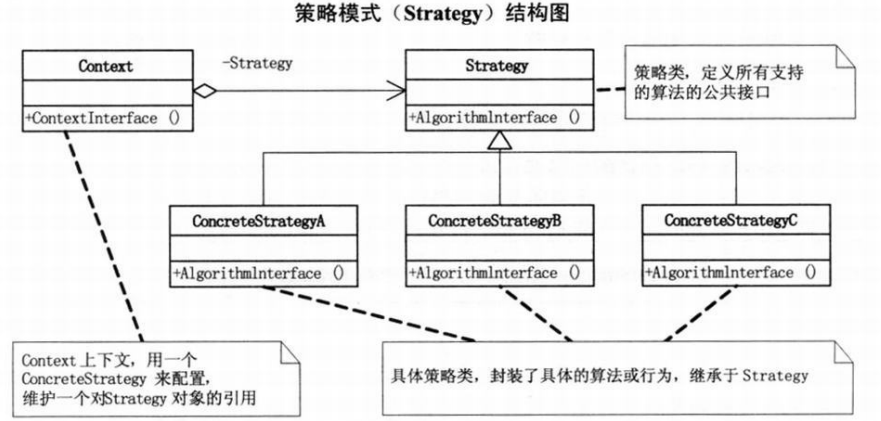
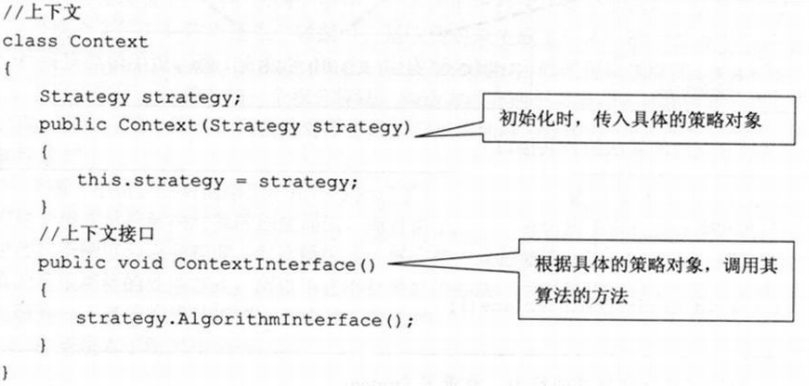
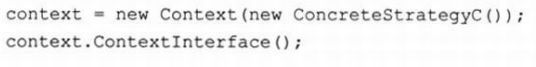
题目：POS机上根据用户购买商品的单价和数量，计算总价，考虑商场打折的各种优惠情况。

分析：

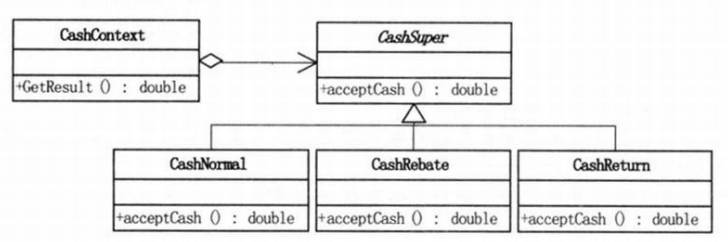




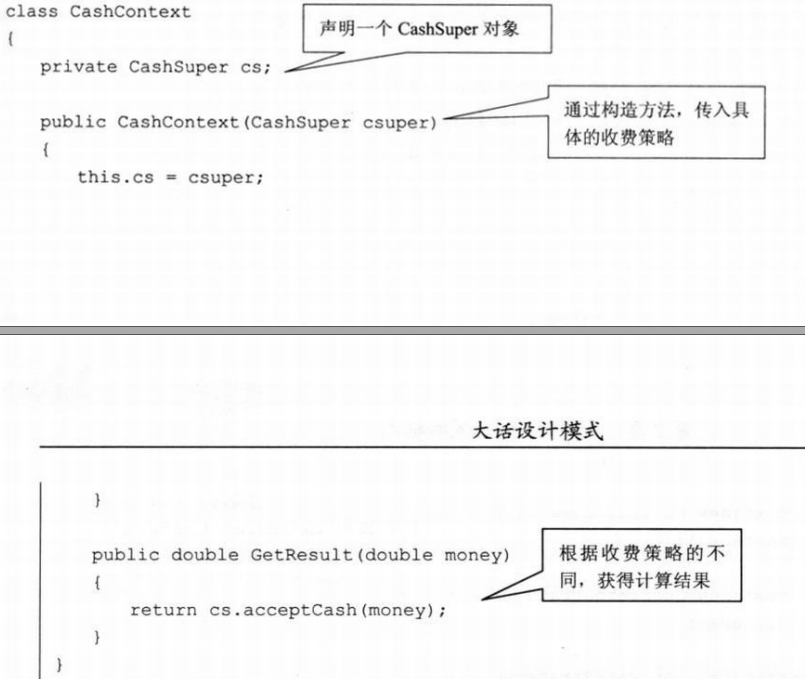
//客户端



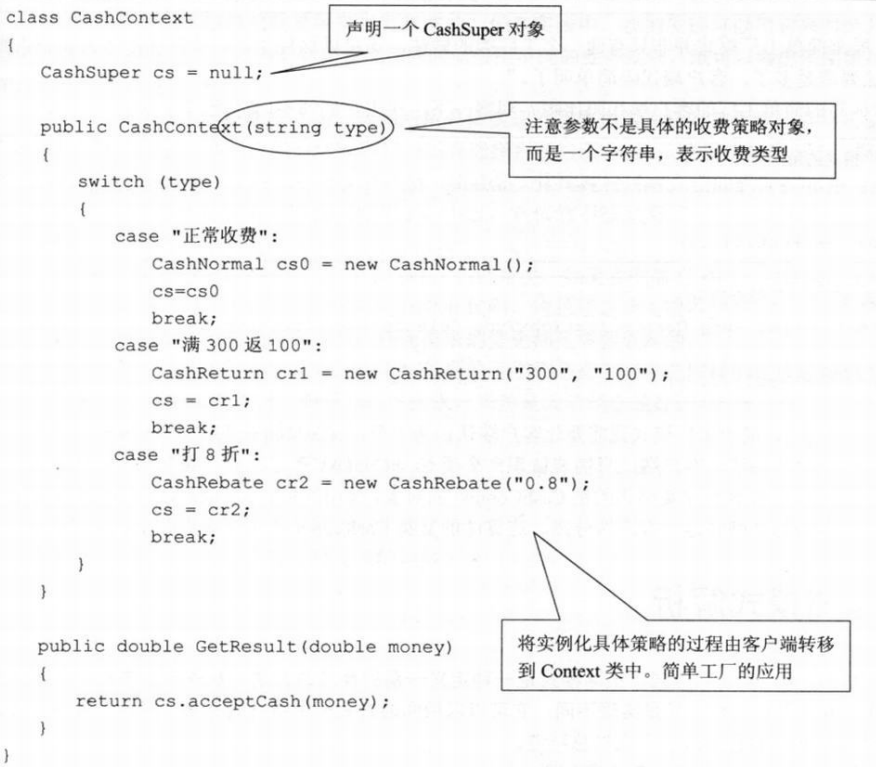
1. 只利用工厂模式，但是打折算法太多，创建很多子类并不是特别适合。但是可以与策略模式结合起来，将Context作为一个工厂，生成对应的strategy。比如上面在对Context初始化时，参数是一个策略对象，可以变成如下：



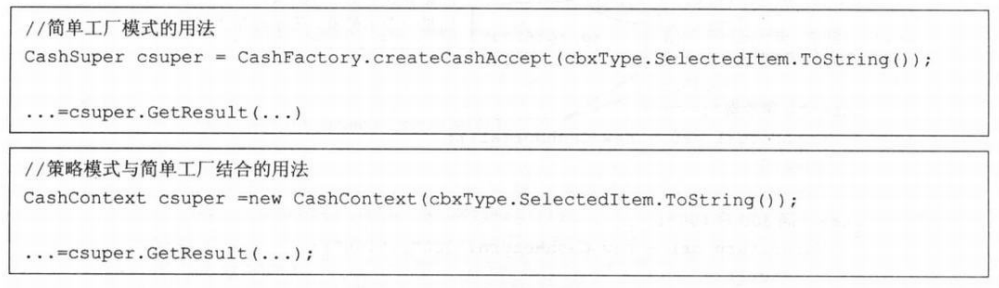
利用策略：



策略和工厂结合：



区别：



1. 策略模式Strategy层为Context层定义一系列算法，所有这些算法完成的都是相同的工作，只是实现不同，它可以以相同的方式调用所有的算法，减少了各种算法类与使用算法类之间的耦合，并且便于对各个算法进行单元测试。
2. 在实践中，一旦听到需要在不同的时间应用不同的业务规则，就可以用策略模式封装各种规则，从而避免将这些规则放在一个类中时，使用过多的switch条件分支。