面试中会被考察的具体问题有三个影响因素：**部门需求、简历、面试中提到的知识点**。因此不同的部门不同的面试者遇到的面试题很可能是完全不一样的，这也是我为什么不直接把面试题放出来的原因，没有多大的参考价值。

这一部分主要总结在面试中被考察到的知识点，多为高频面试问题，读者可以自行考察是否这些知识都掌握了。需要特别注意的是这**并不是全部的知识体系，仅仅是笔者遇到的面试题**，读者可以按照大体的方向去学习，但切不可认为这样就足够了，这里提到的知识点是**远远不够**的。

面试的过程中更多的时候不会单纯地考察知识点，而是会结合一些需求或者业务场景来多方面考察对知识点的理解。因此笔者建议，对于知识点要以理解为首要任务，而不是背诵答案。

**Android**

总体来说，本次春招问的android内容其实不多，主要还是看着我的简历问，简历写得不多，问得也就比较少。但android的基础内容如handler、view机制等还是要会的，其他的有更加深入的研究可以写在简历加分，如Binder、系统服务、优化等。

* Activity
  1. Activity启动流程。其中考察最多的类似问题是：【从桌面点击一个图标之后，到界面显示，这个过程发生了什么？】。很多时候面试官会结合activity生命周期来考问：在启动流程的哪些阶段哪些生命周期被回调，此时Activity状态如何。
  2. 启动模式。也就是常见的四种启动模式，但面试官更喜欢问何时使用他们，也就是使用场景。
  3. 生命周期。这个很少单独问，一般和启动流程或者具体的业务场景结合考问。
  4. context。主要是内存泄露的考察以及application和activity两种context如何选择。
* Handler
  1. 内部原理。Handler必会的啊，android的消息机制，可以称为android程序的引擎来的。
  2. 同步屏障。涉及到绘制优化、屏幕刷新机制等。
  3. 阻塞唤醒原理。这里一般是会问为何loop()方法是死循环却不会占用cpu时间片 or 为何next()方法阻塞却不会卡死。更深一点会问到Linux的IO多路复用epoll原理。
  4. 卡顿与内存优化。整个主线程的所有任务都必须经过Looper，是排查卡顿和ANR的关键点，以及消息太多会造成的后果等。
  5. 消息复用。
* 图片
  1. 计算一张图片大小。分辨率x像素点大小，考察得很多次。
  2. 加载优化。如LaunchActivity同时加载的图片太多如何优化、view的大小比图片小如何优化等。
  3. 缓存优化。内存缓存、硬盘缓存。
  4. Glide框架。重点就是Glide对上面的优化的实现原理，因此需要着重看Glide的缓存原理。
  5. drawable。对比使用图片和drawable的好处，以及drawable的原理。
* SharePreference
  1. 内部原理
  2. commit和apply的区别
  3. 这部分考察得不多，但建议读者可以深入理解sp的缺点，如导致ANR原理，以及新框架MMKV、Data Store的优点。
* window

考察得非常少，主要window的类型、以及window的真正定义理解即可。

* view
  1. 事件分发流程。考察得最多，基本把整个分发流程讲清楚就好了；其次还会考察如何解决具体的冲突场景。
  2. 应用界面的view层级。
  3. 绘制流程与时机。activity启动时到onResume方法被调用，view依旧还未被绘制。
* IPC
  1. 常见IPC类型以及优缺点。
  2. Binder机制。优点、缺点、特点、和传统IPC比较。Binder涉及到的很多是偏底层，更多的时候考察的是上层的应用，如和socket比较等。
  3. socket。这个会重点问，涉及优缺点、使用场景、和binder相比等。
* 序列化
  1. serializable和parcelable的原理，以及各自的优缺点、应用场景。
* jetpack
  1. 使用jetpack的好处与坏处。
  2. 框架原理。这部分如果写在简历也很少问，看面试官；但如果问的话，一般会问原理。
* okHttp
  1. 拦截器以及责任链思想。这个是问的最多的。
  2. 内部调度器对线程、任务数的并发控制。
  3. 优点缺点，和URLConnection相比的好处，诞生的背景等。
* 优化
  1. 性能优化。问得很多，主要看自己平时有没有做过优化。
  2. 卡顿优化。一般询问如何定位和解决卡顿问题。
  3. 内存优化。一般是内存泄露、或者减少内存占用等。
  4. ANR。一般会考察原理以及如何解决。
  5. 工具：leakcanary、profiler。优化涉及的内容太多，这个属于比较深的内容，还是得看自己平时的项目积累。

**Java**

Java的考察是一个非常考验功底的地方，很多的知识点都是基础中的基础，同时又都可以延伸到底层的逻辑。这些知识点在实际的运用中，也有很多的情况需要考虑，考察我们对于知识点的运用。例如线程池，可以从参数作用、到线程池原理、到阻塞唤醒机制、到实际项目的参数配置，有非常多的知识点可以考察。因而这一块就看各位的造诣了。

当然，一些基础或者常考的知识点还是得会的。

* 集合框架

1. HashMap。几乎每家公司都问，主要是内部原理如hash算法、冲突解决方案、扩容方案、红黑树的优缺点等。必会的内容，不会就直接当场去世了。
2. HashSet。内部使用HashMap来实现，value设置为object。记住这个就好了。
3. ConcurrentHashMap。必问。他的并发原理以及好处，同时有些面试官也会问缺点等问题。
4. Hashtable、SychronizeMap。一般和ConcurrentHashMap一起问，进行对比。
5. CopyOnWriteArrayList。一般会作为线程安全方法来进行比较优缺点。
6. 集合框架重点还是在Map，但是其他的框架List和queue的原理也是要了解的。

* 访问限制符

public protect default private 四个要懂，基础知识了。（笔者就是忽略了这些当时回答错了）特别注意protect是可以跨包访问的。

* 类
  1. 4种内部类，特别注意每个class编译后都会产生一个class文件，不管静态或非静态。面试踩坑了
  2. lambda的本质。就是匿名内部类。
  3. 抽象类和接口的区别。这个很看理解，如果有开发过具体项目的会回答得更加深刻，这是背八股文体现不出来的。
* 异常
  1. 异常体系、分类、机制。
  2. 与error的区别。
* IO

主要还是问NIO的原理以及优缺点。建议把缓冲流的原理也得学一学并进行比较。

* 线程池
  1. 内部原理。必会的啊。
  2. 关键参数作用及如何配置。重点在如何配置，需要结合具体的机器情况、任务情况等等考量。
  3. 线程池的作用。不仅仅只是线程复用，更重要的是管理线程、控制线程数量。这个也比较考察具体的项目运用理解。
  4. 常见的四种线程池。
* 并发
  1. sychronize。必问，java的锁机制。特别是jdk6之后的锁优化以及运用场景。为什么是重量级的、JVM层如何实现如果了解可以加分。
  2. Lock。必问，AQS的原理最好懂。一般会拿来和synchronize比较。
  3. volatile。必问，会拿来和锁比较，他的两个重要作用。更深点会问到cpu缓存一致性协议、以及指令重排的类型与原理。
  4. CAS。必问，问原理以及ABA问题。
  5. 死锁。一般询问如何解决或者产生的条件。
  6. Object的wait和notify。阻塞唤醒，一般会用一个代码或者具体的场景来询问如何保证多线程同步。
  7. ThreadLocal。原理、内存泄露等
  8. 这一块问的还是比较多，而且大都可以深入去问，看自己的学习程度了。
* JVM
  1. GC机制。必问。
  2. 类加载机制。必问，同时还会问双亲委托机制。
  3. 方法调用过程。这个也问的挺多，也看对JVM的学习程度了。
  4. 线程与进程的内存关系。如一个线程占多少内存、一个进程可以开多少线程、一个进程占用多少内存等。
  5. 内存分布。JMM、运行时数据区、native内存分布。很看对JVM的理解程度。

**Kotlin**

kotlin其实问的很少，现在大部分的项目还是使用Java开发的。一些有kotlin业务的公司会问，但是问的内容也不多。这一块如果还没学习其实不急的，把Java学好才是王道，kotlin和java在原理上，都是互通的。

* 协程

问的是最多的，主要是理解线程和协程的关系、协程的优缺点。这个也看个人的学习程度了。

* run、let、also、with、apply

注意返回值以及作用域

* 特性的具体实现

如默认参数的具体实现。有了解过最好了，没有的话就联想Java是如何实现的，如默认参数可以联想方法重载。

**计算机网络**

计网这一部分主要还是HTTP和TCP的内容了，经典中的经典。需要注意的是链路层、网络层的一些协议要了解，也是计算机基础了，被问到不会的话会比较尴尬。其次是一些新的协议如QUIC、http3.0等可以了解一下，面试会很加分，同时也可以进一步去理解TCP的优缺点。

* HTTP

http在android上的体现并不多，因为大部分的工作都给框架解决了，问的问题其实很少。

* 1. 历代http的优化以及原因
  2. 请求方法、响应码
* HTTPS

必问。一定要会了

* 1. 原理以及和http的区别。加密算法、hash摘要、ca证书验证都要了解
  2. 建立连接过程
  3. 破解：中间人攻击等
* 数据链路层、IP层
  1. ARP和RARP协议
  2. NAT协议
  3. DNS
  4. 这部分主要问一些常见的协议，考察计网功底，这里列出来的是笔者考察过的，读者需要比价系统地去学习这一块。
* TCP
  1. 握手挥手
  2. 拥塞控制
  3. 可靠传输原理
  4. 缺点以及如何改进。这个是比较重要的，对应http3.0的优化就是针对TCP的缺点来入手的。
  5. TCP的连接数目上限
  6. TCP非常重要，必问的内容，不会的读者一定要去好好学习一下。
* UDP
  1. 优缺点
  2. 和TCP比较
  3. 应用场景
  4. 一般和TCP一起出现，询问他们的区别，以及如何通过UDP来优化TCP的缺点。
* 数据格式

json的优缺点，为什么要使用json而不是XML。熟记就可以了。

**设计模式**

设计模式这一块问的不是很多，原因是设计模式需要实际的编码才能去真正理解他的好处。所以大部分情况下面试官都是询问在项目中用了哪些设计模式、或者哪些开源框架用了什么设计模式。其次问的最多的就是三大架构模式了，必会的。

* 常见的设计模式

一般面试官会让我们讲自己认识的设计模式，按照自己的理解讲出来就好了。

* 在项目和源码中的体现

问的最多的就是：你看过的框架源码都使用了哪些设计模式。不仅考察了设计模式、还考察了源码。另外也会问在项目中运用了哪些设计模式。

* MVC\MVP\MVVM

基本必问了。结合具体的项目实现以及自己的思考，分析他们的优缺点、适用场景等。

**计算机基础**

计算机基础这里包括了c语言、操作系统、数据库库、以及前面的计网等等。这部分直接问的内容不多，但却是非常重要的，也是考验我们基础扎不扎实的地方。其次，面试官也会用一些业务场景来考验我们的理解程度，如聊到java并发时，问具体为何线程切换代价高；又如聊到JVM内存分布时，问在native层是怎样的内存分布以及逻辑内存物理内存等；再如聊到volatile，会问到cpu缓存一致性的问题等等。这些都是非常考验功底的内容。因而，这些基础内容要学好，不仅是为了面试，更是为了让我们自己可以更好地学习理解知识。

* C语言

问的很少，偶尔会问到。笔者只被问到：同构结构体的内存占用、指针相关问题。

C语言作为语言鼻祖，一些基础还是要会的。

* 操作系统
  1. 中断原理、作用。
  2. 死锁原理以及如何解决。
  3. 操作系统这一块确实问的不多，同样也是考察基础功底了。可以不问，但是问到一定要会。

**项目**

项目是非常重要的，一个拥有亮点的项目可为面试加分不少。面试前需要把自己的项目总结一下，提炼出项目的亮点难点。假如有优秀的项目，一定要把他写在简历上，并把亮点表现出来。

在面试中面试官一般会先让我们介绍项目，介绍项目一般可以按照这个模板来介绍：**项目的背景、项目的内容、自己负责的内容、项目亮点**。技术面的面试官会询问项目有什么技术亮点难点，这个时候可以把我们提前准备的亮点给面试官表述出来。例如做了什么优化、自定义view等，体现出自己在技术上的能力。

当然，很多时候我们的项目并没有什么亮点，甚至就是一个Demo，例如笔者。在被面试官问到项目有什么技术难点时支支吾吾说不出来。这时候可以把项目自己实现的功能讲出来，例如一个wanAndroid App可以这样描述：首页使用viewPager+recyclerView来展示后台数据，点击后可跳转到对应文章详情页；登录注册需要跳转到另一个Activity，采用okHttp拦截器和SharePreference来实现记住登录 。千万不要说：额、、就是一个练习demo，拉数据展示，没有什么亮点。这样让面试官完全无法了解我们的项目到底如何，也无法从项目开始提问来了解我们，面试效果就大打折扣了。

面试官对于我们的项目有了一定的了解之后，会继续深问一些相关的知识点或者一些情况是如何解决的。以wanAndroid App为例子，面试官可以进一步提问：viewPager有做缓存吗？recyclerView缓存机制如何？首页图片过多首页加载缓慢导致卡顿如何解决？为什么使用拦截器实现，他有什么优缺点？SharePreference的原理如何等等。

非技术面试官可能会问一些如：项目中学习到什么？你们是如何合作的？项目过程中遇到过什么问题等等。常见的问题可以提前准备，把自己的心得整理一下。

那，没有项目怎么办？那赶紧去写一个。正如某个大厂的面试官说的：**到现在都不知道你们这些实习生，到底是真的会，还是背八股文的**。的确如此，有时候我们甚至都不知道自己是不是真的懂，唯一的办法就是，是骡子是马拉出来遛一遛，写个项目。项目是面试官一个了解我们的非常好的方式。

需要注意的是：**面试前要把自己的项目总结好，包括背景、内容、技术难点、亮点；面试时不要觉得自己项目很捞，而要表现出自己的自信，把项目内容有条不紊地表达出来**。后面回答好面试官提出的问题就可以了。

**算法**

算法这一块，先看看笔者在面试中遇到的算法题：(点击可跳转leetcode对应题目)

* [重排链表](https://leetcode-cn.com/problems/reorder-list/)
* [出现一次的字符](https://leetcode-cn.com/problems/di-yi-ge-zhi-chu-xian-yi-ci-de-zi-fu-lcof/)
* [K个一组链表反转](https://leetcode-cn.com/problems/reverse-nodes-in-k-group/)
* [凑零钱](https://leetcode-cn.com/problems/coin-change/)
* [全路径简化](https://leetcode-cn.com/problems/simplify-path/)
* [字符串拆分](https://leetcode-cn.com/problems/word-break/)
* [二叉树路径和](https://leetcode-cn.com/classic/problems/path-sum-iii)
* [链表是否有环](https://leetcode-cn.com/problems/linked-list-cycle/)
* [二叉树深度](https://leetcode-cn.com/problems/er-cha-shu-de-shen-du-lcof/)

算法难度基本就是力扣的中等题，且很多都属于经典题目。面试的过程中，只要别太紧张基本不会有太大的问题。笔者在前期的面试过程中过于紧张，导致大脑空白，根本无法思考，连题意都没看明白。此时，可以深吸一口气，冷静一下，旁边有水可以喝一口水，努力让自己冷静下来。如果是在牛客网面试，可以打开代码全屏，可以遮掉面试官的视频，眼不见心不烦（狗头）。

如何提升算法，其实就是**刷题，刷题，还是塔喵的，刷题**。算法没有捷径可走，需要靠自己勤奋去做题。如何刷以及刷到什么程度，可以参考笔者的经验：

1. 分标签刷。题目的类型总体就分为那么几类，每一类题型都有特定的模板。因此在一段时间内，对同一类型的题目重复刷，然后总结思路，效率会更好。
2. 可以参考[大厂算法合集](https://codetop.cc/#/home)、[剑指offer](https://leetcode-cn.com/problemset/lcof/)，在面试前一段时间把这些题目刷一刷，很多时候可以做到原题或者类型题。
3. 多做总结。实习考察的算法大都是有套路，每次刷题都把关键的点记录下来，做总结，很快可以提炼出一个题型的整体思路。这里推荐[一个博主的算法框架](https://labuladong.gitee.io/algo/),帮助我们提炼算法思路。
4. 学习官解，精简代码。这里不是要实现那种一行代码解决问题的骚操作，而是去除掉我们代码中的那些无用的判断、循环等操作，增加代码可阅读性和提升效率。

关于算法，不同的企业对算法的要求是不同的，有些企业的算法都是简单题甚至没有算法题，例如美团的算法题是很简单的，腾讯的算法题也基本都是经典题目；而字节跳动对于算法考察较为严格，多为非常见算法题，且基本每轮技术面都有算法题。

其次，企业的不同的部门对算法的考察偏好也是不同的。例如我同学面试wxg一次性来了5道算法题，而笔者面试pcg则没有遇到这个情况。对算法看重的企业多会更深地去考察我们的算法功底，而对算法没有要求的更多是走个流程。

总之，学好算法对我们是一件终身受益的事情，而不管企业考察难度如何。

​

**简历**

关于简历，网络上有很多的博客写得很好，读者可以自行去搜查一下。这里主要和读者们分享笔者对于简历的一些心得。

* 简历的作用是**展示自己的亮点**给面试官，让面试官可以在**短时间内**迅速了解我们。

简历一定要把自己的亮点写上去，展示自己的能力。说人话就是：告诉面试官自己有多牛叉，这不是表现我们谦虚的时候。不管是项目亮点、还是个人技能，都通通写上去。

其次，一定要控制篇幅，不宜过长。面试官一般10s内阅读完一份简历，只需要把有亮点的内容展示出来，而一些类似“熟悉Java类与接口、了解使用xx框架”等基础知识就不要写了。

另外，有**开源项目或博客**也一定要写上去，特别加分。有时候即使面试回答得一般，但是看到我们的博客或者开源项目可能就认为只是紧张，随手就给我们一个过了。因此平时多写博客、项目，是有好处的。

* 技能点描述要**精准**，不要模糊。

举个例子：

* 1. 熟悉JVM。
  2. 熟悉JVM的GC标记算法及其分代回收理论、方法调用的解析及其入栈过程、类加载流程、内存模型与运行时数据区等。

对比这两种描述，第二种面试官可以很明显知道我们到底对JVM掌握到什么程度以及哪些知识点，并针对性提问，而第一种面试官甚至会觉得我们根本不懂JVM，连JVM中有什么知识点都不知道。

因此描述技能一定要精准，不要模糊，让面试官猜。面试官不会掀起我们的盖头，只会掀起我们的头盖[狗头]。

* 无用信息不要写。

纵使在校园内参加了非常多的大型社团活动，包括但不限于三下乡、主持大会等。我们面试的是android岗位，不是社团。只需要简单一句参加过校园各类社团活动，让面试官知道我们是一个active boy/gird 就好了。

* 写上简历的内容一定要是自己掌握的内容。

这是简历最基本的要求，不用多说了吧。

**非技术问题**

这部分的内容，大都会在三面或者HR面提问到，到这里其实面试已经差不多稳了，但也不可以乱讲话哦，不然HR姐姐卡我们简历就难受了。这些问题多为考察我们的软实力，重点是要知道面试官提问这个问题的目的是什么，了解他的“潜台词”。

顺带提一下HR面试，基本上到这一步，我们的面试就基本通过了。HR面试的主要目的是了解我们这个人本身的情况，如学习、周边环境、未来打算、价值观等等。只需要真诚和HR姐姐聊聊天，同时避开一些雷别踩就好了。嗯？这还能有雷？有的哇，例如HR姐姐问你拿了几家offer之后你会选择我们吗？你直接说：如果另一家过了就不要你们了。那人家还给你发offer？

但其他的个人情况问题，还是需要真诚和HR沟通，其实也是看企业文化是否适合自己。

下面看看笔者被提问到的一些软实力问题：

* 你项目中遇到的最大的困难是什么？

这个问题问的最多，多为考察面试者的解决问题的能力；问题本身并不是重点，解决的过程才是重点。

* 你在项目中学到了什么？

考问学习总结的能力，回答一些自己的收获，可以是技术的，也可以是非技术的例如团队合作的经验。

* 平时是如何学习的？

考察自学能力，可以把自己平时学习的方法讲出来，向面试官展示自己学习的能力。

* 你为什么选android？

考察对android的兴趣。不要讲得太浮夸哇，什么从小就xxx，真诚一点讲出来就好了。

* 未来的计划如何？考研吗？

根据具体情况回答就好了，如先实习锻炼业务能力，毕业后一两年希望有单独负责一个模块的能力等等；一般来说，企业更愿意招聘未来可以转正且有规划的实习生，更有培养空间。

* 工作城市有什么考虑吗？父母怎么态度等

这个很明显了，根据自己的实际情况来回答；当然，面试官更愿意招聘希望去他们部门所在城市的面试者。

* 你还在面试其他公司吗？拿到其他公司的offer了吗？你会选谁？

这里可以讲一两个比较优秀的公司来提高自己的身价，让HR感受到你的市场竞争力，但切不可全部十几个说出来哇，我们都讨厌海王，包括HR。

选谁这个问题，如果此时自己还没确定，可以把自己考虑的因素讲出来，如业务内容、地点、实习生培养制度等讲出来，然后告诉HR最后会权衡考虑，并在最后向HR了解这些内容（如果你都不问，怎么权衡？HR一看就知道你在养鱼）。如果已经确定好了要来这家公司，那就直接告诉他一定来。

千万记住不要撒谎，例如告诉HR我一定来，但心里已经有了想去的公司。到时候要拒offer会非常尴尬，而且有可能进入企业黑名单。[狗头]

当然面试官肯定更加倾向于想要选他们的面试者，不然发了offer又不来，那何必发呢？所以这里就看各位的情商发挥了。

* 你有什么优缺点？在别人看来，你有什么优缺点？

一定要提前做好准备，笔者当时就是没做准备，乱讲一通。

需要注意的就是别太浮夸，真实一点；什么叫真实一点？例如：我的缺点是总是控制不好自己的学习节奏，老是学习太晚，伤害身体。虽然存在这种情况，但别说HR，可能我们自己都不信不是吗[狗头]

其次是讲缺点的时候可以聊一下自己是怎么对待自己的缺点的，因为发现是一回事，而最终怎么处理又是另一回事了；把自己怎么对待自己的缺点的可以讲出来。

最后是注意题目，是有两个角度：自己和他人。一个是自我反省，另一个是外在表现，最好准备好这两道题的答案。

* 前面的面试体会如何？你觉得你前面的面试表现如何？

主要看在前面的面试中有没有什么体会，然后有没有去处理发现的问题。这里可以讲发现了很多的不会的知识点，然后自己都去再次学习了等等

* 自我介绍

几乎在每一场的面试都会有，但在三面总监面一般没有。

自我介绍是除了简历之外面试官第二个快速了解我们的地方。最好不要对着简历念，而是补充一些简历没有的内容。

例如可以讲自己从何时开始自学，自学能力非常强；加入过什么老师团队，做了什么项目；近段时间学习到什么程度等等。

在面试前可以做好一份自我介绍，然后后面所有的面试都用这份自我介绍就可以了。这样也不会说在自我介绍的时候不知道说啥。

但是千万不要太长哦，抓重点一两分钟就好了。

* 你还有什么问题吗？

这部分可以向面试官提问一些自己想了解的情况。一般来说，需要了解的有：部门业务内容、实习生具体的工作内容（sdk、业务开发、跨平台、转语言等）、实习生的培养内容、学习建议等等。

不同的面试官也需要区分问，例如HR可以问多一些工作地点、转正机会等问题，而技术面试官可以问一些工作内容的问题。

这些因人而异，最好不要说没问题咯，让面试官看起来对岗位没有多大的兴趣~

**面试中**

未知，所以恐惧。

在没开始面试前，可能很多的读者都是人生第一次入职面试，肯定非常紧张的，在面试前甚至想着，要不算了，不面了。这都是正常的。解决的办法就是：**去面一次**。经历过一次之后就不会紧张了。或者说，没那么紧张了。

大厂的面试官的体验一般都是非常好的，笔者到目前为止遇到的所有面试官，都给我非常好的面试体验。

在面试的时候要注意语速与咬字清晰问题。笔者本身是一个语速很快的人，加上面试紧张，面试官很难听清楚我到底在讲什么。面试一定要控制语速，可以提前给自己录音看看什么程度的语速是最合适的。

其次，要注意自己的表达清晰。仔细听好面试官的问题，然后先组织好思路，再慢慢表述出来。回答问题，也是相当于和一个讲解知识点的过程，思路一定要清晰。一些口头禅、冗余的表达尽量去除。

有一个需要注意的点就是：**不了解的东西，千万不要讲出来，切记！！！**。例如，我们不懂NIO，在面试官问到传统IO的时候，就不要拿NIO去比较或者提起NIO。如果提起了，面试官99%会问一些NIO的问题，这时候就会灰常尴尬。而如果自己掌握了NIO，那么就可以不断提起NIO，把面试官的询问重点往我们熟悉的知识点牵引。

最后需要注意一点的是：**拒offer是一个很考验情商的事情**。这不是在凡尔赛，而是一个非常重要的事情，处理不好可能会上企业黑名单。举个最简单的例子：

现在手上拿到了企业a的offer，但是自己更加向往的企业是b，但是b还在面试。这个时候企业a的HR问你：a和b你选择哪个？然后你坚定地说肯定选择a啊。然后拖了一两个星期，b通过了，然后跟a的HR讲我不去，我去b了。不仅影响了企业a的招聘，而且HR还会觉得你言而无信。

所以，拒offer最好是趁早，**不要影响人家的招聘**；其次是**语言前后要一致**，不要当渣男or渣女。[狗头]

**春招时间线**

实习生招聘一般集中在春季，秋季招聘主要是校招，也就是招正式工。

* 最先的一批开始时间是在年前，也就是12月。例如CVTE在11月、字节跳动在春节前就开启了实习招聘，属于最早的一批。
* 第二批在春节后，也就是2月份，很多的部门有内推直达。多关注牛客网等的一些内推信息，此时公司官网可能还没有开放投递通道。
* 三月份几乎是所以一二线大厂开始招聘的时间。大厂还会分两个批次：提前批和正式批。
  + 提前批：一般通过内推直达没有笔试，且提前批hc较多，是最推荐的投递简历时机。
  + 正式批：提前批挂了，正式批可以继续面。缺点就是一般都需要笔试
  + 但很多的公司没有提前批与正式批之分，这个要看具体的公司具体安排。强烈建议，越早投越好。
* 四月份是正式批开始时间，但热门岗位几乎已经招满人了。此时hc更少，要求也会比三月份更高。

**最后**

最后想要告诉读者的是：影响面试结果的因素有非常多，如面试官心情、口味，当天自己的状态，岗位要求等等。笔者的两个同学，都是前面连续挂了两次一面，但后面也都成功拿到了BAT offer，甚至不止一个。挂掉面试绝对不是简单的自己能力不行，而是多方面的因素，总结好经验，继续面试就好了。

迈出第一步，开启第一轮面试非常重要。不要觉得自己“好菜啊，投了肯定不过的，不投了”。不投，怎么知道不过呢？面试也是一个学习的过程，会慢慢发现自己的不足，包括表达、知识等等。所以，如果你还在犹豫，赶紧投起来吧。

希望这篇文章能够给正在找实习的读者提供一点帮助，先预祝各位实习面试顺利吧。有什么问题或者想了解的内容可以评论区交流，后续笔者继续补充或者新开一篇文章做解答。

**如果文章对你有帮助，还希望给笔者一个赞鼓励一下。**

全文到此，原创不易，觉得有帮助可以点赞收藏评论转发。 有任何想法欢迎评论区交流指正。 如需转载请评论区或私信告知。

另外欢迎光临笔者的个人博客：[传送门](https://qwerhuan.gitee.io)

文章分类

[Android](/android" \t "_blank)

文章标签

[面试](/tag/%E9%9D%A2%E8%AF%95" \t "_blank)

相关推荐

[干货满满张哈希](/user/501033033545053" \t "_blank)

5小时前

[Java](/tag/Java) [面试](/tag/%E9%9D%A2%E8%AF%95)

[Spring Cloud 升级之路 - 2020.0.x - 5. 理解 NamedContextFactory](/post/6966414251934089253)

[spring-cloud-commons 中参考了 spring-cloud-netflix 的设计，引入了 NamedContextFactory 机制，一般用于对于不同微服务的客户端模块使用不同的](/post/6966414251934089253)

* + 2
  + 评论
  + 分享

微博

微信扫一扫

[敖丙](/user/4406498333825357" \t "_blank)

1天前

[Java](/tag/Java) [面试](/tag/%E9%9D%A2%E8%AF%95)

[字节面试：什么是责任链模式？](/post/6966023225167839262)

[前言 面试经历大家肯定都有过，但是面试的流程其实跟一种设计模式很像，每一轮的面试官都有自己的职责，一个求职者面试经历的过程就好比一次客户端的请求过程。 在设计模式系列的文章中之前已经为大家分享了创建型](/post/6966023225167839262)

* + 35
  + 6
  + 分享

微博

微信扫一扫

[黄轶](/user/2137106333044861" \t "_blank)

4天前

[面试](/tag/%E9%9D%A2%E8%AF%95) [前端](/tag/%E5%89%8D%E7%AB%AF)

[聊聊前端面试](/post/6964658454543728647)

[最近 Zoom 国内又开放招聘了，我们组有了前端的 HC，所以我也参加了几场面试。在面试的过程中我也发现了很多问题，因此作为一个 9 年工作经验和 7 年面试经验的老兵，我想分享一些经验给你。](/post/6964658454543728647)

* + 390
  + 61
  + 分享

微博

微信扫一扫

[业志陈](/user/923245496518439" \t "_blank)

17小时前

[面试](/tag/%E9%9D%A2%E8%AF%95) [Android](/tag/Android)

[聊聊 Context 的一些知识点](/post/6966223613057826852)

[Context 在我们的日常开发中经常会使用到，在代码层次上指的是一个抽象类，在概念上指的是上下文环境，在功能上则起到了访问系统服务及系统资源的作用。Activity、Service 和 Applic](/post/6966223613057826852)

* + 2
  + 评论
  + 分享

微博

微信扫一扫

[白毛欧尼酱](/user/888061128343934" \t "_blank)

5小时前

[面试](/tag/%E9%9D%A2%E8%AF%95)

[web前端面试之——原型与原型链](/post/6966406883129163790)

[web前端面试之——原型与原型链 包括对原型与原型链的知识点的理解与记忆，扩展小知识，面试表达方式等内容](/post/6966406883129163790)

* + 1
  + 评论
  + 分享

微博

微信扫一扫

[程序新视界](/user/3368559359568696" \t "_blank)

2天前

[面试](/tag/%E9%9D%A2%E8%AF%95)

[筛选了几百份简历，明白了为什么大厂要设学历门槛了](/post/6965596533655863303)

[最近一直在为公司招聘前端工程师，前前后后筛选了几百份简历，面试了大几十人，这过程中也发生了很多奇特的事情。渐渐明白，为什么大厂招聘时会在学历上设置门槛。 这篇文章就给大家聊聊，从筛选简历到最终面试中发](/post/6965596533655863303)

* + 7
  + 21
  + 分享

微博

微信扫一扫

[神奇的程序员](/user/3984285870859614" \t "_blank)

26天前

[程序员](/tag/%E7%A8%8B%E5%BA%8F%E5%91%98) [面试](/tag/%E9%9D%A2%E8%AF%95)

[我离职了](/post/6956728664562073630)

[大家看到这篇文章的时候，我已经从老东家离职了，从有离职想法到找到新工作，期间经历了很多事情，也成长了许多。 本文就跟大家分享下我的这段经历，欢迎各位感兴趣的开发者阅读本文。](/post/6956728664562073630)

* + 946
  + 747
  + 分享

微博

微信扫一扫

[程序员十三](/user/3808363978174302" \t "_blank)

2天前

[面试](/tag/%E9%9D%A2%E8%AF%95) [Spring Boot](/tag/Spring%20Boot)

[【建议收藏】缺少 Vue3 和 Spring Boot 的实战项目经验？我这儿有啊！](/post/6965469101636452388)

[缺少 Vue3 和 Spring Boot 的实战项目经验？缺少学习项目和练手项目？我这儿有啊！ 从 2018年到 2021 年，空闲时间里陆陆续续做了一些开源项目，推荐给大家啊！记得点赞和收藏噢！](/post/6965469101636452388)

* + 17
  + 4
  + 分享

微博

微信扫一扫

[程序员十三](/user/3808363978174302" \t "_blank)

1月前

[Vue.js](/tag/Vue.js) [面试](/tag/%E9%9D%A2%E8%AF%95)

[Vue3发布半年我不学，摸鱼爽歪歪，哎~就是玩儿](/post/6955129410705948702)

[是从 Vue 2 开始学基础还是直接学 Vue 3 ？尤雨溪给出的答案是：“直接学 Vue 3 就行了，基础概念是一模一样的。“尤雨溪喊你学Vue3了！](/post/6955129410705948702)

* + 1298
  + 173
  + 分享

微博

微信扫一扫

[执鸢者](/user/3210229683336983" \t "_blank)

1天前

[面试](/tag/%E9%9D%A2%E8%AF%95)

[前端百题斩【009】——函数执行过程](/post/6965871302327336967)

[前端百题斩【8】中阐述了JavaScript代码的执行过程，但是并没有具体阐述一个函数的执行过程，对于函数执行过程如下所示：（注意：具体的内容本文将不会展开，只会全面讲述一下）](/post/6965871302327336967)

* + 5
  + 评论
  + 分享

微博

微信扫一扫

关于作者

[安卓学习者 @ 支付宝](/user/3931509313252552" \t "_blank)

获得点赞709

文章被阅读35,256

相关文章

[Android事件分发机制五：面试官你坐啊](/post/6922300686638153736" \o "Android事件分发机制五：面试官你坐啊" \t "_blank)

[47](/post/6922300686638153736" \o "Android事件分发机制五：面试官你坐啊" \t "_blank)

[18](/post/6922300686638153736" \o "Android事件分发机制五：面试官你坐啊" \t "_blank)

[Android事件分发机制一：事件是如何到达activity的？](/post/6918272111152726024" \o "Android事件分发机制一：事件是如何到达activity的？" \t "_blank)

[36](/post/6918272111152726024" \o "Android事件分发机制一：事件是如何到达activity的？" \t "_blank)

[8](/post/6918272111152726024" \o "Android事件分发机制一：事件是如何到达activity的？" \t "_blank)

[Android全面解析之Window机制](/post/6888688477714841608" \o "Android全面解析之Window机制" \t "_blank)

[41](/post/6888688477714841608" \o "Android全面解析之Window机制" \t "_blank)

[10](/post/6888688477714841608" \o "Android全面解析之Window机制" \t "_blank)

[Android全面解析之Handler机制（终篇）：常见问题汇总](/post/6887933281686421518" \o "Android全面解析之Handler机制（终篇）：常见问题汇总" \t "_blank)

[34](/post/6887933281686421518" \o "Android全面解析之Handler机制（终篇）：常见问题汇总" \t "_blank)

[7](/post/6887933281686421518" \o "Android全面解析之Handler机制（终篇）：常见问题汇总" \t "_blank)

[这一篇TCP总结请收下](/post/6932842985616834568" \o "这一篇TCP总结请收下" \t "_blank)

[42](/post/6932842985616834568" \o "这一篇TCP总结请收下" \t "_blank)

[16](/post/6932842985616834568" \o "这一篇TCP总结请收下" \t "_blank)

目录

分享

窗体顶端

提示

根据我国[《网络安全法》](http://www.cac.gov.cn/2016-11/07/c_1119867116.htm)，您需要绑定手机号后才可在掘金社区内发布内容。



获取验证码

绑定手机

窗体底端