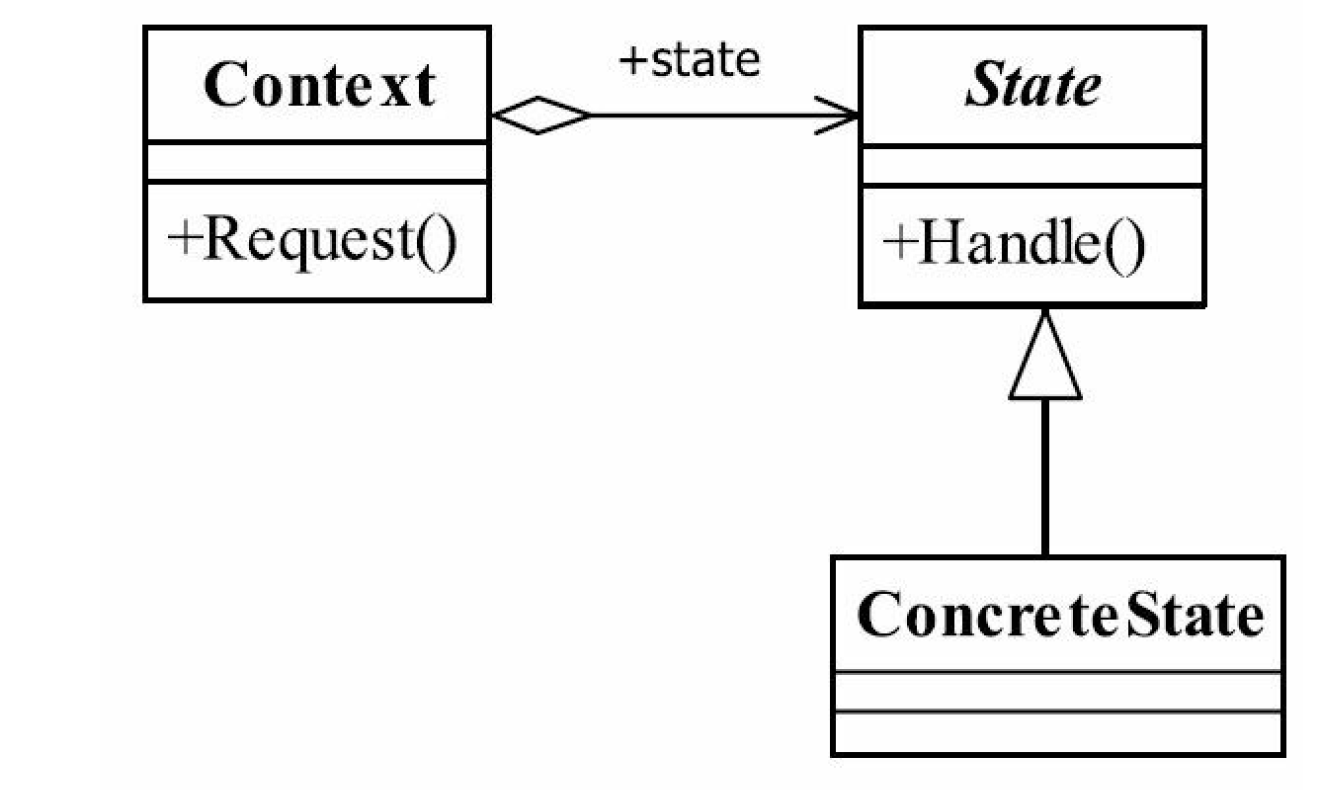
状态模式

当一个对象内在状态改变时允许其改变行为，这个对象看起来像改变了其类。

状态模式的核心是封装，状态的变更引起了行为的变更，从外部看起来就好像这个对象对应的类发生了改变一样。



我们先来看看状态模式中的3个角色。

● State——抽象状态角色

接口或抽象类，负责对象状态定义，并且封装环境角色以实现状态切换。

● ConcreteState——具体状态角色

每一个具体状态必须完成两个职责：本状态的行为管理以及趋向状态处理，通俗地说，

就是本状态下要做的事情，以及本状态如何过渡到其他状态。

● Context——环境角色

定义客户端需要的接口，并且负责具体状态的切换。

环境角色有两个不成文的约束：

● 把状态对象声明为静态常量，有几个状态对象就声明几个静态常量。

● 环境角色具有状态抽象角色定义的所有行为，具体执行使用委托方式。