介绍

前面几篇文章分别介绍了装饰器模式，桥接模式和适配器模式这三种设计模式。这三种设计模式，从设计思想角度上看是非常相似的。三者都是结构型的设计模式，而且都存在依赖抽象的情况。但是三者之间却又存在一些微妙的区别，这也是本文重点关注的内容。

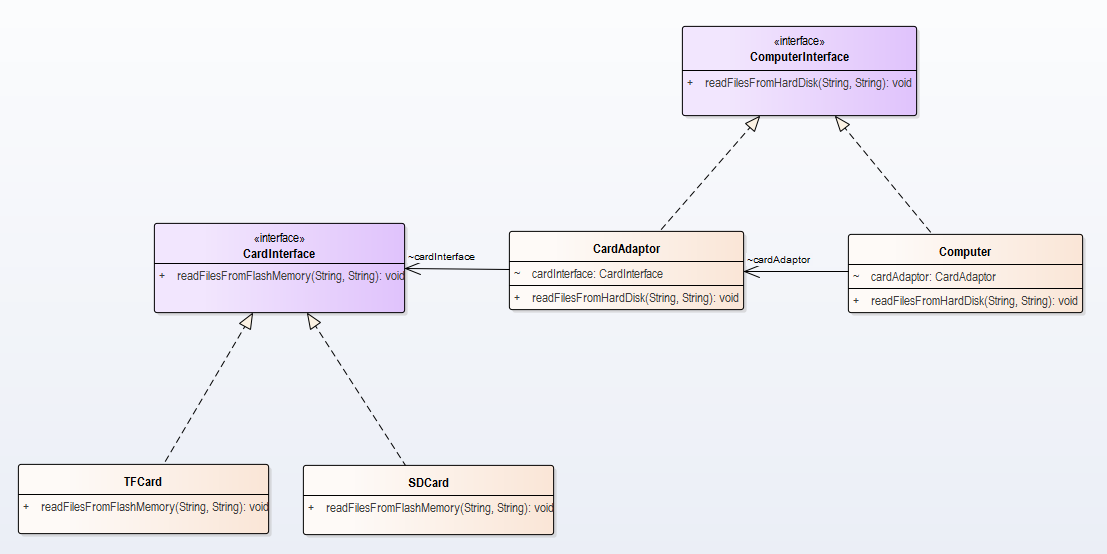
适配器模式

该模式重点强调的是适配的功能。

该模式的关键点是：

* 主体类和适配器类实现相同的接口A
* 主体类依赖适配器类
* 适配器类依赖抽象接口B
* 被适配的类实现抽象接口B

最终的效果就是，主体类可以使用之前不相关的被适配类中的某些功能。

把前文[《Java设计模式（6）----------适配器模式》](https://www.imooc.com/article/24001)中的UML图拿过来，结构是这样的  


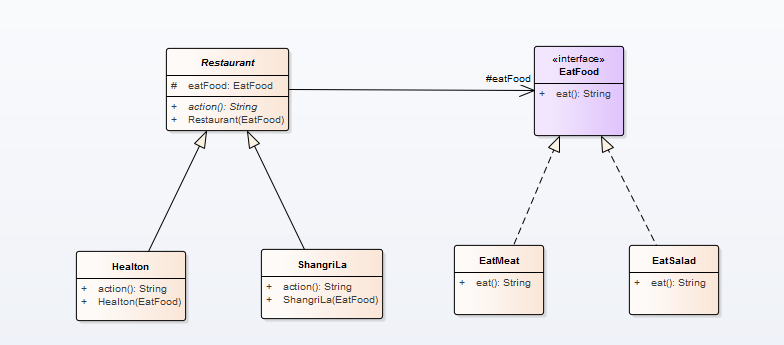
桥接模式

该模式重点强调的是多维度的变化。

该模式的关键点是：

* 主体类依赖抽象A
* 主体类具有多个不同的实现类
* 抽象A具有多个不同的实现类

最终的效果就是，主体类的实现类和抽象的实现类分别可以在两个维度上进行各自的变化。如果主体类依赖多个抽象，则维度进行增加，方便扩展。

把前文[Java设计模式（7）----------桥接模式](https://www.imooc.com/article/24008)中的UML图拿过来，结构是这样的  


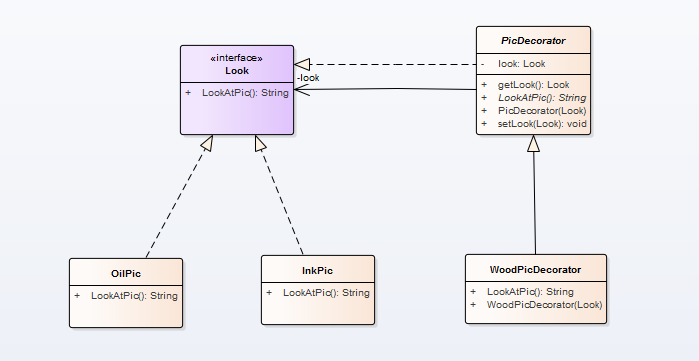
装饰器模式

该模式重点强调的是装饰功能。

该模式的关键点是：

* 抽象A具有多个具体子类
* 装饰器类依赖抽象A
* 装饰器类实现抽象A
* 装饰器类存在不同子类

最终的效果就是，（装饰器实现类）对（原抽象的子类）进行某些方法的功能加强。

把前文[《Java设计模式（9）----------装饰器模式》](https://www.imooc.com/article/24027)中的UML图拿过来，结构是这样的  


小结

其实从三者的UML图可以看出三种模式都是2个三角结构，但是位置不同。

* 适配器模式三角结构的位置是一上一下**（因为适配器依赖抽象）。**
* 桥接模式三角结构的位置是并列的**（主体类直接依赖抽象）。**
* 装饰器模式三角结构的位置也是并列的，但是多出了一条线**（主体类不仅依赖抽象，而且实现该抽象接口）**