

19 最少知识原则

最少知识原则LKP说的是一个软件实体（类，对象，系统，模块等等）尽可能少地与其他实体发生相互作用。

该原则也叫迪米特法则！是描述的一种设计思想

减少对象之间的联系

SPR原则知道我们要减小对象的粒度，提高复用性。但是越来越多对象如果不加以控制会产生错综复杂的联系和相互引用，导致修改任何一个对象的影响面非常广，反而降低了可复用性。“一人犯法，株连九族”

如果两个对象之间不必批次直接通信，就不要直接发生联系。

常见的作法是引入一个第三者对象来承担这些对象之间的通信作用。如果需要通信和请求可以委托给第三者对象来进行

设计模式中的LKP原则

1. 中介者模式

通过增加一个中介对象，让所有相关对象都通过中介相互应用。当一个对象改变时，只需要通知中介者对象即可

2. 外观模式

外观模式在JS中用的不多，主要是为子系统中的一组接口提供一个一致的界面，而外观模式定义了一个高层的接口，从而对客户屏蔽一组子系统的内部复杂性，高层接口相对简单易用，并指导如何和子系统打交道。

例如全自动洗衣机通过一个自动洗衣按钮来完成了原先的浸泡，洗衣，漂洗，脱水4个步骤

这里隔绝客户与子系统之间的联系是符合LKP原则的

封装在最少知识原则的体现

封装表达的是数据隐藏，只对外暴露必要的接口API，通过这些API让对象之间的联系限制在最小的范围。

在JS中封装也可以用来限制变量作用域，减少对其他不相干模块的影响