单一职责

概述

一个类，应该仅有一个引起他变化的原因。而违反这一个原则的例子有，把所有的逻辑全部都写到了一个方法或者函数里面。如果在一个类承担了过多的职责，那么会出现严重的耦合。一个职责的变化可能会引起其他职责的变化

开闭原则 The Open-Closed Principle

概述

不修改，但可以扩展。当一个类完成的时候，尽量不去修改，而去增加类。 必须猜出来哪些最有可能变化的类，然后构造抽象去隔离那些变化 面对需求，对程序的该改动是通过新增加代码，而不是更改现有的代码 开发人员应该对程序中频繁变化的那些部分作出抽象

依赖倒置原则

概述

抽象不应该依赖于细节，细节应该依赖于抽象。针对接口编程，而不是针对实现编程 高层模块依赖于底层模块，两者都依赖于抽象

里式代换原则

子类可以替换父类，软件没有产生影响，父类才可以得到复用，子类也可以在父类上增加新行为 跟JAVA里面的向下转型是一样的，子类可以实现父类允许实现的所有方法，同时程序不发生变动 子类型的可替换性使得父类的模块在无需修改的情况之下就可以扩展