**设计模式笔记**

1. **设计模式入门**

OO基础:抽象、封装、多态、继承

OO原则:

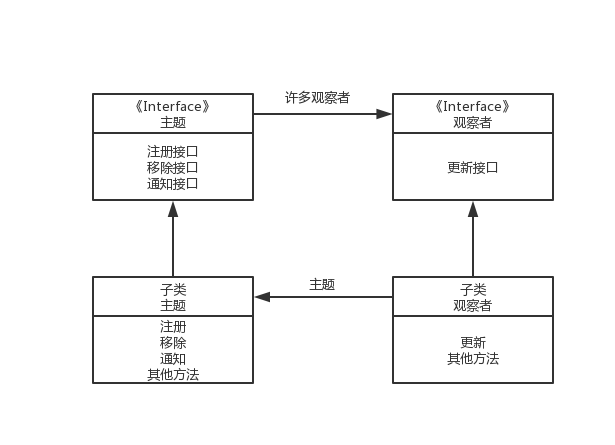
1. 封装变化
2. 多用组合,少用继承
3. 针对接口编程而非实现

OO模式之策略模式：

定义算法族,分别封装起来,让其之间可以相互替换,次模式让算法的变化独立于使用算法的客户

1. **观察者模式**

**性质:**观察者模式的本质就是出版者+订阅者。或者说,主题(出版者)和观察者(订阅者)。

**OO模式之观察者模式：**

定义了对象之间的一对多依赖，这样一来，当一个对象改变状态时，它的所有依赖都会收到通知且自动更新。

**观察者模式的类图：**

**松耦合：**

当两个对象之间松耦合，他们依然可以交互，但是不太清楚彼此的细节

**设计原则：**

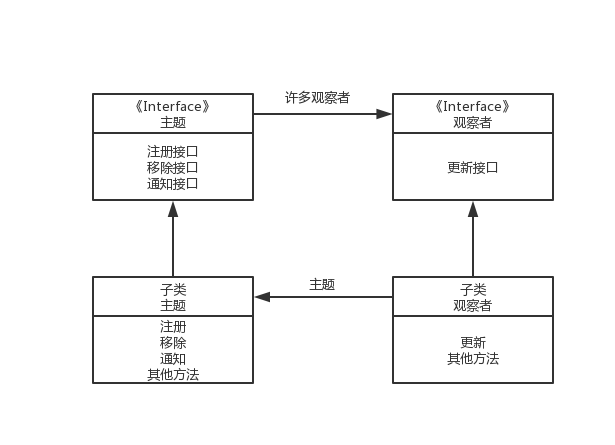
为了交互对象间的松耦合而努力

松耦合的设计之所以能够建立富有弹性的OO系统以应对变化，是因为对象之间的依赖降到了最低

1. **装饰者模式**

**设计原则:**

类应当对扩展开放，对修改关闭

**OO模式之装饰者模式：**

动态地将责任附加到对象上,若要扩展功能，装饰者提供比继承更有弹性的替代方案

**装饰者模式的类图:**

实际上装饰者模式利用了组合来替代继承

**OO原则(扩展)：**

1. 封装变化
2. 多用组合、少用继承
3. 针对接口编程而非实现
4. 为交互对象之间松耦合而努力
5. 对扩展开放、对修改关闭