## 面向对象的六大原则

1. 单一职责原则（Simple Responsibility Principle）

两个完全不一样的功能不应该放在同一个类中；一个类应该是一组相关性很高的函数、数据的封装

1. 开闭原则（Open Close Principle）

软件中的对象（类、模块、函数等）应该对于扩展是开放的，但是对于修改是封闭的。当软件需求变化时，应该尽量通过扩展的方式来实现变化，而不是通过修改已有的代码来实现。

1. 里氏替换原则（Liskov Substitution principle）

所有引用基类的地方必须能透明地使用其子类的对象；即只要父类出现的地方子类就可以出现，并且替换成子类也不会产生任何错误或异常，使用者根本就不知道是父类还是子类。

1. 依赖倒置原则（Dependence Inversion principle）

模块间的依赖通过抽象发生，实现类之间不发生直接的依赖关系，其依赖关系是通过接口或抽象类发生的。抽象指的是抽象类或者接口。

1. 接口隔离原则（Interface Segregation principle）

客户端不应该依赖它不需要的接口。另一种定义是：类间的依赖关系应该建立在最小的接口上。接口隔离原则将非常庞大、臃肿的接口拆分成更小的和更具体的接口。客户端只需要依赖它感兴趣的接口。

1. 迪米特原则（Low of Demeter）最少知识原则

一个对象应该对其他对象有最少的了解。类于类之间的关系越密切，耦合度越大，当一个类发生变化时，对另一个类的影响就越大。