> handleSearchRequest(Request r)
- boolean isRequestAvailable(Request r, List<Date> dates)
+ Reservation makeReservation(ReservationRequest r)
+ void cancelReservation(Reservation r)
+ boolean isSameCity(string city)

<<Enumeration>> RoomType

SINGLE
DOUBLE

Room

- RoomType type
- Boolean available

+ boolean isAvailable()

Reservation

- List<Room> rooms
- Date startDate
- Date endDate

BookingSystem

- List<Hotel> hotels

ReservationRequest

- Date startDate
- Date endDate
- Map<RoomType, int> roomsNeeded

SearchHotelRequest

- Date startDate
- Date endDate
- String city
- Int groupSize

NotEnoughRoomForReservationException

Use cases

- Search for available rooms
- Make reservation
- Cancel reservation
- Search for hotels
- Make reservation (Booking System)
- Cancel reservation (Booking System)

Copyright © www.jiuzhang.com

Classes

- Search for hotels

- Based on request (start date + end date + city + group size)
- Check for all hotels in this city
- For such hotels, if this is enough capacity for group size during dates

Copyright © www.jiuzhang.com

Class

九章算法

Request

- Date startDate
- Date endDate

Reservation

- List<Room> rooms
- Date startDate
- Date endDate

Hotel

- List<Room> rooms
- List<Reservation> reservations
- Map<Room, List<Date>> roomReservations
- LRUCache<Request, Map<RoomType, Set<Room>>> cache
- String city

+ Map<RoomType, int> handleSearchRequest(Request r)
- boolean isRequestAvailable(Request r, List<Date> dates)
+ Reservation makeReservation(ReservationRequest r)
+ void cancelReservation(Reservation r)
+ boolean isSameCity(string city)

<<Enumeration>> RoomType

SINGLE
DOUBLE

- Int capacity

Room

- RoomType type
- Boolean available

+ boolean isAvailable()

BookingSystem

- List<Hotel> hotels

ReservationRequest

- Date startDate
- Date endDate
- Map<RoomType, int> roomsNeeded

SearchHotelRequest

- Date startDate
- Date endDate
- String city
- Int groupSize

NotEnoughRoomForReservationException

Use cases

- Search for available rooms
- Make reservation
- Cancel reservation
- Search for hotels
- Make reservation (Booking System)
- Cancel reservation (Booking System)

Copyright © www.jiuzhang.com

Class

Request

- Date startDate
- Date endDate

Hotel

- List<Room> rooms
- List<Reservation> reservations
- Map<Room, List<Date>> roomReservations
- LRUCache<Request, Map<RoomType, Set<Room>>> cache
- String city

+ Map<RoomType, int> handleSearchRequest(Request r)
- boolean isRequestAvailable(Request r, List<Date> dates)
+ Reservation makeReservation(ReservationRequest r)
+ void cancelReservation(Reservation r)
+ boolean isSameCity(string city)
+ boolean isValidRequest(SearchHotelRequest r)

<<Enumeration>> RoomType

SINGLE
DOUBLE

- Int capacity

Room

- RoomType type
- Boolean available

+ boolean isAvailable()

Reservation

- List<Room> rooms
- Date startDate
- Date endDate

BookingSystem

- List<Hotel> hotels

ReservationRequest

- Date startDate
- Date endDate
- Map<RoomType, int> roomsNeeded

SearchHotelRequest

- Date startDate
- Date endDate
- String city
- Int groupSize

NotEnoughRoomForReservationException

Use cases

- Search for available rooms
- Make reservation
- Cancel reservation
- Search for hotels
- Make reservation (Booking System)
- Cancel reservation (Booking System)

Copyright © www.jiuzhang.com

Classes

- Search for hotels

- Based on request (start date + end date + city + group size)
- Check for all hotels in this city
- For such hotels, if this is enough capacity for group size during dates
- List all hotels that satisfy above criteria

Copyright © www.jiuzhang.com

Class

九章算法

Request

- Date startDate
- Date endDate

Hotel

- List<Room> rooms
- List<Reservation> reservations
- Map<Room, List<Date>> roomReservations
- LRUCache<Request, Map<RoomType, Set<Room>>> cache
- String city

+ Map<RoomType, int> handleSearchRequest(Request r)
- boolean isRequestAvailable(Request r, List<Date> dates)
+ Reservation makeReservation(ReservationRequest r)
+ void cancelReservation(Reservation r)
+ boolean isSameCity(string city)
+ boolean isValidRequest(SearchHotelRequest r)

<<Enumeration>> RoomType

SINGLE
DOUBLE

- Int capacity

Room

- RoomType type
- Boolean available

+ boolean isAvailable()

Reservation

- List<Room> rooms
- Date startDate
- Date endDate

BookingSystem

- List<Hotel> hotels

+ List<Hotel> searchHotel(SearchHotelRequest r)

ReservationRequest

- Date startDate
- Date endDate
- Map<RoomType, int> roomsNeeded

SearchHotelRequest

- Date startDate
- Date endDate
- String city
- Int groupSize

NotEnoughRoomForReservationException

Use cases

- Search for available rooms
- Make reservation
- Cancel reservation
- Search for hotels
- Make reservation (Booking System)
- Cancel reservation (Booking System)

Copyright © www.jiuzhang.com

Use case

- Make reservation

- Pick a hotel
- Send reservation request to that hotel

Copyright © www.jiuzhang.com

Class

Request

- Date startDate
- Date endDate

Hotel

- List<Room> rooms
- List<Reservation> reservations
- Map<Room, List<Date>> roomReservations
- LRUCache<Request, Map<RoomType, Set<Room>>> cache
- String city

+ Map<RoomType, int> handleSearchRequest(Request r)
- boolean isRequestAvailable(Request r, List<Date> dates)
+ Reservation makeReservation(ReservationRequest r)
+ void cancelReservation(Reservation r)
+ boolean isSameCity(string city)
+ boolean isValidRequest(SearchHotelRequest r)

<<Enumeration>> RoomType

SINGLE
DOUBLE

- Int capacity

Room

- RoomType type
- Boolean available

+ boolean isAvailable()

Reservation

- List<Room> rooms
- Date startDate
- Date endDate

BookingSystem

- List<Hotel> hotels

+ List<Hotel> searchHotel(SearchHotelRequest r)
+ Reservation makeReservation(Hotel h, ReservationRequest r)

ReservationRequest

- Date startDate
- Date endDate
- Map<RoomType, int> roomsNeeded

SearchHotelRequest

- Date startDate
- Date endDate
- String city
- Int groupSize

NotEnoughRoomForReservationException

Use cases

- Search for available rooms
- Make reservation
- Cancel reservation
- Search for hotels
- Make reservation (Booking System)
- Cancel reservation (Booking System)

Copyright © www.jiuzhang.com

Classes

- Cancel reservation

- Cancel reservation for a hotel

Copyright © www.jiuzhang.com

Class

Request

- Date startDate
- Date endDate

Hotel

- List<Room> rooms
- List<Reservation> reservations
- Map<Room, List<Date>> roomReservations
- LRUCache<Request, Map<RoomType, Set<Room>>> cache
- String city

+ Map<RoomType, int> handleSearchRequest(Request r)
- boolean isRequestAvailable(Request r, List<Date> dates)
+ Reservation makeReservation(ReservationRequest r)
+ void cancelReservation(Reservation r)
+ boolean isSameCity(string city)
+ boolean isValidRequest(SearchHotelRequest r)

<<Enumeration>> RoomType

SINGLE
DOUBLE

- Int capacity

Room

- RoomType type
- Boolean available

+ boolean isAvailable()

Reservation

- Hotel hotel
- List<Room> rooms
- Date startDate
- Date endDate

BookingSystem

- List<Hotel> hotels

+ List<Hotel> searchHotel(SearchHotelRequest r)
+ Reservation makeReservation(Hotel h, ReservationRequest r)
+ void cancelReservation(Reservation r)

ReservationRequest

- Date startDate
- Date endDate
- Map<RoomType, int> roomsNeeded

SearchHotelRequest

- Date startDate
- Date endDate
- String city
- Int groupSize

NotEnoughRoomForReservationException

Use cases

Search for available rooms
Make reservation
Cancel reservation
Search for hotels
Make reservation (Booking System)
Cancel reservation (Booking System)

Copyright © www.jiuzhang.com

Class – Final view

九章算法

Request

- Date startDate
- Date endDate

Hotel

- List<Room> rooms
- List<Reservation> reservations
- Map<Room, List<Date>> roomReservations
- LRUCache<Request, Map<RoomType, Set<Room>>> cache
- String city

+ Map<RoomType, int> handleSearchRequest(Request r)
- boolean isRequestAvailable(Request r, List<Date> dates)
+ Reservation makeReservation(ReservationRequest r)
+ void cancelReservation(Reservation r)
+ boolean isSameCity(string city)
+ boolean isValidRequest(SearchHotelRequest r)

<<Enumeration>> RoomType

SINGLE
DOUBLE

- Int capacity

Room

- RoomType type
- Boolean available

+ boolean isAvailable()

Reservation

- Hotel hotel
- List<Room> rooms
- Date startDate
- Date endDate

BookingSystem

- List<Hotel> hotels

+ List<Hotel> searchHotel(SearchHotelRequest r)
+ Reservation makeReservation(Hotel h, ReservationRequest r)
+ void cancelReservation(Reservation r)

ReservationRequest

- Date startDate
- Date endDate
- Map<RoomType, int> roomsNeeded

SearchHotelRequest

- Date startDate
- Date endDate
- String city
- Int groupSize

NotEnoughRoomForReservationException

Use cases

Search for available rooms
Make reservation
Cancel reservation
Search for hotels
Make reservation (Booking System)
Cancel reservation (Booking System)

Copyright © www.jiuzhang.com

Challenge

- Can you add payment in your system?

Copyright © www.jiuzhang.com

Challenge

- Can you add payment in your system?

- Need to get price for each reservation
- Need to take a payment method

Copyright © www.jiuzhang.com

Challenge

- Can you add payment in your system?

- Need to get price for each reservation

Copyright © www.jiuzhang.com

Class

Request

- Date startDate
- Date endDate

Hotel

- List<Room> rooms
- List<Reservation> reservations
- Map<Room, List<Date>> roomReservations
- LRUCache<Request, Map<RoomType, Set<Room>>> cache
- String city

+ Map<RoomType, int> handleSearchRequest(Request r)
- boolean isRequestAvailable(Request r, List<Date> dates)
+ Reservation makeReservation(ReservationRequest r)
+ void cancelReservation(Reservation r)
+ boolean isSameCity(string city)
+ boolean isValidRequest(SearchHotelRequest r)

<<Enumeration>> RoomType

SINGLE
DOUBLE

- Int capacity
- Float price

Room

- RoomType type
- Boolean available

+ boolean isAvailable()

Reservation

- Hotel hotel
- List<Room> rooms
- Date startDate
- Date endDate

BookingSystem

- List<Hotel> hotels

+ List<Hotel> searchHotel(SearchHotelRequest r)
+ Reservation makeReservation(Hotel h, ReservationRequest r)
+ void cancelReservation(Reservation r)

ReservationRequest

- Date startDate
- Date endDate
- Map<RoomType, int> roomsNeeded

SearchHotelRequest

- Date startDate
- Date endDate
- String city
- Int groupSize

NotEnoughRoomForReservationException

Use cases

Search for available rooms
Make reservation
Cancel reservation
Search for hotels
Make reservation (Booking System)
Cancel reservation (Booking System)

Copyright © www.jiuzhang.com

Enum code sample

九章算法

```
public enum RoomType
{
    SINGLE(1, 129),
    DOUBLE(2, 199);

    private int capacity;
    private float price;

    RoomType(int capacity, float price)
    {
        this.capacity = capacity;
        this.price = price;
    }

    public int getCapacity()
    {
        return capacity;
    }

    public float getPrice()
    {
        return price;
    }
}
```

Copyright © www.jiuzhang.com

Class

Request

- Date startDate
- Date endDate

Hotel

- List<Room> rooms
- List<Reservation> reservations
- Map<Room, List<Date>> roomReservations
- LRUCache<Request, Map<RoomType, Set<Room>>> cache
- String city

+ Map<RoomType, int> handleSearchRequest(Request r)
- boolean isRequestAvailable(Request r, List<Date> dates)
+ Reservation makeReservation(ReservationRequest r)
+ void cancelReservation(Reservation r)
+ boolean isSameCity(string city)
+ boolean isValidRequest(SearchHotelRequest r)

<<Enumeration>> RoomType

SINGLE
DOUBLE

- Int capacity
- Float price

Room

- RoomType type
- Boolean available

+ boolean isAvailable()

Reservation

- Hotel hotel
- List<Room> rooms
- Date startDate
- Date endDate

BookingSystem

- List<Hotel> hotels

+ List<Hotel> searchHotel(SearchHotelRequest r)
+ Reservation makeReservation(Hotel h, ReservationRequest r)
+ void cancelReservation(Reservation r)
+ void payByPaypal (Reservation r)

ReservationRequest

- Date startDate
- Date endDate
- Map<RoomType, int> roomsNeeded

SearchHotelRequest

- Date startDate
- Date endDate
- String city
- Int groupSize

NotEnoughRoomForReservationException

Use cases

Search for available rooms
Make reservation
Cancel reservation
Search for hotels
Make reservation (Booking System)
Cancel reservation (Booking System)

Copyright © www.jiuzhang.com

Class

Request

- Date startDate
- Date endDate

Hotel

- List<Room> rooms
- List<Reservation> reservations
- Map<Room, List<Date>> roomReservations
- LRUCache<Request, Map<RoomType, Set<Room>>> cache
- String city

+ Map<RoomType, int> handleSearchRequest(Request r)
- boolean isRequestAvailable(Request r, List<Date> dates)
+ Reservation makeReservation(ReservationRequest r)
+ void cancelReservation(Reservation r)
+ boolean isSameCity(string city)
+ boolean isValidRequest(SearchHotelRequest r)

<<Enumeration>> RoomType

SINGLE
DOUBLE

- Int capacity
- Float price

Room

- RoomType type
- Boolean available

+ boolean isAvailable()

Reservation

- Hotel hotel
- List<Room> rooms
- Date startDate
- Date endDate

BookingSystem

- List<Hotel> hotels

+ List<Hotel> searchHotel(SearchHotelRequest r)
+ Reservation makeReservation(Hotel h, ReservationRequest r)
+ void cancelReservation(Reservation r)
+ void payByPaypal (Reservation r)
+ void payByCreditCard (Reservation r)

ReservationRequest

- Date startDate
- Date endDate
- Map<RoomType, int> roomsNeeded

SearchHotelRequest

- Date startDate
- Date endDate
- String city
- Int groupSize

NotEnoughRoomForReservationException

Use cases

Search for available rooms
Make reservation
Cancel reservation
Search for hotels
Make reservation (Booking System)
Cancel reservation (Booking System)

Copyright © www.jiuzhang.com

Good practice:

- Strategy Pattern

Copyright © www.jiuzhang.com

Class

Request

- Date startDate
- Date endDate

Hotel

- List<Room> rooms
- List<Reservation> reservations
- Map<Room, List<Date>> roomReservations
- LRUCache<Request, Map<RoomType, Set<Room>>> cache
- String city

+ Map<RoomType, int> handleSearchRequest(Request r)
- boolean isRequestAvailable(Request r, List<Date> dates)
+ Reservation makeReservation(ReservationRequest r)
+ void cancelReservation(Reservation r)
+ boolean isSameCity(string city)
+ boolean isValidRequest(SearchHotelRequest r)

<<Enumeration>> RoomType

SINGLE
DOUBLE

- Int capacity
- Float price

Room

- RoomType type
- Boolean available

+ boolean isAvailable()

Reservation

- Hotel hotel
- List<Room> rooms
- Date startDate
- Date endDate

BookingSystem

- List<Hotel> hotels

+ List<Hotel> searchHotel(SearchHotelRequest r)
+ Reservation makeReservation(Hotel h, ReservationRequest r)
+ void cancelReservation(Reservation r)
+ void payByPaypal (Reservation r)
+ void payByCreditCard (Reservation r)

ReservationRequest

- Date startDate
- Date endDate
- Map<RoomType, int> roomsNeeded

SearchHotelRequest

- Date startDate
- Date endDate
- String city
- Int groupSize

NotEnoughRoomForReservationException

Use cases

Search for available rooms
Make reservation
Cancel reservation
Search for hotels
Make reservation (Booking System)
Cancel reservation (Booking System)

Copyright © www.jiuzhang.com

Class

Request

- Date startDate
- Date endDate

Hotel

- List<Room> rooms
- List<Reservation> reservations
- Map<Room, List<Date>> roomReservations
- LRUCache<Request, Map<RoomType, Set<Room>>> cache
- String city

+ Map<RoomType, int> handleSearchRequest(Request r)
- boolean isRequestAvailable(Request r, List<Date> dates)
+ Reservation makeReservation(ReservationRequest r)
+ void cancelReservation(Reservation r)
+ boolean isSameCity(string city)
+ boolean isValidRequest(SearchHotelRequest r)

<<Enumeration>>
RoomType

SINGLE
DOUBLE

- Int capacity
- Float price

Room

- RoomType type
- Boolean available

+ boolean isAvailable()

Reservation

- Hotel hotel
- List<Room> rooms
- Date startDate
- Date endDate

BookingSystem

- List<Hotel> hotels

+ List<Hotel> searchHotel(SearchHotelRequest r)
+ Reservation makeReservation(Hotel h, ReservationRequest r)
+ void cancelReservation(Reservation r)

ReservationRequest

- Date startDate
- Date endDate
- Map<RoomType, int> roomsNeeded

SearchHotelRequest

- Date startDate
- Date endDate
- String city
- Int groupSize

<<interface>>
PaymentStrategy

NotEnoughRoomForReservationException

Use cases

Search for available rooms
Make reservation
Cancel reservation
Search for hotels
Make reservation (Booking System)
Cancel reservation (Booking System)

Copyright © www.jiuzhang.com

Class

九章算法

Request

- Date startDate
- Date endDate

Hotel

- List<Room> rooms
- List<Reservation> reservations
- Map<Room, List<Date>> roomReservations
- LRUCache<Request, Map<RoomType, Set<Room>>> cache
- String city

+ Map<RoomType, int> handleSearchRequest(Request r)
- boolean isRequestAvailable(Request r, List<Date> dates)
+ Reservation makeReservation(ReservationRequest r)
+ void cancelReservation(Reservation r)
+ boolean isSameCity(string city)
+ boolean isValidRequest(SearchHotelRequest r)

<<Enumeration>>
RoomType

SINGLE
DOUBLE

- Int capacity
- Float price

Room

- RoomType type
- Boolean available

+ boolean isAvailable()

Reservation

- Hotel hotel
- List<Room> rooms
- Date startDate
- Date endDate

BookingSystem

- List<Hotel> hotels

+ List<Hotel> searchHotel(SearchHotelRequest r)
+ Reservation makeReservation(Hotel h, ReservationRequest r)
+ void cancelReservation(Reservation r)

ReservationRequest

- Date startDate
- Date endDate
- Map<RoomType, int> roomsNeeded

SearchHotelRequest

- Date startDate
- Date endDate
- String city
- Int groupSize

<<interface>>
PaymentStrategy

+ void pay(Reservation r)

Use cases

Search for available rooms
Make reservation
Cancel reservation
Search for hotels
Make reservation (Booking System)
Cancel reservation (Booking System)

NotEnoughRoomForReservationException

Copyright © www.jiuzhang.com

Class

Request

- Date startDate
- Date endDate

Hotel

- List<Room> rooms
- List<Reservation> reservations
- Map<Room, List<Date>> roomReservations
- LRUCache<Request, Map<RoomType, Set<Room>>> cache
- String city

+ Map<RoomType, int> handleSearchRequest(Request r)
- boolean isRequestAvailable(Request r, List<Date> dates)
+ Reservation makeReservation(ReservationRequest r)
+ void cancelReservation(Reservation r)
+ boolean isSameCity(string city)
+ boolean isValidRequest(SearchHotelRequest r)

<<Enumeration>>
RoomType

SINGLE
DOUBLE

- Int capacity
- Float price

Room

- RoomType type
- Boolean available

+ boolean isAvailable()

Reservation

- Hotel hotel
- List<Room> rooms
- Date startDate
- Date endDate

BookingSystem

- List<Hotel> hotels
- PaymentStrategy strategy

+ List<Hotel> searchHotel(SearchHotelRequest r)
+ Reservation makeReservation(Hotel h, ReservationRequest r)
+ void cancelReservation(Reservation r)

ReservationRequest

- Date startDate
- Date endDate
- Map<RoomType, int> roomsNeeded

SearchHotelRequest

- Date startDate
- Date endDate
- String city
- Int groupSize

<<interface>>
PaymentStrategy

+ void pay(Reservation r)

Use cases

Search for available rooms
Make reservation
Cancel reservation
Search for hotels
Make reservation (Booking System)
Cancel reservation (Booking System)

NotEnoughRoomForReservationException

Copyright © www.jiuzhang.com

Class

Request

- Date startDate
- Date endDate

Hotel

- List<Room> rooms
- List<Reservation> reservations
- Map<Room, List<Date>> roomReservations
- LRUCache<Request, Map<RoomType, Set<Room>>> cache
- String city

+ Map<RoomType, int> handleSearchRequest(Request r)
- boolean isRequestAvailable(Request r, List<Date> dates)
+ Reservation makeReservation(ReservationRequest r)
+ void cancelReservation(Reservation r)
+ boolean isSameCity(string city)
+ boolean isValidRequest(SearchHotelRequest r)

<<Enumeration>>
RoomType

SINGLE
DOUBLE

- Int capacity
- Float price

Room

- RoomType type
- Boolean available

+ boolean isAvailable()

Reservation

- Hotel hotel
- List<Room> rooms
- Date startDate
- Date endDate

BookingSystem

- List<Hotel> hotels
- PaymentStrategy strategy

+ List<Hotel> searchHotel(SearchHotelRequest r)
+ Reservation makeReservation(Hotel h, ReservationRequest r)
+ void cancelReservation(Reservation r)
+ void setStrategy(Payment strategy)

ReservationRequest

- Date startDate
- Date endDate
- Map<RoomType, int> roomsNeeded

SearchHotelRequest

- Date startDate
- Date endDate
- String city
- Int groupSize

<<interface>>
PaymentStrategy

+ void pay(Reservation r)

NotEnoughRoomForReservationException

Use cases

Search for available rooms
Make reservation
Cancel reservation
Search for hotels
Make reservation (Booking System)
Cancel reservation (Booking System)

Copyright © www.jiuzhang.com

Class

Request

- Date startDate
- Date endDate

Hotel

- List<Room> rooms
- List<Reservation> reservations
- Map<Room, List<Date>> roomReservations
- LRUCache<Request, Map<RoomType, Set<Room>>> cache
- String city

+ Map<RoomType, int> handleSearchRequest(Request r)
- boolean isRequestAvailable(Request r, List<Date> dates)
+ Reservation makeReservation(ReservationRequest r)
+ void cancelReservation(Reservation r)
+ boolean isSameCity(string city)
+ boolean isValidRequest(SearchHotelRequest r)

<<Enumeration>>
RoomType

SINGLE
DOUBLE

- Int capacity
- Float price

Room

- RoomType type
- Boolean available

+ boolean isAvailable()

Reservation

- Hotel hotel
- List<Room> rooms
- Date startDate
- Date endDate

BookingSystem

- List<Hotel> hotels
- PaymentStrategy strategy

+ List<Hotel> searchHotel(SearchHotelRequest r)
+ Reservation makeReservation(Hotel h, ReservationRequest r)
+ void cancelReservation(Reservation r)
+ void setStrategy(Payment strategy)
+ void makePayment(Reservation r)

ReservationRequest

- Date startDate
- Date endDate
- Map<RoomType, int> roomsNeeded

SearchHotelRequest

- Date startDate
- Date endDate
- String city
- Int groupSize

<<interface>>
PaymentStrategy

+ void pay(Reservation r)

NotEnoughRoomForReservationException

Use cases

Search for available rooms
Make reservation
Cancel reservation
Search for hotels
Make reservation (Booking System)
Cancel reservation (Booking System)

Copyright © www.jiuzhang.com

Q & A

扫描二维码关注微信/微博
获取最新面试题及权威解答

微信: ninechapter
知乎专栏： http://zhuanlan.zhihu.com/jiuzhang
微博: http://www.weibo.com/ninechapter
官网: www.jiuzhang.com

Copyright © www.jiuzhang.com