

设计原则

开闭原则

软件实体（模块、类、方法等）应该“对扩展开放，对修改关闭”

单一职责原则

一个类或者模块只负责完成一个职责（或者功能）

里氏替换原则

子类对象能够替换程序中父类对象出现的任何地方，并且保证原来程序的逻辑行为不变及正确性不被破坏（按照协议来设计）

接口隔离原则

客户端不应该强迫依赖它不需要的接口（一个类对另一个类的依赖应该建立在最小的接口上）

依赖倒置原则

高层模块不要依赖低层模块。高层模块和低层模块应该通过抽象来互相依赖。除此之外，抽象不要依赖具体实现细节，具体实现细节依赖抽象

迪米特法则

又称最小知识原则。每个模块只应该了解那些与它关系密切的模块的有限知识。或者说，每个模块只和自己的朋友“说话”，不和陌生人说话

合成复用原则

软件复用时，要尽量先使用组合或者聚合等关联关系来实现，其次才考虑使用继承关系来实现

KISS 原则

尽量保持简单

YAGNI 原则

不要做过度设计

DRY 原则

不要写重复的代码