

# 2020 阿里巴巴最新 Android 面试真题集锦

## 一、Java 面试题

### 1. 重载函数的签名(区别是否是重载函数)

方法名+参数类型+参数顺序(返回值不是)

### 2. finalize 的工作原理

一旦垃圾收集器准备好释放对象占用的存储空间，它首先调用 `finalize()`，而且只有在下一次垃圾收集过程中，才会真正回收对象的内存。所以如果使用 `finalize()`，就可以在垃圾收集期间进行一些重要的清除或清扫工作。此外系统进行垃圾回收时并不保证一定会调用它，所以可以说它和对象被回收没有必然关系。

### 3. 一个对象的创建过程

例如有一个 `Dog` 类：

- 1.即使没有 `static`，构造器其实也是静态方法，所以当首次创建 `Dog` 时或者 `Dog` 的静态域或者静态方法被访问的时候，`Dog` 的 `class` 对象会被加载。
- 2.在加载 `Dog` 的 `class` 的时候可能会发现其有基类，此时先暂停 `Dog` 的加载，转而去加载其基类。所以基类的 `static` 域比子类先初始化。
- 3.载入 `class` 文件之后，会创建一个该类的 `class` 对象，所有关于静态初始化的动作会被执行，所以静态初始化只在 `Class` 对象首次加载时进行一次。
- 4.当使用 `new` 的时候，首先为 `Dog` 对象在堆上分配足够内存空间。这块内存控件会被清零，所有实例都被初始化成了默认值，包括基类。
- 5.执行所有字段定义处的初始化，从基类开始。
- 6.执行构造器。(在第一行会执行基类的构造器)

### 4. 对象的销毁是按照创建的逆序来进行的。

### 5. `final` 的方法和类都不允许覆盖和继承。`private` 默认实现了 `final`。

### 6. 简述内部类的特点

- 1.内部类只能在与其外围类的对象关联的情况下才能被创建(非 `static`)

- 2.内部类在创建的时候，捕获了一个外围类的对象，所以能访问外围类的 `private` 对象。(非 `static`)
- 3.如果其他类对内部类有访问权限的话(如 `public`)，那么可以通过 外围类对象.new 内部类()，这样的方式创建内部对象，注意如果没有外围类的对象，内部类是不允许创建的。(非 `static`)
- 4.多层嵌套的内部类能够透明的访问所有外围类的所有成员。
- 5.内部类不能被覆盖
- 6.内部类让 `java` 模拟了多继承，让外围类继承一个类，然后内部类继承一个类，又要有内部类对象必须有对应的外围类对象。
- 7.`static` 的内部类，和普通的类基本相同。

7. 形式参数可被视为 `local variable`，也就是说形式参数相当于在方法中定义了一个局部变量 `a`，当传入 `c` 时，只是将 `c` 指向的对象给 `a`####

8. `finally` 语句块是在 `try` 或者 `catch` 中的 `return` 语句之前执行的

## 9. 简述 `jvm` 中默认的 `ClassLoader` 与功能

- 1.Bootstrap ClassLoader: 负责加载 `java` 基础类，主要是 `%JRE_HOME/lib/` 目录下的 `rt.jar`、`resources.jar`、`charsets.jar` 和 `class` 等
- 2.Extension ClassLoader: 负责加载 `java` 扩展类，主要是 `%JRE_HOME/lib/ext` 目录下的 `jar` 和 `class`
- 3.App ClassLoader: 负责加载当前 `java` 应用的 `classpath` 中的所有类。
- 4.classloader 加载类用的是全盘负责双亲委托机制。
  - 1.所谓全盘负责，即是当一个 `classloader` 加载一个 `Class` 的时候，这个 `Class` 所依赖的和引用的所有 `Class` 也由这个 `classloader` 负责载入，除非是显式的使用另外一个 `classloader` 载入。所以，当我们自定义的 `classloader` 加载成功了 `com.company.MyClass` 以后，`MyClass` 里所有依赖的 `class` 都由这个 `ClassLoader` 来加载完成。
  - 2.双亲委托，是指子类加载器如果没有加载过该目标类，就先委托父类加载器加载该目标类，只有在父类加载器找不到字节码文件的情况下才从自己的类路径中查找并装载目标类。

10. switch 语句后的控制表达式只能是 short、char、int、long 整数类型和枚举类型，不能是 float，double 和 boolean 类型。String 类型是 java7 开始支持

## 11. 重写的特性

- 1.方法名相同，参数类型和顺序相同
- 2.子类返回类型小于等于父类方法返回类型。即：子类返回的类型要为父类的子类
- 3.子类抛出异常小于等于父类方法抛出异常。即：子类抛出的异常类型要为父类的子类
- 4.子类访问权限大于等于父类方法访问权限。

12. !=和==，当用于基本类型时候，是比较值是否相同；当用于引用类型的时候，是比较对象是否相同，比较内存地址

13. null 可以被强制类型转换成任意类型的对象，于是通过它来执行静态方法

## 14. 如何终止一个线程###

- 1.如果线程中是循环或者线程已经 sleep()，可以通过 Excutor 产生一个中断，
- 2.处于循环中的话，还可以设置一个标志位，一旦为 false 就让循环退出此时线程也会接着终止。
- 3.如果线程处于 io 之中，那么可以通过关闭 io 来使线程结束。

## 15. 静态属性和静态方法是否可以被继承

非静态属性、静态属性、静态方法 都可以被子类继承，但是不会被覆盖，转为哪个类，调用的就是那个类的东西。也就是说这几个东西不具有多态。

16. 列出各种 List，Set，Map，Queue 并说出他们的同与不同。

- 1.List: ArrayList、LinkedList、Vector、CopyOnWriteArrayList

- 1.ArrayList:常用的 List，非线程安全、内部使用数组来实现、对于随机访问 get 和 set，ArrayList 优于 LinkedList
  - 2.LinkedList: 线程非安全、内部用链表来实现、对于新增和删除操作 add 和 remove，LinkedList 比较占优势，因为 ArrayList 要移动数据。
  - 3.Vector: Vector 的方法都是同步的、是线程安全的、当 Vector 或 ArrayList 中的元素超过它的初始大小时,Vector 会将它的容量翻倍,而 ArrayList 只增加 50%的大小，这样,ArrayList 就有利于节约内存空间。
  - 4.CopyOnWriteArrayList: ArrayList 的一个线程安全的变体，其中所有可变操作（add、set 等等）都是通过对底层数组进行一次新的复制来实现的，在 CopyOnWriteArrayList 中，写入将导致创建整个底层数组的副本，而源数组将保留在原地，使得复制的数组在被修改时，读取操作可以安全的执行。当修改完成时，一个原子性的操作将把新的数组换入，使得新的读取操作可以看到这个新的修改。这对于读操作远远多于写操作的应用非常适合
- 2.Set: HashSet、TreeSet、LinkedHashSet、CopyOnWriteArraySet、EnumSet
    - 1.HashSet: 非线程安全、放入的元素需要重写 equals()和 hashCode()方法，才能避免放入元素重复，并且难以被发现。
    - 2.TreeSet: 非线程安全、其内部使用红黑树储存元素、TreeSet 储存的类型必须实现 comparable 接口。
    - 3.LinkedHashSet: 非线程安全、以元素插入的顺序来维护集合的链接表，其他地方和 HashMap 类似。
    - - 1. CopyOnWriteArraySet : 线程安全、CopyOnWriteArraySet 基于 CopyOnWriteArrayList 实现。
    - - 1. EnumSet 是一个专门为枚举类设计的集合类,EnumSet 中所有元素都必须是指定枚举类型的枚举值，该枚举类型在创建 EnumSet 时显式、或隐式地指定。EnumSet 的集合元素也是有序的，它们以枚举值在 Enum 类内的定义顺序来决定集合元素的顺序。
- 3.Map: HashMap、TreeMap、LinkedHashMap、HashTable、ConcurrentHashMap、WeakHashMap、EnumMap
    - 1.HashMap:非线程安全、其内部使用链表法解决哈希冲突、key 需要重写 equals()和 hashCode()方法，其内部在获取元素的时候先通过 hashCode()获取到某条链表，再通过 equals()方法在这条链表中找到该元素。
    - 2.TreeMap: 非线程安全、TreeMap 的实现是红黑树算法的实现，key 必须实现 comparable 接口。
    - 3.LinkedHashMap: 非线程安全、这个 map 是按插入顺序遍历该 map，以散列插入。还可以在构造函数中设置参数，使其以 LRU 算法排序。其获取的 Set 能体现出排序的方式。

- 4.HashTable: 线程安全、所有的方法都是同步的、key 和 value 不能为 null。
  - 5.ConcurrentHashMap: 线程安全、ConcurrentHashMap 融合了 hashtable 和 hashmap 二者的优势、实现了细粒度的锁，也就是不是所有地方都上锁，性能比 HashTable 好。
  - 6.WeakHashMap 与 HashMap 的用法基本相似。区别在于，HashMap 的 key 保留了对实际对象的"强引用"，这意味着只要该 HashMap 对象不被销毁，该 HashMap 所引用的对象就不会被垃圾回收。但 WeakHashMap 的 key 只保留了对实际对象的弱引用，这意味着如果 WeakHashMap 对象的 key 所引用的对象没有被其他强引用变量所引用，则这些 key 所引用的对象可能被垃圾回收，当垃圾回收了该 key 所对应的实际对象之后，WeakHashMap 会自动删除这些 key 所对应的 key-value 对
  - 7.EnumMap: EnumMap 是一个与枚举类一起使用的 Map 实现，EnumMap 中的所有 key 都必须是单个枚举类的枚举值。创建 EnumMap 时必须显式或隐式指定它对应的枚举类。EnumMap 根据 key 的自然顺序(即枚举值在枚举类中的定义顺序)。
- 4.Queue :   ArrayBlockingQueue 、   LinkedBlockingQueue 、   DelayQueue 、   PriorityBlockingQueue、PriorityQueue。
  - 1.ArrayBlockingQueue : 线程安全、基于数组的阻塞队列实现，在 ArrayBlockingQueue 内部，维护了一个定长数组，以便缓存队列中的数据对象。同一时间只能有一个线程进行操作。
  - 2.LinkedBlockingQueue: 基于链表的阻塞队列，同 ArrayListBlockingQueue 类似，其内部也维持着一个数据缓冲队列，当生产者往队列中放入一个数据时，队列会从生产者手中获取数据，并缓存在队列内部，而生产者立即返回、只有当队列缓冲区达到最大值缓存容量时（LinkedBlockingQueue 可以通过构造函数指定该值），才会阻塞生产者队列，直到消费者从队列中消费掉一份数据，生产者线程会被唤醒。可以有多个线程进行同时操作。
  - 3.DelayQueue: DelayQueue 里面放的是实现了 Delayed 的接口，Delayed 接口需要实现两个方法：getDelay(TimeUnit unit)是为了返回该任务所需延长的时间，compareTo(Delayed o)是为了比较到底那个任务需要排在前面，当使用 queue.take()的时候延时开始，此时本线程挂起，直到延时结束开始运行 run() 中的任务。
  - 4.PriorityBlockingQueue: 基于优先级的阻塞队列（优先级的判断通过构造函数传入的 Compator 对象来决定），但需要注意的是 PriorityBlockingQueue 并不会阻塞数据生产者，而只会在没有可消费的数据时，阻塞数据的消费者。因此使用的时候要特别注意，生产者生产数据的速度绝对不能快于消费者消费数据的速度，否则时间一长，会最终耗尽所有的可用堆内存空间。
  - 5.PriorityQueue: 非线程安全的优先队列。

## 17. 列出几种线程池

- 1.ThreadPoolExecutor: 通过 Executors 可以构造单线程池、固定数目线程池、不固定数目线程池。

- 2.ScheduledThreadPoolExecutor：可以延时调用线程或者延时重复调度线程。

## 18. 解释一下 java 内存模型

- 1.每个线程有自己的共享变量副本(实例域、静态域、数组元素)
- 2.Java 线程之间的通讯由 java 内存模型控制(JMM)，JMM 决定了一个线程对共享变量的写入在什么时候对另一个线程可见。
- 3.当一个 A 线程改变了一个共享变量，此时只是改变了自己本地共享变量的副本，所以线程之间要实现通讯需要先将 A 线程的本地副本，刷入到主存之中，然后线程 B 去主存中读取线程 A 刷入的改变的变量。

## 19. 解释一下 java 代码的原子性

- 1.除了 long 和 double，其他基本类型的简单操作都是原子性操作：读取、写入数字。
- 2.java1.5 之后使用 volatile 关键字能够让 long 和 double 像其他基本类型一样。
- 3.不能因为有原子性，就放弃同步。

## 20. 解释一下 volatile 字段###

- 1.在了解了 java 内存模型之后，我们知道 volatile 关键字保证了，某个共享变量改变之后，另一个线程中本地共享变量的副本也会立即刷新。
- 2.当某个域的值依赖于它之前的值，如计数递增那么 volatile 就会失效。
- 3.使用 volatile 而不使用 synchronized 的唯一情况就是类中只有一个可变域，其他时候都应该使用 synchronized。

## 21. 解释一下 synchronized 字段

- 1.synchronized 可以用于修饰方法，在某线程从某个用 synchronized 修饰的方法返回之前，其他所有要调用这个对象中任意使用 synchronized 修饰的方法的线程都会被阻塞。即同一时刻某个对象中只能有一个 synchronized 修饰的方法被调用。
- 2.对于某个对象，其所有 synchronized 方法共享一把锁，也就是说一个对象中含有一把锁(也叫监视器)。
- 3.一个线程可以多次获取某个对象的锁，比如一个 synchronized 方法中调用了该对象中另一个 synchronized 方法。那么这个锁的计数就会变成 2，每从一个 synchronized 方法离开都会使计数减一，每进入一个 synchronized 方法都会使计数加一。
- 4.每个类也有一把锁(属于类的 Class 的一部分)，所以 synchronized static 可以防止类范围内对于 static 数据的并发访问。

## 22. Lock 的使用

- 1.在 `lock()`和 `unlock()`之间我们可以创建一个临界资源，同时一时刻只能有一个线程访问这个资源。。
- 2.我们可以对临界区域进行异常捕捉，注意 `return` 必须在 `try` 里面，防止过早释放临界资源。
- 3.使用 `Lock` 的好处就在于我们能处理某些异常，而使用 `synchronized` 我们无法对异常进行清理工作。

## 23. `wait()` 和 `sleep()` 区别

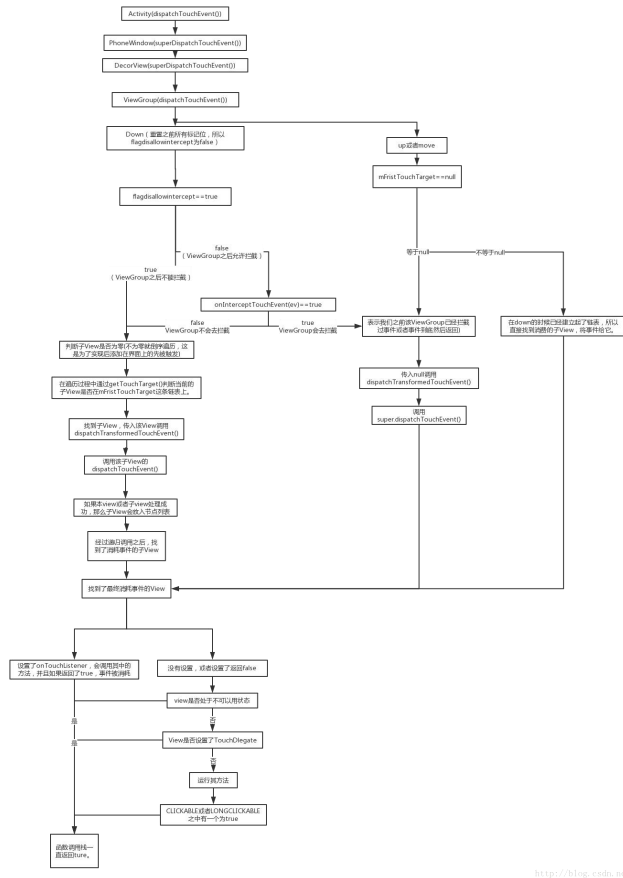
- 1.该方法在 `Object` 中，可以将当前线程挂起，需要在 `synchronized` 控制的块中，因为 `wait` 需要获取需要挂起的线程所要处理对象的锁，否则会报错。在 `wait()`中锁是被释放的。
- 2.该方法是属于 `Thread` 类中、`sleep()`方法导致了程序暂停执行指定的时间，让出 `cpu` 该其他线程，线程不会释放对象锁。

## 24. 死锁产生的条件

- 1.至少有一个资源是不能被共享的
- 2.至少有一个任务持有一个资源并且在等待另一个资源
- 3.资源不能被抢占
- 4.必须有循环等待

# 二、Android 面试题

1.Android 事件分发机制，请详细说下整个流程



## 2.Android view 绘制机制和加载过程，请详细说下整个流程

1.ViewRootImpl 会调用 performTraversals(),其内部会调用 performMeasure()、performLayout、performDraw()。

2.performMeasure() 会调用最外层的 ViewGroup 的 measure()-->onMeasure(),ViewGroup 的 onMeasure()是抽象方法，但其提供了 measureChildren()，这之中会遍历子 View 然后循环调用 measureChild()这之中会调用 getChildMeasureSpec()+父 View 的 MeasureSpec+子 View 的 LayoutParam 一起获取本 View 的 MeasureSpec，然后调用子 View 的 measure()到 View 的 onMeasure()-->setMeasureDimension(getDefaultSize(),getDefaultSize()),getD



efaultSize()默认返回 measureSpec 的测量数值，所以继承 View 进行自定义的 wrap\_content 需要重写。

3.performLayout()会调用最外层的 ViewGroup 的 layout(l,t,r,b),本 View 在其中使用 setFrame()设置本 View 的四个顶点位置。在 onLayout(抽象方法)中确定子 View 的位置，如 LinearLayout 会遍历子 View，循环调用 setChildFrame()--> 子 View.layout()。

4.performDraw() 会调用最外层 ViewGroup 的 draw(): 其中会先后调用 background.draw()(绘制背景)、onDraw()(绘制自己)、dispatchDraw()(绘制子 View)、onDrawScrollBars()(绘制装饰)。

5.MeasureSpec 由 2 位 SpecMode(UNSPECIFIED、EXACTLY(对应精确值和 match\_parent)、AT\_MOST(对应 wrap\_content)) 和 30 位 SpecSize 组成一个 int,DecorView 的 MeasureSpec 由窗口大小和其 LayoutParams 决定 其他 View 由父 View 的 MeasureSpec 和本 View 的 LayoutParams 决定。ViewGroup 中有 getChildMeasureSpec()来获取子 View 的 MeasureSpec。

6.三种方式获取 measure()后的宽高：

- 1.Activity#onWindowFocusChange()中调用获取
- 2.view.post(Runnable)将获取的代码投递到消息队列的尾部。
- 3.ViewTreeObservable.

3.activity 的加载过程 请详细介绍下:

- 1.Activity 中 最 终 到 startActivityForResult()  
( mMainThread.getApplicationThread() 传入了一个 ApplicationThread 检查 APT )

->Instrumentation#execStartActivity()和 checkStartActivityResult()(这是在启动了 Activity 之后判断 Activity 是否启动成功，例如没有在 AM 中注册那么就会报错)

->ActivityManagerNative.getDefault().startActivity() ( 类似 AIDL , 实现了 IAM , 实际是由远端的 AMS 实现 startActivity() )

->ActivityStackSupervisor#startActivityMayWait()

->ActivityStack#resumeTopActivityInnerLocked

->ActivityStackSupervisor#realStartActivityLocked() ( 在 这 里 调 用 APT 的 scheduleLaunchActivity,也是 AIDL , 不过是在远端调起了本进程 Application 线程 )

->ApplicationThread#scheduleLaunchActivity() ( 这是本进程的一个线程，用于作为 Service 端来接受 AMS client 端的调起 )

->ActivityThread#handleLaunchActivity() ( 接 收 内 部 类 H 的 消 息 , ApplicationThread 线程发送 LAUNCH\_ACTIVITY 消息给 H )

->最终在 ActivityThread#performLaunchActivity()中实现 Activity 的启动完成了以下几件事：

- 2.从传入的 ActivityClientRecord 中获取待启动的 Activity 的组件信息
- 3.创建类加载器，使用 Instrumentation#newActivity()加载 Activity 对象
- 4.调用 LoadedApk.makeApplication 方法尝试创建 Application，由于单例所以不会重复创建。

- 5.创建 Context 的实现类 ContextImpl 对象 ,并通过 Activity#attach()完成数据初始化和 Context 建立联系 , 因为 Activity 是 Context 的桥接类 ,  
最后就是创建和关联 window ,让 Window 接收的事件传给 Activity ,在 Window 的创建过程中会调用 ViewRootImpl 的 performTraversals()初始化 View。
- 6.Instrumentation#callActivityOnCreate()->Activity#performCreate()->Activity#onCreate().onCreate() 中 会 通 过 Activity#setContentView() 调用 PhoneWindow 的 setContentView()  
更新界面。

#### 4.Activity 的启动模式：

- 1.standard:默认标准模式，每启动一个都会创建一个实例，
- 2.singleTop：栈顶复用，如果在栈顶就调用 onNewIntent 复用，从 onResume() 开始
- 3.singleTask：栈内复用，本栈内只要用该类型 Activity 就会调到栈顶复用，从 onResume()开始
- 4.singleInstance：单例模式，除了 3 中特性，系统会单独给该 Activity 创建一个栈，

#### 5.Activity 缓存方法：

- 1.配置改变导致 Activity 被杀死，横屏变竖屏：在 onStop 之前会调用 onSaveInstanceState()保存数据在重建 Activity 之后，会在 onStart()之后调用

onRestoreInstanceState(),并把保存下来的 Bundle 传给 onCreate()和它会默认重建

Activity 当前的视图，我们可以在 onCreate()中，回复自己的数据。

- 2.内存不足杀掉 Activity，优先级分别是：前台可见，可见非前台，后台。

6.Service 的生命周期，两种启动方法，有什么区别：

- 1.context.startService() ->onCreate()->onStart()->Service running-->(如果调用 context.stopService() )->onDestroy() ->Service shut down
- 1.如果 Service 还没有运行，则调用 onCreate()然后调用 onStart()；
- 2.如果 Service 已经运行 则只调用 onStart() ,所以一个 Service 的 onStart 方法可能会重复调用多次。
- 3.调用 stopService 的时候直接 onDestroy，
- 4.如果是调用者自己直接退出而没有调用 stopService 的话，Service 会一直在后台运行。该 Service 的调用者再启动起来后可以通过 stopService 关闭 Service。
- 2.context.bindService()->onCreate()->onBind()->Service running-->onUnbind() -> onDestroy() ->Service stop
- 1.onBind 将返回给客户端一个 IBind 接口实例 ,IBind 允许客户端回调服务的方法，比如得到 Service 运行的状态或其他操作。
- 2.这个时候会把调用者和 Service 绑定在一起，Context 退出了,Service 就会调用 onUnbind->onDestroy 相应退出。
- 3.所以调用 bindService 的生命周期为 :onCreate --> onBind(只一次，不可多次绑定) --> onUnbind --> onDestory。

## 7.怎么保证 service 不被杀死

- 1.提升 service 优先级
- 2.提升 service 进程优先级
- 3.onDestroy 方法里重启 service

## 8.广播的两种注册方法，有什么区别。

- 1.静态注册:如果有广播信息来，你写的广播接收器同样的能接受到,比如系统的一些广播。
- 2.动态注册:当应用程序结束了，广播自然就没有了，一些自己定义的广播

## 9.Intent 可以传递哪些数据类型

- 1.Serializable
- 2.charsequence: 主要用来传递 String，char 等
- 3.parcelable
- 4.Bundle

## 10.Json 有什么优劣势

- 1.JSON 的速度要远远快于 XML
- 2.JSON 相对于 XML 来讲，数据的体积小
- 3.JSON 对数据的描述性比 XML 较差

## 11.动画有哪几类，各有什么特点：

- 1.动画的基本原理：其实就是利用插值器和估值器，来计算出各个时刻 View 的属性，然后通过改变 View 的属性来，实现 View 的动画效果。
- 2.View 动画:只是影像变化，view 的实际位置还在原来的地方。
- 3.帧动画是在 xml 中定义好一系列图片之后，使用 AnimationDrawable 来播放的动画。
- 4.View 的属性动画：
  - 1.插值器：作用是根据时间的流逝的百分比来计算属性改变的百分比
  - 2.估值器：在 1 的基础上由这个东西来计算出属性到底变化了多少数值的类

## 12.Handler、Loop 消息队列模型，各部分的作用。

- 1.MessageQueue：读取会自动删除消息，单链表维护，在插入和删除上有优势。在其 next()中会无限循环，不断判断是否有消息，有就返回这条消息并移除。
- 2.Looper：Looper 创建的时候会创建一个 MessageQueue，调用 loop()方法的时候消息循环开始，loop()也是一个死循环，会不断调用 messageQueue 的 next()，当有消息就处理，否则阻塞在 messageQueue 的 next()中。当 Looper 的 quit()被调用的时候会调用 messageQueue 的 quit()，此时 next()会返回 null，然后 loop()方法也跟着退出。
- 3.Handler：在主线程构造一个 Handler，然后在其他线程调用 sendMessage()，此时主线程的 MessageQueue 中会插入一条 message，然后被 Looper 使用。
- 4.系统的主线程在 ActivityThread 的 main()为入口开启主线程，其中定义了内部类 Activity.H 定义了一系列消息类型，包含四大组件的启动停止。

13. 怎样退出终止 App:自己设置一个 Activity 的栈，然后一个个 finish()。

#### 14. Android IPC:Binder 原理

- 1.在 Activity 和 Service 进行通讯的时候，用到了 Binder。
- 1.当属于同个进程我们可以继承 Binder 然后在 Activity 中对 Service 进行操作
- 2.当不属于同个进程，那么要用到 AIDL 让系统给我们创建一个 Binder，然后在 Activity 中对远端的 Service 进行操作。
- 2.系统给我们生成的 Binder：
- 1.Stub 类中有：接口方法的 id，有该 Binder 的标识，有 asInterface(IBinder)(让我们在 Activity 中获取实现了 Binder 的接口，接口的实现在 Service 里，同进程时候返回 Stub 否则返回 Proxy)，有 onTransact()这个方法是在不同进程的时候让 Proxy 在 Activity 进行远端调用实现 Activity 操作 Service
- 2.Proxy 类是代理，在 Activity 端，其中有:IBinder mRemote(这就是远端的 Binder)，两个接口的实现方法不过是代理最终还是要远端的 onTransact()中进行实际操作。
- 3.哪一端的 Binder 是副本，该端就可以被另一端进行操作，因为 Binder 本体在定义的时候可以操作本端的东西。所以可以在 Activity 端传入本端的 Binder，让 Service 端对其进行操作称为 Listener，可以用 RemoteCallbackList 这个容器来装 Listener，防止 Listener 因为经历过序列化而产生的问题。

- 4.当 Activity 端向远端进行调用的时候，当前线程会挂起，当方法处理完毕才会唤醒。
- 5.如果一个 AIDL 就用一个 Service 太奢侈，所以可以使用 Binder 池的方式，建立一个 AIDL 其中的方法是返回 IBinder，然后根据方法中传入的参数返回具体的 AIDL。
- 6.IPC 的方式有：Bundle（在 Intent 启动的时候传入，不过是一次性的），文件共享(对于 SharedPreferences 是特例，因为其在内存中会有缓存)，使用 Messenger(其底层用的也是 AIDL，同理要操作哪端，就在哪端定义 Messenger)，AIDL，ContentProvider(在本进程中继承实现一个 ContentProvider，在增删改查方法中调用本进程的 SQLite，在其他进程中查询)，Socket

## 15.android 的优化

## 16.一个 singleton 如何实现线程的同步问题

- 1.单例类确保自己只有一个实例(构造函数私有:不被外部实例化,也不被继承)。
- 2.单例类必须自己创建自己的实例。
- 3.单例类必须为其他对象提供唯一的实例。

## 17.android 重要术语解释

- 1.ActivityManagerServices，简称 AMS，服务端对象，负责系统中所有 Activity 的生命周期
- 2.ActivityThread，App 的真正入口。当开启 App 之后，会调用 main()开始运行，开启消息循环队列，这就是传说中的 UI 线程或者叫主线程。与 ActivityManagerServices 配合，一起完成 Activity 的管理工作



- 3.ApplicationThread , 用来实现 ActivityManagerService 与 ActivityThread 之间的交互。在 ActivityManagerService 需要管理相关 Application 中的 Activity 的生命周期时 , 通过 ApplicationThread 的代理对象与 ActivityThread 通讯。
- 4.ApplicationThreadProxy , 是 ApplicationThread 在服务器端的代理 , 负责和客户端的 ApplicationThread 通讯。AMS 就是通过该代理与 ActivityThread 进行通信的。
- 5.Instrumentation , 每一个应用程序只有一个 Instrumentation 对象 , 每个 Activity 内都有一个对该对象的引用。Instrumentation 可以理解为应用进程的管家 , ActivityThread 要创建或暂停某个 Activity 时 , 都需要通过 Instrumentation 来进行具体的操作。
- 6.ActivityStack , Activity 在 AMS 的栈管理 , 用来记录已经启动的 Activity 的先后关系 , 状态信息等。通过 ActivityStack 决定是否需要启动新的进程。
- 7.ActivityRecord , ActivityStack 的管理对象 , 每个 Activity 在 AMS 对应一个 ActivityRecord , 来记录 Activity 的状态以及其他的管理信息。其实就是服务器端的 Activity 对象的映像。
- 8.TaskRecord , AMS 抽象出来的一个 “任务” 的概念 , 是记录 ActivityRecord 的栈 , 一个 “Task” 包含若干个 ActivityRecord。AMS 用 TaskRecord 确保 Activity 启动和退出的顺序。如果你清楚 Activity 的 4 种 launchMode , 那么对这个概念应该不陌生。

## 18.理解 Window 和 WindowManager

- 1.Window 用于显示 View 和接收各种事件 ,Window 有三种类型 :应用 Window(每个 Activity 对应一个 Window)、子 Window(不能单独存在 , 附属于特定 Window)、系统 window Toast 和状态栏)
- 2.Window 分层级 , 应用 Window 在 1-99、子 Window 在 1000-1999、系统 Window 在 2000-2999.WindowManager 提供了增删改 View 三个功能。
- 3.Window 是个抽象概念 : 每一个 Window 对应着一个 View 和 ViewRootImpl , Window 通过 ViewRootImpl 来和 View 建立联系 , View 是 Window 存在的实体 , 只能通过 WindowManager 来访问 Window。
- 4.WindowManager 的实现是 WindowManagerImpl 其再委托给 WindowManagerGlobal 来对 Window 进行操作 , 其中有四个 List 分别储存对应的 View、ViewRootImpl、WindowManger.LayoutParams 和正在被删除的 View
- 5.Window 的实体是存在于远端的 WindowMangerService 中 , 所以增删改 Window 在本端是修改上面的几个 List 然后通过 ViewRootImpl 重绘 View , 通过 WindowSession(每个应用一个)在远端修改 Window。
- 6.Activity 创建 Window : Activity 会在 attach()中创建 Window 并设置其回调 (onAttachedToWindow()、dispatchTouchEvent()),Activity 的 Window 是由 Policy 类创建 PhoneWindow 实现的。然后通过 Activity#setContentView() 调用 PhoneWindow 的 setContentView。

19.Bitmap 的处理 :

- 1.当使用 ImageView 的时候，可能图片的像素大于 ImageView，此时就可以通过 BitmapFactory.Options 来对图片进行压缩，inSampleSize 表示缩小  $2^{(inSampleSize-1)}$  倍。
- 2.Bitmap 的缓存：
  - 1.使用 LruCache 进行内存缓存。
  - 2.使用 DiskLruCache 进行硬盘缓存。
  - 3.实现一个 ImageLoader 的流程：同步异步加载、图片压缩、内存硬盘缓存、网络拉取
    - 1.同步加载只创建一个线程然后按照顺序进行图片加载
    - 2.异步加载使用线程池，让存在的加载任务都处于不同线程
    - 3.为了不开启过多的异步任务，只在列表静止的时候开启图片加载

## 20.综合技术：

- 1.CrashHandler：获取 app crash 的信息保存在本地然后在下一次打开 app 的时候发送到服务器。
- 2.multidex 解决方法数过大的问题

## 21.如何实现一个网络框架(参考 Volley)

- 1.缓存队列,以 url 为 key 缓存内容可以参考 Bitmap 的处理方式，这里单独开启一个线程。
- 2.网络请求队列，使用线程池进行请求。

- 3.提供各种不同类型的返回值的解析如 String , Json , 图片等等。

## 22.ClassLoader 的基础知识：

- 1.双亲委托：一个 ClassLoader 类负责加载这个类所涉及的所有类，在加载的时候会判断该类是否已经被加载过，然后会递归去他父 ClassLoader 中找。
- 2.可以动态加载 Jar 通过 URLClassLoader
- 3.ClassLoader 隔离问题 JVM 识别一个类是由：ClassLoader id+PackageName+ClassName。
- 4.加载不同 Jar 包中的公共类：
  - 1.让父 ClassLoader 加载公共的 Jar，子 ClassLoader 加载包含公共 Jar 的 Jar，此时子 ClassLoader 在加载公共 Jar 的时候会先去父 ClassLoader 中找。(只适用 Java)
  - 2.重写加载包含公共 Jar 的 Jar 的 ClassLoader，在 loadClass 中找到已经加载过公共 Jar 的 ClassLoader，也就是把父 ClassLoader 替换掉。(只适用 Java)
  - 3.在生成包含公共 Jar 的 Jar 时候把公共 Jar 去掉。

## 23. 插件化框架描述：dynamicLoadApk 为例子

- 1.可以通过 DexClassLoader 来对 apk 中的 dex 包进行加载访问
- 2.如何加载资源是个很大的问题，因为宿主程序中并没有 apk 中的资源，所以调用 R 资源会报错，所以这里使用了 Activity 中的实现 ContextImpl 的 getAssets()和 getResources()再加上反射来实现。

- 3.由于系统启动 Activity 有很多初始化动作要做，而我们手动反射很难完成，所以可以采用接口机制，将 Activity 的大部分生命周期提取成接口，然后通过代理 Activity 去调用插件 Activity 的生命周期。同时如果像增加一个新生命周期方法的时候，只需要在接口中和代理中声明一下就行。
- 4.缺点：
  - 1.慎用 this，因为在 apk 中使用 this 并不代表宿主中的 activity，当然如果 this 只是表示自己的接口还是可以的。除此之外可以使用 that 代替 this。
  - 2.不支持 Service 和静态注册的 Broadcast
  - 3.不支持 LaunchMode 和 Apk 中 Activity 的隐式调用。

#### 24.热修复：Andfix 为例子

- 1.大致原理：apkpatch 将两个 apk 做一次对比，然后找出不同的部分。可以看到生成的 apatch 了文件，后缀改成 zip 再解压开，里面有一个 dex 文件。通过 jadx 看一下源码，里面就是被修复的代码所在的类文件，这些更改过的类都加上了一个 \_CF 的后缀，并且变动的方法都被加上了一个叫 @MethodReplace 的 annotation，通过 clazz 和 method 指定了需要替换的方法。然后客户端 sdk 得到补丁文件后就会根据 annotation 来寻找需要替换的方法。最后由 JNI 层完成方法的替换。
- 2.无法添加新类和新的字段、补丁文件很容易被反编译、加固平台可能会使热补丁功能失效

## 三、数据库

### 1.Sqlite 升级，增加字段的语句

[blog.csdn.net/qq\\_26287435...](http://blog.csdn.net/qq_26287435...)

SQLite supports a limited subset of ALTER TABLE.

The ALTER TABLE command in SQLite allows the user to rename a table or to add a new column to an existing table.

```
alter table table_name  
add column 'new_col' varchar(100)  
default '1'
```

## 2. 数据库框架对比和源码分析

[www.codexiu.cn/android/blo...](http://www.codexiu.cn/android/blo...)

ORMLite: [github.com/j256/ormlite...](https://github.com/j256/ormlite...)

Afinal: [github.com/yangfuhai/a...](https://github.com/yangfuhai/a...)

ActiviteAndroid: [github.com/pardom/Acti...](https://github.com/pardom/Acti...)

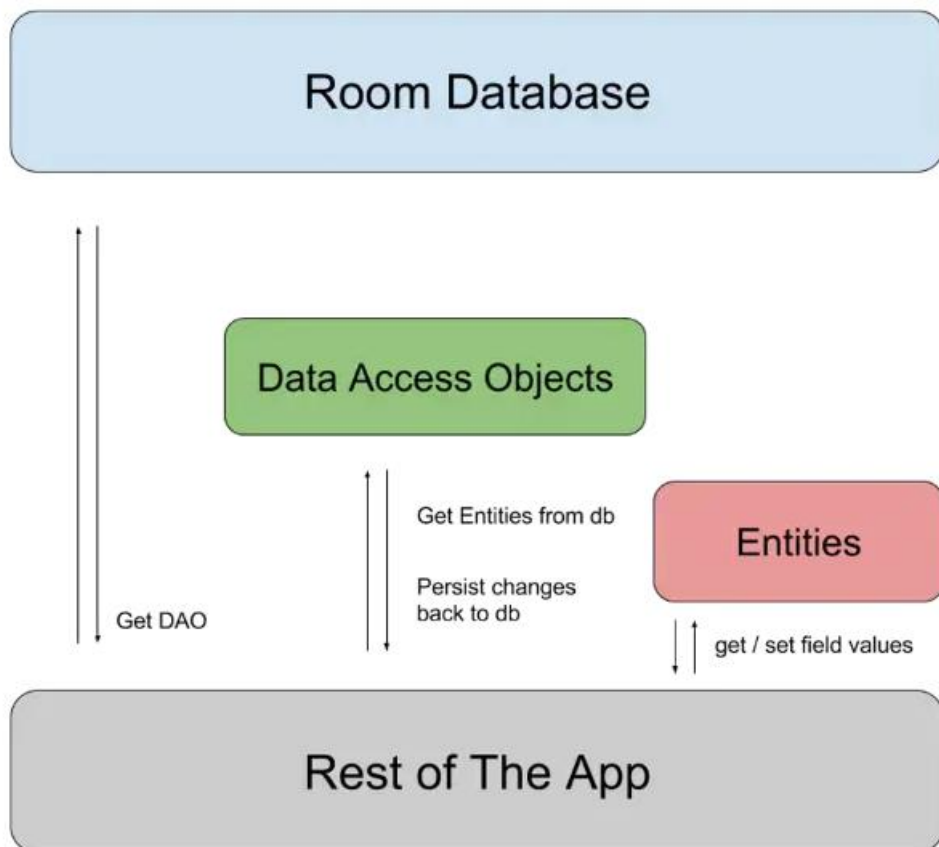
SugarORM: Android 平台专用 ORM

GreenDao: [github.com/greenrobot/...](https://github.com/greenrobot/...)

Realm: [github.com/realm/realm...](https://github.com/realm/realm...)

	A	B	C	D	E	F
1	1条		添加	删除	查询	
2	GreenDao		6毫秒	7毫秒	2毫秒	
3	Realm		5毫秒	8毫秒	1毫秒	
4						
5	10条		添加	删除	查询	
6	GreenDao		6毫秒	9毫秒	2毫秒	
7	Realm		6毫秒	11毫秒	1毫秒	
8						
9	100条		添加	删除	查询	
10	GreenDao		530毫秒	11毫秒	2毫秒	
11	Realm		15毫秒	18毫秒	1毫秒	
12						
13	1000条		添加	删除	查询	
14	GreenDao		5400毫秒	10毫秒	11毫秒	
15	Realm		68毫秒	150毫秒	1毫秒	
16						
17	10000条		添加	删除	查询	
18	GreenDao		55023毫秒	13毫秒	207毫秒	
19	Realm		800毫秒	4500毫秒	4毫秒	
20						
21	100000条		添加	删除	查询	
22	GreenDao		无	无	无	
23	Realm		10253毫秒	290217毫秒	42毫秒	
24						
25						

[www.jianshu.com/p/5eab05820...](http://www.jianshu.com/p/5eab05820...)



### GreenDao

主键使用@Id 注解标注，可以设置 `autoincrement = true` 让主键自增，注意：这里主键类型必须是 Long，因为只有 `id=null` 时自增才会生效

如果不需要时使用 `Transient` 注解忽略此属性

支持对象之间的关联（一对一，一对多等等）

### Room

主键使用 `PrimaryKey` 注解，可以设置 `autoGenerate = true` 表示主键自增

使用 `@Ignore` 注解，表示忽略此属性

允许 `entity` 对象相互引用，Room 仍然允许你定义 `entity` 之间的外键（Foreign Key）约束



**Room:** 1、使用 Room 时需要创建自己的 AppDatabasele 抽象类，并且继承自 RoomDatabase。

2、使用 @Database 注解标注，其中可以指定 entitys，将这个库中所有的表对应的实体放在 entitys 中，version 字段可以指定数据库版本。

3、Dao 对象需要开发者编写各 entity 对应的 Dao 接口，并用 @Dao 注解标注，接口中提供需要的方法声明。Room 会自动生成对应的实现类，并实现所有的方法。

```
@Daopublic interface ProductDao {

    @Query("select * from products")
    List<ProductEntity> queryProducts();

    @Query("select * from products where id = :productId")
    ProductEntity getProductById(long productId);

    @Insert(onConflict = OnConflictStrategy.REPLACE)
    void insertAll(ProductEntity... entity);

    @Update(onConflict = OnConflictStrategy.REPLACE)
    void updateAll(ProductEntity... entity);

    @Delete
    void deleteAll(ProductEntity... entity);

}
```

4、在 AppDatabase 中提供对应实体的 Dao 的方法声明。这样就可以通过 APPDatabase 实例获取到指定的 dao 对应从而操作表了。说明：Room 会自动生成该类的实现类。使用时直接调用即可。代码如下：

```
@Database(entities = {ProductEntity.class,
CommentEntity.class},version =
1)@TypeConverters(DateConverter.class)public abstract class
AppDatabase extends RoomDatabase {
    public static String DATABASE_NAME = "room-sample-db";

    public abstract ProductDao productDao();

    public abstract CommentDao commentDao();
}
```

5、获取数据库实例

使用 Room 提供的静态方法创建数据库。代码如下：

```
appDatabase = Room.databaseBuilder(context.getApplicationContext(),
    AppDatabase.class, DATABASE_NAME)
    .addMigrations(MIGRATION_1_2)
    .build();复制代码
```

## GreenDao

1、使用 GreenDao 时会自动生成 DaoMaster，DaoSession，XXXEntityDao 三个类。

### DaoMaster

- 是 GreenDao 的入口，也是 greenDao 顶级对象，对于一个指定的表单持有数据库对象（SQLite 数据库）并且能够管理 DAO 类，能够创建表和删除表
- 其内部类OpenHelper与DevOpenHelper是创建SQLite数据库的SQLiteOpenHelper的具体实现

### DaoSession

- 对于一个指定的表单可以管理所有的 Dao 对象。
- 也能够对实体类执行 insert load update refresh delete 操作。
- DaoSession 也能跟踪 identity scope: 即 session 查询后的实体会存在缓存中，并给该实体生成一个 flag 来追踪该实体，下次再次查询时会直接从缓存中取出来而不是从数据库中取出来

### Daos

- 能够持久访问和查询实体类
- 比起 DaoSession 有更多的持久化方法 count loadAll insertInt 等等；

2、获取数据库实例

1)、首先需要获取 SQLiteOpenHelper 对象。可以通过 DaoMaster 中的 DevOpenHelper 来实例化，也可以自定义类实现 DaoMaster 中的 OpenHelper

```
SQLiteOpenHelper oh = new DaoMaster.DevOpenHelper(context, DB_NAME, null);
复制代码
```

2)、获取 DaoSession: 通过 DaoMaster 的 newSession()获取

```
mDatabase = oh.getWritableDatabase();
daoSession = new DaoMaster(mDatabase).newSession();复制代码
```

3)、获取 Dao 对象：通过 DaoSession 获取

`daoSession.getProductEntityDao();`复制代码

操作 **Sql**(增删改查) **insert, update, delete Room**: 使用 **@Insert, @Update, @Delete** 注解,并且可以指定相应的策略(**replace, rollback, fail** 等) **GreenDao**:直接生成对应的方法。没有提供相应的策略。但是提供了执行 **sql** 的方法。

**query Room**:使用 **@Query** 注解标注, 直接传 **sql**

**GreenDao**: 以对象的方式操作。可以使用 **QueryBuilder** 类执行各种复杂的查询。

两者做法各有千秋, 就看使用者的习惯。就我个人而言还更喜欢使用纯 **sql**, 可读性高。

加密 **Room** 不支持 **SqlCipher** 加密, 不过后续 **SqlCipher** 应该会提供针对 **Room** 的加密方式, 这只是时间的问题。

**GreenDao** 集成了 **SqlCipher** 加密。

数据库升级

**Room**

生成 **AppDatabase** 实例时可以通过 **addMigrations()**添加对应的 **Migration** 来支持迁移操作。代码如下:

```
appDatabase = Room.databaseBuilder(context.getApplicationContext(),
    AppDatabase.class, DATABASE_NAME)
    .addMigrations(MIGRATION_1_2)
    .build();
    private static Migration MIGRATION_1_2 = new Migration(1,2) {
    @Override
    public void migrate(SupportSQLiteDatabase database) {
        // TODO:
    }
};
```

复制代码

**GreenDao** 通过重写 **DaoMaster.OpenHelper** 类实现迁移操作:

```
public static class AppSQLiteOpenHelper extends DaoMaster.OpenHelper {
    public AppSQLiteOpenHelper(Context context, String name) {
        super(context, name);
    }

    public AppSQLiteOpenHelper(Context context, String name,
        SQLiteDatabase.CursorFactory factory) {
```

```

        super(context, name, factory);
    }

    @Override
    public void onUpgrade(Database db, int oldVersion, int newVersion)
    {
        //数据库迁移操作
        MigrationHelper.getInstance().migrate(db,
        ProductEntityDao.class, CommentEntityDao.class);
    }
}复制代码

```

**Room 添加对 liveData 的支持**

**GreenDao 不支持 LiveData，需要自己实现。**

**2-0.realm 使用**

```

@Entity(indices = {@Index(value = {"first_name", "last_name"},
    unique = true)})public class User {
    @PrimaryKey
    public int id;

    @ColumnInfo(name = "first_name")
    public String firstName;

    @ColumnInfo(name = "last_name")
    public String lastName;

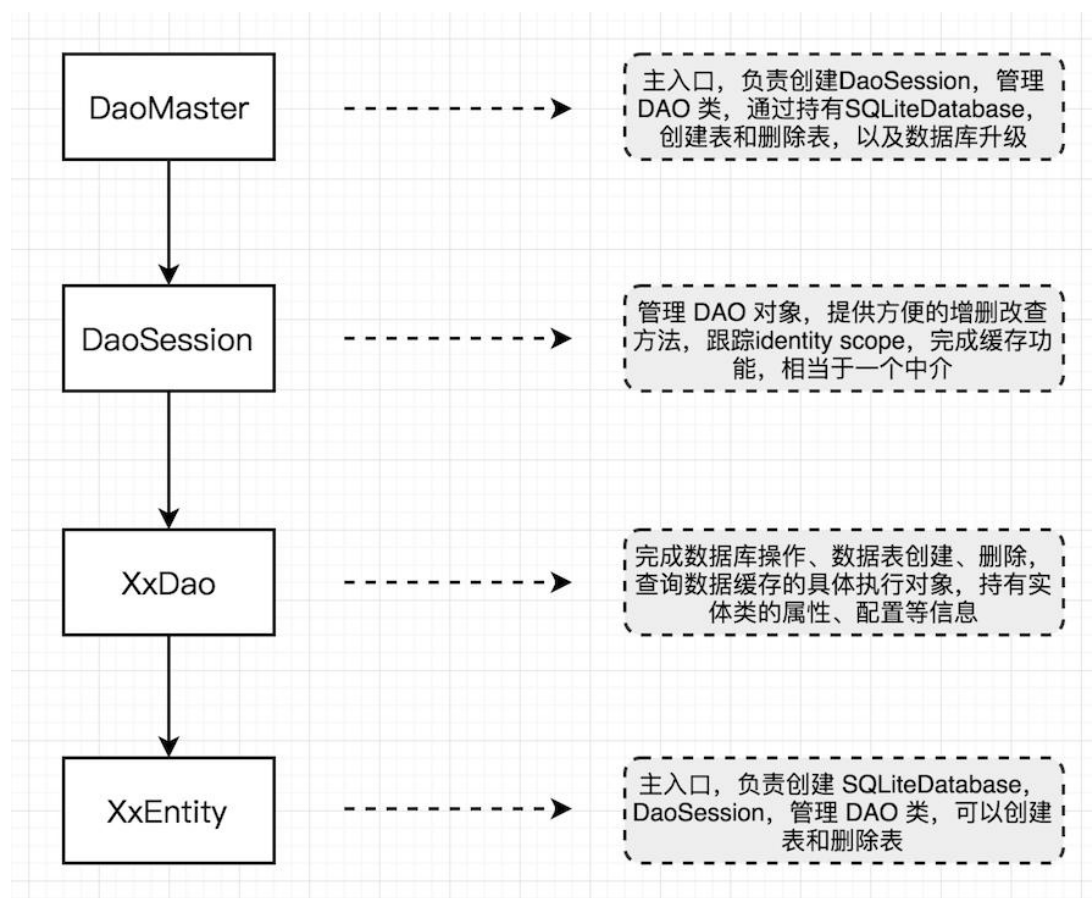
    @Ignore
    Bitmap picture;
}
@Entity(foreignKeys = @ForeignKey(entity = User.class,
    parentColumns = "id",
    childColumns = "user_id"))public class Book
{
    @PrimaryKey
    public int bookId;

    public String title;

    @ColumnInfo(name = "user_id")
    public int userId;
}复制代码

```

**2-1.GreenDao 源码分析**



**DaoMaster**: 使用 greenDAO 的切入点。 DaoMaster 保存数据库对象 (SQLiteDatabase) 并管理特定模式的 DAO 类 (而不是对象)。 它有静态方法来创建表或删除它们。 它的内部类OpenHelper和DevOpenHelper都是SQLiteOpenHelper的实现，用来在SQLite数据库中创建和升级等操作。

**DaoSession**: 管理特定模式的所有可用 DAO 对象，你可以使用其中一个的getter方法获取 DAO 对象。 DaoSession 还为实体提供了一些通用的持久性方法，如插入，加载，更新，刷新和删除。 最后，DaoSession 对象也跟踪 identity scope。

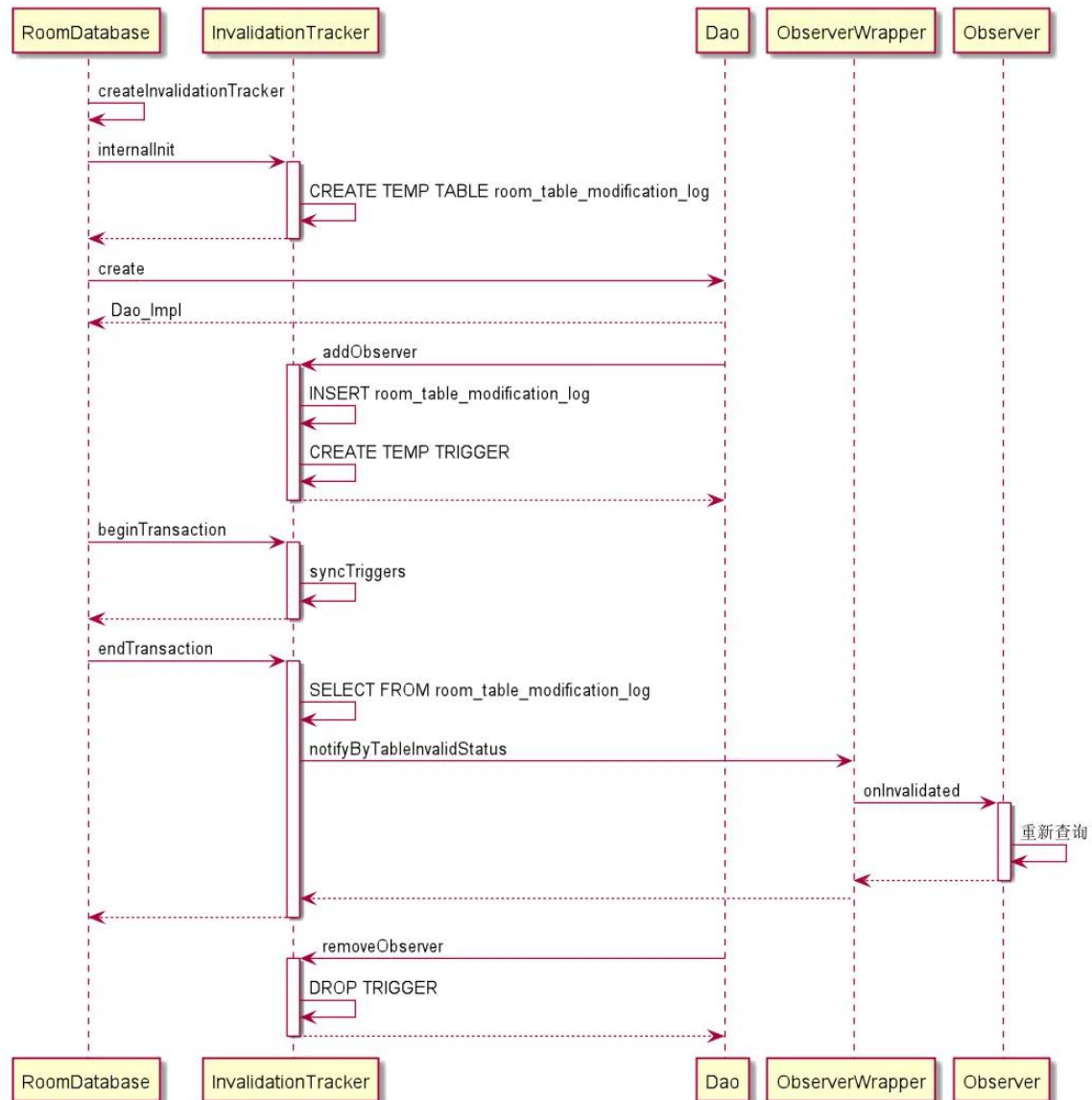
**xxDAO**: 数据访问对象 (DAO)，用于实体的持久化和查询。 对于每个实体，greenDAO 会生成一个 DAO。 它比 DaoSession 拥有更多的持久化方法，例如：count，loadAll 和 insertInTx。

DevOpenHelper --> OpenHelper --> DatabaseOpenHelper --> SQLiteDatabaseOpenHelper

理解了 DaoMaster、DaoSession、xxDAO 这几个类， greenDAO 也就基本理解了，这里给出一个比喻， DaoMaster 相当于一个 Boss，统领大局，但是具体的工作会交给 DaoSession， DaoSession 相当于一个 Leader，会负责管理一些 xxDAO，当然也可以有多个 DaoSession， DaoSession 会提供数据库操作的一些方便的入口方法，其内部具体的执行会交给 xxDAO 执行，xxDAO 负责自己对应的实体类对象的增删改查，其中执行过程中需要 DaoConfig 和

DatabaseStatement，拿到拼接的 sql 语句和要存储的实体类的参数值，最终会通过原生数据库完成操作，基本就是这样一个过程，可以理解为职责一步步向下传递的过程。

## 2-1.Room 源码分析



图中有一些不合适的地方，removeObserver 并不是有 Dao 调用的，而是查询返回的 RxJava Observable 或者 LiveData “不再用”了的时候被各自调用的。

其中，“重新查询”的意思就是数据流被更新，数据流相关的 RxJava Observable 或者 LiveData 都没有在包含在图中。复制代码

[my.oschina.net/sfshine/blo...](https://my.oschina.net/sfshine/blo...)

Room 是 Google 提供的一个 ORM 库。Room 提供了三个主要的组件：

**@Database:** 用来注解类，并且注解的类必须是继承自 RoomDatabase 的抽象类。该类主要作用是创建数据库和创建 Daos（data access objects，数据访问对象）。

**@Entity**: 用来注解实体类, **@Database** 通过 **entities** 属性引用被**@Entity** 注解的类, 并利用该类的所有字段作为表的列名来创建表。

**@Dao**: 用来注解一个接口或者抽象方法, 该类的作用是提供访问数据库的方法。在使用**@Database** 注解的类中必须定一个不带参数的方法, 这个方法返回使用**@Dao** 注解的类

数据库的创建

数据库必须是一个抽象类 **RoomDatabase** 的扩展类

在注释中包括与数据库关联的实体列表 必须包含一个具有 0 个参数且返回带**@Dao** 注释的类的抽象方法

通过调用 **Room.databaseBuilder()** 或 **Room.inMemoryDatabaseBuilder()** 创建数据库实例

```
@Database(entities = {User.class}, version = 1) // 注释 public abstract
class AppDatabase extends RoomDatabase {
    public abstract UserDao userDao(); // 抽象方法
}复制代码
```

以单例形式对外提供 **RoomDataBase** 实例

```
public static UserDataBase getInstance(Context context) {
    if (userDataBase == null) {
        synchronized (UserDataBase.class) {
            if (userDataBase == null) {
                userDataBase =
Room.databaseBuilder(context.getApplicationContext()
                        , UserDataBase.class, "user_data").build();
            }
        }
    }
    return userDataBase;
}
```

定义实体数据: 表示数据库中的表 主键

```
@Entity(tableName = "userDataBase")class User {
    @PrimaryKey(autoGenerate = true) // 单个主键设置为自增长
    public var id = 0
    @ColumnInfo(name = "nameUser") // 定义列名
    public var name: String? = null
}
```

```

@Entity(primaryKeys = ["id", "name"]) // 组合主键复制代码
添加索引@Entity
使用 @Entity 的 indices 属性，列出要包含在索引或复合索引中的列的名称
@Entity(indices = [Index("nameUser"), Index(value = ["name"])])) // 创建索引
@Entity(indices = [Index("nameUser"), Index(value = ["name"], unique = true)]) //唯一索引

```

外键约束@ForeignKey

使用@ForeignKey 注释定义其与实体的 关系；ForeignKey 中 entity 为要关联的父实体类；

parentColumns 为关联父实体类的列名；childColumns 此实体类中的列名

// 在 User 实体中引入 Address

@Embedded

public var address: Address? = null 复制代码

```

@Entity(foreignKeys = [ForeignKey(entity = User::class,
    parentColumns = ["id"],
    childColumns = ["user_id"])]))class Book {

```

@PrimaryKey

var bookId: Int = 0

var title: String? = null

@ColumnInfo(name = "user\_id")

var userId: Int = 0

}

public class Converters {

@TypeConverter

public static Date fromTimestamp(Long value) {

return value == null ? null : new Date(value);

}

@TypeConverter

public static Long dateToTimestamp(Date date) {

return date == null ? null : date.getTime();

}

}复制代码

在抽象数据库类中添加转换注解

@TypeConverters({Converters.class})

使用 类型转换器

```

@Query("SELECT * FROM user WHERE birthday BETWEEN :from AND :to") List
findUsersBornBetweenDates(Date from, Date to);

```



### 3. 数据库的优化

[www.jianshu.com/p/8bef9e422...](http://www.jianshu.com/p/8bef9e422...)

```
String sql = "insert into tb_test values (?, 'test')";
```

```
SQLiteStatement sqlStatement  
=getReadableDatabase().compileStatement(sql);  
for (int count =0 ; count <1000 ; count++){  
    sqlStatement.clearBindings();  
    sqlStatement.bindLong(1, count);  
    sqlStatement.executeInsert();  
}
```

```
CREATE INDEX index_name ON table_name;复制代码
```

#### 四、分库分表

五、其他 少用 `cursor.getColumnIndex()`：使用 `cursor.getColumnIndex()`，系统会根据列名来获取列所在的下标，比较耗时，所以我们可以使用 `static` 定义下标，直接通过下标来获取某列。

使用 `StringBuilder` 或 `StringBuffer` 来拼接字符串：SQL 语句字符串的拼接或创建多个临时变量，此时我们可以使用 `StringBuilder` 或 `StringBuffer` 来拼接字符串，减少不必要的资源占用。

查询时，只返回自己需要的值或结果：如果在查询数据时直接将全部数据获取出来，这样的操作会带来不必要的系统资源开销和浪费。在查询时，我们尽量只取自己需要的字段和结果。

`cursor` 使用后要及时关闭：即在查询完结果后，调用 `cursor.close()` 将资源关闭。

### 4. 数据库数据迁移问题

[blog.csdn.net/github\\_3713...](http://blog.csdn.net/github_3713...)

数据库升级，主要有以下几种情况：

增加表 删除表 修改表：分为“增加表字段”以及“删除表字段”

先将旧的表删除再创建新的表，这是最简单暴力的，但前面提过这不是我们想要的结果。

主要思路是：首先将原来的表进行改名称 `rename table`（临时表），接着创建新的表 `create table`，再者将旧表内的数据迁移到新表内，最后 `drop table` 删除临时表。

[blog.csdn.net/a714530833/...](http://blog.csdn.net/a714530833/...)

## 5. Sqlite 常见异常

[blog.csdn.net/Gaugamela/a...](http://blog.csdn.net/Gaugamela/a...)

android.database.sqlite.SQLiteCantOpenDatabaseException: unable to open database file

[blog.csdn.net/gaugamela/a...](http://blog.csdn.net/gaugamela/a...)

android.database.sqlite.SQLiteCantOpenDatabaseException: unable to open database file

[blog.csdn.net/Aaren\\_Jiang...](http://blog.csdn.net/Aaren_Jiang...)

java.lang.IllegalStateException: attempt to re-open an already-closed object: SQLiteDatabase: /data/data/com.ant.sqlite3/databases/mydata.db

[echo.vars.me/android/sql...](http://echo.vars.me/android/sql...)

database is locked

attempt to re-open an already-closed object

SQLiteDatabase created and never closed

attempt to re-open an already-closed object

database disk image is malformed

[codeday.me/bug/2018072...](http://codeday.me/bug/2018072...)

Caused by: android.database.sqlite.SQLiteException: no such table: ligas\_bd: , while compiling: SELECT \* FROM ligas\_bd

[www.cnblogs.com/wangmars/p/...](http://www.cnblogs.com/wangmars/p/...)

java.lang.IllegalStateException: Cannot perform this operation because the connection pool has been closed. java.lang.IllegalStateException: attempt to re-open an already-closed object: SQLiteDatabase:

realm 无事务插入 1W 条 亲测 java.lang.IllegalStateException: Changing Realm data can only be done from inside a transaction.

[juejin.im/post/5a5f18...](http://juejin.im/post/5a5f18...)

Distinct

## 四、计算机网络

### 1. 网络分层

OSI 七层模型

OSI 七层协议模型主要是：应用层（Application）、表示层（Presentation）、会话层（Session）、传输层（Transport）、网络层（Network）、数据链路层（Data Link）、物理层（Physical）。

### 2. TCP/IP 五层模型

TCP/IP 五层模型：应用层（Application）、传输层（Transport）、网络层（Network）、数据链路层（Data Link）、物理层（Physical）。

### 3. 三次握手与四次挥手

第一次握手：客户端发送 syn 包(syn=j)到服务器，并进入 SYN\_SEND 状态，等待服务器确认；

第二次握手：服务器收到 syn 包，必须确认客户的 SYN (ack=j+1)，同时自己也发送一个 SYN 包 (syn=k)，即 SYN+ACK 包，此时服务器进入 SYN\_RECV 状态；

第三次握手：客户端收到服务器的 SYN+ACK 包，向服务器发送确认包 ACK(ack=k+1)，此包发送完毕，客户端和服务器进入 ESTABLISHED 状态，完成三次握手。

握手过程中传送的包里不包含数据，三次握手完毕后，客户端与服务器才正式开始传送数据。理想状态下，TCP 连接一旦建立，在通信双方中的任何一方主动关闭连接之前，TCP 连接都将被一直保持下去。断开连接时服务器和客户端均可以主动发起断开 TCP 连接的请求，断开过程需要经过“四次握手”

第一次挥手：客户端发送报文告诉服务器没有数据要发送了

第二次挥手：服务端收到，再发送给客户端告诉它我收到了

第三次挥手：服务端向客户端发送报文，请求关闭连接

第四次挥手：客户端收到关闭连接的请求，向服务端发送报文，服务端关闭连接

### 4. TCP 为什么三次握手不是两次握手，为什么两次握手不安全

为了实现可靠数据传输，TCP 协议的通信双方，都必须维护一个序列号，以标识发送出去的数据包中，哪些是已经被对方收到的。三次握手的过程即是通信双方相互告知序列号起始值，并确认对方已经收到了序列号起始值的必经步骤

如果只是两次握手，至多只有连接发起方的起始序列号能被确认，另一方选择的序列号则得不到确认

## 5. 为什么 TCP 是可靠的,UDP 早不可靠的?为什么 UDP 比 TCP 快?

TCP/IP 协议拥有三次握手双向机制，这一机制保证校验了数据，保证了他的可靠性。

UDP 就没有了，udp 信息发出后,不验证是否到达对方,所以不可靠。

## 6. http 协议

http 协议是一个基于请求与响应模式的无连接，无状态，应用层的协议，支持 c/s 模式，简单快速，灵活

简单快速：协议简单，通信速度快

灵活：允许传输任意类型的数据对象，由 Content-Type 标记

无连接：每次处理一个请求，处理完成后既断开

无状态：对事务处理没有记忆能力

http 有两种报文：请求报文和响应报文

请求报文由请求行，请求报头，和请求数据组成

请求行：抓包第一行，包括请求方法，url 和 http 版本

请求报头：指的就是题目中“里面的协议头部”

请求数据：指 post 方式提交的表单数据

响应报文由状态行，响应报头，响应正文组成

状态行：状态码

响应报头：同请求报头

响应正文：服务器返回的资源数据

接下来是 http 头部，既请求报头和响应报头，统称消息报头，消息报头可以分为通用报头，请求报头，响应报头，实体报头等

通用报头和实体报头既可以出现在请求报头中，也可以出现在响应报头中，通用报头包含的字段如：Date Connection Cache-Control,实体报头中有 Content-Type Content-Length Content-Language Content-Encoding.

请求报头中包含的字段有：

Host,User-Agent,Accept-Encoding,Accept-Language,Connection

响应报头包含的字段：

Location, Server

## 7. http 的 get 和 post 的区别

http 是应用层的协议，底层基于 TCP/IP 协议，所以本质上，get 和 post 请求都是 TCP 请求。所以二者的区别都是体现在应用层上（HTTP 的规定和浏览器/服务器的限制）

- 1.参数的传输方式：GET 参数通过 URL 传递，POST 放在 Request body 中。
- 2.GET 请求在 URL 中传送的参数是有长度限制的，而 POST 没有。
- 3.对于 GET 方式的请求，浏览器会把 http header 和 data 一并发送出去，服务器响应 200（返回数据）；而对于 POST，浏览器先发送 header，服务器响应 100 continue，浏览器再发送 data，服务器响应 200 ok（返回数据）。不过要注意，并不是所有浏览器都会在 POST 中发送两次包，比如火狐
- 4.对参数的数据类型，GET 只接受 ASCII 字符，而 POST 没有限制。
- 5.GET 比 POST 更不安全，因为参数直接暴露在 URL 上，所以不能用来传递敏感信息。
- 6.GET 请求只能进行 url 编码，而 POST 支持多种编码方式。
- 7.GET 在浏览器回退时是无害的，而 POST 会再次提交请求。
- 8.GET 产生的 URL 地址可以被 Bookmark，而 POST 不可以。
- 9.GET 请求会被浏览器主动 cache，而 POST 不会，除非手动设置。

## 8. socket 和 http 的区别：

Http 协议：简单的对象访问协议，对应于应用层。Http 协议是基于 TCP 链接的。

tcp 协议：对应于传输层

ip 协议：对应与网络层

TCP/IP 是传输层协议，主要解决数据如何在网络中传输；而 Http 是应用层协议，主要解决如何包装数据。

Socket 是对 TCP/IP 协议的封装，Socket 本身并不是协议，而是一个调用接口(API)，通过 Socket，我们才能使用 TCP/IP 协议。

Http 连接：http 连接就是所谓的短连接，及客户端向服务器发送一次请求，服务器端相应后连接即会断掉。

**socket 连接：**socket 连接及时所谓的长连接，理论上客户端和服务端一旦建立连接，则不会主动断掉；但是由于各种环境因素可能会是连接断开，比如说：服务器端或客户端主机 down 了，网络故障，或者两者之间长时间没有数据传输，网络防火墙可能会断开该链接已释放网络资源。所以当在一个 socket 连接中没有数据的传输，那么为了位置连续的连接需要发送心跳消息，具体心跳消息格式是开发者自己定义的。

## 9. TCP 与 UDP 区别总结：

- 1、TCP 面向连接（如打电话要先拨号建立连接）；UDP 是无连接的，即发送数据之前不需要建立连接
  - 2、TCP 提供可靠的服务。也就是说，通过 TCP 连接传送的数据，无差错，不丢失，不重复，且按序到达；UDP 尽最大努力交付，即不保证可靠交付
  - 3、TCP 面向字节流，实际上是 TCP 把数据看成一连串无结构的字节流；UDP 是面向报文的
- UDP 没有拥塞控制，因此网络出现拥塞不会使源主机的发送速率降低（对实时应用很有用，如 IP 电话，实时视频会议等）
- 4、每一条 TCP 连接只能是点到点的；UDP 支持一对一，一对多，多对一和多对多的交互通信
  - 5、TCP 首部开销 20 字节；UDP 的首部开销小，只有 8 个字节
  - 6、TCP 的逻辑通信信道是全双工的可靠信道，UDP 则是不可靠信道

## 10. https

HTTPS(全称：Hyper Text Transfer Protocol over Secure Socket Layer)，是以安全为目标的 HTTP 通道，简单讲是 HTTP 的安全版。HTTP 是应用层协议，位于 HTTP 协议之下是传输协议 TCP。TCP 负责传输，HTTP 则定义了数据如何进行包装，在 HTTP 跟 TCP 中间加多了一层加密层 TLS/SSL，SSL 是个加密套件，负责对 HTTP 的数据进行加密。TLS 是 SSL 的升级版。现在提到 HTTPS，加密套件基本指的是 TLS。

传输加密的流程：http 是应用层将数据直接给到 TCP 进行传输，https 是应用层将数据给到 TLS/SSL，将数据加密后，再给到 TCP 进行传输。

HTTPS 是如何加密数据的：

一般来说，加密分为对称加密、非对称加密

### 1. 对称加密：

对称加密的意思就是，加密数据用的密钥，跟解密数据用的密钥是一样的。

对称加密的优点在于加密、解密效率通常比较高。缺点在于，数据发送方、数据接收方需要协商、共享同一把密钥，并确保密钥不泄露给其他人。传输过程中容易被截获。

网上一个很形象的例子：假如现在小客与小服要进行一次私密的对话，他们不希望这次对话内容被其他外人知道。可是，我们平时的数据传输过程中又是明文传输的，万一被某个黑客把他们的对话内容给窃取了，那就难受了。为了解决这个问题，小服这家伙想到了一个方法来加密数据，让黑客看不到具体的内容。该方法是这样子的：在每次数据传输之前，小服会先传输小客一把密钥，然后小服在之后给小客发消息的过程中，会用这把密钥对这些消息进行加密。小客在收到这些消息后，会用之前小服给的那把密钥对这些消息进行解密，这样，小客就能得到密文里面真正的数据了。如果小客要给小服发消息，也同样用这把密钥来对消息进行加密，小服收到后也用这把密钥进行解密。这样，就保证了数据传输的安全性。

### 2. 非对称加密

非对称加密的意思就是，加密数据用的密钥（公钥），跟解密数据用的密钥（私钥）是不一样的。

网上一个很形象的例子：小服还是挺聪明的，得意了一会之后，小服意识到了密钥会被截取这个问题。倔强的小服又想到了另外一种方法：用非对称加密的方法来加密数据。该方法是这样的：小服和小客都拥有两把钥匙，一把钥匙的公开的（全世界都知道也没关系），称之为公钥；而另一把钥匙是保密（也就是只有自己才知道），称之为私钥。并且，用公钥加密的数据，只有对应的私钥才能解密；用私钥加密的数据，只有对应的公钥才能解密。所以在传输数据的过程中，小服在给小客传输数据的过程中，会用小客给他的公钥进行加密，然后小客收到后，再用自己的私钥进行解密。小客给小服发消息的时候，也一样会用小服给他的公

钥进行加密，然后小服再用自己的私钥进行解密。这样，数据就能安全着到达双方。是什么原因导致非对称加密这种方法的不安全性呢？它和对称加密方法的不安全性不同。非对称加密之所以不安全，是因为小客收到了公钥之后，无法确定这把公钥是否真的是小服。

解决的办法就是数字证书：小服再给小客发公钥的过程中，会把公钥以及小服的个人信息通过 Hash 算法生成消息摘要，为了防止摘要被人调换，小服还会用 CA 提供的私钥对消息摘要进行加密来形成数字签名，当小客拿到这份数字证书之后，就会用 CA 提供的公钥来对数字证书里面的数字签名进行解密得到消息摘要，然后对数字证书里面小服的公钥和个人信息进行 Hash 得到另一份消息摘要，然后把两份消息摘要进行对比，如果一样，则证明这些东西确实是小服的，否则就不是。

## 11. 加密算法

### 1. 对称加密算法

#### Data Encryption Standard(DES)

DES 是一种典型的块加密方法：将固定长度的明文通过一系列复杂的操作变成同样长度的密文，块的长度为 64 位。同时，DES 使用的密钥来自定义变换过程，因此算法认为只有持有加密所用的密钥的用户才能解密密文。DES 的密钥表面上是 64 位的，实际有效密钥长度为 56 位，其余 8 位可以用于奇偶校验。

DES 现在已经不被视为一种安全的加密算法，主要原因是它使用的 56 位密钥过短。

为了提供实用所需的安全性，可以使用 DES 的派生算法 3DES 来进行加密 (虽然 3DES 也存在理论上的攻击方法)。

#### Advanced Encryption Standard(AES)

AES 在密码学中又称 Rijndael 加密法，用来替代原先的 DES，已经被多方分析且广泛使用。



## DES 与 AES 的比较

自 DES 算法公诸于世以来，学术界围绕它的安全性等方面进行了研究并展开了激烈的争论。在技术上，对 DES 的批评主要集中在以下几个方面：

- 1、作为分组密码，DES 的加密单位仅有 64 位二进制，这对于数据传输来说太小，因为每个分组仅含 8 个字符，而且其中某些位还要用于奇偶校验或其他通讯开销。
- 2、DES 的密钥的位数太短，只有 56 比特，而且各次迭代中使用的密钥是递推产生的，这种相关必然降低密码体制的安全性，在现有技术下用穷举法寻找密钥已趋于可行。
- 3、DES 不能对抗差分和线性密码分析。
- 4、DES 用户实际使用的密钥长度为 56bit，理论上最大加密强度为 256。DES 算法要提高加密强度（例如增加密钥长度），则系统开销呈指数增长。除采用提高硬件功能和增加并行处理功能外，从算法本身和软件技术方面都无法提高 DES 算法的加密强度。

## 2. 非对称加密算法

### RSA

1977 年由 MIT 的 Ron Rivest、Adi Shamir 和 Leonard Adleman 一起提出，以他们三人姓氏开头字母命名，是一种获得广泛使用的非对称加密算法。

对极大整数做因数分解的难度 (The Factoring Problem) 决定了 RSA 算法的可靠性。换言之，对一个极大整数做因数分解愈困难，RSA 算法就愈可靠。假如有人找到一种快速因数分解的算法的话，那么用 RSA 加密的信息的可靠性就肯定会极度下降。目前看来找到这样的算法的可能性非常小。

## DES 与 RSA 的比较

RSA 算法的密钥很长，具有较好的安全性，但加密的计算量很大，加密速度较慢限制了其应用范围。为减少计算量，在传送信息时，常采用传统加密方法与公开密钥加密方法相结合的方式，即信息采用改进的 DES 对话密钥加密，然后使用 RSA 密钥加密对话密钥和信息摘要。对方收到信息后，用不同的密钥解密并可核对信息摘要。

采用 DES 与 RSA 相结合的应用，使它们的优缺点正好互补，即 DES 加密速度快，适合加密较长的报文，可用其加密明文；RSA 加密速度慢，安全性好，应用于 DES 密钥的加密，可解决 DES 密钥分配的问题。

目前这种 RSA 和 DES 结合的方法已成为 EMAIL 保密通信标准。

## 12. Volley

### 1、Volley 的特点

Volley 是谷歌大会上推出的网络通信框架（2.3 之前使用 HttpClient，之后使用 HttpURLConnection），它既可以访问网络获取数据，也可以加载图片，并且在性能方面进行了大幅度的调整，它的设计目的就是适合进行数据量不大但通信频繁的网络操作，而对于大数据量的操作，比如文件下载，表现很糟糕，因为 volley 处理 http 返回的默认实现是 BasicNetwork，它会把返回的流全部导入内存中，下载大文件会发生内存溢出

### 2、Volley 执行的过程：

默认情况下，Volley 中开启四个网络调度线程和一个缓存调度线程，首先请求会加入缓存队列，缓存调度线程从缓存队列中取出线程，如果找到该请求的缓存就直接读取该缓存并解析，然后回调给主线程，如果没有找到缓存的响应，则将这个请求加入网络队列，然后网络调度线程会轮询取出网络队列中的请求，发起 http 请求，解析响应并将响应存入缓存，回调给主线程

### 3、Volley 为什么不适合下载上传大文件？为什么适合数据量小的频率高的请求？

1.volley 基于请求队列，Volley 的网络请求线程池默认大小为 4。意味着可以并发进行 4 个请求，大于 4 个，会排在队列中。并发量小所以适合数据量下频率高的请求

2.因为 Volley 下载文件会将流存入内存中（是一个小于 4k 的缓存池），大文件会导致内存溢出，所以不能下载大文件，不能上传大文件的原因和 1 中差不多，

设想你上传了四个大文件，同时占用了 volley 的四个线程，导致其他网络请求都阻塞在队列中，造成反应慢的现象

## 13. OKHttp

### 1、 OKHttp 的特点

- 1.相较于 Volley，它的最大并发量为 64
- 2.使用连接池技术，支持 5 个并发的 socket 连接默认 keepAlive 时间为 5 分钟，解决 TCP 握手和挥手的效率问题，减少握手次数
- 3.支持 Gzip 压缩，且操作对用户透明，可以通过 header 设置，在发起请求的时候自动加入 header，Accept-Encoding: gzip，而我们的服务器返回的时候 header 中有 Content-Encoding: gzip
- 4.利用响应缓存来避免重复的网络请求
- 5.很方便的添加拦截器，通常情况下，拦截器用来添加，移除，转换请求和响应的头部信息，比如添加公参等
- 6.请求失败，自动重连，发生异常时重连，看源码调用 recover 方法重连了一次
- 7.支持 SPDY 协议(SPDY 是 Google 开发的基于 TCP 的应用层协议，用以最小化网络延迟，提升网络速度，优化用户的网络使用体验。SPDY 并不是一种用于替代 HTTP 的协议，而是对 HTTP 协议的增强。新协议的功能包括数据流的多路复用、请求优先级以及 HTTP 报头压缩。谷歌表示，引入 SPDY 协议后，在实验室测试中页面加载速度比原先快 64%)
- 8.使用 Okio 来简化数据的访问与存储，提高性能

### 2、 OkHttp 的缺点

- 1.消息回来需要切到主线程，主线程要自己去写。
- 2.调用比较复杂，需要自己进行封装。
- 3.缓存失效：网络请求时一般都会获取手机的一些硬件或网络信息，比如使用的网络环境。同时为了信息传输的安全性，可能还会对请求进行加密。在这些情况下 OkHttp 的缓存系统就会失效了，导致用户在无网络情况下不能访问缓存。

### 3、 OkHttp 框架中都用到哪些设计模式

- 1.最明显的 Builder 设计模式，如构建对象 OkHttpClient，还有单利模式
- 2.工厂方法模式，如源码中的接口 Call
- 3.观察者模式如 EventListener，监听请求和响应
- 4.策略模式
- 5.责任链模式，如拦截器

## 14. Retrofit

Retrofit 底层是基于 OkHttp 实现的，与其他网络框架不同的是，它更多使用运行时注解的方式提供功能

### 1、 原理

通过 java 接口以及注解来描述网络请求，并用动态代理的方式生成网络请求的 request，然后通过 client 调用相应的网络框架（默认 okhttp）去发起网络请求，并将返回的 response 通过 converterFactory 转换成相应的数据 model，最后通过 callAdapter 转换成其他数据方式（如 rxjava Observable）

### 2、 Retrofit 流程

- （1）通过解析 网络请求接口的注解 配置 网络请求参数
- （2）通过 动态代理 生成 网络请求对象
- （3）通过 网络请求适配器 将 网络请求对象 进行平台适配
- （4）通过 网络请求执行器 发送网络请求
- （5）通过 数据转换器 解析服务器返回的数据
- （6）通过 回调执行器 切换线程（子线程 ->>主线程）
- （7）用户在主线程处理返回结果

### 3、 Retrofit 优点

- 1.可以配置不同 HTTP client 来实现网络请求，如 okhttp、httpClient 等；
- 2.请求的方法参数注解都可以定制；

- 3.支持同步、异步和 RxJava;
- 4.超级解耦;
- 5.可以配置不同的反序列化工具来解析数据, 如 json、xml 等
- 6.框架使用了很多设计模式

## 五、程序员简历

---

### 程序员简历就该这样写

本篇文章除了教大家用 Markdown 如何写一份程序员专属的简历, 后面还会给大家推荐一些不错的用来写 Markdown 简历的软件或者网站, 以及如何优雅的将 Markdown 格式转变为 PDF 格式或者其他格式。

推荐大家使用 Markdown 语法写简历, 然后再将 Markdown 格式转换为 PDF 格式后进行简历投递。

如果你对 Markdown 语法不太了解的话, 可以花半个小时简单看一下 Markdown 语法说明: <http://www.markdown.cn>。

### 为什么说简历很重要?

一份好的简历可以在整个申请面试以及面试过程中起到非常好的作用。在不夸大自己能力的情况下, 写出一份好的简历也是一项很棒的能力。为什么说简历很重要呢?

### 先从面试前来说

- 假如你是网申, 你的简历必然会经过 HR 的筛选, 一张简历 HR 可能也就花费 10 秒钟看一下, 然后 HR 就会决定你这一关是 Fail 还是 Pass。

- 假如你是内推，如果你的简历没有什么优势的话，就算是内推你的人再用心，也无能为力。

另外，就算你通过了筛选，后面的面试中，面试官也会根据你的简历来判断你究竟是否值得他花费很多时间去面试。

所以，简历就像是我们的一个门面一样，它在很大程度上决定了你能否进入到下一轮的面试中。

## 再从面试中来说

我发现大家比较喜欢看面经，这点无可厚非，但是大部分面经都没告诉你很多问题都是在特定条件下才问的。举个简单的例子：一般情况下你的简历上注明你会的东西才会被问到（Java、数据结构、网络、算法这些基础是每个人必问的），

比如写了你会 redis，那面试官就很大概率会问你 redis 的一些问题。比如：redis 的常见数据类型及应用场景、redis 是单线程为什么还这么快、redis 和

memcached 的区别、redis 内存淘汰机制等等。

所以，首先，你要明确的一点是：**你不会的东西就不要写在简历上**。另外，你要**考虑你该如何才能让你的亮点在简历中凸显出来**，比如：你在某某项目做了什么事情解决了什么问题（只要有项目就一定有要解决的问题）、你的某一个项目里使用了什么技术后整体性能和并发量提升了很多等等。

面试和工作是两回事，聪明的人会把面试官往自己擅长的领域领，其他人则被面试官牵着鼻子走。虽说面试和工作是两回事，但是你要想要获得自己满意的 offer，你自身的实力必须要强。

## 下面这几点你必须知道

1. 大部分公司的 HR 都说我们不看重学历（骗你的！），但是如果你的学校不出众的话，很难在一堆简历中脱颖而出，除非你的简历上有特别的亮点，比如：某某大厂的实习经历、获得了某某大赛的奖等等。
2. 大部分应届生找工作的硬伤是没有工作经验或实习经历，所以如果你是应届生就不要错过秋招和春招。一旦错过，你后面就极大可能会面临社招，这个时候没有工作经验的你可能就会面临各种碰壁，导致找不到一个好的工作
3. 写在简历上的东西一定要慎重，这是面试官大量提问的地方；

4. 将自己的项目经历完美的展示出来非常重要。

## 必须了解的两大法则

### STAR 法则 (Situation Task Action Result)

- **Situation:** 事情是在什么情况下发生;
- **Task:** 你是如何明确你的任务的;
- **Action:** 针对这样的情况分析, 你采用了什么行动方式;
- **Result:** 结果怎样, 在这样的情况下你学习到了什么。

简而言之, STAR 法则, 就是一种讲述自己故事的方式, 或者说, 是一个清晰、条理的作文模板。不管是什么, 合理熟练运用此法则, 可以轻松的对面试官描述事物的逻辑方式, 表现出自己分析阐述问题的清晰性、条理性和逻辑性。

### FAB 法则 (Feature Advantage Benefit)

- **Feature:** 是什么;
- **Advantage:** 比别人好在哪些地方;
- **Benefit:** 如果雇佣你, 招聘方会得到什么好处。

简单来说, 这个法则主要是让你的面试官知道你的优势、招了你之后对公司有什么帮助。

## 项目经历怎么写?

简历上有一两个项目经历很正常, 但是真正能把项目经历很好的展示给面试官的非常少。对于项目经历大家可以考虑从如下几点来写:

1. 对项目整体设计的一个感受
2. 在这个项目中你负责了什么、做了什么、担任了什么角色
3. 从这个项目中你学会了那些东西, 使用到了那些技术, 学会了那些新技术的使用

4. 另外项目描述中，最好可以体现自己的综合素质，比如你是如何协调项目组成员协同开发的或者在遇到某一个棘手的问题的时候你是如何解决的  
又或者说你在这个项目用了什么技术实现了什么功能比如:用 redis 做缓存提高访问速度和并发量、使用消息队列削峰和降流等等。

## 专业技能该怎么写？

先问一下你自己会什么，然后看看你意向的公司需要什么。一般 HR 可能并不太懂技术，所以他在筛选简历的时候可能就盯着你专业技能的关键词来看。对于公司有要求而你不会的技能，你可以花几天时间学习一下，然后在简历上可以写上自己了解这个技能。比如你可以这样写(下面这部分内容摘自我的简历，大家可以根据自己的情况做一些修改和完善)：

- 计算机网络、数据结构、算法、操作系统等课内基础知识：掌握
- Java 基础知识：掌握
- JVM 虚拟机（Java 内存区域、虚拟机垃圾算法、虚拟垃圾收集器、JVM 内存管理）：掌握
- 高并发、高可用、高性能系统开发：掌握
- Struts2、Spring、Hibernate、Ajax、Mybatis、JQuery ：掌握
- SSH 整合、SSM 整合、 SOA 架构：掌握
- Dubbo： 掌握
- Zookeeper: 掌握
- 常见消息队列: 掌握
- Linux： 掌握
- MySQL 常见优化手段：掌握
- Spring Boot +Spring Cloud +Docker:了解



- Hadoop 生态相关技术中的 HDFS、Storm、MapReduce、Hive、Hbase : 了解
- Python 基础、一些常见第三方库比如 OpenCV、wxpy、wordcloud、matplotlib: 熟悉

## 排版注意事项

1. 尽量简洁，不要太花里胡哨；
2. 一些技术名词不要弄错了大小写比如 MySQL 不要写成 mysql，Java 不要写成 java。这个在我看来还是比较忌讳的，所以一定要注意这个细节；
3. 中文和数字英文之间加上空格的话看起来会舒服一点；

## 其他的一些小 tips

1. 尽量避免主观表述，少一点语义模糊的形容词，尽量要简洁明了，逻辑结构清晰。
2. 如果自己有博客或者个人技术栈点的话，写上去会为你加分很多。
3. 如果自己的 Github 比较活跃的话，写上去也会为你加分很多。
4. 注意简历真实性，一定不要写自己不会的东西，或者带有欺骗性的内容
5. 项目经历建议以时间倒序排序，另外项目经历不在于多，而在于有亮点。
6. 如果内容过多的话，不需要非把内容压缩到一页，保持排版干净整洁就可以了。
7. 简历最后最好能加上：“感谢您花时间阅读我的简历，期待能有机会和您共事。”这句话，显的你会很有礼貌。

## 推荐的工具/网站

- 冷熊简历(MarkDown 在线简历工具，可在线预览、编辑和生成 PDF):<http://cv.ftqq.com/>
- Typora+[Java 程序员简历模板](#)

## 六、面试准备

不论是校招还是社招都避免不了各种面试、笔试，如何去准备这些东西就显得格外重要。不论是笔试还是面试都是有章可循的，我这个“有章可循”说的意思只是说应对技术面试是可以提前准备。我其实特别不喜欢那种临近考试就提前背啊记啊各种题的行为，非常反对！我觉得这种方法特别极端，而且在稍有一点经验的面试官面前是根本没有用的。建议大家还是一步一个脚印踏踏实实地走。

- 1 如何获取大厂面试机会？
- 2 面试前的准备
  - 2.1 准备自己的自我介绍
  - 2.2 关于着装
  - 2.3 随身带上自己的成绩单和简历
  - 2.4 如果需要笔试就提前刷一些笔试题
  - 2.5 花时间一些逻辑题
  - 2.6 准备好自己的项目介绍
  - 2.7 提前准备技术面试
  - 2.7 面试之前做好定向复习
- 3 面试之后复盘

# 1 如何获取大厂面试机会？

在讲如何获取大厂面试机会之前，先来给大家科普/对比一下两个校招非常常见的概念——春招和秋招。

1. 招聘人数：秋招多于春招；
2. 招聘时间：秋招一般 7 月左右开始，大概一直持续到 10 月底。但是大厂（如 BAT）都会早开始早结束，所以一定要把握好时间。春招最佳时间为 3 月，次佳时间为 4 月，进入 5 月基本就不会再有春招了（金三银四）。
3. 应聘难度：秋招略大于春招；
4. 招聘公司：秋招数量多，而春招数量较少，一般为秋招的补充。

综上，一般来说，秋招的含金量明显是高于春招的。

下面我就说一下我自己知道的一些方法，不过应该也涵盖了大部分获取面试机会的方法。

1. 关注大厂官网，随时投递简历（走流程的网申）；
2. 线下参加宣讲会，直接投递简历；
3. 找到师兄师姐/认识的人，帮忙内推（能够让你避开网申简历筛选，笔试筛选，还是挺不错的，不过也还是需要你的简历够棒）；
4. 博客发文被看中/**Github** 优秀开源项目作者，大厂内部人员邀请你面试；
5. 求职类网站投递简历（不是太推荐，适合海投）；

除了这些方法，我也遇到过这样的经历：有些大公司的一些部门可能暂时没招够人，然后如果你的亲戚或者朋友刚好在这个公司，而你正好又在寻求 offer，那么面试机会基本上是有，而且这种面试的难度好像一般还普遍比其他正规面试低很多。

## 2 面试前的准备

### 2.1 准备自己的自我介绍

从 HR 面、技术面到高管面/部门主管面，面试官一般会让你先自我介绍一下，所以好好准备自己的自我介绍真的非常重要。网上一般建议的是准备好两份自我介绍：一份对 hr 说的，主要讲能突出自己的经历，会的编程技术一语带过；另一份对技术面试官说的，主要讲自己会的技术细节，项目经验，经历那些就一语带过。

我这里简单分享一下我自己的自我介绍的一个简单的模板吧：

面试官，您好！我叫某某。大学时间我主要利用课外时间学习某某。在校期间参与过一个某某系统的开发，另外，自己学习过程中也写过很多系统比如某某系统。在学习之余，我比较喜欢通过博客整理分享自己所学知识。我现在是某某社区的认证作者，写过某某很不错的文章。另外，我获得过某某奖，我的 Github 上开源的某个项目已经有多少 Star 了。

### 2.2 关于着装

穿西装、打领带、小皮鞋？NO！NO！NO！这是互联网公司面试又不是去走红毯，所以你只需要穿的简单大方就好，不需要太正式。

### 2.3 随身带上自己的成绩单和简历

有的公司在面试前都会让你交一份成绩单和简历当做面试中的参考。

### 2.4 如果需要笔试就提前刷一些笔试题

平时空闲时间多的可以刷一下笔试题目（牛客网上有很多）。但是不要只刷面试题，不动手 code，程序员不是为了考试而存在的。

### 2.5 花时间一些逻辑题

面试中发现有些公司都有逻辑题测试环节，并且都把逻辑笔试成绩作为很重要的一个参考。

## 2.6 准备好自己的项目介绍

如果有项目的话，技术面试第一步，面试官一般都是让你自己介绍一下你的项目。你可以从下面几个方向来考虑：

1. 对项目整体设计的一个感受（面试官可能会让你画系统的架构图）
2. 在这个项目中你负责了什么、做了什么、担任了什么角色
3. 从这个项目中你学会了那些东西，使用到了那些技术，学会了那些新技术的使用
4. 另外项目描述中，最好可以体现自己的综合素质，比如你是如何协调项目组成员协同开发的或者在遇到某一个棘手的问题的时候你是如何解决的  
又或者说你在这个项目用了什么技术实现了什么功能比如：用 `redis` 做缓存提高访问速度和并发量、使用消息队列削峰和降流等等。

## 2.7 提前准备技术面试

搞清楚自己面试中可能涉及哪些知识点、哪些知识点是重点。面试中哪些问题会被经常问到、自己该如何回答。（强烈不推荐背题，第一：通过背这种方式你能记住多少？能记住多久？第二：背题的方式的学习很难坚持下去！）

## 2.7 面试之前做好定向复习

所谓定向复习就是专门针对你要面试的公司来复习。比如你在面试之前可以在网上找找有没有你要面试的公司的面经。

举个栗子：在我面试 `ThoughtWorks` 的前几天我就在网上找了一些关于

`ThoughtWorks` 的技术面的一些文章。然后知道了 `ThoughtWorks` 的技术面会让我们在之前做的作业的基础上增加一个或两个功能，所以我提前一天就把我之前做的程序重新重构了一下。然后在技术面的时候，简单的改了几行代码之后写个测试就完事了。如果没有提前准备，我觉得 20 分钟我很大几率会完不成这项任务。

### 3 面试之后复盘

如果失败，不要灰心；如果通过，切勿狂喜。面试和工作实际上是两回事，可能很多面试未通过的人，工作能力比你强的多，反之亦然。我个人觉得面试也像是一场全新的征程，失败和胜利都是平常之事。所以，劝各位不要因为面试失败而灰心、丧失斗志。也不要因为面试通过而沾沾自喜，等待你的将是更美好的未来，继续加油！

## 七、面试困惑

### （1）Android 开发前景？

我们通过智联招聘、拉勾网、Boss 直聘、前程无忧发布的移动互联网招聘岗位需求发现：移动端的招聘量变少，但中高端的职位却多了起来，这说明行业只是变得成熟规范起来了。竞争激烈，但产品质量与留存更加重要，我们进入了技术赋能业务的时代。大前端正在跨平台，移动开发者的未来更可能是跨终端，产品、运营、数据分析、后端，技多不压身。

### （2）我是双非/三本/专科学校的，我有机会进入大厂吗？

我自己也是非 985 非 211 学校的，结合自己的经历以及一些朋友的经历，我觉得让我回答这个问题再好不过。首先，我觉得学校歧视很正常，真的太正常了，如果要抱怨的话，你只能抱怨自己没有进入名校。

但是，千万不要动不动说自己学校差，动不动拿自己学校当做自己进不了大厂的借口，学历只是筛选简历的很多标准中的一个而已，如果你够优秀，简历够丰富，你也一样可以和名校同学一起同台竞争。企业 HR 肯定是更喜欢高学历的人，

毕竟 985、211 优秀人才比例肯定比普通学校高很多，HR 团队肯定会优先在这些学校里选。这就好比相亲，你是愿意在很多优秀的人中选一个优秀的，还是愿意在很多普通的人中选一个优秀的呢？

双非本科甚至是二本、三本甚至是专科的同学也有很多进入大厂的，不过比率相比于名校的低很多而已。从大厂招聘的结果上看，高学历人才的数量占据大头，那些成功进入 BAT、美团，京东，网易等大厂的双非本科甚至是二本、三本甚至

是专科的同学往往是因为具备丰富的项目经历或者在某个含金量比较高的竞赛比如 ACM 中取得了不错的成绩。

一部分学历不突出但能力出众的面试者能够进入大厂并不是说明学历不重要，而是学历的软肋能够通过其他的优势来弥补。

所以，如果你的学校不够好而你自己又想去大厂的话，建议你可以从这几点来做：

尽量在面试前最好有一个可以拿的出手的项目；

有实习条件的话，尽早出去实习，实习经历也会是你的简历的一个亮点（有能力在大厂实习最佳！）；

参加一些含金量比较高的比赛，拿不拿得到名次没关系，重在锻炼。

### **（3）非计算机专业的学生能学好 Android 吗？我能进大厂吗？**

当然可以！现在非科班的程序员很多，很大一部分原因是互联网行业的工资比较高。我们学校外面的培训班里面 90%都是非科班，我觉得他们很多人学的都还不错。另外，我的一个朋友本科是机械专业，大一开始自学安卓，技术贼溜，在我看来他比大部分本科是计算机的同学学的还要好。参考（2）的回答，即使你是非科班程序员，如果你想进入大厂的话，你也可以通过自己的其他优势来弥补。我觉得我们不应该因为自己的专业给自己划界限或者贴标签，说实话，很多科班的同学可能并不如你，你以为科班的同学就会认真听讲吗？还不是几乎全靠自己课下自学！不过如果你是非科班的话，你想要学好，那么注定就要舍弃自己本专业的一些学习时间，这是无可厚非的。建议非科班的同学，首先要打好计算机基础知识基础：①计算机网络、②操作系统、③数据机构与算法，我个人觉得这 3 个对你最重要。这些东西就像是内功，对你以后的长远发展非常有用。

当然，如果你想要进大厂的话，这些知识也是一定会被问到的。另外，“一定学好数据结构与算法！一定学好数据结构与算法！一定学好数据结构与算法！”，重要的事情说 3 遍。

### **（4）Android 学习该学哪些东西？**

掌握架构师筑基必备技能

掌握 Android 框架体系架构

学会 360°全方位性能调优

通过设计思想解读开源框架

学习 NDK 模块开发

通过实战炼成移动架构师

学会开发微信小程序

掌握 hybrid 开发与 Flutter

## （5）我没有实习经历的话找工作是不是特别艰难？

没有实习经历没关系，只要有拿得出手的项目或者大赛经历的话，你依然有可能拿到大厂的 offer。笔主当时找工作的时候就没有实习经历以及大赛获奖经历，单纯就是凭借自己的项目经验撑起了整个面试。如果你既没有实习经历，又没有拿得出手的项目或者大赛经历的话，我觉得在简历关，除非你有其他特别的亮点，不然，你应该就会被刷。

## （6）我该如何准备 Android 面试呢？

面试刷题

Android 基础知识

Activity

Service



Broadcast

Receiver

ContentProvider

Intent

Fragment

## Java 面试题

Java 基础

Java 中的设计模式

## Android 高级

Android 性能优化

Android 屏幕适配

AIDL

Android 中的事件处理

Android 中的动画

ContentObserver

内容观察者作用及特点

## 项目框架的使用

自我介绍

开发中都使用过哪些框架、平台

都使用过哪些自定义控件

项目的流程

项目中常见的问题

即时通讯是是怎么做的

常见框架分析

## Flutter 相关面试题全解析

Dart 部份

Flutter 部份

腾讯、阿里、百度、美团、字节跳动、京东等) Android 高频面试真题集锦

(7) 面试的简历准备注意事项有哪些呢? 下面是我总结的一些准备面试的 Tips 以及面试必备的注意事项:

简历

简历的重要性就不言而喻了，怎么样写好简历是个技术活，当然如果你有很好的背景（学校或者公司），那么不管你怎么写，基本上都不刷掉你，我们作为一般的人还是需要下一番功夫的。拿我的简历作为例子，大概有以下几个部分：

个人信息：姓名、出生日期、教育背景、博客地址、github 地址、联系方式（手机、邮箱和微信号）

工作经历：毕业后待过哪些公司，一般是倒序，项目尽量精简明了，可以参考 [SMART 原则](#)

专业技能：自己熟悉的一些技能，这个为什么我写到最后，因为对于工作三年的同学来说，面试官更加注重的是你的项目经历，大部分面试都是看你的项目经历来提问。

[怎样写简历](#)，这个开源网站不错，教你怎么写简历，而且有一个在线 markdown 在线网站，可以导出 pdf。

## 简历投递注意事项

一般 HR 都会集中处理简历，正序或者倒序都有可能，所以选择最合适的时间段去投递就好。

总体上来说，比较推荐的时间段就是周二至周四，上午 9 点以后。

除了投递时间之外，还有一些注意事项可以提高你的简历被阅读率：

1. 尽量选择 HR 邮箱直投：不论是哪个招聘网站，整体的回复速度都要低于 HR 直投邮箱，大部分 HR 打开自己邮箱的频率也要高于其他招聘类网站。2. 邮箱实名：最好将发件人改为自己的名字，可以让 HR 一眼就看到，而不是乱七八糟的其他内容，不仅看起来更加整洁，也可以增加印象。3. 邮件名称固定格式：如果企业方有要求，要严格按照企业方的要求来，如果没有，应届生建议“姓名+

学校+专业+应聘岗位+最快到岗时间（尤其是标注尽快到岗）”，言简意赅。 4.

正文不要空着，记得添加附件.pdf：有些 HR 习惯下载一段时间内的所有简历附件，然后一起查看，如果你没有附件会直接被忽略，pdf 是为了避免简历跑版的重要措施；也有些 HR 更倾向于一封邮件一封邮件地查看，这时正文如果有内容就会更加直接，正文和附件都有，双重保障。

不要认为一些小细节不重要~投递简历的很多小细节会决定第一印象的，看到一封简洁、信息明确的邮件，HR 第一反应都会是“WOW，这个不错/还行，让我看看”

## （8）没有项目经历/博客/Github 开源项目怎么办？

从现在开始做！网上有很多非常不错的项目视频，你就跟着一步一步做，不光要做，还要改进，改善。另外，如果你的老师有相关 Java 后台项目的话，你也可以主动申请参与进来。

如果有自己的博客，也算是简历上的一个亮点。建议可以在掘金、Segmentfault、CSDN、知乎等技术交流社区写博客，当然，你也可以自己搭建一个博客（采用 Hexo+Github Pages 搭建非常简单）。写一些什么？学习笔记、实战内容、读书笔记等等都可以。多用 Github，用好 Github，上传自己不错的项目，写好 readme 文档，在其他技术社区做好宣传。相信你也会收获一个不错的开源项目！

## （9）大厂到底青睐什么样的应届生？

从阿里、腾讯等大厂招聘官网对于 Java 后端方向/后端方向的应届实习生的要求，我们大概可以总结归纳出下面这 4 点能给简历增加很多分数：

参加过竞赛（含金量超高的是 ACM）：

对数据结构与算法非常熟练；

参与过实际项目（比如学校网站）；

参与过某个知名的开源项目或者自己的某个开源项目很不错；

## (10)有哪些有价值的问题值得问？

针对这个问题。笔主专门找了几个专门做 HR 工作的小哥哥小姐姐们询问并且查阅了挺多前辈们的回答，然后结合自己的实际经历，我概括了下面几个比较适合问的问题。

### 面对 HR 或者其他 Level 比较低的面试官时

能不能谈谈你作为一个公司老员工对公司的感受？(这个问题比较容易回答，不会让面试官陷入无话可说的尴尬境地。另外，从面试官的回答中你可以加深对这个公司的了解，让你更加清楚这个公司到底是不是你想的那样或者说你是否能适应这个公司的文化。除此之外，这样的问题在某种程度上还可以拉近你与面试官的距离。)

能不能问一下，你当时因为什么原因选择加入这家公司的呢或者说这家公司有哪些地方吸引你？有什么地方你觉得还不太好或者可以继续完善吗？（类似第一个问题，都是问面试官个人对于公司的看法。）

我觉得我这次表现的不是太好，你有什么建议或者评价给我吗？(这个是我常问的。我觉得说自己表现不好只是这个语境需要这样来说，这样可以显的你比较谦虚好学上进。)

接下来我会有一段空档期，有什么值得注意或者建议学习的吗？（体现出你对工作比较上心，自主学习意识比较强。）

这个岗位为什么还在招人？(岗位真实性和价值咨询)

大概什么时候能给我回复呢？(终面的时候，如果面试官没有说的话，可以问一下)

## 面对部门领导

部门的主要人员分配以及对应的主要工作能简单介绍一下吗？

未来如果我要加入这个团队，你对我的期望是什么？（部门领导一般情况下是你的直属上级了，你以后和他打交道的机会应该是最多的。你问这个问题，会让他感觉你是一个对他的部门比较上心，比较有团体意识，并且愿意倾听的候选人。）

公司对新入职的员工的培养机制是什么样的呢？（正规的公司一般都有培养机制，提前问一下是对你自己的负责也会显的你比较上心）

以您来看，这个岗位未来在公司内部的发展如何？(在我看来，问这个问题也是对你自己的负责吧，谁不想发展前景更好的岗位呢？)

团队现在面临的最大挑战是什么？(这样的问题不会暴露你对公司的不了解，并且也能让你对未来工作的挑战或困难有一个提前的预期。)

## 面对 Level 比较高的(比如总裁,老板)

贵公司的发展目标和方向是什么？（看下公司的发展是否满足自己的期望）

与同行业的竞争者相比，贵公司的核心竞争优势在什么地方？（充分了解自己的优势和劣势）

公司现在面临的最大挑战是什么？

## 八、经典 HR 面试问题解析

### 问题 1：“请你自我介绍一下”

思路：

- 1、这是面试的必考题目。
- 2、介绍内容要与个人简历相一致。
- 3、表述方式上尽量口语化。
- 4、要切中要害，不谈无关、无用的内容。
- 5、条理要清晰，层次要分明。
- 6、事先最好以文字的形式写好背熟。

### 问题 2：“你有什么业余爱好？”

思路：

- 1、 业余爱好能在一定程度上反映应聘者的性格、观念、心态，这是招聘单位问该问题的主要原因。
- 2、 最好不要说自己没有业余爱好。

- 3、不要说自己有那些庸俗的、令人感觉不好的爱好。
- 4、最好不要说自己仅限于读书、听音乐、上网，否则可能令面试官怀疑应聘者性格孤僻。
- 5、最好能有一些户外的业余爱好来“点缀”你的形象。

### 问题 3：“谈谈你的缺点”

思路：

- 1、不宜说自己没缺点。
- 2、不宜把那些明显的优点说成缺点。
- 3、不宜说出严重影响所应聘工作的缺点。
- 4、不宜说出令人不放心、不舒服的缺点。
- 5、可以说出一些对于所应聘工作“无关紧要”的缺点，甚至是一些表面上看是缺点，从工作的角度看却是优点的缺点。

### 问题 4：“谈一谈你的一次失败经历”

思路：

- 1、不宜说自己没有失败的经历。
- 2、不宜把那些明显的成功说成是失败。
- 3、不宜说出严重影响所应聘工作的失败经历，
- 4、所谈经历的结果应是失败的。
- 5、宜说明失败之前自己曾信心百倍、尽心尽力。
- 6、说明仅仅是由于外在客观原因导致失败。



- 7、 失败后自己很快振作起来，以更加饱满的热情面对以后的工作。

## **问题 5：“你为什么选择我们公司？”**

**思路：**

- 1、 面试官试图从中了解你求职的动机、愿望以及对此项工作的态度。
- 2、 建议从行业、企业和岗位这三个角度来回答。
- 3、 参考答案——“我十分看好贵公司所在的行业，我认为贵公司十分重视人才，而且这项工作很适合我，相信自己一定能做好。”

## **问题 6：“项目开发中遇到的最大的一个难题和挑战，你是如何解决的。”**

**思路：**

- 1、 不宜直接说出具体的困难，否则可能令对方怀疑应聘者不行。
- 2、 可以尝试迂回战术，说出应聘者对困难所持有的态度——“工作中出现一些困难是正常的，也是难免的，但是只要有坚忍不拔的毅力、良好的合作精神以及事前周密而充分的准备，任何困难都是可以克服的。”

## **问题 7：“你为什么会离开上家公司？”**

**思路：**

- 1、 最重要的是：应聘者要使找招聘单位相信，应聘者在过往的单位的“离职原因”在此家招聘单位里不存在。

- 2、 避免把“离职原因”说得太详细、太具体。
- 3、 不能掺杂主观的负面感受，如“太辛苦”、“人际关系复杂”、“管理太混乱”、“公司不重视人才”、“公司排斥我们某某的员工”等。
- 4、 但也不能躲闪、回避，如“想换换环境”、“个人原因”等。
- 5、 不能涉及自己负面的人格特征，如不诚实、懒惰、缺乏责任感、不随和等。
- 6、 尽量使解释的理由为应聘者个人形象添彩。
- 7、 如“我离职是因为这家公司倒闭。我在公司工作了三年多，有较深的感情。从去年始，由于市场形势突变，公司的局面急转直下。到眼下这一步我觉得很遗憾，但还要面对现实，重新寻找能发挥我能力的舞台。”

## 问题 8：“你能为公司带来什么经济效益？”

思路：

- 1、 基本上原则上“投其所好”。
- 2、 回答这个问题前应聘者最好能“先发制人”，了解招聘单位期待这个职位所能发挥的作用。
- 3、 应聘者可以根据自己的了解，结合自己在专业领域的优势来回答这个问题。

## 问题 9：你对未来的职业规划？

**思路：**很多面试官都会问，“你的职业规划是什么？”这个问题往往会难倒很多求职者。今天我跟大家分享，求职者怎样回答，才能更给自己加分。**（一）什么是职业规划？**

想回答好这个问题，首先要清楚，什么是职业规划。职业规划( Career Planning )是指对职业生涯乃至人生进行持续的、系统的计划过程，它包括职业定位、目标设定和通道设计三个要素。

职业规划也叫“职业生涯规划”，规划的好坏可能将影响整个生命历程。

**职业规划有三个要素：**

- 1、个人内在要素，包括职业性格、兴趣、职业价值观等，也就是“我想做什么”；
- 2、商业价值要素，包括已具备的知识，技能，经历，人脉，也就是“我能做什么”；
- 3、外在环境要素，包括宏观产业、组织、家庭等方面，也就是“环境支持我做什么”；

综上，职业规划就是在综合分析与权衡的基础上，确定出一个人当下最适合的职业发展方向，并为实现这一目标做出有效的合理的安排、计划与努力。

所以说，谈到职业规划时，求职者要综合考虑到自己的兴趣、技能，职业的目标，以及为实现目标所需要的计划。

## （二）提及职业规划，面试官想考核什么？

所谓“知己知彼，百战百胜。”要想知道怎样回复更合适，首先应该知道面试官这么问，是为了考核什么？

一般来说，面试官考核内容不外乎以下几点：

- 1、求职者对自我的认知
- 2、求职者对岗位的了解程度，对职业的理解程度
- 3、求职者的反应能力、逻辑能力和语言能力
- 4、考察求职者工作的稳定性
- 5、考察求职者的上进心、目标感和自我驱动力

对于应届毕业生来说，实习经验不多，社会阅历尚浅，对自己认知不足，很多想法都不成熟，让他们说清楚自己的职业规划，其实是一件很有难度的事情。<b>在某些场合，求职者说的“内容”并不重要，“怎么说”才是更重要的。

大多数面试官主要是希望通过求职者的回答，了解求职者对自我的认知，对该行业的看法，对应聘岗位的认知，判断求职者的逻辑思维和语言表述能力，从而对求职者的性格和价值观有个大致的了解。

对于有一定工作经验的求职者来说，这个问题主要考核稳定性、上进心、目标感和驱动力。没有职业规划的人对自己定位不清晰，没有发展方向，很可能遇到问题就会退缩，一不开心就要辞职。没有自己目标的人，在工作业绩上也难出彩，他无法实现自我驱动，只能靠外在的动力来驱动，比如只能完成领导确定交代的

任务，不能突破既定任务的天花板。而目标感强、自我驱动力强的人是能够充分利用资源、充分提升自我的，能够在工作任务之外也持续为这个职业目标而奋斗。

### **（三）如何回复面试官？**

在了解面试官心理状态的基础上，回复职业规划问题可从以下几个角度考虑：

- 1、充分认知自己的性格、兴趣、爱好、特长、知识、能力等，并结合当下的环境，选择可以将个人爱好与职业发展结合起来的行业/职业。
- 2、了解应聘公司的背景、现状与未来，在谈到规划时，可以适当的与公司发展相贴合。
- 3、不要说“我想几年当主管，几年当经理”，这种毫无意义的答案。职业规划更应该考虑专业技能方面的提升计划和步骤，而不是仅仅在于职级提升。
- 4、说明自己有长远规划的能力，但在表述中主要着眼于最近的3年5年即可，说明自己当下会努力做好应聘的岗位。
- 5、一个有竞争力的应聘者对于职业规划问题一定要有清晰的想法，大的方向和短期的目标必须明确，同时尽可能给自己预留调整的空间。

**根据以上原则，对于职业规划这个问题的回复，可以参考下面这个表述：**

“感谢你提出这么深刻的问题。我的兴趣是XXX，优势是XXX，因此我选择了XXX行业/职业，这是一个可以将我的兴趣和工作结合起来的行业，是我非常喜欢的，所以我会很用心对待XXX岗位。”

“说到职业规划，近期三到五年，我打算在 XXX 行业做到 XXX，希望可以稳定提升，持续学到更多的知识，后续可以在 XXX 行业/XXX 岗位独当一面，独立负责 XXXX 事务,解决 XXXX 问题。”

“谈到远期规划，我会根据环境的变化，工作内容的变化，以及我自身能力的变化，不断进行调整的。对于职业规划，我暂时的考虑是这样子的。谢谢！”

## 问题 10：“你最崇拜谁？”

思路：

- 1、 最崇拜的人能在一定程度上反映应聘者的性格、观念、心态，这是面试官问该问题的主要原因。
- 2、 不宜说自己谁都不崇拜。
- 3、 不宜说崇拜自己。
- 4、 不宜说崇拜一个虚幻的、或是不知名的人。
- 5、 不宜说崇拜一个明显具有负面形象的人。
- 6、 所崇拜的人人最好与自己所应聘的工作能“搭”上关系。
- 7、 最好说出自己所崇拜的人的哪些品质、哪些思想感染着自己、鼓舞着自己。

## 问题 11：“对这项工作，你有哪些可预见的困难？”

思路：

- 1、 不宜直接说出具体的困难，否则可能令对方怀疑应聘者不行。

2、 可以尝试迂回战术，说出应聘者对困难所持有的态度——“工作中出现一些困难是正常的，也是难免的，但是只要有坚忍不拔的毅力、良好的合作精神以及事前周密而充分的准备，任何困难都是可以克服的。”

## **问题 12：“如果我录用你，你将怎样开展工作”**

思路：

1、 如果应聘者对于应聘的职位缺乏足够的了解，最好不要直接说出自己开展工作的具体办法。

2、 可以尝试采用迂回战术来回答，如“首先听取领导的指示和要求，然后就有有关情况进行了解和熟悉，接下来制定一份近期的工作计划并报领导批准，最后根据计划开展工作。”

## **问题 13：“与上级意见不一是，你将怎么办？”**

思路：

1、 一般可以这样回答“我会给上级以必要的解释和提醒，在这种情况下，我会服从上级的意见。”

2、 如果面试你的是总经理，而你所应聘的职位另有一位经理，且这位经理当时不在场，可以这样回答：“对于非原则性问题，我会服从上级的意见，对于涉及公司利益的重大问题，我希望能向更高层领导反映。”

## **问题 14：“我们为什么要录用你？”**

思路：

1、 应聘者最好站在招聘单位的角度来回答。

2、 招聘单位一般会录用这样的应聘者：基本符合条件、对这份共组感兴趣、有足够的信心。

3、 如“我符合贵公司的招聘条件，凭我目前掌握的技能、高度的责任感和良好的适应能力及学习能力，完全能胜任这份工作。我十分希望能为贵公司服务，如果贵公司给我这个机会，我一定能成为贵公司的栋梁！”

## 问题 15：“你是应届毕业生，缺乏经验，如何能胜任这项工作？”

思路：

- 1、如果招聘单位对应届毕业生的应聘者提出这个问题，说明招聘单位并不真正在乎“经验”，关键看应聘者怎样回答。
- 2、对这个问题的回答最好要体现出应聘者的诚恳、机智、果敢及敬业。
- 3、如“作为应届毕业生，在工作经验方面的确会有所欠缺，因此在读书期间我一直利用各种机会在这个行业里做兼职。我也发现，实际工作远比书本知识丰富、复杂。但我有较强的责任心、适应能力和学习能力，而且比较勤奋，所以在兼职中均能圆满完成各项工作，从中获取的经验也令我受益非浅。请贵公司放心，学校所学及兼职的工作经验使我一定能胜任这个职位。”

•

## 问题 16：“你希望与什么样的上级共事？”

思路：

- 1、通过应聘者对上级的“希望”可以判断出应聘者对自我要求的意识，这既上一个陷阱，又上一次机会。
- 2、最好回避对上级具体的希望，多谈对自己的要求。
- 3、如“做为刚步入社会新人，我应该多要求自己尽快熟悉环境、适应环境，而不应该对环境提出什么要求，只要能发挥我的专长就可以了。”

### 1、请你简单介绍一下？

- 回答提示：一般人回答这个问题过于平常，只说姓名、年龄、爱好、工作经验，这些在简历上都有，其实，企业最希望知道的是求职者能否胜任工作，包括：最强的技能、最深入研究的知识领域、个性中最积极的部分、做过的最成功的事、主要的成就等，这些都可以和学习无关，也可以和学习有关，但要突出积极的个性和做事的能力，说得合情合理企业才会相信。企业很重视一个人的礼貌，求职者要尊重考官，在回答每个问题之后都说一句“谢谢”。企业喜礼貌的求职者。



## 2、你觉得你个性上最大的优点是什么？

- 回答提示:沉着冷静、条理清楚、立场坚定、顽强向上。  
乐于助人和关心他人、适应能力和幽默感、乐观和友爱.我在培训机构经过一到两年的培训及项目实战,加上实习工作,使我适合这份工作.我相信我能成功.

## 3、说说你最大的缺点？

- 回答提示:这个问题企业问的概率很大,通常不希望听到直接回答的缺点是什么等,如果求职者说自己小心眼、爱忌妒人、非常懒、脾气大、工作效率低,企业肯定不会录用你.绝对不要自作聪明地回答“我最大的缺点是过于追求完美”,有的人以为这样回答会显得自己比较出色,但事实上,他已经岌岌可危了.企业喜欢求职者从自己的优点说起,中间加一些小缺点,最后再把问题转回到优点上,突出优点的部分.企业喜欢聪明的求职者.

## 4、你对加班的看法？

- 回答提示:实际上好多公司问这个问题,并不证明一定要加班.只是想测试你是否愿意为公司奉献.
- 回答样本:如果是工作需要我会义不容辞加班.我现在单身,没有任何家庭负担,可以全身心的投入工作.但同时,我也会提高工作效率,减少不必要的加班

## 5、你对薪资的要求？

•

回 答提示:如果你对薪酬的要求太低,那显然贬低自己的能力;如果你对薪酬的要求太高,那又会显得你分量过重,公司受用不起.一些雇主通常都事先对求聘的职位定下开支预算,因而他们第一次提出的价钱往往是他们所能给予的最高价钱.他们问你只不过想证实一下这笔钱是否足以引起你对该工作的兴趣.

•

•

回答样本一:我对工资没有硬性要求.我相信贵公司在处理我的问题上会友善合理.我注重的是找对工作机会,所以只要条件公平,我则不会计较太多

•

•

回答样本二:我受过系统的软件编程的训练,不需要进行大量的培训.而且我本人也对编程特别感兴趣.因此,我希望公司能根据我的情况和市场标准的水平,给我合理的薪水.

- 
- 

回答样本三:如果你必须自己说出具体数目,请不要说一个宽泛的范围,那样你将只能得到最低限度的数字.最好给出一个具体的数字,这样表明你已经对当今的人才市场作了调查,知道像自己这样学历的雇员有什么样的价值.

- 

## 6、在五年的时间内,你的职业规划?

- 回答提示:这是每一个应聘者都不希望被问到的问题,但是几乎每个人都会被问到.比较多的答案是“管理者”.但是近几年来,许多公司都已经建立了专门的技术途径.这些工作地位往往被称作“顾问”、“参议技师”或“高级软件工程师”等等.当然,说出其他一些你感兴趣的职位也是可以的,比如产品销售部经理,生产部经理等一些与你的专业有相关背景的工作.要知道,考官总是喜欢有进取心的应聘者,此时如果说“不知道”,或许就会使你丧失一个好机会.最普通的回答应该是“我准备在技术领域有所作为”或“我希望能按照公司的管理思路发展”.

## 7、你朋友对你的评价?

- 回答提示:想从侧面了解一下你的性格及与人相处的问题.
- 回答样本:“我的朋友都说我是一个可以信赖的人.因为,我一旦答应别人的事情,就一定会做到.如果我做不到,我就不会轻易许诺.”  
回答样本:“我觉的我是一个比较随和的人,与不同的人都可以友好相处.在我与人相处时,我总是能站在别人的角度考虑问题”

## 8、你还有什么问题要问吗?

- 回答提示:企业的这个问题看上去可有可无,其实很关键,企业不喜欢说“没有问题”的人,因为其很注重员工的个性和创新能力.企业不喜欢求职者问个人福利之类的问题,如果有人这样问:贵公司对新入公司的员工有没有什么培训项目,我可以参加吗?或者说贵公司的晋升机制是什么样的?企业将很欢迎,因为体现出你对学习的热情和对公司的忠诚度以及你的上进心.

## 9、如果通过这次面试我们单位录用了你,但工作一段时间却发现你根本

## 不适合这个职位, 你怎么办?

回答提示:一段时间发现工作不适合我,有两种情况:

- 1、如果你确实热爱这个职业,那你就不断学习,虚心向领导和同事学习业务知识和处事经验,了解这个职业的精神内涵和职业要求,力争减少差距;
- 2、你觉得这个职业可有可无,那还是趁早换个职业,去发现适合你的,你热爱的职业,那样你的发展前途也会大点,对单位和个人都有好处。

## 10、在完成某项工作时,你认为领导要求的方式不是最好的,自己还有更好的方法,你应该怎么做?

- 回答提示:
- ①.原则上我会尊重和服从领导的工作安排;同时私底下找机会以请教的口吻,婉转地表达自己的想法,看看领导是否能改变想法;
- ②如果领导没有采纳我的建议,我也同样会按领导的要求认真地去完成这项工作;
- ③.还有一种情况,假如领导要求的方式违背原则,我会坚决提出反对意见;如领导仍固执己见,我会毫不犹豫地再向上级领导反映。

## 11、如果你的工作出现失误,给本公司造成经济损失,你认为该怎么办?

- 回答提示:
- ①我本意是为公司努力工作,如果造成经济损失,我认为首要的问题是想方设法去弥补或挽回经济损失.如果我无能力负责,希望单位帮助解决;
- ②是责任问题.分清责任,各负其责,如果是个人的责任,我甘愿受罚;如果是一个我负责的团队中别人的失误,也不能幸灾乐祸,作为一个团队,需要互相提携共同完成工作,安慰同事并且帮助同事查找原因总结经验.
- ③总结经验教训,一个人的一生不可能不犯错误,重要的是能从自己的或者是别人的错误中吸取经验教训,并在今后的工作中避免发生同类的错误.检讨自己的工作方法、分析问题的深度和力度是否不够,以致出现了本可以避免的错误。

## 12、如果你在这次考试或面试中没有被录用,你怎么打算?

- 回答提示:现在的社会是一个竞争的社会,从这次面试中也可看出这一点,有竞争就必然有优劣,有成功必定就会有失败.往往成功的背后有许多的困难和挫折,如果这次失败了也仅仅是一次而已,只有经过经验经历的积累才能塑造出一个完全的成功者.我会从以下几个方面来正确看待这次失败。

- 第一、要敢于面对,面对这次失败不气馁,接受已经失去了这次机会就不会回头这个现实,从心理意志和精神上体现出对这次失败的抵抗力.要有自信,相信自己经历了这次之后经过努力一定能行.能够超越自我.
- 第二、善于反思,对于这次面试经验要认真总结,思考剖析,能够从自身的角度找差距.正确对待自己,实事求是地评价自己,辩证的看待自己的长短得失,做一个明白人.
- 第三、走出阴影,要克服这一次失败带给自己的心理压力,时刻牢记自己弱点,防患于未然,加强学习,提高自身素质.
- 第四、认真工作,回到原单位岗位上后,要实实在在、踏踏实实地工作,三十六行,行行出状元,争取在本岗位上做出一定的成绩.
- 第五、再接再厉,成为软件工程师或网络工程师一直是我的梦想,以后如果有机会我仍然再次参加竞争.

### 13、如果你做的一项工作受到上级领导的表扬,但你主管领导却说是他做的,你该怎样?

- 回答提示:我首先不会找那位上级领导说明这件事,我会主动找我的主管领导来沟通,因为沟通是解决人际关系的最好办法,但结果会有两种:
- 1.我的主管领导认识到自己的错误,我想我会视具体情况决定是否原谅他;
- 2.他更加变本加厉的来威胁我,那我会毫不犹豫地找我的上级领导反映此事,因为他这样做会造成负面影响,对今后的工作不利.

### 14、谈谈你对跳槽的看法?

- 回答提示:
- (1) 正常的"跳槽"能促进人才合理流动,应该支持;
- (2) 频繁的跳槽对单位和个人双方都不利,应该反对.

### 15、工作中你难以和同事、上司相处,你该怎么办?

- 回答提示:
- ①我会服从领导的指挥,配合同事的工作.
- ②我会从自身找原因,仔细分析是不是自己工作做得不好让领导不满意,同事看不惯.还要看看是不是为人处世方面做得不好.如果是这样的话 我会努力改正.
- ③如果我找不到原因,我会找机会跟他们沟通,请他们指出我的不足.有问题就及时改正.
- ④作为优秀的员工,应该时刻以大局为重,即使在一段时间内,领导和同事对我不理解,我也会做好本职工作,虚心向他们学习,我相信,他们会看见我在努力,总有一天会对我微笑的!

### 16、假设你在某单位工作,成绩比较突出,得到领导的肯定.但同时你发

现同事们越来越孤立你, 你怎么看这个问题?你准备怎么办?

- 回答提示:
- ①成绩比较突出,得到领导的肯定是件好事情,以后更加努力
- ②检讨一下自己是不是对工作的热心度超过同事间交往的热心了,加强同事间的交往及共同的兴趣爱好.
- ③工作中,切勿伤害别人的自尊心
- ④不再领导前拨弄是非
- ⑤乐于助人对面

17、你最近是否参加了培训课程?谈谈培训课程的内容. 是公司资助还是自费参加?

- 回答提示:自费参加,在线培训机构 VIP 课程 (可以多谈谈自己学的技术)

18、你对于我们公司了解多少?

- 回答提示:在去公司面试前上网查一下该公司主营业务.如回答:贵公司有意改变策略,加强与国外大厂的 OEM 合作,自有品牌的部分则透过海外经销商.

19、请说出你选择这份工作的动机?

- 回答提示:这是想知道面试者对这份工作的热忱及理解度,并筛选因一时兴起而来应试的人,如果是无经验者,可以强调“就算职种不同,也希望有机会发挥之前的经验”.

20、你最擅长的技术方向是什么?

- 回答提示:说和你要应聘的职位相关的课程,表现一下自己的热诚没有什么坏处.

21、你能为我们公司带来什么呢?

- 回答提示:
- ①假如你可以的话,试着告诉他们你可以减低他们的费用——“我已经接受过培训机构近两年专业的培训,立刻就可以上岗工作”.
- ②企业很想知道未来的员工能为企业做什么,求职者应再次重复自己的优势,然后说:“就我的能力,我可以做一个优秀的员工在组织中发挥能力,给组织带来高效率和更多的收益”.企业喜欢求职者就申请的职位表明自己的能力,比如申请营销之类的职位,

可以说:“我可以开发大量的新客户,同时,对老客户做更全面周到的服务,开发老客户的新需求和消费.”等等.

## 22、最能概括你自己的三个词是什么?

- 回答提示:
- 我经常用的三个词是:适应能力强,有责任心和做事有始终,结合具体例子向主考官解释,

## 23、你的业余爱好是什么?

- 回答提示:找一些富于团体合作精神的,这里有一个真实的故事:有人被否决掉,因为他的爱好是深海潜水.主考官说:因为这是一项单人活动,我不敢肯定他能否适应团体工作.

## 24、作为被面试者给我打一下分

- 回答提示:试着列出四个优点和一个非常非常非常小的缺点,(可以抱怨一下设施,没有明确责任人的缺点是不会有人介意的).

## 25、你怎么理解你应聘的职位?

- 回答提示:把岗位职责和任务及工作态度阐述一下

## 26、喜欢这份工作的哪一点?

- 回答提示:相信其实大家心中一定都有答案了吧!每个人的价值观不同,自然评断的标准也会不同,但是,在回答面试官这个问题时可不能太直接就把自己心理的话说出来,尤其是薪资方面的问题,不过一些无伤大雅的回答是不错的考虑,如交通方便,工作性质及内容颇能符合自己的兴趣等等都是不错的答案,不过如果这时自己能仔细思考出这份工作的与众不同之处,相信在面试上会大大加分.

## 27、为什么要离职?

- 回答提示:
- ①回答这个问题时一定要小心,就算在前一个工作受到再大的委屈,对公司有多少的怨言,都千万不要表现出来,尤其要避免对公司本身主管的批评,避免面试官的负面情绪及印象;建议此时最好的回答方式是将问题归咎在自己身上,例如觉得工作没有学习发展的空间,自己想面试工作的相关产业中多加学习,或是前一份工作与自己的生涯规划不合等等,回答的答案最好是积极正面的.

- ②我希望能获得一份更好的工作,如果机会来临,我会抓住;我觉得目前的工作,已经达到顶峰,即没有升迁机会.

## 28、说说你对行业、技术发展趋势的看法?

回答提示:企业对这个问题很感兴趣,只有有备而来的求职者能够过关.求职者可以直接在网上查找对你所申请的行业部门的信息,只有深入了解才能产生独特的见解.企业认为最聪明的求职者是对所面试的公司预先了解很多,包括公司各个部门,发展情况,在面试回答问题的时候可以提到所了解的情况,企业欢迎进入企业的人是“知己”,而不是“盲人”.

•

## 29、对工作的期望与目标何在?

•

•

回答提示:这是面试者用来评断求职者是否对自己有一定程度的期望、对这份工作是否了解的问题.对于工作有确实学习目标的人通常学习较快,对于新工作自然较容易进入状况,这时建议你,最好针对工作的性质找出一个确实的答案,如业务员的工作可以这样回答:“我的目标是能成为一个超级业务员,将公司的产品广泛的推销出去,达到最好的业绩成效;为了达到这个目标,我一定会努力学习,而我相信以我认真负责的态度,一定可以达到这个目标.”其他类的工作也可以比照这个方式来回答,只要在目标方面稍微修改一下就可以了.

•

## 30、说说你的家庭.

- 回答提示:企业面试时询问家庭问题不是非要知道求职者家庭的情况,探究隐私,企业不喜欢探究个人隐私,而是要了解家庭背景对求职者的塑造和影响.企业希望听到的重点也在于家庭对求职者的积极影响.企业最喜欢听到的是:我很爱我的家庭!我的家庭一向很和睦,虽然我的父亲和母亲都是普通人,但是从小,我就看到我父亲起早贪黑,每天工作特别勤劳,他的行动无形中培养了我认真负责的态度和勤劳的精神.我母亲为人善良,对人热情,特别乐于助人,所以在单位人缘很好,她的一言一行也一直在教导我做人的道理.企业相信,和睦的家庭关系对一个人的成长有潜移默化的影响.

## 31、就你申请的这个职位,你认为你还欠缺什么?

- 回答提示:企业喜欢问求职者弱点,但精明的求职者一般不直接回答.他们希望看到这样的求职者:继续重复自己的优势,然后说:“对于这个职位和我的能力来说,我相信自己是可以胜任的,只是缺乏经验,这个问题我想我可以进入公司以后以最短的时间来解决,我的学习能力很强,我相信可以很快融入公司的企业文化,进入工作状态.”企业喜欢能够巧妙地躲过难题的求职者.

### 32、你欣赏哪种性格的人?

- 回答提示:诚实、不刻板而且容易相处的人、有“实际行动”的人.

### 33、你通常如何处理别人的批评?

- 回答提示
- ①沈默是金.不必说什么,否则情况更糟,不过我会接受建设性的批评;
- ②我会等大家冷静下来再讨论.

### 34、你怎样对待自己的失败?

- 回答提示:我们大家生来都不是十全十美的,我相信我有第二个机会改正我的错误.

### 35、什么会让你有成就感?

回答提示:为贵公司竭力效劳;尽我所能,完成一个项目

### 36、眼下你生活中最重要的是什么?

- 回答提示:对我来说,能在这个领域找到工作是最重要的;望能在贵公司任职对我说最重要.

### 37、你为什么愿意到我们公司来工作?

- 回答提示:对于这个问题,你要格外小心,如果你已经对该单位作了研究,你可以回答一些详细的原因,像“公司本身的高技术开发环境很吸引我.”“我同公司出生在同样的时代,我希望能够进入一家与我共同成长的公司.”“你们公司一直都稳定发展,在近几年来在市场上很有竞争力.”或者“我认为贵公司能够给我提供一个与众不同的发展道路.”这都显示出你已经做了一些调查,也说明你对自己的未来有了较为具体的远景规划.

### 38、你和别人发生过争执吗?你是怎样解决的?



- 回答提示:这是面试中最险恶的问题.其实是考官布下的一个陷阱.千万不要说任何人的过错.应知成功解决矛盾是一个协作团体中成员所必备的能力.假如你工作在一个服务行业,这个问题简直成了最重要的一个环节.你是否能获得这份工作,将取决于这个问题的回答.考官希望看到你是成熟且乐于奉献的.他们通过这个问题了解你的成熟度和处世能力.在没有外界干涉的情况下,通过妥协的方式来解决才是正确答案.

### 39、问题:你做过的哪件事最令自己感到骄傲?

- 回答提示:这是考官给你的一个机会,让你展示自己把握命运的能力.这会体现你潜在的领导能力以及你被提升的可能性.假如你应聘于一个服务性质的单位,你很可能被邀请去午餐.记住:你的前途取决于你的知识、你的社交能力和综合表现.

### 40、你新到一个部门,一天一个客户来找你解决问题,你努力想让他满意,可是始终达不到群众得满意,他投诉你们部门工作效率低,你这个时候怎么作?

- 回 答提示:
- 1)首先,我会保持冷静.作为一名工作人员,在工作中遇到各种各样的问题是正常的,关键是如何认识它,积极应对,妥善处理.
- 2)其次,我会反思一下客户不满意的原因.一是看是否是自己在解决问题上的确有考虑的不周到的地方,二是看是否是客户不太了解相关的服务规定而提出超出规定的要求,三是看是否是客户了解相关的规定,但是提出的要求不合理.
- 3)再次,根据原因采取相对的对策.如果是自己确有不周到的地方,按照服务规定作出合理的安排,并向客户作出解释;如果是客户不太了解政策规定而造成的误解,我会向他作出进一步的解释,消除他的误会;如果是客户提出的要求不符合政策规定,我会明确地向他指出.
- 4)再次,我会把整个事情的处理情况向领导作出说明,希望得到他的理解和支持.
- 5)我不会因为客户投诉了我而丧失工作的热情和积极性,而会一如既往地牢记为客户服务的宗旨,争取早日做一名领导信任、公司放心、客户满意的职员.

### 41、对这项工作,你有哪些可预见的困难?”

- 回答提示:
- ①不宜直接说出具体的困难,否则可能令对方怀疑应聘者不行;
- ②可以尝试迂回战术,说出应聘者对困难所持有的态度——“工作中出现一些困难是正常的,也是难免的,但是只要有坚忍不拔的毅力、良好的合作精神以及事前周密而充分的准备,任何困难都是可以克服.”
- 分析:一般问这个问题,面试者的希望就比较大了,因为已经在谈工作细节.但常规思路中的回答,又被面试官“骗”了.当面试官询问这个问题的时候,有两个目的.第一,看看应聘者是不是在行,说出的困难是不是在这个职位中一般都不可避免的问题.第二,是想

看一下应聘者解决困难的手法对不对,及公司能否提供这样的资源.而不是想了解应聘者对困难的态度.

## 42、如果我录用你,你将怎样开展工作?”

- 回答提示:
- ①如果应聘者对于应聘的职位缺乏足够的了解,最好不要直接说出自己开展工作的具体办法;
- ②可以尝试采用迂回战术来回答,如“首先听取领导的指示和要求,然后就有关情况进行了解和熟悉,接下来制定一份近期的工作计划并报领导批准,最后根据计划开展工作.”
- 分析:这个问题的主要目的也是了解应聘者的工作能力和计划性、条理性,而且重点想要知道细节.如果向思路中所讲的迂回战术,面试官会认为回避问题,如果引导了几次仍然是回避的话.此人绝对不会录用了.

## 43、“你希望与什么样的上级共事?”

- 回答提示:
- ①通过应聘者对上级的“希望”可以判断出应聘者对自我要求的意识,这既上一个陷阱,又是一次机会;
- ②最好回避对上级具体的希望,多谈对自己的要求;
- ③如“做为刚步入社会的新人,我应该多要求自己尽快熟悉环境、适应环境,而不应该对环境提出什么要求,只要能发挥我的专长就可以了
- 分析:这个问题比较好的回答是,希望我的上级能够在工作中对我多指导,对我工作中的错误能够立即指出.总之,从上级指导这个方面谈,不会有大的纰漏.

## 44、与上级意见不一是,你将怎么办?”

- 回答提示:
- ①一般可以这样回答“我会给上级以必要的解释和提醒,在这种情况下,我会服从上级的意见.”
- ②如果面试你的是总经理,而你所应聘的职位另有一位经理,且这位经理当时不在场,可以这样回答:“对于非原则性问题,我会服从上级的意见,对于涉及公司利益的重大问题,我希望能向更高层领导反映.”
- 分析:这个问题的标准答案是思路1,如果用2的回答,必死无疑.你没有摸清楚改公司的内部情况,先想打小报告,这样的人没有人敢要.

## 45、您在前一家公司的离职原因是什么?”

- 回答提示:

- ①最重要的是:应聘者要使招聘单位相信,应聘者在过往的单位的“离职原因”在此家招聘单位里不存在;
- ②避免把“离职原因”说得太详细、太具体;
- ③不能掺杂主观的负面感受,如“太辛苦”、“人际关系复杂”、“管理太混乱”、“公司不重视人才”、“公司排斥我们某某的员工”等;
- ④但也不能躲闪、回避,如“想换环境”、“个人原因”等;
- ⑤不能涉及自己负面的人格特征,如不诚实、懒惰、缺乏责任感、不随和等;
- ⑥尽量使解释的理由为应聘者个人形象添彩;
- ⑦相关例子:如“我离职是因为这家公司倒闭;我在公司工作了三年多,有较深的感情;从去年始,由于市场形势突变,公司的局面急转直下;到眼下这一步我觉得很遗憾,但还要面对现实,重新寻找能发挥我能力的舞台。”同一个面试问题并非只有一个答案,而同一个答案并不是在任何面试场合都有效,关键在应聘者掌握了规律后,对面试的具体情况把握,有意识地揣摩面试官提出问题的心理背景,然后投其所好。
- 分析:除非是薪资太低,或者是最初的工作,否则不要用薪资作为理由。“求发展”也被考官听得太多,离职理由要根据每个人的真实离职理由来设计,但是在回答时一定要表现得真诚.实在想不出来的时候,家在外地可以说是因为家中有事,须请假几个月,公司又不可能准假,所以辞职.这个答案一般面试官还能接受。

#### 46、“你工作经验欠缺,如何能胜任这项工作?”

- 回答提示:
- ①如果招聘单位对应届毕业生的应聘者提出这个问题,说明招聘公司并不真正在乎“经验”,关键看应聘者怎样回答;
- ②对这个问题的回答最好要体现出应聘者的诚恳、机智、果敢及敬业;
- ③如“作为应届毕业生,在工作经验方面的确会有所欠缺,因此在读书期间我一直利用各种机会在这个行业里做兼职.我也发现,实际工作远比书本知识丰富、复杂.但我有较强的责任心、适应能力和学习能力,而且比较勤奋,所以在兼职中均能圆满完成各项工作,从中获取的经验也令我受益非浅.请贵公司放心,学校所学及兼职的工作经验使我一定能胜任这个职位。”
- 分析:这个问题思路中的答案尚可.突出自己的吃苦能力和适应性以及学习能力（不是学习成绩）为好。

#### 47、为了做好你工作份外之事,你该怎样获得他人的支持和帮助?

- 回答提示:每个公司都在不断变化发展的过程中;你当然希望你的员工也是这样.你希望得到那些希望并欢迎变化的人,因为这些人明白,为了公司的发展,变化是公司日常生活中重要组成部分.这样的员工往往很容易适应公司的变化,并会对变化做出积极的响应.此外,他们遇到矛盾和问题时,也能泰然处之.下面的问题能够考核应聘者这方面的能力。
- 据说有人能从容避免正面冲突.请讲一下你在这方面的经验和技巧.有些时候,我们得和我们不喜欢的人在一起共事.说说你曾经克服了性格方面的冲突而取得预期工作效果的经历。

#### 48、假如你晚上要去送一个出国的同学去机场,可单位临时有事非你办不可,你怎么办?

- 回答提示:我觉得工作是第一位的,但朋友间的情谊也是不能偏废的.这个问题我觉得要按照当时具体的情况来决定.
- 1)、如果我的朋友晚上 9 点中的飞机,而我的 加班八点就能够完成的话,那就最理想了,干完工作去机场,皆大欢喜.
- 2)、如果说工作不是很紧急,加班仅仅是为了明天上班的时候能把报告交到办公室,那完全可以跟领导打声招呼,先去机场然后回来加班,晚点睡就是了.
- 3)、如果工作很紧急,两者不可能兼顾的情况下,我觉得可以由两种选择.。A、如果不是全单位都加班的话,是不是可以要其他同事来代替以下工作,自己去机场,哪怕就是代替你离开的那一会儿.2、如果连这一点都做不到的话,那只好忠义不能两全了,打电话给朋友解释一下,小心他会理解,毕竟工作做完了就完了,朋友还是可以再见面的.

#### 49、如果通过这次面试我们单位录用了你,但工作一段时间却发现你根本不适合这个职位,你怎么办?

- 回答提示:一段时间发现工作不适合我,有两种情况:˙
- 1、如果你确实热爱这个职业,那你就不断学习,虚心向领导和同事学习业务知识和处事经验,了解这个职业的精神内涵和职业要求,力争减少差距;
- 2、你觉得这个职业可有可无,那还是趁早换个职业,去发现适合你的,你热爱的职业,那样你的发展前途也会大点,对单位和个人都有好处.

#### 50、谈谈你过去做过的成功案例

- 回答提示:举一个你最有把握的例子,把来龙去脉说清楚,而不要说了很多却没有重点.切忌夸大其词,把别人的功劳到说成自己的,很多主管为了确保要用的人是最适合的,会打电话向你的前一个主管征询对你的看法及意见,所以如果说谎,是很容易穿梆的.

#### 51、谈谈你过去的工作经验中,最令你挫折的事情

- 回答提示:曾经接触过一个客户,原本就有耳闻他们以挑剔出名,所以事前的准备功夫做得十分充分,也投入了相当多的时间与精力,最后客户虽然并没有照单全收,但是接受的程度已经出乎我们意料之外了.原以为从此可以合作愉快,却得知客户最后因为预算关系选择了另一家代理商,之前的努力因而付诸流水.尽管如此,我还是从这次的经验学到很多,如对该产业的了解,整个 team 的默契也更好了.
- 分析:借此了解你对挫折的容忍度及调解方式.

## 52、如何安排自己的时间?会不会排斥加班?

- 回答提示:基本上,如果上班工作有效率,工作量合理的话,应该不太需要加班.可是我也知道有时候很难避免加班,加上现在工作都采用责任制,所以我会调配自己的时间,全力配合.
- 分析:虽然不会有人心甘情愿的加班,但依旧要表现出高配合度的诚意.

## 53、为什么我们要在众多的面试者中选择你?

- 回答提示:根据我对贵公司的了解,以及我在这份工作上所累积的专业、经验及人脉,相信正是贵公司所找寻的人才.而我在工作态度、EQ上,也有圆融、成熟的一面,和主管、同事都能合作愉快.
- 分析:别过度吹嘘自己的能力,或信口开河地乱开支票,例如一定会为该公司带来多少钱的业务等,这样很容易给人一种爱说大话、不切实际的感觉.

## 54、对这个职务的期许?

- 回答提示:希望能借此发挥我的所学及专长,同时也吸收贵公司在这方面的经验,就公司、我个人而言,缔造“双赢”的局面.
- 分析:回答前不妨先询问该公司对这项职务的责任认定及归属,因为每一家公司的状况不尽相同.以免说了一堆理想抱负却发现牛头不对马嘴.

## 55、为什么选择这个职务?

- 回答提示::这一直是我的兴趣和专长,经过这几年的磨练,也累积了一定的经验及人脉,相信我一定能胜任这个职务的.
- 分析:适时举出过去的“丰功伟业”,表现出你对这份职务的熟稔度,但避免过于夸张的形容或流于炫耀.

## 56、为什么选择我们这家公司?

- 回答提示:曾经在报章杂志看过关于贵公司的报道,与自己所追求的理念有志一同.而贵公司在业界的业绩也是有目共睹的,而且对员工的教育训练、升迁等也都很有制度.
- 分析:去面试前先做功课,了解一下该公司的背景,让对方觉得你真的很有心想得到这份工作,而不只是探探路.

## 57、你认为你在学校属于好学生吗?

- 回答提示:企业的招聘者很精明,问这个问题可以试探出很多问题:如果求职者学习成绩好,就会说:“是的,我的成绩很好,所有的成绩都很优异.当然,判断一个学生是不是好学生有很多标准,在学校期间我认为成绩是重要的,其他方面包括思想道德、实践经验、团队精神、沟通能力也都是很重要的,我在这些方面也做得很好,应该说我是一个全面发展的学生.”如果求职者成绩不尽理想,便会说:“我认为是不是一个好学生的标准是多元化的,我的学习成绩还可以,在其他方面我的表现也很突出,比如我去很多地方实习过,我很喜欢在快节奏和压力下工作,我在学生会组织过 xx活动,锻炼了我的团队合作精神和组织能力.”有经验的招聘者一听就会明白,企业喜欢诚实的求职者.

## 58、请谈谈如何适应办公室工作的新环境?

- 回答提示
- ①办公室里每个人有各自的岗位与职责,不得擅离岗位.
- ②根据领导指示和工作安排,制定工作计划,提前预备,并按计划完成.
- ③多请示并及时汇报,遇到不明白的要虚心请教
- ④抓间隙时间,多学习,努力提高自己的政治素质和业务水平.

## 59、在工作中学习到了些什么?

- 回答提示:这是针对转职者提出的问题,建议此时可以配合面试工作的特点作为主要依据来回答,如业务工作需要与人沟通,便可举出之前工作与人沟通的例子,经历了哪些困难,学习到哪些经验,把握这些要点做陈述,就可以轻易过关了

## 60、有想过创业吗?

- 回答提示:这个问题可以显示你的冲劲,但如果你的回答是“有”的话,千万小心,下一个问题可能就是“那么为什么你不这样做呢?”

## 61、除了本公司外,还应聘了哪些公司?

- 回答提示:很奇怪,这是相当多公司会问的问题,其用意是要概略知道应徵者的求职志向,所以这并非绝对是负面答案,就算不便说出公司名称,也应回答“销售同种产品的公司”,如果应聘的其他公司是不同业界,容易让人产生无法信任的感觉.

## 62、何时可以到职?

- 回答提示:大多数企业会关心就职时间,最好是回答,如果被录用的话,到职日可按公司规定上班”,但如果还未辞去上一个工作、上班时间又太近,似乎有些强人所难,因为交接至少要一个月的时间,应进一步说明原因,录取公司应该会通融的

### 63、你并非毕业于名牌院校？

- 回答提示:是否毕业于名牌院校不重要,重要的是有能力完成您交给我的工作,比如说:接受了职业培训机构学习或系统学习了相关的技术知识体系,掌握的技能完全可以胜任贵公司现在工作,而且我比一些名牌院校的应届毕业生的动手能力还要强,我想我更适合贵公司这个职位.

### 64、你怎样看待学历和能力？

- 回答提示:学历我想只要是大学专科的学历,就表明觉得我具备了根本的学习能力.剩下的,你是学士也好,还是博士也好,对于这一点的讨论,不是看你学了多少知识,而是看你在这个领域上发挥了什么,也就是所说的能力问题.一个人工作能力的高低直接决定其职场命运,而学历的高低只是进入一个企业的敲门砖,如果贵公司把学历卡在博士上,我就无法进入贵公司,当然这不一定只是我个人的损失,如果一个专科生都能完成的工作,您又何必非要招聘一位博士生呢?

### 65 你对薪资有什么要求？

- 我觉得向薪资这样的问题，考官应该考量的是面试者的思维和能力。每一个公司都有自己的薪酬体系，都是规定过的，赔本的买卖公司肯定不会做的。我觉得这个问题我回答的话，就是应该切入到现在和以后的关系：现在我想要的薪资是和自己的付出相协调的薪资，能够维持自己日常的基本开支之后有盈余（要给自己留有一定的活动资金，以备不时之需）；以后的话，我希望能随着公司的发展以及我个人的能力的提高薪资不断的提高。不希望日日涨工资，只是希望工资的增长能够跟上物价的上涨，不会沦落到拿以前的积蓄来养活自己

### 66 这几年你有什么职业规划？

- 我的长期目标是在一家优秀的企业担任领导职务，为了实现这一目标，我每五年都会制订一项个人职业发展计划，这个计划中包含一个阶段性目标和一系列短期目标。我每六个月会对目标的达成情况做出评估，然后对阶段性目标做出必要的修改。这些目标的设定与达成，使我积累了一些成功的经验，也掌握了一定的工作技能。

