

10、装饰模式

Presented by 马庆 沈月云 彭怡然 姚敏

DECORATOR—Mary过完轮到Sarly过生日，还是不要叫她自己挑了，不然这个月伙食费肯定玩完，拿出我去年在华山顶上照的照片，在背面写上“最好的的礼物，就是爱你的Fita”，再到街上礼品店买了个像框（卖礼品的MM也很漂亮哦），再找隔壁搞美术设计的Mike设计了一个漂亮的盒子装起来.....，我们都是Decorator，最终都在修饰我这个人呀，怎么样，看懂了吗？

装饰模式：装饰模式以对客户端透明的方式扩展对象的功能，是继承关系的一个替代方案，提供比继承更多的灵活性。动态给一个对象增加功能，这些功能可以再动态的撤消。增加由一些基本功能的排列组合而产生的非常大量的功能。

1. 装饰模式介绍



意图

动态地给一个对象添加一些额外的职责。就增加功能来说，Decorator模式比生成子类更为灵活

工作原理

可以创建始于Decorator对象（负责新功能对象）终于原对象的一个对象“链”

1. 装饰模式介绍

- 装饰模式的定义
 - 装饰模式可以动态的给一个对象附加一些功能。
 - 使用装饰模式扩展功能不会产生类爆炸。它采用的是合成方式，比继承方式更加灵活。
- 装饰模式要解决的问题
 - 提供一种修改类的行为，而避免创建众多的派生类的途径。

3. 讨论

❖ 面向对象设计的开-闭原则

❖ 类应该对扩展开放，对修改封闭。

- 目的是在不需修改已有代码的情况下方便的扩展类的功能。按照这一原则设计的系统具有以下优点：
 - 具有一定的适应性和灵活性。
 - 具有一定的稳定性和延续性。

❖ “对扩展开放，对修改封闭。”这听起来来自相矛盾啊！有哪些方法可以不修改已有代码但又能扩展其功能？

❖ 何时使用开闭原则？

3. 讨论

❖ 何谓装饰？如何装饰？

- 所谓装饰就是将一个对象包装起来。
- 在程序上也就是让一个对象**a**包含另一个对象**b**。
- **a**对应的类**A**是“装饰类”，**b**对应的类**B**是被装饰类。
- 如果要想让装饰可以重复，且不必考虑次序，那么装饰的要点是装饰者和被装饰者具有相同的类型（有共同的父类）。

3. 讨论

首先生成一个**DarkRoast**对象

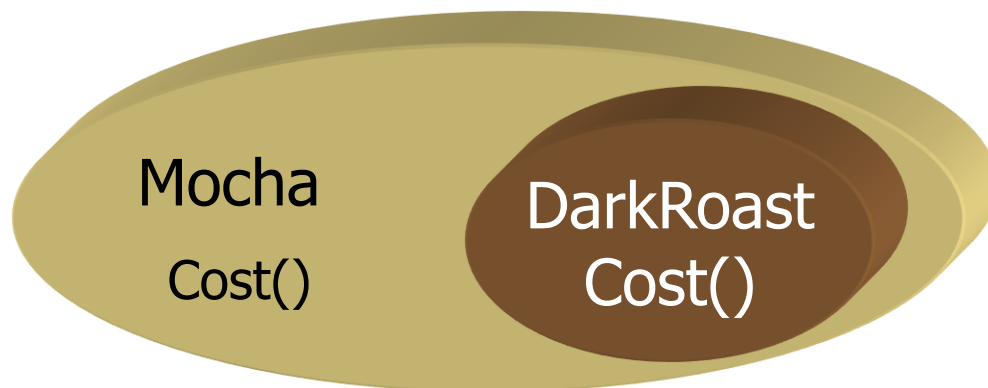


DarkRoast继承了**Beverage**，拥有一个计算饮料价格的方法**cost()**。

3. 讨论

然后

顾客想要mocha，
所以我们创建一个Mocha对象，
并用它包装DarkRoast.

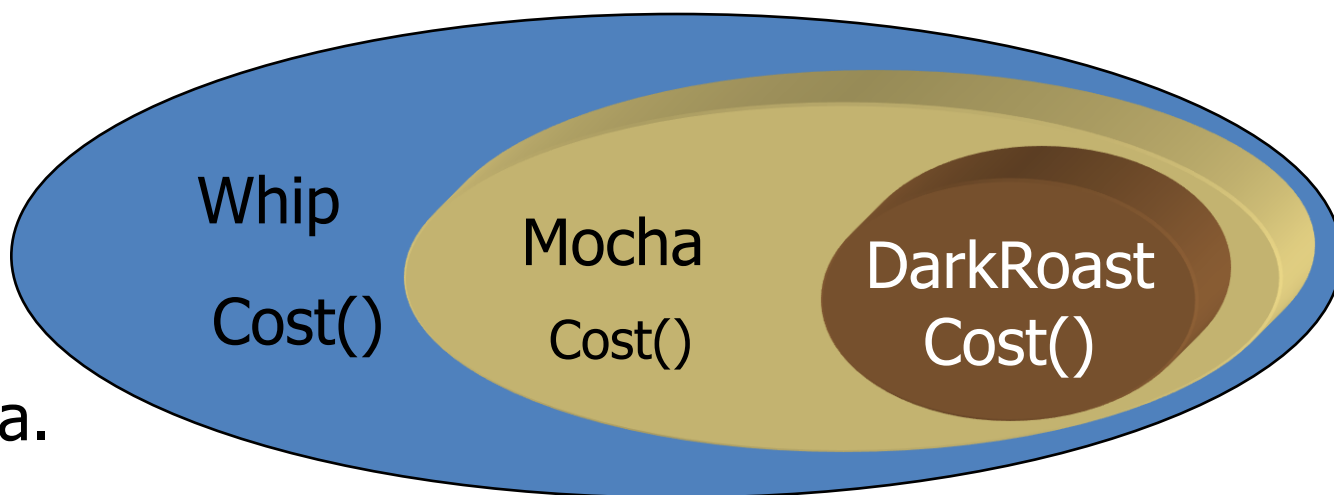


Mocha对象是装饰者，他与被它装饰的对象DarkRoast具有相同的类型（是Beverage的子类），也有一个cost()方法。

3. 讨论

再然后

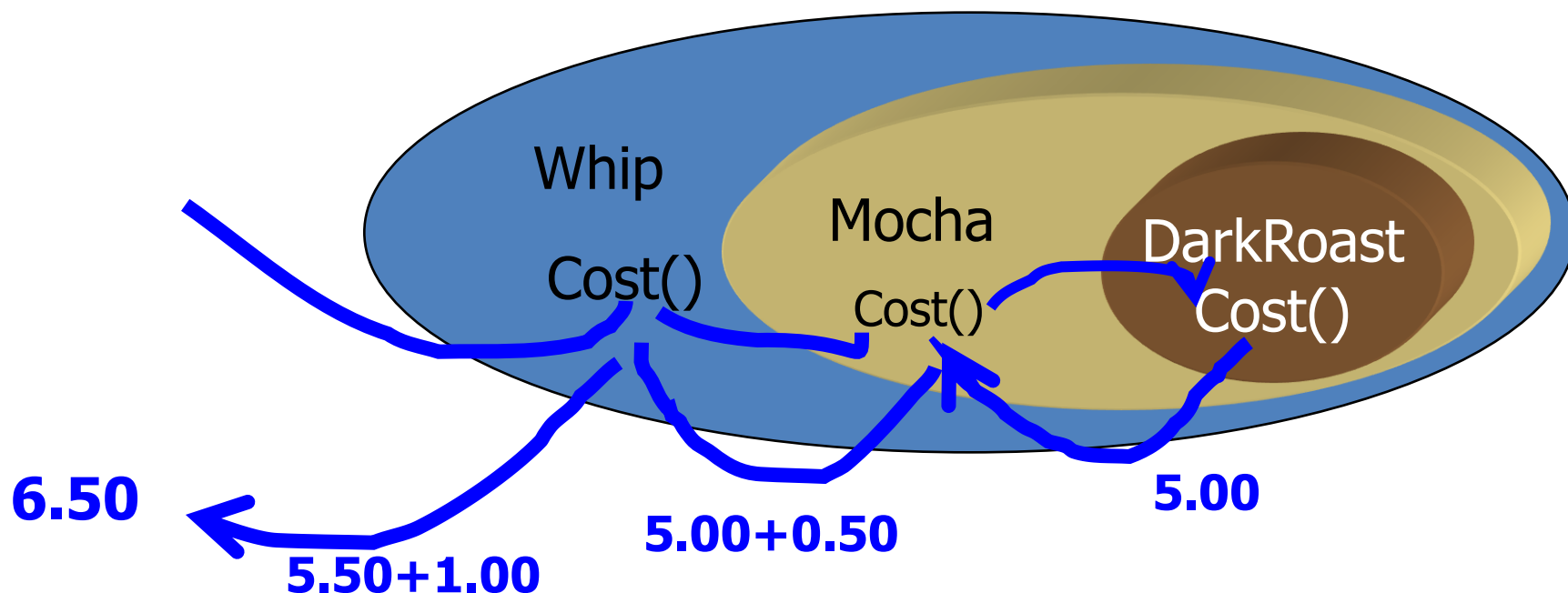
顾客还想要
whip，所以我
们创建一个
Whip对象，并
用它包装**Mocha**.



whip对象是装饰者，他与被它装饰的对象**DarkRoast**具有相同的类型，也有一个**cost()**方法。

3. 讨论

计算饮料的价格



调用最外层的装饰者whip的cost(),whip再将计算任务委派给被它包装的对象,得到一个价格后,再加上whip自己的价格...

Diagram

❖ 何谓委派？

Design

所谓委派就是一个对象将工作（或工作的一步分）交给另一个对象来完成。

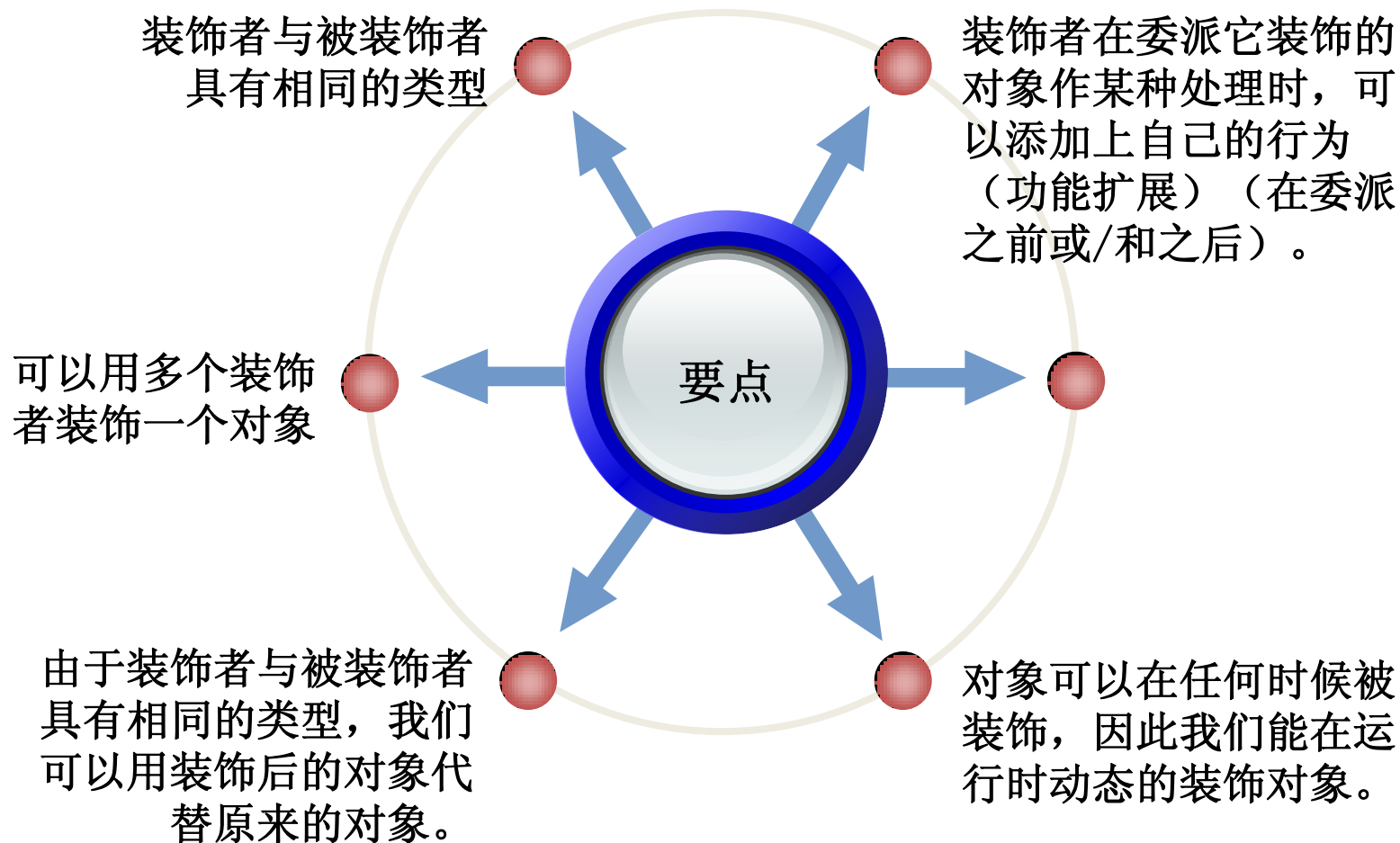
Design

在装饰模式中，委派是指**装饰对象**将任务交给**被装饰对象**来完成。

Design

委派可以传递，最终必须要有一个**干实事的对象**。

3. 讨论



咖啡店的类图

