# 设计模式

### 介绍工厂模式（[参考链接](http://wiki.xuchongyang.com/%E8%AE%BE%E8%AE%A1%E6%A8%A1%E5%BC%8F/%E5%88%9B%E5%BB%BA%E5%9E%8B%E6%A8%A1%E5%BC%8F.html)）

工厂模式分为：简单工厂、工厂方法、抽象工厂三种。

* 简单工厂模式有一个单独的工厂角色类，根据传入的不同参数，负责创建不同的实例
* 工厂方法模式中工厂角色类不再负责创建实例，它自己只负责声明接口，把创建实例的任务给了它的工厂子类
* 抽象工厂模式的工厂会创建出一组具体产品，构成一个产品族

### 介绍单例模式

* 单例模式确保一个类只有一个实例
* 自行实例化
* 自行向整个系统提供这个实例

写法：[参考链接](http://wuchong.me/blog/2014/08/28/how-to-correctly-write-singleton-pattern/)

* 懒汉式，线程不安全
* 懒汉式，线程安全，效率不高
* 双重校验锁（推荐）
* 饿汉式，不是懒加载
* 静态内部类（推荐）
* 枚举 Enum

### 介绍观察者模式

* 定义了对象间的一种一对多的依赖关系
* 观察目标发生改变时，会自动通知观察者，观察者做出相应的反应
* 观察目标会将观察者的引用保存在一个集合中，通知观察者时循环调用

# 数据结构与算法

# 其他

### 最近使用过的开源库有什么，讲讲实现原理（源码）

### HTTP 和 HTTPS 的区别

1、HTTP 是 HTTP 协议运行在 TCP 之上，信息是明文传输的，客户端和服务端也都无法验证对方身份

2、HTTPS 是 HTTP 协议运行在 SSL/TLS 之上，SSL/TLS 运行在 TCP 之上，所有传输内容经过了加密，客户端和服务端也可以验证对方身份

### Gradle 的优势在哪

Gradle 的功能：依赖管理、多模块构建、

* Maven 基于 XML 配置繁琐，阅读性差，Gradle 基于 Groovy，简化了构建代码的行数，易于阅读

1、依赖管理方面：Gradle 支持依赖动态版本管理，解决依赖冲突机制更明确

2、多模块构建方面：Gradle 使用 allprojects 和 subprojects 来定义里面的配置是应用于所有项目还是子项目，更加灵活

3、构建周期方面：Gradle 本身与项目构建周期是解耦的，可以灵活的增删 task

### 对网络通信协议的理解

1、网络由下往上分为物理层、链路层、网络层、传输层、会话层、表示层、应用层

2、IP 协议位于网络层，TCP 协议位于传输层、HTTP 协议位于应用层

3、Socket 是对 TCP/IP 协议的封装，是一个调用接口，通过 Socket 才能使用 TCP/IP 协议

4、TCP 是面向连接的，属于可靠连接；UDP 不是面向连接的，属于不可靠连接。

5、UDP 发送数据前不与对方建立连接，收到数据后也不发送确认信号，所以开销低，实时性好

### 对 TCP 和 HTTP 的理解

1、TCP 为传输层协议，解决数据如何在网络中传输

2、HTTP 为应用层协议，解决如何包装数据

2、HTTP 协议需要依靠 TCP 协议来传输数据

3、HTTP 对 TCP 的使用分为两种：短连接和长连接，HTTP 1.0 默认短连接，1.1 默认长连接

# 其他面试题

### 对 Android 哪一方面的研究比较深入

### 介绍下你之前所做的这些项目

### 如何把控一个项目，架构如何设计

### 医疗的架构是如何的

### 如何优化医疗这个产品

### 医疗的通讯机制

### 门禁机运作过程

### 云门禁这个项目你负责了哪些和 Android 有关的

### 为什么转行