

“白云社区” 设计模式举例

1. 组合模式（结构型模式）

在社区的“案例”和“问答”模块中，用户可以对案例或回答进行评论，还可以对评论进行评论，这样就构成了一个树状结构。这里，一条评论在作为对象的同时，还可以作为其它评论对象的容器，但它对于外界的接口是不依赖于它的容器结构的。因此可以采用组合模式，使得可以像处理单条评论对象一样处理复杂的容器对象（评论子树）。

设计抽象类 `Comment` 和继承它的子类 `AtomComment` 和 `CompositeComment`，分别表示具体的评论和评论子树。一个 `CompositeComment` 对象的属性 `root` 是一个 `AtomComment` 对象，表示该 `CompositeComment` 对象对应的具体评论；属性 `children` 是一个列表，包含零个或多个 `CompositeComment` 对象，表示该评论下的评论子树。`AtomComment` 的方法 `OperateSingle` 实现了限于该评论的处理（如点赞、举报），方法 `OperateAll` 实现了涉及某棵评论子树的处理（如删除）；`CompositeComment` 的方法 `OperateSingle` 调用了 `root` 的 `OperateSingle` 方法，方法 `OperateAll` 调用了 `root` 的 `OperateAll` 方法和 `children` 中所有对象的 `OperateAll` 方法，方法 `Add`, `Remove` 接收 `Comment` 类型参数以添加或删除元素，方法 `GetChildren` 返回容器保存的所有子结点。UML 类图如图 1 所示。

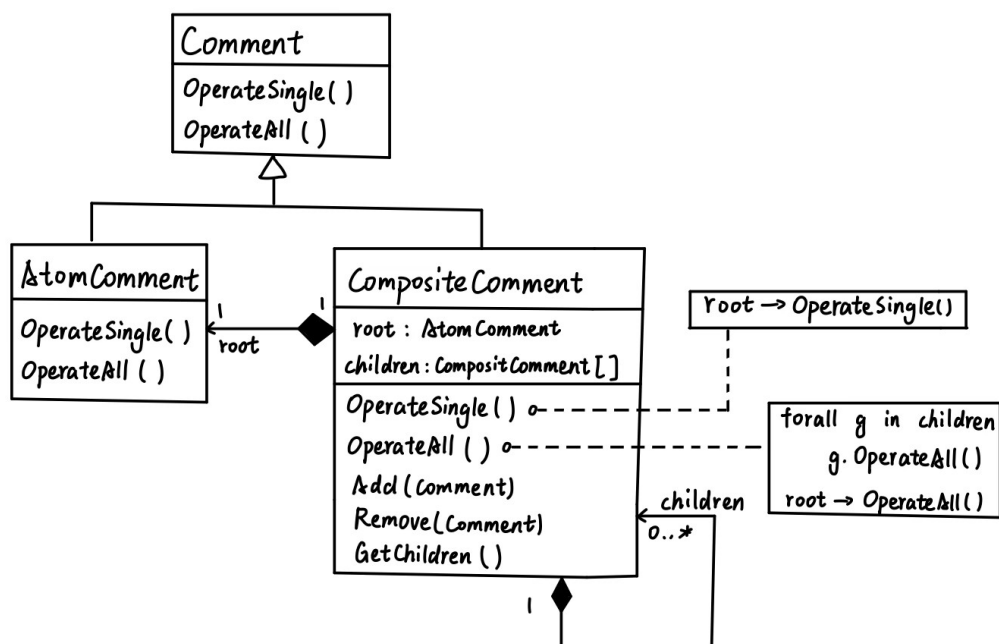


图 1 组合模式类图

2. 观察者模式（行为型模式）

在社区的“问答”模块中，一个问题下的回答的状态变化将会影响到问题。例如，当回答被创建或删除时需要通知提问的用户，当回答的热度变化时问题的热度也会变化。此时，“问题”就是一个观察者，“回答”是被观察者。因此可以采用观察者模式，使得回答的状态变化时，问题可以得到通知并自动更新。

设计抽象类 Subject, Observer 和分别继承它们的子类 Answer 和 Question，分别表示作为被观察者的回答和作为观察者的问题。当被观察者的状态发生变化时，会调用 Notify 方法通知观察者，观察者调用 Update 方法获得被观察者的状态并据此更新自身状态。UML 类图如图 2 所示。

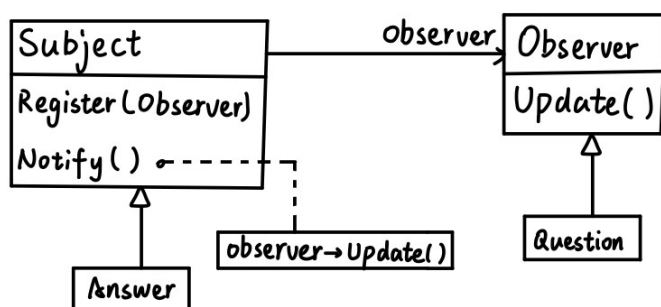


图 2 观察者模式类图