# 面向对象设计原则

## 依赖倒置原则（DIP）

高层模块（稳定）不应依赖于低层模块（变化），二者都应该依赖于抽象（稳定）。

抽象（稳定）不应该依赖于实现细节（变化），实现细节（变化）依赖于抽象（稳定）。

## 开放封闭原则（OCP）

对扩展开放，对更改封闭。

类模块应该时可扩展的，不可修改。

## 单一职责原则（SRP）

一个类应该仅有一个引起它变化的原因。

变化的方向隐含着类的责任。

## Liskov替换原则（LSP）

子类必须能够替换它们的基类。

继承表达类型抽象。

## 接口隔离原则（ISP）

不应该强迫客户程序依赖它们不用的方法。

接口应该小而完备。

## 优先使用对象组合，而不是类继承

## 封装变化点

## 针对接口编程，而不是针对实现编程

## “组件协作”模式：

Template Method 模板方法

Strategy 策略模式

Observer / Event 观察者模式

## “单一职责”模式：

Decorator 装饰模式

Bridge 桥模式

## “对象创建”模式：

Factory Method 工厂方法

Abstract Factory 抽象工厂

Prototype 原型模式 不常用

Builder 构建器 不常用

## “对象性能”模式：

Singleton 单件模式

Flyweight 享元模式

## “接口隔离”模式：

Facade 门面模式

Proxy 代理模式

Adapter 适配器

Mediator 中介者

## “状态变化”模式：

State 状态模式

Memento 备忘录

## “数据结构”模式：

Composite 组合模式

Iterator 迭代器

Chain of Resposibility职责链