引言

其实本来真的没打算写这篇文章，主要是LZ得记忆力不是很好，不像一些记忆力强的人，面试完以后，几乎能把自己和面试官的对话都给记下来。LZ自己当初面试完以后，除了记住一些聊过的知识点以外，具体的内容基本上忘得一干二净，所以写这篇文章其实是很有难度的。

但是，最近问LZ的人实在是太多了，为了避免重复回答，给自己省点力气，干脆就在这里统一回复了。

其实之前LZ写过一篇文章，但是那篇文章更多的是在讨论“面试前该不该刷题”这个话题，而这篇文章将会更加聚焦在面试前如何准备，以及工作当中如何学习这个话题上，而且会尽量写出一些干货。

第一个问题：阿里面试都问什么？

这个是让LZ最头疼的一个问题，也是群里的猿友们问的最多的一个问题。

说实话，LZ只能隐约想起并发、JVM、分布式、TCP/IP协议这些个关键字，具体的问题真的是几乎都没记住。而且就算LZ记住了，也告诉你了，你也背会了，但LZ觉得，在面试中，你被问到一模一样问题的可能性依然很小。

甚至，就算你运气好被问到了，你也照着背下来了，也不一定就能对你的面试起到正面的作用，因为面试官万一多问一句，你可能就露馅了，那还不如干脆点说不会更好。

LZ参加的是阿里的社招面试，而社招不同于校招，问题的范围其实是很随机的。因为能参加一些比较知名的互联网公司社招的人，70%以上都会有个3-5年的经验。这倒不是说一两年经验的同学没有机会进这些公司，而是因为这种公司，大部分情况下只招一些比较资深的开发和应届生，而不招那些处于中间阶段的人。而1-2年经验的同学，往往就刚好处于这个尴尬的阶段。

对于能有3-5年经验的这部分人中，每个人的经历又都不同，所擅长的点也不一样，因此这就会导致每个人的问题和范围都不太一样。

很少说有哪个知名的互联网公司，比如BAT、京东、360、搜狐、网易等这些公司，其社招面试还有固定的问题和模式，让你可以像应届生面试一样，在面试前靠临时抱佛脚度过这一关。

大部分公司在社招的时候，不光是阿里，其它公司也都一样（因为LZ在一年多前也参加过很多其它知名互联网公司的面试，详情见《记录2015年年初跳槽的经历！》），基本上都分为两个阶段的提问。

第一个阶段是主语言本身以及它的高级特性，第二个阶段是讲述自己的项目，并在中间穿插着问题。

所以，LZ不妨就这两个阶段，谈谈社招面试的准备，而不是去把阿里面试的过程背一遍。说实话，LZ也确实记不住，所以不要再问LZ阿里面试都会问哪些问题了，你看看上面那个连接里的文章，也会发现，LZ里面也基本上没有写具体的问题，原因是一样的，真的记不住啊。（就是因为记忆力的问题，导致LZ从小偏科，文科成绩一直堪忧，-_-）

社招面试如何准备

LZ会分为四个部分来谈论这个问题，由于LZ本身是Java出身，因此关于主语言的问题，都是与Java相关，其它语言的同学可以选择性忽略。此外，面试的时候一般面试官的问题都是环环相扣，逐渐深入的，这点在下面大家可以更明显的感受出来。

1、主语言本身以及它的高级特性。

主语言当然就是你平日里拿来赚钱的家伙。不要告诉LZ你没有主语言，你会N多种语言，或者是你精通N多种语言，你要非这么说的话，你可以来杭州试试，LZ保证不打死你，最多打残。

LZ的主语言很显然是Java，那么对于Java来说，它的语言本身以及它的高级特性，都有哪些比较容易在面试中问到呢？

一般情况下，主要有以下知识点很容易被问到。（PS：以下所列举的，都是一些Java相对而言比较高级一点的知识点，因为这里谈的是社招，而不是校招）

1）Java的数据结构相关的类实现原理，比如LinkedList，ArrayList，HashMap，TreeMap这一类的。以下简单模拟一个数据结构的连环炮。

比如，面试官先问你HashMap是不是有序的？

你肯定回答说，不是有序的。那面试官就会继续问你，有没有有顺序的Map实现类？

你如果这个时候说不知道的话，那这个问题就到此结束了。如果你说有TreeMap和LinkedHashMap。

那么面试官接下来就可能会问你，TreeMap和LinkedHashMap是如何保证它的顺序的？

如果你回答不上来，那么到此为止。如果你依然回答上来了，那么面试官还会继续问你，你觉得它们两个哪个的有序实现比较好？

如果你依然可以回答的话，那么面试官会继续问你，你觉得还有没有比它更好或者更高效的实现方式？

如果你还能说出来的话，那么就你所说的实现方式肯定依然可以问你很多问题。

以上就是一个面试官一步一步提问的例子。所以，如果你了解的不多，千万不要敷衍，因为可能下一个问题你就暴露了，还不如直接说不会，把这个问题结束掉，赶紧切换到你熟悉的领域。

2）Java并发包当中的类，它们都有哪些作用，以及它们的实现原理，这些类就是java.concurrent包下面的。与上面一样，咱们也简单的模拟一个并发包的连环炮。

比如面试官可能会先问你，如果想实现所有的线程一起等待某个事件的发生，当某个事件发生时，所有线程一起开始往下执行的话，有什么好的办法吗？
这个时候你可能会说可以用栅栏（Java的并发包中的CyclicBarrier），那么面试官就会继续问你，你知道它的实现原理吗？
如果你继续回答的话，面试官可能会继续问你，你还知道其它的实现方式吗？
如果你还能说出很多种实现方式的话，那么继续问你，你觉得这些方式里哪个方式更好？
如果你说出来某一个方式比较好的话，面试官依然可以继续问你，那如果让你来写的话，你觉得还有比它更好的实现方式吗？
如果你这个时候依然可以说出来你自己更好的实现方式，那么面试官肯定还会揪着这个继续问你。
为什么说面试的时候要引导面试官，原因就在这了。因为面试官的提问很多时候都是有迹可循的，你如果抓住了他的轨迹，能够猜到他下面很可能会问什么，那你在回答的时候就可以往你想要谈的方向去说。这样面试时就会显得更加从容，更加的游刃有余。
3）IO包和NIO包中的内容。这部分里面NIO会是重点，IO包大部分都会比较熟悉，因此可能会直接略过，直接问你NIO的内容。
IO包和NIO包的内容相对来说不是很多，首先NIO模型要熟悉，特别是其中的selector一定要非常清楚它的职责和实现原理。其实NIO的核心是IO线程池，一定要记住这个关键点。有的时候，面试官可能也会问你IO包的设计模式（装饰器模式），为什么要这样设计？
有的面试官还会问你有没有更好的设计，这个时候如果你不知道请果断说自己现在的水平有限，想不出来更好的设计，千万不要信口开河，随意YY。
4）Java的虚拟机的内容。这部分主要包括三部分，GC、类加载机制，以及内存。
一个GC部分简单的连环炮。
面试官可以先问你什么时候一个对象会被GC？
接着继续问你为什么要在这种时候对象才会被GC？
接着继续问你GC策略都有哪些分类？
你如果说出来了，继续问你这些策略分别都有什么优劣势？都适用于什么场景？
你继续说出来了以后，给你举个实际的场景，让你选择一个GC策略？
你如果选出来了，继续问你，为什么要选择这个策略？
下面是关于类加载机制的简单连环炮。
首先肯定是先问你Java的类加载器都有哪些？
回答了这些以后，可能会问你每个类加载器都加载哪些类？
说完以后，可能会问你这些类加载之间的父子关系是怎样的？
你在回答的时候可能会提到双亲委派模型，那么可以继续问你什么是双亲委派模型？
你解释完了以后，可能会继续问你，为什么Java的类加载器要使用双亲委派模型？

你回答完以后，可能会继续问你如何自定义自己的类加载器，自己的类加载器和Java自带的类加载器关系如何处理？

再来一个关于内存的连环炮。

首先肯定就是问你内存分为哪几部分，这些部分分别都存储哪些数据？

然后继续问你一个对象从创建到销毁都是怎么在这些部分里存活和转移的？

接着可能会问你，内存的哪些部分会参与GC的回收？

完事以后，可能还会问你Java的内存模型是怎么设计的？

你回答了以后，还会继续问你为什么要这么设计？

问完以后，还可能会让你结合内存模型的设计谈谈volatile关键字的作用？

你在谈的时候，肯定会提到可见性，那么接着可见性这三个字，还可以继续问你并发的内容。

基本上Java语言本身以及语言稍微高级点的内容就是以上部分，如果你能把以上四部分了解的非常透彻，那基本上Java这部分就没啥问题了，因为光以上的内容就够你跟面试官聊很久了。你聊这些聊得久了，自然问你其它问题的时间就会短点。

你从LZ写这些问题的过程也应该能感受出来，很多时候，面试官都是顺着一条线一路问下去的，如果你觉得这条线你不熟悉的话，就要及时拐弯，引导面试官去问其它方面的问题。千万不要一直往下深入，直到自己跳不出来为止，那就尴了个尬了。

2、讲述自己的项目，并在中间穿插着问题

这一部分是面试过程中必问，也是聊得最久的一个阶段。除非你前面的语言部分非常扎实，扎实到面试官问了一两个小时，依旧没有探出你对语言本身的了解到底有多深。否则的话，你一定逃不过自己的项目这一关，而且一般情况下聊得时间不会太短。

这一部分内容，一般的模式就是你自己去讲你做过的项目，然后面试官会冷不丁的让你去解释其中某一部分，比如让你解释当时为什么要这么做，或者问你现在觉得有没有更好的办法。而这些穿插的问题，大部分与你的项目所用到的技术有关。而你需要做的，就是充分、再充分的去总结自己做过的项目（尤其是最近的一两个项目），挖掘出一个甚至N个亮点，以备于到时候可以让面试官产生眼前一亮的感觉。如果你能达到这种效果的话，基本上离你成功就不远了。

这部分内容由于和每个人自己的经历息息相关，因此这里也没法列举可能问到的问题。这篇文章《程序员面经：面试前到底该不该刷题以及面试前该如何准备》是LZ之前写的，里面大概讨论了下如何在面试前总结，有兴趣的可以去了解一下。

3、额外的加分项

上面两个阶段基本上是必问的，还有一些加分项。这些加分项中，有些内容面试官也会问你（比如TCP/IP协议、算法），但更多的是会先问你了解不了解，你了解的话再继续聊，不了解的话就直接略过了，不至于因为这种问题而直接把你打入地狱。

下面LZ列举一下这些加分项，如果可以的话，这些加分项还是要争取一下的。

1、计算机系统原理。
2、网络通信协议（TCP/IP，HTTP等）。
3、数据结构与算法。
4、著名开源项目的源码。
5、你自己有很棒的开源项目。
6、你的个人博客。
7、待评论区补充。

这几项当中，对于前1-3项，如果你之前就比较了解，只是由于时间问题忘记了的话，还是可以临时抱佛脚一下的。至于后面4-6项，就需要你日常的积累了，不是一时半会儿能做到的。如果你平日里没有积累，那么后面这三个加分项只能抛弃了。

4、与你职位相关的内容

其实这最后一项是对前面三项的补充，你应该尽量去主攻和你面试的职位相关的内容。比如你面试一个实时计算的职位，那么你的算法最好要厉害，对于著名的实时计算开源项目要熟悉，最好阅读过源码，而且还要对分布式系统有一定的见解。

因此，这个第4部分没有具体的内容，只是提醒你，如果你很明确自己的面试职位，最好在面试前准备的时候，尽量朝职位的需求方向靠拢，这样成功的可能性更大。

对于Java程序猿学习的建议

这一部分其实也算是今天的重点，这一部分用来回答很多群里的朋友所问过的问题，那就是LZ你是如何学习Java的，能不能给点建议？

今天LZ是打算来点干货，因此咱们就不说一些学习方法和技巧了，直接来谈每个阶段要学习的内容甚至是一些书籍。这一部分的内容，同样适用于一些希望转行到Java的同学。

在大家看之前，LZ要先声明两点。

1、由于LZ本人是Java后端开发出身，因此所推荐的学习内容是Java Web和Java后端开发的路线，非Java Web和Java后端开发的同学请适当参考其学习思想即可，切勿照搬。

2、下面对于【第一部分】的推荐内容，目的是让你尽快成为一个可以参加工作的Java开发者，更适用于处于待业状态，准备转行Java的同学。如果你是在校学生，务必要在学好基础（比如计算机系统、算法、编译原理等等）的前提下，再考虑去进行下面的学习。

第一部分：对于尚未做过Java工作的同学，包括一些在校生以及刚准备转行Java的同学。

一、Java基础

首先去找一个Java的基础教程学一下，这里可以推荐一个地址，或者你也可以参照这个地址上去找相应的视频，地址为http://www.runoob.com/java/java-tutorial.html。

学习Java基础的时候，应该尽量多动手，很多时候，你想当然的事情，等你写出来运行一下，你就会发现不是这么回事儿，不信你就试试。

学完以上内容以后，你应该对Java有一个基本的了解了，你可以用Java语言写出一些简单的程序，并且你用的是最简单的编辑器，比如记事本。

这个时候，不要急于进入下一部分，留下几天好好写一些程序，尽可能熟悉这些基础内容。

二、Web开发

等你写上几天程序以后，你往往会比较迷茫，因为你写的东西似乎看起来毫无用处，比如实现一个简单的计算器，读取一个文件等。这个时候你就应该去学着写一些让你觉得有意思的东西了，所以你应该学习更多的知识。

这些内容主要是Web开发相关的内容，包括HTML/CSS/JS（前端页面）、Servlet/JSP（J2EE）以及Mysql（数据库）相关的知识。

它们的学习顺序应该是从前到后，因此最先学习的应该是HTML/CSS/JS（前端页面），这部分内容你可以去上面的那个runoob网站上找。你可以试着自己写一些页面，当然，你可以尽你最大的努力让它变得最漂亮。这部分内容对于后端Java来说，理论上不是特别重要，但至少要达到可以自己写出一些简单页面的水平。

接下来，你需要学习的是Servlet/JSP（J2EE）部分，这部分是Java后端开发必须非常精通的部分，因此这部分是这三部分中最需要花精力的，而且这个时候，你要学会使用开发工具，而不能再使用记事本了，可以选择eclipse。

当你下载安装好eclipse以后，请视频中的教程一步一步去学习，一定要多动手。关于Servlet/Jsp部分视频的选择，业界比较认可马士兵的视频，因此推荐给大家。当然了，LZ本人并没有看过他的视频，所以不好说的太绝对，如果大家自己有更好的选择，可以坚持自己的，不要被LZ干扰。

原本LZ也是打算出教学视频的，但是由于时间问题，还是决定放弃了。但是如果你看视频的过程中遇到了问题，欢迎来LZ的交流群提问，或者去斗鱼观看LZ的直播提出你的问题，直播地址和群号都在LZ的个人博客左侧。

最后一步，你需要学会使用数据库，mysql是个不错的入门选择，而且Java领域里主流的关系型数据库就是mysql。这部分一般在你学习Servlet/Jsp的时候，就会接触到的，其中的JDBC部分就是数据库相关的部分。你不仅要学会使用JDBC操作数据库，还要学会使用数据库客户端工具，比如navicat，sqlyog，二选一即可。

三、开发框架

当你学会以上内容以后，这个时候你还不足以参加工作，你还需要继续深造。公司里为了提高开发的效率，会使用一些Java Web框架，因此你还需要学习一些开发框架。

目前比较主流的是SSM框架，即spring、springmvc、mybatis。你需要学会这三个框架的搭建，并用它们做出一个简单的增删改查的Web项目。你可以不理解那些配置都是什么含义，以及为什么要这么做，这些留着后面你去了解。但你一定要可以快速的利用它们三个搭建出一个Web框架，你可以记录下你第一次搭建的过程，相信我，你一定会用到的。

还要提一句的是，你在搭建SSM的过程中，可能会经常接触到一个叫maven的工具。这个工具也是你以后工作当中几乎是必须要使用的工具，所以你在搭建SSM的过程中，也可以顺便了解一下maven的知识。在你目前这个阶段，你只需要在网络上了解一下maven基本的使用方法即可，一些高端的用法随着你工作经验的增加，会逐渐接触到的。

关于学习SSM框架的地址给大家推荐一个，这里面有视频，大家可以去观看，地址是 http://edu.51cto.com/lesson/id-76468.html。

四、找工作

当你完成开发框架的学习以后，你就该找工作了，在校的找实习，毕业的找全职。与此同时，在找工作的同时，你不应该停下你的学习，准确的说，是你在以后都不能停下学习。

上面这些内容你只是囫囵吞枣的学会了使用，你可以逐步尝试着去了解更多的东西，网络是你最重要的老师。

第二部分：对于参加工作一年以内的同学。

恭喜你，这个时候，你已经拥有了一份Java的工作。这个阶段是你成长极快的阶段，而且你可能会经常加班。

但是加班不代表你就可以松懈了，永远记得LZ说的那句话，从你入行那一刻起，你就要不停的学习。在这一年里，你至少需要看完《Java编程思想》这本书。这本书的内容是帮助你对于Java有一个更加深入的了解，是Java基础的升级版。

这本书很厚，当初看这本书，LZ花了整整三个月。正常速度的话，应该可以在半年左右看完。LZ这里不要求过高，只要你在一年以内把这本书看完即可。当然了，LZ所说的看完，是充分吸收，而不是读一遍就完事了，因此有些内容你可能会看不止一遍。

总而言之，这个阶段的核心学习思想就是，在工作中实践，并且更加深入的了解Java基础。

第二部分：对于参加工作1年到2年的同学。

这部分时间段的同学，已经对Java有了一个更加深入的了解。但是对于面向对象的体会可能还不够深刻，编程的时候还停留在完成功能的层次，很少会去考虑设计的问题。

于是这个时候，设计模式就来了。LZ当时看的是《大话设计模式》这本书，并且写了完整版的设计模式博客。因此，LZ要求大家，最多在你工作一年的时候，必须开始写博客，而设计模式就是你博客的开端。

请记住，LZ所提的基本都是最低要求，因此不要有任何松懈的心理，否则五年后，你不要去羡慕别人高于你的工资，也不要去羡慕别人进入了某公司。

这一年，你必须对于设计模式了如指掌，《大话设计模式》可以作为你的开端。当然了，你也可以去看LZ的个人博客去学习，地址是 http://www.cnblogs.com/zuoxiaolong/p/pattern26.html。

此外，设计模式并不是你这一年唯一的任务，你还需要看一些关于代码编写优化的书。比如《重构 改善既有代码的设计》，《effective java》。

总而言之，这个阶段，你的核心任务就是提高你的代码能力，要能写出一手优雅的代码。

第三部分：对于参加工作2年到3年的同学

有的同学在这个时候觉得自己已经很牛逼了，于是忍不住开始慢慢松懈。请记住，你还嫩的多。

这个阶段，有一本书是你必须看的，它叫做《深入理解Java虚拟机》。这本书绝对是Java开发者最重要的书，没有之一。在LZ眼里，这本书的重要性还要高于《Java编程思想》。

这本书的内容是帮助你全面的了解Java虚拟机，在这个阶段，你一定已经知道Java是运行在JVM之上的。所以，对于JVM，你没有任何理由不了解它。LZ之前有写过JVM系列的知识，可以去看一下，地址是http://www.cnblogs.com/zuoxiaolong/category/508918.html。

另外，在过去2年的工作当中，你肯定或多或少接触过并发。这个时候，你应该去更加深入的了解并发相关的知识，而这部分内容，LZ比较推荐《Java并发编程实战》这本书。只要你把这本书啃下来了，并发的部分基本已经了解了十之六七。

与此同时，这个阶段你要做的事情还远不止如此。这个时候，你应该对于你所使用的框架应该有了更深入的了解，对于Java的类库也有了更深入的了解。因此，你需要去看一些JDK中的类的源码，也包括你所使用的框架的源码。

这些源码能看懂的前提是，你必须对设计模式非常了解。否则的话，你看源码的过程中，永远会有这样那样的疑问，这段代码为什么要这么写？为什么要定义这个接口，它看起来好像很多余？

由此也可以看出，这些学习的过程是环环相扣的，如果你任何一个阶段拉下来了，那么你就真的跟不上了，或者说是一步慢步步慢。而且LZ很负责的告诉你，LZ在这个阶段的时候，所学习的东西远多于这里所罗列出来的。因此千万不要觉得你已经学的很多了，LZ所说的这些都只是最低要求，不光是LZ，很多人在这个时间段所学习的内容都远超本文的范围。

如果你不能跟上节奏的话，若干年后，如果不是程序猿市场还不错的话，你很可能不仅仅是工资比别人低，公司没别人好，而是根本就找不到工作。

总而言之，这个阶段，你需要做的是深入了解Java底层和Java类库（比如并发那本书就是Java并发包java.concurrent的内容），也就是JVM和JDK的相关内容。而且还要更深入的去了解你所使用的框架，方式比较推荐看源码或者看官方文档。

另外，还有一种学习的方式，在2年这个阶段，也应该启用了，那就是造轮子。

不要听信那套“不要重复造轮子”的论调，那是公司为了节省时间成本编造出来的。重复造轮子或许对别人没有价值，因为你造的轮子可能早就有了，而且一般情况下你造出来的轮子还没有现存的好。但是对别人没有价值，不代表对你自己没有价值。

一个造轮子的过程，是一个从无到有的过程。这个过程可以对你进行系统的锻炼，它不仅考察你的编码能力，还考察你的框架设计能力，你需要让你的轮子拥有足够好的扩展性、健壮性。

而且在造轮子的过程中，你会遇到各种各样的难题，这些难题往往又是你学习的契机。当你把轮子造好的时候，你一定会发现，其实你自己收获了很多。

所以，这个阶段，除了上面提到的了解JVM、JDK和框架源码以外，也请你根据别人优秀的源码，去造一个任何你能够想象出来的轮子。

第四部分：参加工作3年到4年的同学

这个阶段的同学，提升已经是很难了，而且这个阶段的学习往往会比较多样化。

因为在前3年的过程中，你肯定或多或少接触过一些其它的技术，比如大数据、分布式缓存、分布式消息服务、分布式计算、软负载均衡等等。这些技术，你能精通任何一项，都将是你未来面试时巨大的优势，因此如果你对某一项技术感兴趣的话，这个时候可以深入去研究一下。这项技术不一定是你工作所用到的，但一定是相关的。

而且在研究一门新技术时，切忌朝三暮四。有的同学今天去整整大数据，搞搞Hadoop、hbase一类的东西。过不了一段时间，就觉得没意思，又去研究分布式缓存，比如redis。然后又过不了一段时间，又去研究分布式计算，比如整整Mapreduce或者storm。

结果到最后，搞得自己好像什么都会一样，在简历上大言不惭的写上大数据、分布式缓存、分布式计算都了解，其实任何一个都只是浮于表面。到时候面试官随便一问，就把你给识破了。

一定要记住，作为一个程序猿，平日里所接触的技术可能会很多，但是想要让一门技术成为你的优势，那么一定是你对这门技术的了解强过绝大多数人才行。

因此在这个阶段，你就不能再简单的去学习前3年的内容了，虽然前面的学习如果还不够深入的话依旧要继续，但这个时候你应该更多的考虑建立你的优势，也可以称为差异性。

差异性相信不难理解，就是让你自己变得与众不同。你前面三年的学习足够你成为一名基本合格的Java开发者，但你离成为一名优秀的Java开发者还有很大的距离。

所谓优秀，即能别人所不能。而你前三年所学习的内容，是很多做过几年的Java开发都能够掌握的。那么为了让自己有差异性，你就需要另辟蹊径，找一个方向深入研究下去，以期在将来，你能够成为这个领域的专家，比如分布式计算领域的专家，大数据领域的专家，并发领域的专家等等。

此外，你除了建立你的差异性之外，还要去弥补你基础上的不足，直到现在，LZ都没有提及基础知识。原因是基础是很枯燥无味的，学的太早不仅容易懵逼，而且懵逼的同时还容易产生心理阴影，以至于以后再不想去研究这些基础。但基础又是你深入研究一些领域时所必须掌握的，比如你去研究分布式计算，你不懂算法你玩个毛毛？比如你去做分布式缓存，你对计算机系统的内存不了解，你如何去做缓存？

如果你的基础本来就非常强，那么恭喜你，相信你在之前的工作中已经充分体会到了这些基础对你的帮助。但LZ相信大部分人的基础都很薄弱，哪怕是科班毕业的人，很多人也不敢说自己当初的基础学的多么强大，比如算法、计算机系统原理、编译原理这些。

但是每个人时间都是有限的，而且这些基础的书籍每一本读下来，没个一年半载的，还真拿不下来，因此还是要有所抉择的。虽然艺多不压身，但问题是艺多是有代价的，是需要你付出时间和精力的，而LZ个人更赞成在同等代价的情况下获取最大的收获。

首先，LZ比较推崇的基础书籍有三本，分别是《深入理解计算机系统》，《tcp/ip详解 卷一、二、三》，《数据结构与算法》。其中TCP/IP有三本书，但我们这里把这三本看成是一本大书。

这三本分别适合三种人，《深入理解计算机系统》比较适合一直从事Java Web开发和APP后端开发工作的人群。《tcp/ip详解 卷一、二、三》比较适合做网络编程的人群，比如你使用netty去开发的话，那么就要对TCP/IP有更深入的了解。而《数据结构与算法》这本书，则比较适合做计算研究工作的人，比如刚才提到的分布式计算。

另外，LZ要强调的是，这里所说的适合，并不是其它两本对你就没有用。比如你做Java Web和APP后端开发，《tcp/ip详解 卷一、二、三》这本书对你的作用也是很大的。这里只是分出个主次关系而已，你要是时间足够的话，能把三本都精读那当然最好不过了。但如果时间有限的话，那么就先挑对你帮助最大的书去读。

理论上来讲，这一年你能把这三本其中一本精读下来，就已经非常厉害了。有了基础，有了前面的工作经验，你就可以去开拓属于你的领域了。

在这一年里，一定要规划好自己的领域，建立好自己的优势，制造出差异性。如果你对自己的领域不够清晰的话，随着你工作的时间日益增多，你接触的技术会越来越多，这个时候，你很容易被淹死在技术的海洋里，看似接触的技术越来越多，会用的也越来越多，但你毫无优势。

有的同学可能会问，“LZ，我也不知道我的领域是什么啊？怎么办呢？”

对于这种人，LZ只想说，“卧槽，这还问我？要不干脆我替你学习得了，好不好？”

第五部分：参加工作4年到5年的同学

经过前面一年的历练，相信你在自己所钻研的领域已经有了自己一定的见解，这个时候，技术上你应该已经遇到瓶颈了。

这个时候不要着急提高自己的技术，已经是时候提高你的影响力了，你可以尝试去一些知名的公司去提高你的背景，你可以发表一些文章去影响更多的人。当然，你也可以去Github创建一个属于你的开源项目，去打造自己的产品。这次的开源项目不同于之前的造轮子，你这个时候是真的要去尽量尝试造出来真正对别人有价值的轮子。

技术学到这个阶段，很容易遇到瓶颈，而且往往达到一定程度后，你再深入下去的收效就真的微乎其微了，除非你是专门搞学术研究的。然而很可惜，大部分程序猿做不到这一步，那是科学家做的事情。

这个时候提高影响力不仅仅是因为技术上容易遇到瓶颈，更多的是影响力可以给你创造更多的机会。程序猿在某种程度上和明星很像，一个好的电视剧和电影就可以成就一批明星，程序猿有的时候也是，一个好的项目就可以成就一群程序猿。

比如国内几个脍炙人口的项目，像淘宝、支付宝、QQ、百度、微信等等。这每一个项目，都成就了一批程序猿。LZ敢说，这里面任何一个项目，如果你是它的核心开发，光是这样一个

Title，就已经是你非常大的优势。更何况还不止如此，Title说到底也是个名头，更重要的是，这种项目在做的时候，对你的历练一定也是非常给力的。
而你如果想要参与这样的项目，除了靠运气之外，影响力也是很重要的一个手段。比如你在分布式计算领域有一定的影响力，那么如果有什么好的关于分布式计算的项目，对方就很可能会邀请你。就算人家不邀请你，你自己主动去面试的时候，对方如果知道你在这个领域的影响力，也肯定会起到很大的作用，而这个作用，甚至可能会超过你现在的技术能力。
所以，在这个阶段，你最大的任务是提高自己的影响力，为自己未来的十年工作生涯那一天做准备。如果你能够靠你的影响力和以前积累的技术，参与到一个伟大的项目当中，那么你后面的五年也就有着落了。
当然了，LZ现在满打满算，做程序猿也就4年半不到，因此关于4年到5年这一部分，LZ的见解不一定是对的，就算是对的，也不一定是适合任何人的。所以，希望大家自己有的判断力，去决定到底该如何度过这一年。

结语

本文到此就基本结束了，整篇文章很长，但其实主要就说了两部分内容，一个是社招面试的准备，一个是Java生涯的学习。
关于这两部分，LZ已经给出了自己的见解，但是还是那句话，每个人吸收知识的时候，都要有抽取精华，去除糟粕的能力。LZ所说的，可能有些是对的，有些是错的，有些是适合你的，有些是不太适合你的，你要自己能够判断。
其实你在生活和工作当中也是一样的，你身边的人形形色色，有的人你喜欢，有的人你很讨厌。但其实你喜欢的人也有缺点，你讨厌的人也有优点。你要学会从你讨厌的人身上学会他的优点，千万不要一棒子打死，这只会让你失去很多学习成长的机会。
好了，说了这么多了，就到此为止吧，希望本文可以帮助到作为程序猿或即将成为程序猿的你。
最后，如果你对阿里巴巴技术感兴趣，希望到阿里巴巴平台上从事技术相关的岗位，可以试试这里：招聘帖