1 单一职责原则

就一个类而言，应该仅有一个引起它变化的原因。

2 开放-封闭原则

软件实体（类、模块、函数等）应该可以扩展，但是不可修改。面对需求，对程序的改动是通过增加新代码进行的，而不是修改现有的代码，这是开放-封闭原则的精神所在。

3 依赖倒转原则

抽象不应该依赖细节，细节应该依赖于抽象，通俗一点就是，针对接口编程，不要对实现编程。

（A 高层模块不应该依赖低层模块。两个都应该依赖抽象

B 抽象不应该依赖细节，细节应该依赖于抽象）

4 里氏代换原则

一个软件实体如果使用的是一个父类的，那么一定使用与其子类，而且它察觉不出父类对象和子类对象的区别，也就是软件里面，把父类都替换成它的子类，程序的行为没有变化。