策略设计模式-StrategyPattern

# 设计模式

设计模式可以把你的思考架构的层次提高到模式层面，而不是仅仅停留在琐碎的对象上。

利用**设计模式名称**与其他程序员沟通，其他程序员能够马上且清楚地知道你在说什么。模式可以让你用更少的词汇做更充分的沟通。

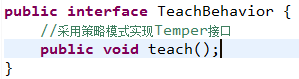
# 策略模式

策略模式定义了**算法族**，分别封装起来，让它们之间可以相互替换，此模式让算法的变化独立于使用算法的客户。

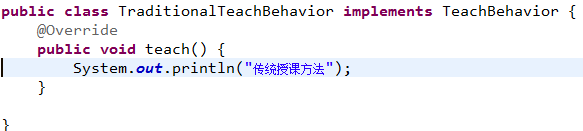
**算法族：某一行为的多种实现或某一目标的多种实现都可以称为算法族。**

**举个简单的例子说明策略模式：**

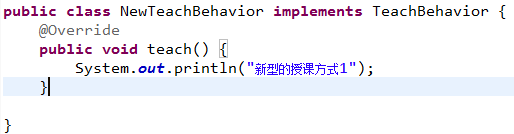
**老师都具有教课teache这个属性，可是对于不同科目的教课方法不同，随着时代变化，教课方法需要更新，有必要对教课方法进行单独封装。且采用策略模式，对教课进行多种实现。对于Teacher对象可以通过setTeach方法实现动态的改变教课方法，这既是策略模式。**



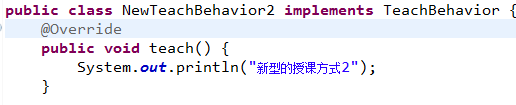
**策略1：**



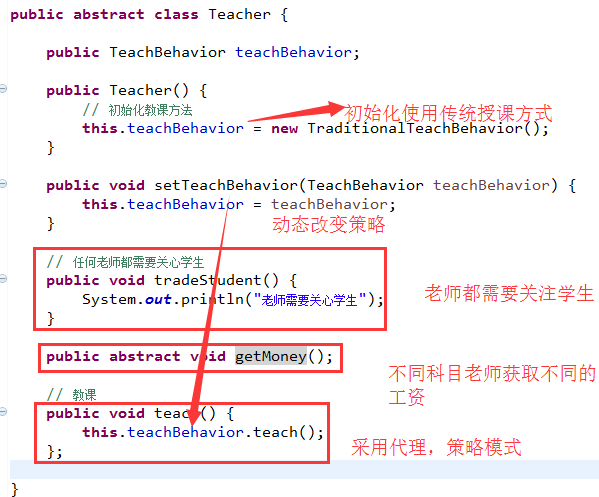
**策略2：**



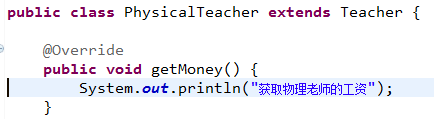
**策略3：**

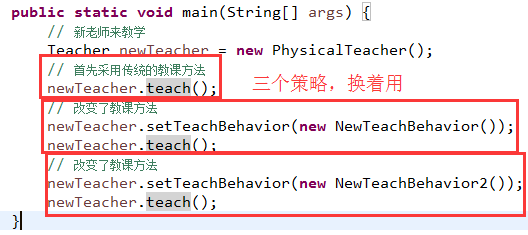


**老师超类（利用抽象类）：**



**物理老师实现类：**



**具体的测试：**

# 策略设计模式的深度理解

就是把某些需要重复使用的行为单独封装起来，定义一个接口，然后对此接口进行不同的实现，这些不同的实现就是不同的策略。

目标对象，通过setXxx方法可以动态地选择所需要使用的策略。