

# 2019Android 设计模式总结

## 1. 设计模式六大原则

### a. 单一职责原则：就一个类来说，应该只有一个引起它变化的原因

一个类做一件事情，避免职责过多。比如这种情况是不太好的，在一个 Activity 中既有 bean 文件，又有 http 请求，还有 adapter 等等，这就导致我们需要修改任何一个东西的时候都会导致 Activity 的改变，这样一来就有多个引起它变化的原因，不符合单一职责原则

### b. 开放封闭原则：类，模块，函数应该是可以扩展的，但是不可以修改

对于扩展是开放的，对于修改是封闭的。尽量做到面对需求的改变时，我们的代码能保持相对稳定，通过扩展的方式应对变化，而不是修改原有代码实现

### c. 里氏替换原则：所有引用基类的地方，必须可以透明的时候其子类的对象

里氏替换原则是实现开放封闭原则的重要方式之一，我们知道，使用基类的地方都可以使用子类去实现，因为子类拥有基类的所有方法，所以在程序设计中尽量使用基类类型对对象进行定义，在运行时确定子类类型。

### d. 依赖倒置原则：高层模块不应该依赖于底层模块，两者都应该依赖于抽象，抽象不应该依赖于细节，细节应该依赖于抽象

依赖倒置原则针对的是模块之间的依赖关系，高层模块指调用端，底层模块指具体的实现类，抽象指接口或抽象类，细节就是实现类。该原则的具体表现就是模块间的依赖通过抽象发生，直线类之间不发生直接依赖关系，依赖通过接口或抽象类产生，降低耦合，比如 MVP 模式下，View 层和 P 层通过接口产生依赖关系

### e. 迪米特原则（最少知识原则）：一个软件实体应该尽可能少的与其他实体发生相互作用

迪米特原则要求我们在设计系统时，尽量减少对象之间的交互

### f. 接口隔离原则：一个类对另一个类的依赖应该建立在最小的接口上

接口隔离原则的关键是接口以及这个接口要小，如何小呢，也就是我们要为专门的类创建专门的接口，这个接口只对它有效，不要试图让一个接口包罗万象，要建立最小的依赖关系

## 2. 设计模式的分类

设计模式分为三类

### 创建型设计模式

与对象创建有关包括单例模式，工厂方法模式，抽象工厂模式，建造者模式，原型模式

### 结构型设计模式

结构性设计模式是从程序的结构上解决模块之间的耦合问题，包括适配器模式，代理模式，装饰模式，外观模式，桥接模式，组合模式和享元模式

### 行为型设计模式

主要处理类或对象如何交互及如何分配职责，包括策略模式，模板方法模式，观察者模式，迭代器模式，责任链模式，命令模式，备忘录模式，状态模式，访问者模式，中介模式，解析器模式