## 状态模式

1. 定义与使用场景

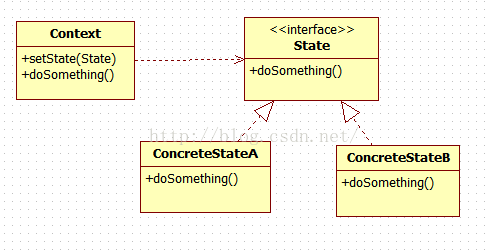
定义：当一个对象的内在状态改变时允许改变其行为，这个对象看起来像是改变了其类。

使用场景：一个对象的行为取决于它的状态，并且它必须在运行时根据状态改变它的行为。

代码中包含大量与对象状态有关的条件语句，例如，一个操作中含有庞大的多分支语句（if-else或switch-case），且这些分支依赖于该对象的状态。

状态模式将每一个条件分支放入一个独立的类中，这使得可以根据对象自身的情况将对象的状态作为一个对象，这一对象可以不依赖于其他对象而独立变化，这样通过多态来去除过大的、重复if-else等分支语句。

1. UML类图



Context：环境类，定义客户感兴趣的接口，维护一个State子类的实例，这个实例定义了对象当前的状态。

State：抽象状态类或者状态接口，定义一个或者一组接口，表示该状态下的行为

ConcreteStatusA、ConcreteStateB：具体的状态类，每一个具体的状态类实现抽象State中定义的接口，从而达到不同状态下的不同行为。

1. 简单实现
2. 源码分析
3. 总结

状态模式将所有与一个特定的状态相关的行为都放入一个状态类中，它提供了一个更好的方法来组织与特定状态相关的代码，将繁琐的状态转换成结构清晰的状态类族，在避免代码膨胀的同时也保证了可扩展性与可维护性。