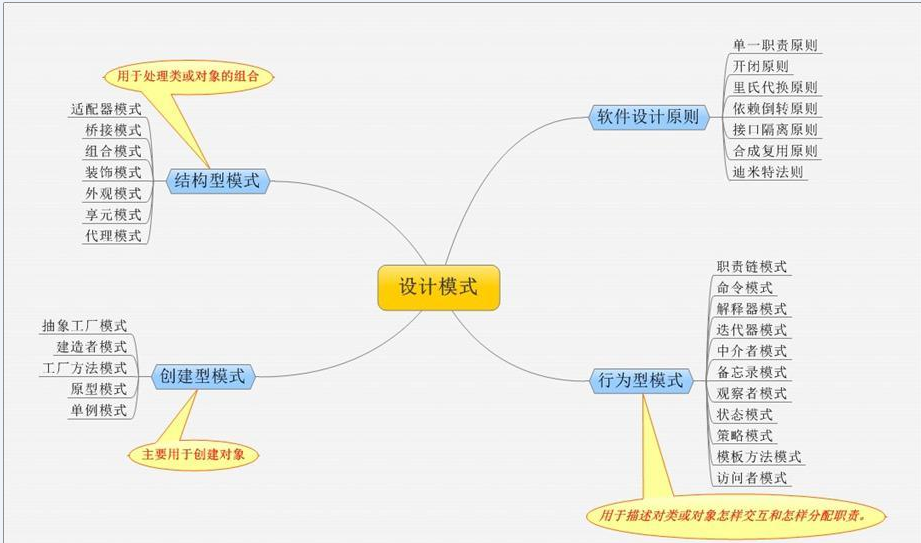
# 设计模式

设计模式之间的关系：



## 什么是设计模式：

设计模式是一套被反复使用的、多数人知晓的、经过分类编码、代码设计经验的总结（使用设计模式：可重用代码、保证代码的可靠性、使代码更容易被人理解）

面向对象设计原则：

SOLID（单一功能、开闭原则、里氏替换、接口隔离以及依赖反转）

[单一功能原则](http://zh.wikipedia.org/wiki/%E5%8D%95%E4%B8%80%E5%8A%9F%E8%83%BD%E5%8E%9F%E5%88%99)

认为[对象](http://zh.wikipedia.org/wiki/%E5%AF%B9%E8%B1%A1_(%E8%AE%A1%E7%AE%97%E6%9C%BA%E7%A7%91%E5%AD%A6))应该仅具有一种单一功能的概念。

[开闭原则](http://zh.wikipedia.org/wiki/%E5%BC%80%E9%97%AD%E5%8E%9F%E5%88%99)

认为“软件体应该是对于扩展开放的，但是对于修改封闭的”的概念。

[里氏替换原则](http://zh.wikipedia.org/wiki/%E9%87%8C%E6%B0%8F%E6%9B%BF%E6%8D%A2%E5%8E%9F%E5%88%99)

认为“程序中的对象应该是可以在不改变程序正确性的前提下被它的子类所替换的”的概念。参考 [契约式设计](http://zh.wikipedia.org/wiki/%E5%A5%91%E7%BA%A6%E5%BC%8F%E8%AE%BE%E8%AE%A1)。

[接口隔离原则](http://zh.wikipedia.org/wiki/%E6%8E%A5%E5%8F%A3%E9%9A%94%E7%A6%BB%E5%8E%9F%E5%88%99)

认为“多个特定客户端接口要好于一个宽泛用途的接口”的概念。

[依赖反转原则](http://zh.wikipedia.org/wiki/%E4%BE%9D%E8%B5%96%E5%8F%8D%E8%BD%AC%E5%8E%9F%E5%88%99)

认为一个方法应该遵从“依赖于抽象而不是一个实例 的概念。  
[依赖注入](http://zh.wikipedia.org/wiki/%E4%BE%9D%E8%B5%96%E6%B3%A8%E5%85%A5)是该原则的一种实现方式。

# 