

设计原则名称	设计原则简介	设计原则之间关系
单一职责原则 SRP Single Responsibility Principle	一个对象应该只包含 单一的职责 ，并且该职责被完整地封装在一个类中。	
开闭原则 OCP Open-Closed Principle	一个软件实体应该 对扩展开放，对修改关闭 。软件实体可以是一个软件模块、一个由多个类组成的局部结构或一个单独的类。	开闭原则是对单一职责原则的加强。
里氏代换原则 LSP Liskov Substitution Principle	所有引用基类（父类）的地方必须能 透明地 使用其子类的对象。 通俗表达： 软件中如果能够使用基类对象，那么一定能够使用其子类对象。	里氏代换原则是开闭原则的具体实现。
依赖倒转原则 DIP Dependency Inversion Principle	高层模块 不应该依赖于低层模块 ，他们都应该依赖抽象。抽象不应该依赖于细节，细节应该依赖于抽象。 另一种表述： 要针对接口编程，而不要针对实现编程。	依赖倒转原则是开闭原则的具体实现。
接口隔离原则 ISP Interface Segregation Principle	客户 不应该依赖 那些它不需要的接口。	拆分时需要满足单一职责原则
合成复用原则 CRP Composite Reuse Principle	在系统中应该尽量多实用组合聚合关联关系，尽量少甚至不使用继承关系。	
迪米特法则 LoD Law of Demter	在一个软件实体对其他实体的引用越少越好，或者说如果两个类不必彼此直接通信，那么这两个类就不应当发生直接的相互作用，而是通过引入一个第三者发生简介交互。	