前言

因为和同事有约定再加上LZ自己也喜欢做完一件事之后进行总结，因此有了这篇文章。这篇文章大部分内容都是面向整个程序员群体的，当然因为LZ本身是做Java开发的，因此有一部分内容也是专门面向咱们Java程序员的。

简单先说一下，LZ坐标杭州，13届本科毕业，算上年前在阿里巴巴B2B事业部的面试，一共有面试了有6家公司（因为LZ不想请假，因此只是每个晚上去其他公司面试，所以面试的公司比较少），其中成功的有4家，另外两家失败的原因在于：

1、阿里巴巴B2B事业部的面试，两轮技术面试都过了，最后一轮面试是对方的主管，由于听说技术面试过了基本上90%都面试成功了，所以LZ在和主管的交谈中也是毫无顾忌，说得天花乱坠，很多自己介于知道和不知道的东西都直接脱口而出了，结果多次被对方一反问就问得哑口无言。事后想来，模棱两可的答案是面试中最忌讳的，这次的失败也让LZ认真地对待后面的每一次面试

2、另外一家失败的是一家小公司，也就20来个人吧，整个团队是支付宝出来创业的，非常厉害。面试完LZ多方了解了一下，对方认为我基本功什么的都不错，但是实际项目经验还是欠缺一些，因为对方是创业型公司，需要人上手就能干活，因此我在这个时候还不是特别适合他们团队

至于其他成功的四家公司，给LZ的面试评价都挺高的貌似，但LZ也不想记流水账，因此就不一一列举每家公司的面试过程了，下面LZ主要谈谈作为一名工作三年左右的Java程序员应该具备的一些技能以及个人的一些其他感悟。

关于程序员的几个阶段

每个程序员、或者说每个工作者都应该有自己的职业规划，如果看到这里的朋友没有自己的职业规划，希望你可以思考一下自己的将来。

LZ常常思考自己的未来，也从自己的思考中总结出了一些东西，作为第一部分来谈谈。LZ认为一名程序员应该有几个阶段（以下时间都算上实习期）：

- 第一阶段----三年
  - 我认为三年对于程序员来说是第一个门槛，这个阶段将会淘汰掉一批不适合写代码的人。这一阶段，我们走出校园，迈入社会，成为一名程序员，正式从书本上的内容迈向真正的企业级开发。我们知道如何团队协作、如何使用项目管理工具、项目版本如何控制、我们写的代码如何测试如何在线上运行等等，积累了一定的开发经验，也对代码有了一定深入的认识，是一个比较纯粹的Coder的阶段
- 第二阶段----五年第三阶段----十年
  - 五年又是区分程序员的第二个门槛。有些人在三年里，除了完成工作，在空余时间基本不会研究别的东西，这些人永远就是个Coder，年纪大一些势必被更年轻的人给顶替；有些人在三年里，除了写代码之外，还热衷于研究各种技术实现细节、看了N多好书、写一些博客、在Github上分享技术，这些人在五年后必然具备在技术上独当一面的能力并且清楚自己未来的发展方向，从一个Coder逐步走向系统分析师或是架构师，成为项目组中不可或缺的人物
-

- 十年又是另一个门槛了，转行或是继续做一名程序员就在这个节点上。如果在前几年就抱定不转行的思路并且为之努力的话，那么在十年的这个节点上，有些人必然成长为一名对行业有着深入认识、对技术有着深入认识、能从零开始对一个产品进行分析的程序员，这样的人在公司基本担任的都是CTO、技术专家、首席架构师等最关键的职位，这对于自己绝对是一件荣耀的事，当然老板在经济上也绝不会亏待你

第一部分总结一下，我认为，随着你工作年限的增长、对生活对生命认识的深入，应当不断思考三个问题：

1. 我到底适不适合当一名程序员？
2. 我到底应不应该一辈子以程序员为职业？
3. 我对编程到底持有的是一种什么样的态度，是够用就好呢还是不断研究？

最终，明确自己的职业规划，对自己的规划负责并为之努力。

关于项目经验

LZ在网上经常看到一些别的朋友有提出项目经验的问题，依照LZ面试的感觉来说，面试主要看几点：项目经验+基本技术+个人潜力（也就是值不值得培养）。

关于项目经验，我认为并发编程网的创始人方腾飞老师讲的一段话非常好：

> 介绍产品时面试官会考察应聘者的沟通能力和思考能力，我们大部分情况都是做产品的一个功能或一个模块，但是即使是这样，自己有没有把整个系统架构或产品搞清楚，并能介绍清楚，为什么做这个系统？这个系统的价值是什么？这个系统有哪些功能？优缺点有哪些？如果让你重新设计这个系统你会如何设计？

我觉得这就已经足以概括了。也许你仅仅工作一年，也许你做的是项目中微不足道的模块，当然这些一定是你的劣势且无法改变，但是如何弥补这个劣势，从方老师的话中我总结几点：

1. 明确你的项目到底是做什么的，有哪些功能
2. 明确你的项目的整体架构，在面试的时候能够清楚地画给面试官看并且清楚地指出从哪里调用到哪里、使用什么方式调用
3. 明确你的模块在整个项目中所处的位置及作用
4. 明确你的模块用到了哪些技术，更好一些的可以再了解一下整个项目用到了哪些技术

在你无法改变自己的工作年限、自己的不那么有说服力的项目经验的情况下（这一定是扣分项），可以通过这种方式来一定程度上地弥补并且增进面试官对你的好感度。

补充一点，在面试中聊你的项目的时候，有一个问题90%是绕不过的：谈一下你在项目中解决过的比较复杂的问题。这需要在工作中不断去发现和探索，不需要多，在你自己目前的项目中只要你找到一两个能说的问题就行。一个小技巧是，即使问题不是你解决的而是别人解决的，但是你把这个问题弄懂、搞透了，在面试的时候你一样可以把这个问题当作是你自己解决的来说----毕竟，谁来管这个问题当时到底是不是你解决的呢？

关于专业技能

写完项目接着写写一名3年工作经验的Java程序员应该具备的技能，这可能是Java程序员们比较关心的内容。我这里要说明一下，以下列举的内容不是都要会的东西----但是如果你掌握得越多，最终能得到的评价、拿到的薪水势必也越高。

1、基本语法

这包括static、final、transient等关键字的作用，foreach循环的原理等等。今天面试我问你static关键字有哪些作用，如果你答出static修饰变量、修饰方法我会认为你合格，答出静态块，我会认为你不错，答出静态内部类我会认为你很好，答出静态导包我会对你很满意，因为能看出你非常热衷研究技术。

最深入的一次，LZ记得面试官直接问到了我volatile关键字的底层实现原理（顺便插一句，面试和被面试本身就是相对的，面试官能问这个问题同时也让面试者感觉到面试官也是一个喜爱研究技术的人，增加了面试者对公司的好感，LZ最终选择的就是问了这个问题的公司），不要觉得这太吹毛求疵了----越简单的问题越能看出一个人的水平，别人对你技术的考量绝大多数都是以深度优先、广度次之为标准的，切记。

2、集合

非常重要，也是必问的内容。基本上就是List、Map、Set，问的是各种实现类的底层实现原理，实现类的优缺点。

集合要掌握的是ArrayList、LinkedList、Hashtable、HashMap、ConcurrentHashMap、HashSet的实现原理，能流利作答，当然能掌握CopyOnWrite容器和Queue是再好不过的了。另外多说一句，ConcurrentHashMap的问题在面试中问得特别多，大概是因为这个类可以衍生出非常多的问题，关于ConcurrentHashMap，我给网友朋友们提供三点回答或者是研究方向：

（1）ConcurrentHashMap的锁分段技术

（2）ConcurrentHashMap的读是否要加锁，为什么

（3）ConcurrentHashMap的迭代器是强一致性的迭代器还是弱一致性的迭代器

3、设计模式

本来以为蛮重要的一块内容，结果只在阿里巴巴B2B事业部面试的时候被问了一次，当时问的是装饰器模式。

当然咱们不能这么功利，为了面试而学习，设计模式在工作中还是非常重要、非常有用的，23种设计模式中重点研究常用的十来种就可以了，面试中关于设计模式的问答主要是三个方向：

（1）你的项目中用到了哪些设计模式，如何使用

（2）知道常用设计模式的优缺点

（3）能画出常用设计模式的UML图

4、多线程

这也是必问的一块了。因为三年工作经验，所以基本上不会再问你怎么实现多线程了，会问得深入一些比如说Thread和Runnable的区别和联系、多次start一个线程会怎么样、线程有哪些状态。当然这只是最基本的，出乎意料地，几次面试几乎都被同时问到了一个问题，问法不尽相同，总结起来是这么一个意思：

> 假如有Thread1、Thread2、Thread3、Thread4四条线程分别统计C、D、E、F四个盘的大小，所有线程都统计完毕交给Thread5线程去做汇总，应当如何实现？

聪明的网友们对这个问题是否有答案呢？不难，java.util.concurrent下就有现成的类可以使用。

另外，线程池也是比较常问的一块，常用的线程池有几种？这几种线程池之间有什么区别和联系？线程池的实现原理是怎么样的？实际一些的，会给你一些具体的场景，让你回答这种场景该使用什么样的线程池比较合适。

最后，虽然这次面试问得不多，但是多线程同步、锁这块也是重点。synchronized和ReentrantLock的区别、synchronized锁普通方法和锁静态方法、死锁的原理及排查方法等等，关于多线程，我在之前有些过文章总结过多线程的40个问题，可以参看40个Java多线程问题总结

过多线程的40个问题，可以参看40个Java多线程问题总结。

5、IO

再次补充IO的内容，之前忘了写了。

IO分为File IO和Socket IO，File IO基本上是不会问的，问也问不出什么来，平时会用就好了，另外记得File IO都是阻塞IO。

Socket IO是比较重要的一块，要搞懂的是阻塞/非阻塞的区别、同步/异步的区别，借此理解阻塞IO、非阻塞IO、多路复用IO、异步IO这四种IO模型，Socket IO如何和这四种模型相关联。这是基本一些的，深入一些的话，就会问NIO的原理、NIO属于哪种IO模型、NIO的三大组成等等，这有些难，当时我也是研究了很久才搞懂NIO。提一句，NIO并不是严格意义上的非阻塞IO而应该属于多路复用IO，面试回答的时候要注意这个细节，讲到NIO会阻塞在Selector的select方法上会增加面试官对你的好感。

如果用过Netty，可能会问一些Netty的东西，毕竟这个框架基本属于当前最好的NIO框架了（Mina其实也不错，不过总体来说还是比不上Netty的），大多数互联网公司也都在用Netty。

6、JDK源码

要想拿高工资，JDK源码不可不读。上面的内容可能还和具体场景联系起来，JDK源码就是实打实地看你平时是不是爱钻研了。LZ面试过程中被问了不少JDK源码的问题，其中最刁钻的一个问了LZ，String的hashCode()方法是怎么实现的，幸好LZ平时String源代码看得多，答了个大概。JDK源码其实没什么好总结的，纯粹看个人，总结一下比较重要的源码：

（1）List、Map、Set实现类的源代码

（2）ReentrantLock、AQS的源代码

（3）AtomicInteger的实现原理，主要能说清楚CAS机制并且AtomicInteger是如何利用CAS机制实现的

（4）线程池的实现原理

（5）Object类中的方法以及每个方法的作用

这些其实要求蛮高的，LZ去年一整年基本把JDK中重要类的源代码研究了个遍，真的花费时间、花费精力，当然回头看，是值得的----不仅仅是为了应付面试。

7、框架

老生常谈，面试必问的东西。一般来说会问你一下你们项目中使用的框架，然后给你一些场景问你用框架怎么做，比如我想要在Spring初始化bean的时候做一些事情该怎么做、想要在bean销毁的时候做一些事情该怎么做、MyBatis中$和#的区别等等，这些都比较实际了，平时积累得好、有多学习框架的使用细节自然都不成问题。

如果上面你的问题答得好，面试官往往会深入地问一些框架的实现原理。问得最多的就是Spring AOP的实现原理，当然这个很简单啦，两句话就搞定的的事儿，即使你不会准备一下就好了。LZ遇到的最变态的是让LZ画一下Spring的Bean工厂实现的UML图，当然面对这样一个有深度的问题，LZ是绝对答不出来的/(ㄒoㄒ)/~~

8、数据库

数据库十有八九也都会问到。一些基本的像union和union all的区别、left join、几种索引及其区别就不谈了，比较重要的就是数据库性能的优化，如果对于数据库的性能优化一窍不通，那么有时间，还是建议你在面试前花一两天专门把SQL基础和SQL优化的内容准备一下。

不过数据库倒是不用担心，一家公司往往有很多部门，如果你对数据库不熟悉而基本技术又非常好，九成都是会要你的，估计会先把你放到对数据库使用不是要求非常高的部门锻炼一下。

9、数据结构和算法分析

数据结构和算法分析，对于一名程序员来说，会比不会好而且在工作中绝对能派上用场。数组、链表是基础，栈和队列深入一些但也不难，树挺重要的，比较重要的树AVL树、红黑树，可以不了解它们的具体实现，但是要知道什么是二叉查找树、什么是平衡树，AVL树和红黑树的区别。记得某次面试，某个面试官和我聊到了数据库的索引，他问我：

> 你知道索引使用的是哪种数据结构实现吗？

LZ答到用的Hash表吧，答错。他又问，你知道为什么要使用树吗？LZ答到因为Hash表可能会出现比较多的冲突，在千万甚至是上亿级别的数据面前，会大大增加查找的时间复杂度。而树比较稳定，基本保证最多二三十次就能找到想要的数据，对方说不完全对，最后我们还是交流了一下这个问题，我也明白了为什么要使用树，这里不说，网友朋友们觉得索引为什么要使用树来实现呢？

至于算法分析，不会、不想研究就算了，记得某次面试对方问我，Collections.sort方法使用的是哪种排序方法，额，吐血三升。当然为了显示LZ的博学，对算法分析也有一定的研究(⊙_⊙)b，LZ还是硬着头皮说了一句可能是冒泡排序吧。当然答案肯定不是，有兴趣的网友朋友们可以去看一下Collections.sort方法的源代码，用的是一种叫做TimSort的排序法，也就是增强型的归并排序法。

10、Java虚拟机

出乎LZ的意料，Java虚拟机应该是很重要的一块内容，结果在这几家公司中被问到的概率几乎为0。要知道，LZ去年可是花了大量的时间去研究Java虚拟机的，光周志明老师的《深入理解Java虚拟机：JVM高级特性与最佳实践》，LZ就读了不下五遍。

言归正传，虽然Java虚拟机没问到，但我觉得还是有必要研究的，LZ就简单地列一个提纲吧，谈谈Java虚拟机中比较重要的内容：

1. Java虚拟机的内存布局
2. GC算法及几种垃圾收集器
3. 类加载机制，也就是双亲委派模型
4. Java内存模型
5. happens-before规则
6. volatile关键字使用规则

也许面试无用，但在走向大牛的路上，不可不会。

11、Web方面的一些问题

Java主要面向Web端，因此Web的一些问题也是必问的。LZ碰到过问得最多的两个问题是：

> 谈谈分布式Session的几种实现方式

常用的四种能答出来自然是让面试官非常满意的，另外一个常问的问题是：

> 讲一下Session和Cookie的区别和联系以及Session的实现原理

这两个问题之外，web.xml里面的内容是重点，Filter、Servlet、Listener，不说对它们的实现原理一清二楚吧，至少能对它们的使用知根知底。另外，一些细节的方面比如get/post的区别、forward/重定向的区别、HTTPS的实现原理也都可能会被考察到。

噢，想起来了，一致性Hash算法貌似也被问到了几次，这个LZ以前专门深入研究过并且写了两篇博文，因此问到这个问题LZ自然是答得高不费力，文章是MemCache详细解读和对一致性Hash算法、Java代码实现的深入研

到这个问题LZ自然是含待笔不贾力。文章是MemCache详细解读和对一致性Hash算法，Java代码实现的探究研究，特别说明，LZ真的不是在为自己以前写的文章打广告啊啊啊啊啊啊。

最后，如果有兴趣有时间，建议学习、研究一下SOA和RPC，面向服务体系，大型分布式架构必备，救命良方、包治百病、屡试不爽。

关于HR面试

如果你过五关斩六将，成功地通过了所有的技术面，那么恭喜你，你离升职加薪、出任CEO、迎娶白富美、走向人生巅峰又进了一步。但是还没有到谈薪资待遇的时候，最后还有一个考验：HR面试。基本所有的大公司都有这一轮的面试，不要小看HR面试，很多公司的HR对于面试者都有一票否决权的----即使前面的面试对你的评价再高。

所以，这轮的面试也必须重视起来，HR面试主要问的是几点：

1. 简历中写的过去工作经历的离职原因
2. 当前公司薪资待遇
3. 期望能到怎样的一家公司
4. 个人未来的发展方向

我专门提一下第2点。可能有人比较排斥也不想说这个，我个人倒是持开放状态，问了就说了，当然一些的夸大还是必要的，当前公司薪资待遇多报个一千块钱完全没问题（毕竟是一家互联网公司总多多少少有些补贴啊什么的嘛）。因为这和你在新公司能拿到的薪水关系不大，新公司能拿到的薪水的决定因素是整个公司的薪资情况以及根据你的面试情况在公司的定位，都是有固定的薪资范围的。HR问这个主要也就是心里有个数并且看你是否诚信----有些公司入职时会要求你提供最近一家单位的银行流水号。

HR面试就说到这里了，总结起来其实就是四个字：滴水不漏。整个面试过程态度积极向上，不要有任何悲观消极的态度（尤其在谈到以前公司情况的时候，即使有再多的不满），就不会有问题。

关于面试心态

这个嘛，LZ其实在公司也面试过几个人，一半以上的面试者回答问题的时候都属于那种双腿发抖、声音颤抖的类型。在LZ看来这大可不必并且这还是扣分项，回答问题的时候最最基本的两个要求：

1. 不紧不慢，平心静气
2. 条理清晰

表达能力绝对是面试的时候重要的考察项目。咱们做的是程序员这一行，讲究的是团队协作，不是写作、画画，一支笔、一个人就行了，一个表达能力不行的程序员，要来又有什么用呢？

除此之外，就是保持良好的心态。古语说得好，只要功夫深，铁杵磨成针，面试的成功与否，在于平时的积累，临时抱抱佛脚，看两道面试题是没有用的，只要平时足够努力，成功是水到渠成的事情，平时不怎么研究技术的，那也就是个听天由命的事情，只要充分地展示平时自己的所学就可以了。

因此在我看来，不要把面试当作面试，当做一次技术交流，把面试的心态从我要找到一份工作转变为我要通过面试去发现不足、提升自己，这样就会平和多了，即使失败也不会有太多失望的感觉。

另外，如果平时自己热衷于研究技术的朋友，真的要有自信，不要觉得别人面试你别人就比你厉害。面试官未