叶钱炜、孙国道、夏佳佳、陈旻昊

19、迭代子模式（Iterator Pattern）

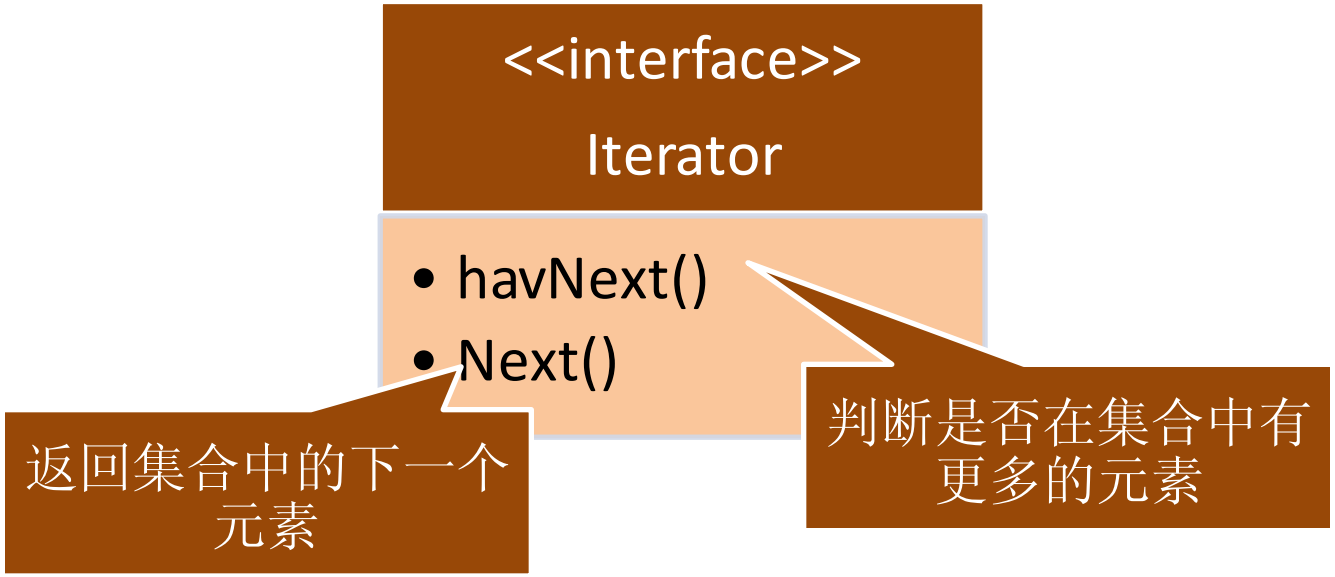
ITERATOR—我爱上了Mary，不顾一切的向她求婚。
Mary："想要我跟你结婚，得答应我的条件"
我："什么条件我都答应，你说吧"
Mary："我看上了那个一克拉的钻石"
我："我买，我买，还有吗？"
Mary："我看上了湖边的那栋别墅"
我："我买，我买，还有吗？"
Mary："我看上那辆法拉利跑车"
我脑袋嗡的一声，坐在椅子上，一咬牙："我买，我买，还有吗？"
.....

迭代子模式：迭代子模式可以顺序访问一个聚集中的元素而不必暴露聚集的内部表象。多个对象聚在一起形成的总体称之为聚集，聚集对象是能够包容一组对象的容器对象。迭代子模式将迭代逻辑封装到一个独立的子对象中，从而与聚集本身隔开。迭代子模式简化了聚集的界面。每一个聚集对象都可以有一个或一个以上的迭代子对象，每一个迭代子的迭代状态可以是彼此独立的。迭代算法可以独立于聚集角色变化。

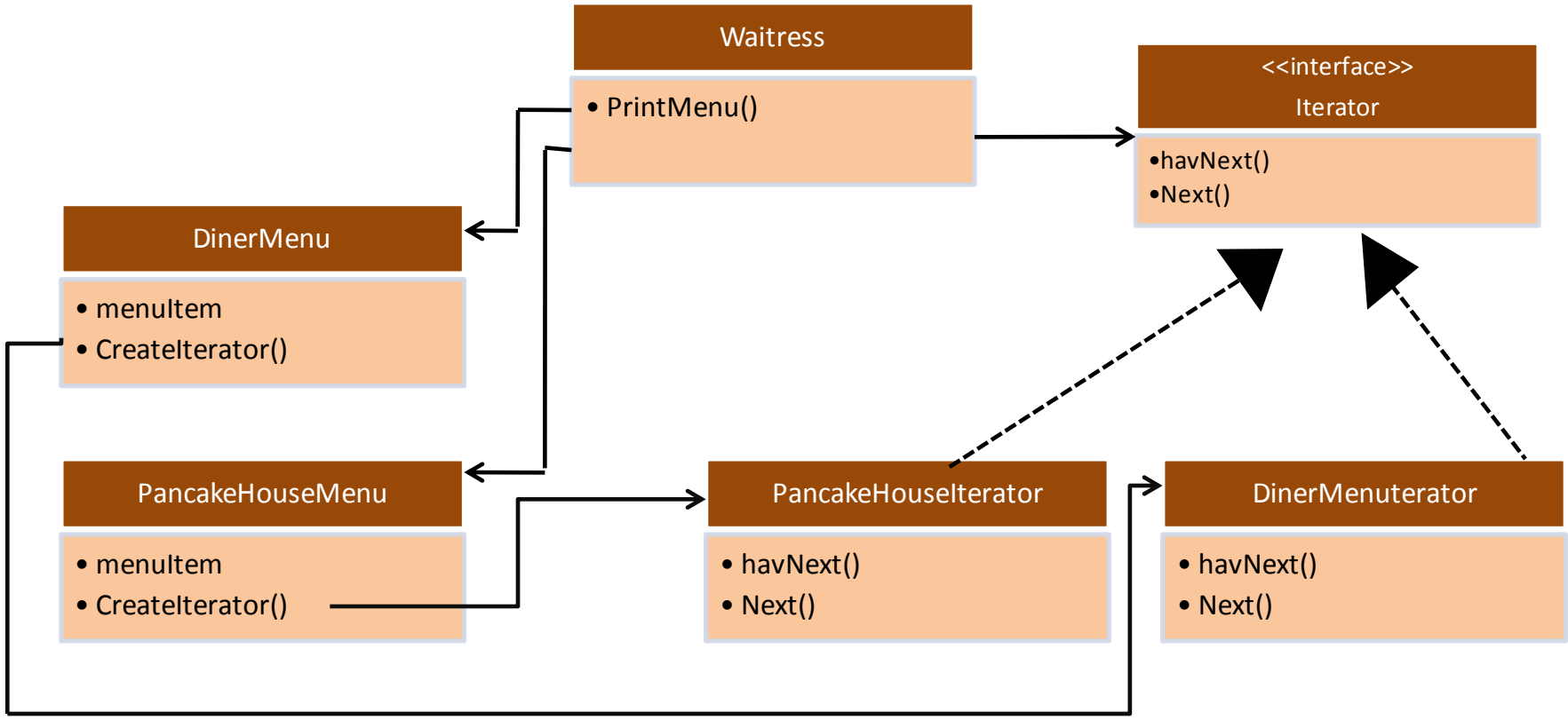
什么是迭代子模式？

提供一种方法顺序访问一个聚合对象中的各个元素，而又不暴露其内部表示。

迭代子模式依赖于一个迭代器接口，有了这个接口后，就可以为各种对象集合（数组、列表、散列表）实现迭代器。于是客户可以直接使用这个迭代器访问这些集合，而不需要关心迭代器是如何遍历这些集合的。



问题的类关系图


Waitress
• PrintMenu()
<<interface>>
Iterator
•havNext()
•Next()
DinerMenu
• menuItem
• CreateIterator()
PancakeHouseMenu
• menuItem
• CreateIterator()
PancakeHouseIterator
• havNext()
• Next()
DinerMenuterator
• havNext()
• Next()

迭代子模式由以下角色组成：

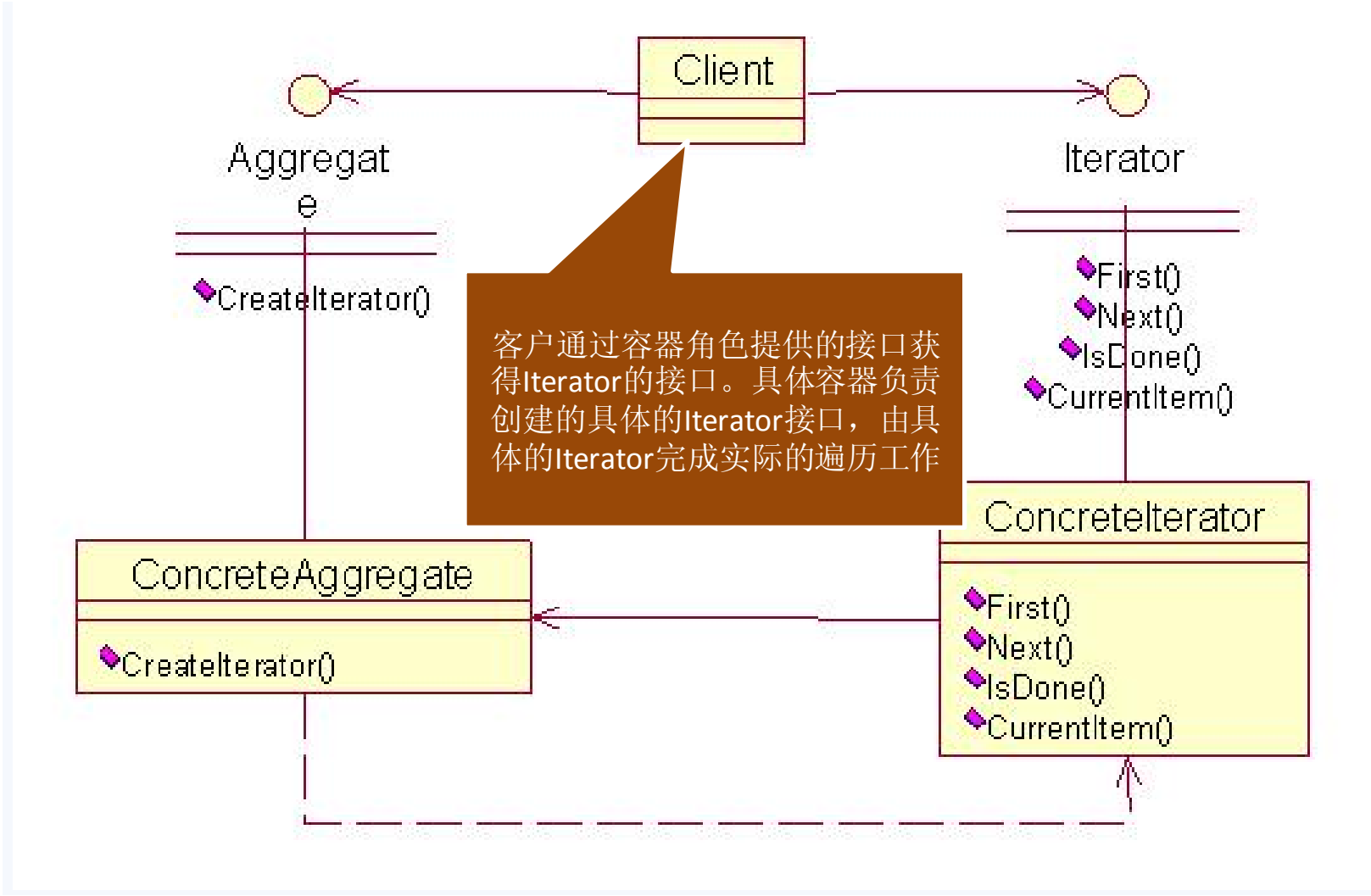
1) 迭代器角色（Iterator）：迭代器角色负责定义访问和遍历元素的接口。例子中的迭代器接口Iterator。

2) 具体迭代器角色（Concrete Iterator）：具体迭代器角色要实现迭代器接口，并要记录遍历中的当前位置。例子中的DinerMenuIterator和PancakeHouseIterator。

3) 容器角色（Container）：容器角色负责提供创建具体迭代器角色的接口。

4) 具体容器角色（Concrete Container）：具体容器角色实现创建具体迭代器角色的接口——这个具体迭代器角色于该容器的结构相关。例子中的DinerMenu和PancakeHouse提供的CreateIterator()接口。

一个迭代子模式类图


Client
Aggregat
e
CreateIterator()
Iterator
First()
Next()
IsDone()
CurrentItem()
客户通过容器角色提供的接口获得Iterator的接口。具体容器负责创建的具体的Iterator接口，由具体的Iterator完成实际的遍历工作
ConcreteAggregate
CreateIterator()
ConcreteIterator
First()
Next()
IsDone()
CurrentItem()

迭代子模式的优点

- 迭代器让我们能游走于聚合内的每个元素，而不暴露其内部表示。
- 使用户能以统一的方法访问聚类中的每一个对象，不管这个聚类中的元素是由数组或者ArraryList（或者其他）组成的。
- 把遍历的任务交迭代器而不交给聚类，不仅让聚类的接口变得简洁，也能使聚类能更专注于它应该专注的事情上