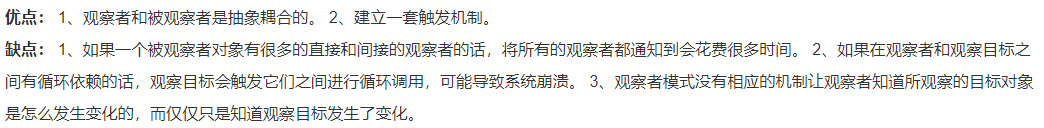
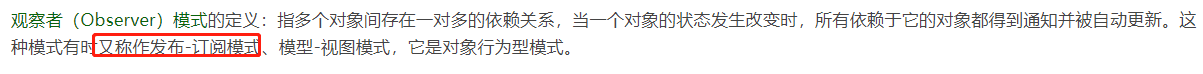
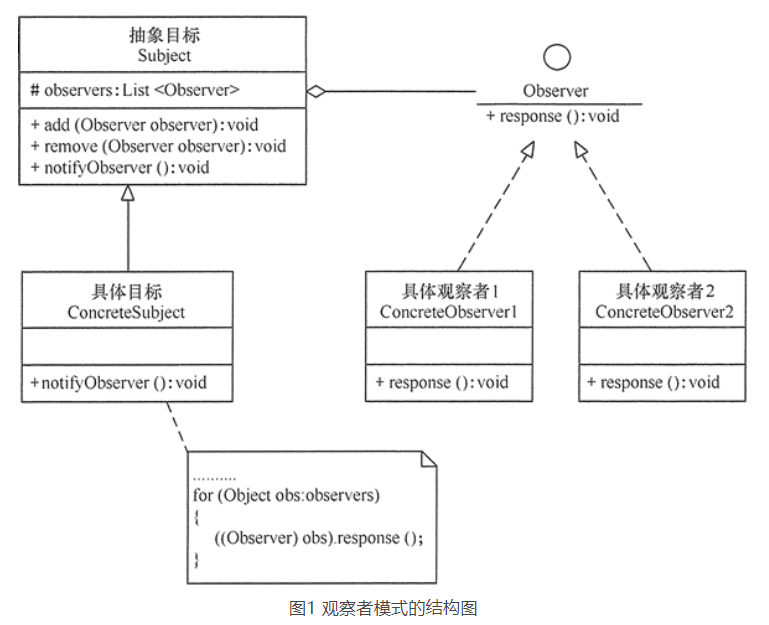
当对象间存在一对多关系时，则使用观察者模式（Observer Pattern）。比如，当一个对象被修改时，则会自动通知依赖它的对象。观察者模式属于行为型模式。







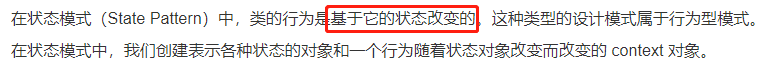
观察者和中介者很像，也是A类保存B类的集合，B类保存A类的引用，

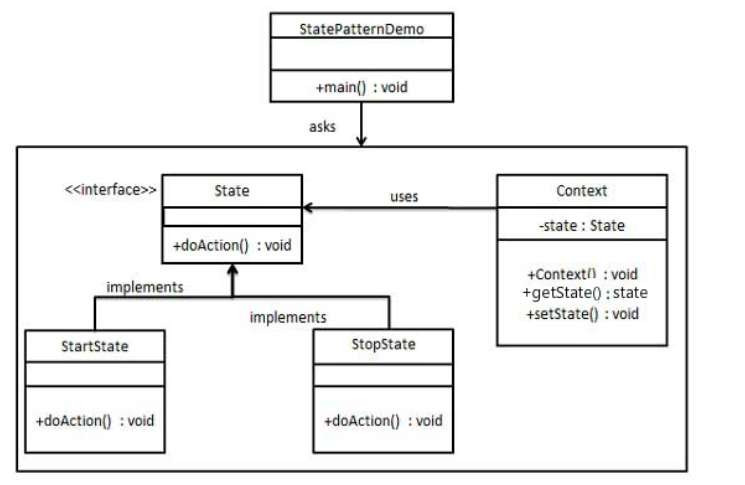
但不同于中介者一个B类可以通过A类通知到其他的B类，（多人聊天室，聊天室是中介者，一个人可以通过中介者发送信息给其余大多数人）

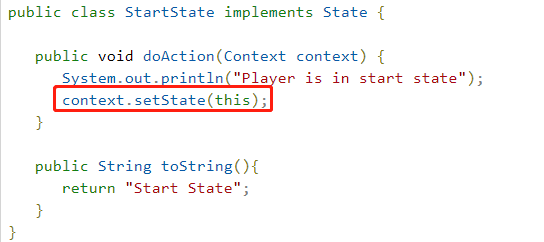
观察者是A类一发生改变，所有的B类都能感知到并做出相应的响应（多人聊天室，聊天室是主体，聊天室发送的官方通知可以被用户（观察者）接收）

<https://www.runoob.com/design-pattern/observer-pattern.html>

枚举可以实现单例，而且不用被序列化和反射干扰

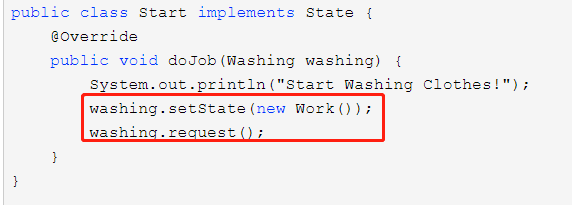






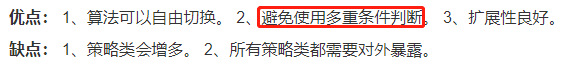
在调用state对象时传入到上下文context中，从而上下文拥有状态对象

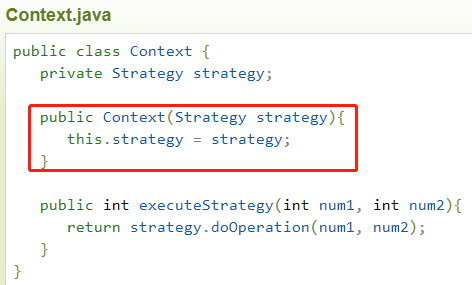
控制该状态到其他状态的转移，通过改变上下文的状态在调用上下文的方法来实现状态转移



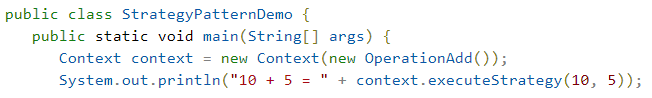
<https://www.cnblogs.com/yssjun/p/11116652.html>

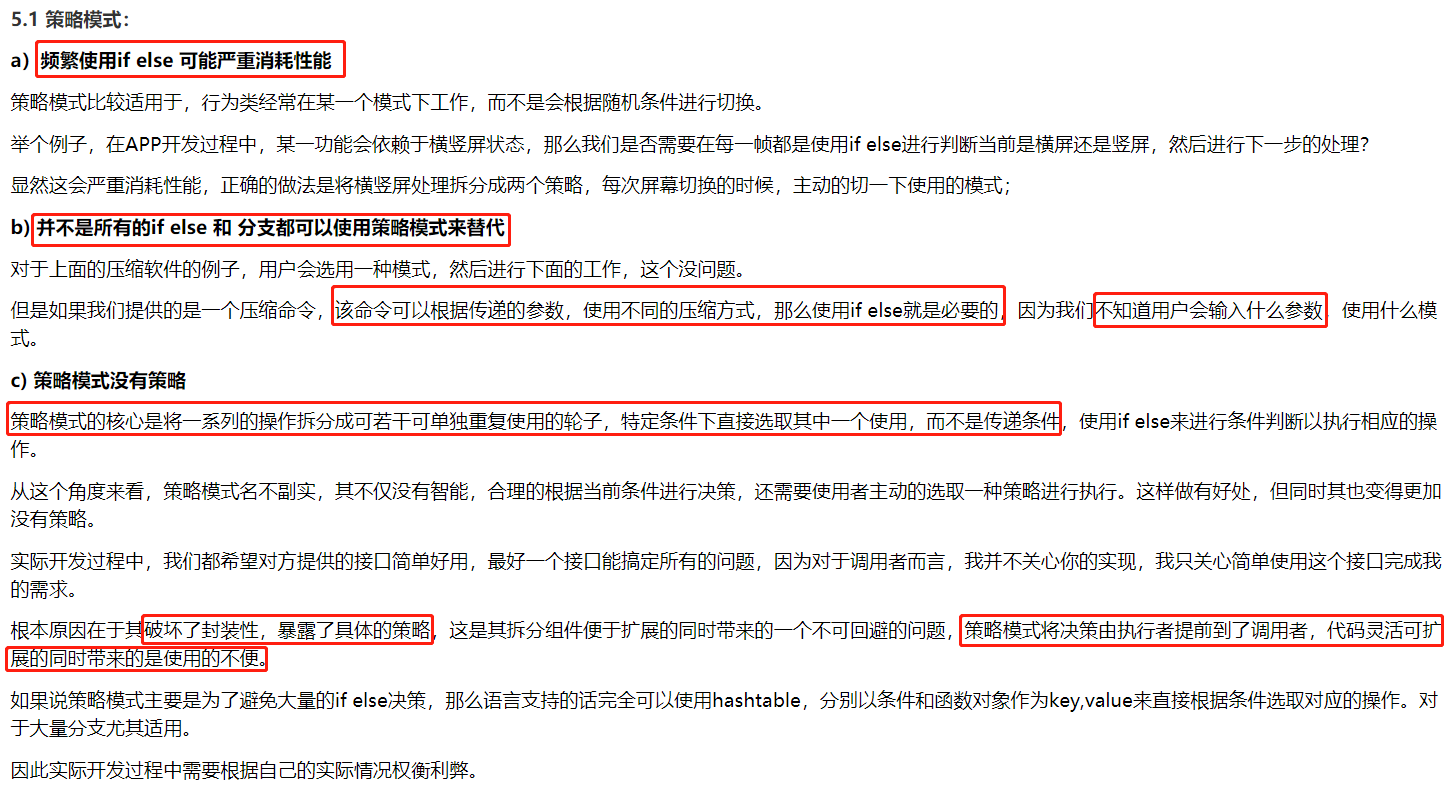
在策略模式（Strategy Pattern）中，一个类的行为或其算法可以在运行时更改。这种类型的设计模式属于行为型模式。（诸葛亮的锦囊妙计，每一个锦囊就是一个策略。）



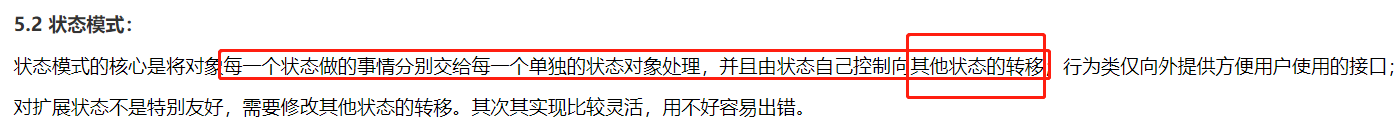


策略模式是在上下文传入策略

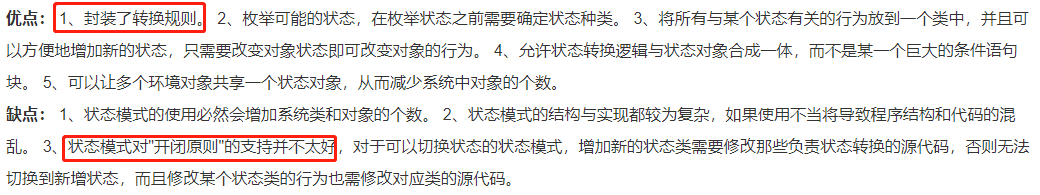




策略模式扩展性好，修改可读性高（比如10到30岁是青年，30到50岁是中年）这种通过岁数来判断的就不能用策略模式



着重于状态的转移，就像洗衣机开始-》工作-》结束有三个状态



<https://www.cnblogs.com/yssjun/p/11116652.html>

在模板模式（Template Pattern）中，一个抽象类公开定义了执行它的方法的方式/模板。它的子类可以按需要重写方法实现

