

**软件体系结构作业15**

姓 名 ： 洪伟鑫

专 业 ： 软件工程

年 级 ： 2022级

学 号 ： 37220222203612

**2025年4月22日**

**1、什么是透明装饰模式，什么是半透明装饰模式？请举例说明。**

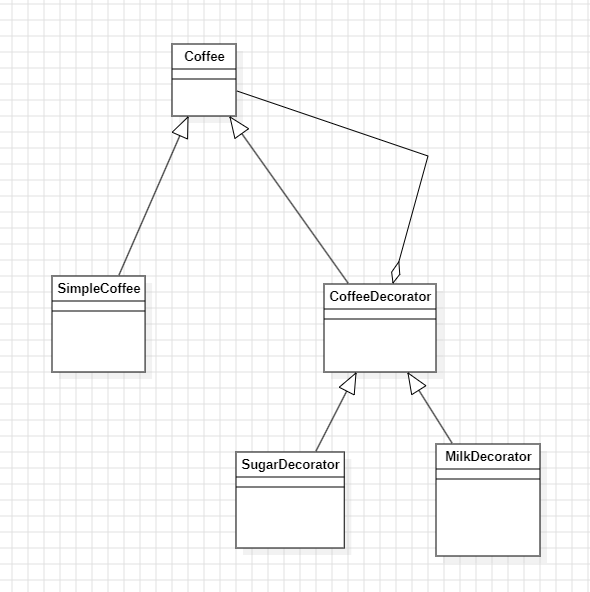
**透明模式**是一种结构型设计模式，旨在动态地向对象添加额外的职责。该模式的一个显著特点是其透明性：装饰器类实现了与被装饰对象相同的接口，这使得装饰器对象能够无缝地替代原始对象，客户端代码甚至察觉不到其差异。这种结构允许在运行时根据需要动态地添加或移除装饰器，为对象赋予新的行为。

在示例结构中，我们定义了一个 Coffee 接口来规范咖啡的基本行为（如获取价格和描述）。SimpleCoffee 类作为基础组件，实现了 Coffee 接口，代表一杯没有任何添加物的咖啡。

为了实现装饰功能，我们引入了抽象的 CoffeeDecorator 基类，它同样实现 Coffee 接口并持有一个 Coffee 对象的引用。具体的装饰器，如 MilkDecorator 和 SugarDecorator，继承自 CoffeeDecorator，它们在调用被包装对象（通过持有的引用）的原始方法前后，添加了各自的特定行为（例如，增加牛奶或糖的成本，并修改描述）。

运行 CoffeeShopDemo 的效果清晰地展示了该模式的工作方式：首先输出基础咖啡的价格和描述；接着，在基础咖啡上添加牛奶装饰器后，输出更新后的价格和描述；同样，在添加了糖装饰器后，也会看到相应的变化；最后，直接创建一个同时添加了牛奶和糖装饰器的咖啡实例，其价格和描述反映了所有装饰效果的累加。

装饰器模式的主要优势在于其高度的灵活性和可扩展性。它允许在不修改现有类代码的情况下动态地为对象添加新功能，完美地遵循了软件设计的开闭原则（对扩展开放，对修改关闭）。通过组合不同的装饰器，可以灵活地实现各种功能组合，有效避免了因功能排列组合而导致的子类数量爆炸性增长的问题。

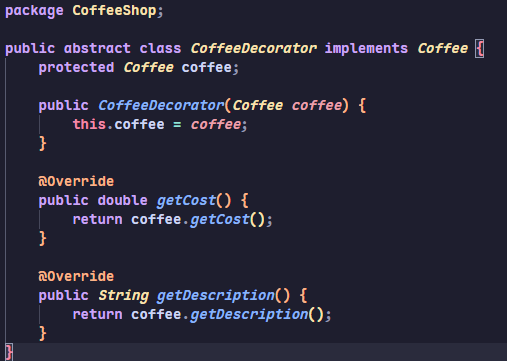


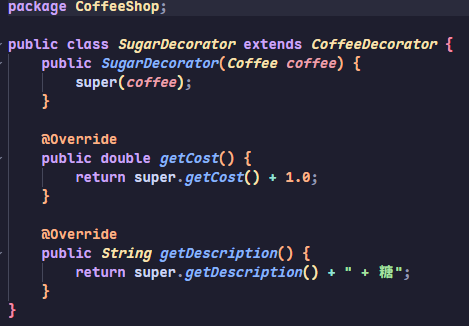
代码：

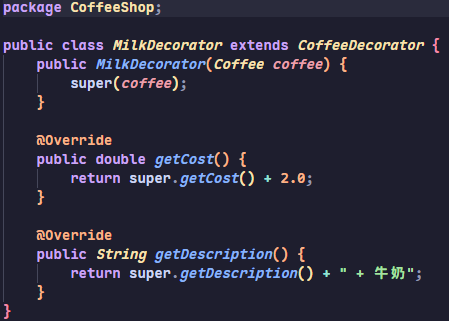
普通的咖啡(基础组件，不是装饰器)：



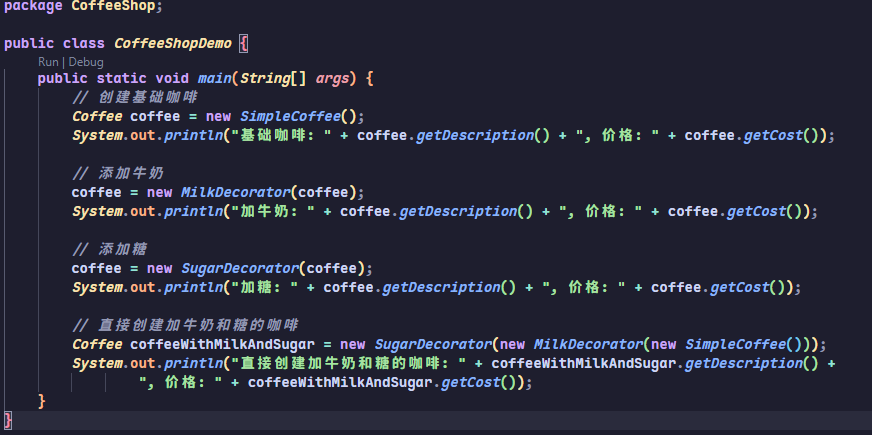
装饰器及具体实现：



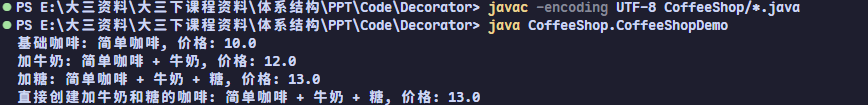




在Demo中测试：



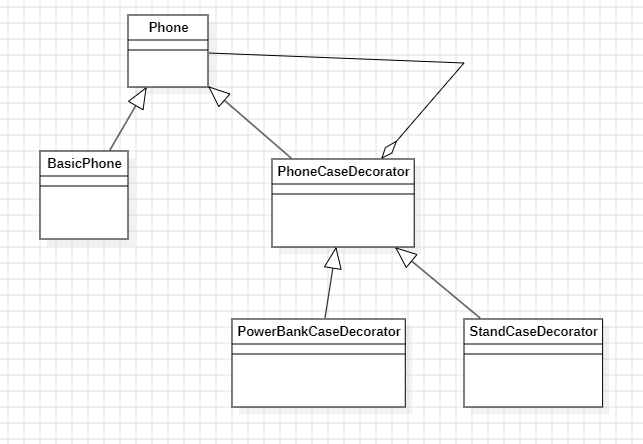
运行结果如下：



**半透明模式**

设定了一个 Phone 接口，它定义了如打电话和发短信等基本通信功能，由 BasicPhone 类实现这些基础能力。接着，我们引入了具体的装饰器，例如 StandCaseDecorator（支架手机壳）和 PowerBankCaseDecorator（充电宝手机壳）。这些装饰器类虽然也实现了 Phone 接口，从而保留了原有的打电话和发短信功能，但它们各自还添加了独特的新功能：StandCaseDecorator 增加了 useStand()（使用支架）和 adjustStandAngle()（调整支架角度）方法；而 PowerBankCaseDecorator 则增加了 chargePhone()（给手机充电）、chargePowerBank()（给充电宝充电）以及 getBatteryLevel()（查看电量）等方法。

这种模式被称为“半透明”的原因在于其部分失去了透明性。虽然客户端仍然可以将装饰后的对象视为原始的 Phone 类型来调用基础的打电话和发短信功能，但如果想要使用装饰器新增的特定功能（如使用支架或给手机充电），客户端代码就必须明确知道当前使用的是哪个具体的装饰器类型，并通常需要进行类型转换才能调用这些新增的方法。这与完全透明的装饰器模式（客户端无需关心具体装饰器类型即可使用所有功能）形成了对比。

****这样只能使用接口中定义的方法

Phone phone = new StandCaseDecorator(new BasicPhone("iPhone"));

phone.call("10086"); // 可以

// phone.useStand(); // 不可以，因为Phone接口中没有这个方法

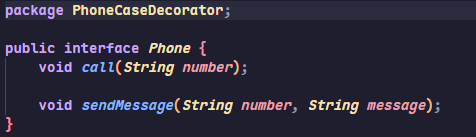
必须这样才能使用新增的方法

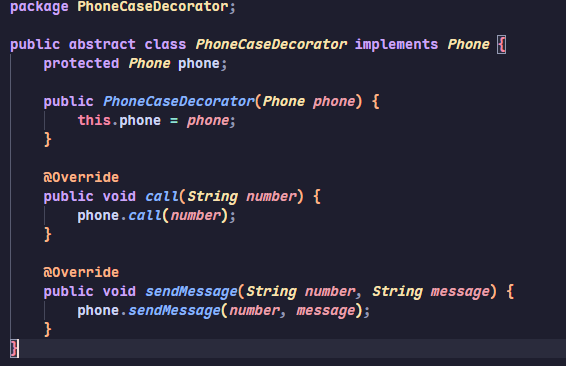
StandCaseDecorator standCase = new StandCaseDecorator(new BasicPhone("iPhone"));

standCase.useStand(); // 可以使用新增的方法

standCase.adjustStandAngle(45); // 可以使用新增的方法

代码:

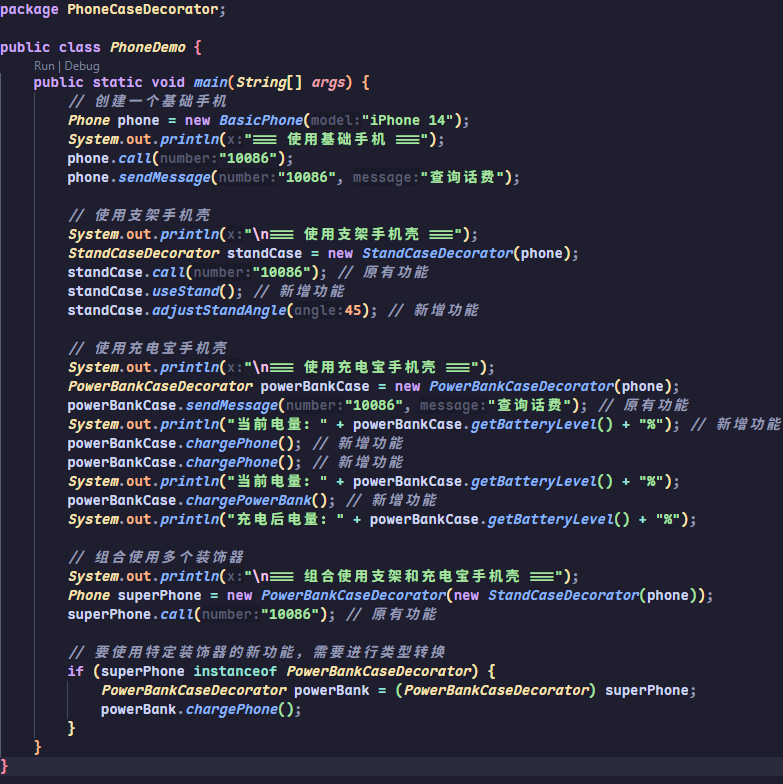




抽象的Decorator只能基本的收发短信和打电话，不能充电或使用手机支架等

以充电为例：





运行结果：

