状态模式(state)

核心：

* 用于解决系统中复杂对象的状态转换以及不同状态下行为的封装问题

结构:

* Context环境类
* 环境类中维护一个State对象，他是定义了当前的状态
* State抽象状态类
* ConcreteState具体状态类
* 每一个类封装了该状态对应的行为

简单点说:

状态模式封装的非常好，状态的变更引起了行为的变更，状态变换放置到了类的内部来实现，外部的调用不用知道类内部如何实现状态和行为的变换，从外部看起来就好像这个对象对应的类发生了改变一样。

观察我写的代码，可以发现状态模式可扩展性极强，若想新增一个状态，源代码逻辑一点都不需要变动。只需要在新增一个状态类并且在接口和环境中新增对应的方法和属性即可。真正的对修改关闭对扩展开放。