

3 年 Java 应该具备的技能体系

Java技术栈 5天前



Java技术栈

www.javastack.cn

优秀的Java技术公众号

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，我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，我最终选择的 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、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，两句话就搞定的事儿，即使你不会准备一下就好了。我遇到的最变态的是让我画一下Spring的Bean工厂实现的UML图，当然面对这样一个有深度的问题，我是答不出来的

4、数据库

数据库十有八九也都会问到。一些基本的像和 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。

5、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：

谈谈分布式Session的几种实现方式。(大家可以聊下你们知道的实现方法)推荐大家看这篇文章《分布式Session共享解决方案》。

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。

6、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？

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，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。

当然为了显示我的博学，对算法分析也有一定的研究(◕‿‿◕)b，我还是硬着头皮说了一句可能是冒泡排序吧。推荐大家看这篇文章《[图文带你了解 8 大排序算法](#)》。

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。

7、Java虚拟机

出乎我的意料，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，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高级特性与较佳实践》，我就读了不下五遍。

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：

(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。

8、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。

9、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。（大家可以留言讨论下用的什么方法实现）

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？推荐大家看这篇文章《史上最全 Java 多线程面试题及答案》。

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。

10、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。

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了我，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：

- (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，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、花费精力，当然回头看，是值得的——不仅仅是为了应付面试。

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。

@All 你工作几年了？这就是差距。

往期干货推荐

-
- 1. [从 0 开始手写一个 Spring MVC 框架](#)
 - 2. [详解 Java 中的三种代理模式](#)
 - 3. [RESTful API 设计技巧经验总结](#)
 - 4. [Java 面试常问计算机网络问题](#)
 - 5. [史上最全 69 道 Spring 面试题和答案](#)
-



阅读原文