

# 阿里 Java 工程师分享 3 年工作经验的程序员应该具备的技能

作者: [wts](#)

原文链接: <https://ld246.com/article/1514355087328>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

每个程序员、或者说每个工作者都应该有自己的职业规划，如果你不是富二代，不是官二代，也没有业规划，希望你思考一下自己的将来。今天小编给大家分享的是一篇来自阿里Java工程师对3年工作经验程序员的职业建议，希望对你有启发。

## \*\* 程序员的三个阶段\*\*

### 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，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

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西，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逐步走向系统分析或是架构师，成为项目组中不可或缺的人物。

### 第三阶段——十年

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你。

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：

\*\* 我到底适不适合当一名程序员？\*\*

---

\*\* 我到底应不应该一辈子以程序员为职业？\*\*

---

\*\* 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。

## 关于专业技能

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，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。

让我印象深刻的一次面试经历，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，

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，我最终选择的就问了这个问题（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，实现的优缺点。

集合要掌握的是ArrayList、LinkedList、Hashtable、HashMap、ConcurrentHashMap、HashSet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，我提供三点回答或者是研究方向：

ConcurrentHashMap的锁分段技术

ConcurrentHashMap的读是否要加锁，为什么

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：

你的项目中用到了哪些设计模式，如何使用

知道常用设计模式的优缺点

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如何实现多线程了，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？

你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...

## \*\* 5、IO\*\*

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，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源码就是实打实地你平时是不是爱钻研了。我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题——String的hashCode()方法是怎么实现的，幸好我平时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：

List、Map、Set实现类的源代码

ReentrantLock、AQS的源代码

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

线程池的实现原理

Object类中的方法以及每个方法的作用

这些其实要求蛮高的，我去年一整年基本把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，两句话就搞定的事儿，即使你不会准备一下就好了。我遇到的最态的是让我画一下Spring的Bean工厂实现的UML图，当然面对这样一个有深度的问题，我是绝对答出来的.....

## \*\* 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，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，可以不了解它们的体实现，但是要想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？

我答到用的Hash表吧，答错。他又问，你知道为什么要使用树吗？我答到因为Hash表可能会出现比多的冲突，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，基本证最多二三十次就能找到想要的数，对方说不完全对，最后我们还是交流了一下这个问题，我也明了为什么要使用树，这里不说，大家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方法使用的是哪种序方法，额，吐血三升。当然为了显示我的博学，对算法分析也有一定的研究，我还是硬着头皮说了句可能是冒泡排序吧。当然答案肯定不是，有兴趣的朋友可以去看一下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\*\*

出乎我意料，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。要道，我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级特性与最佳实践》，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，我就简单地列一个提纲吧，谈谈Java虚拟机中比较重要的内容：

Java虚拟机的内存布局

GC算法及几种垃圾收集器

类加载机制，也就是双亲委派模型

Java内存模型

happens-before规则

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。我碰到过问得最多的两个问题是：

1.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：

2.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/重定的区别、HTTPS的实现原理也都可能会被考察到。

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，命良方、包治百病、屡试不爽。

原文链接[http://mp.weixin.qq.com/s/6AapptJs\\_JTkV7t3Fa1IMQ](http://mp.weixin.qq.com/s/6AapptJs_JTkV7t3Fa1IMQ)