

序言

最近有位做 java 的朋友和我说，他的直属领导离职，然后他被晋升到了领导的职位，不仅要负责团队的管理，还要对来应聘的人员来做面试，由于他之前没有做过面试，所以来问我该如何做面试，那本文就来说下，该如何做好一个合格的 java 技术面试官。

本文将从以下三个大的方面讲述：


1. 从我个人角度出发，我认为应该怎样做好一个合格的 java 面试官。
2. 各个级别的 java 工程师，各自应该具备什么样的技能与素质，这样才能有针对的面试。
3. 我自己在做 java 面试官时，经常问的一些问题，大家可以参考。

作者简介

个人信息

- 姓名： 负学文
- 出生日期： 1988.02.26
- 现居： 吉林长春
- 毕业院校： 黑龙江科技大学
- 学历： 大学本科
- 邮箱： 18612230443@163.com
- 个人的 gitHub 地址： <https://github.com/aloofWolf/JLF>

- 性别： 男
- 户籍： 吉林省四平市
- 工作经验： 11 年
- 专业： 信息与计算科学
- 手机（同微信）： 18612230443
- 爱好： 编程、读书、写作、象棋



获得荣誉

- 曾在技术领域获取过国家专利，估值 800 万元。
- 经常在今日头条上发表一些技术的文章，目前已有 1500+粉丝。今日头条账号：负学文。
- 任职期间多次被评为优秀员工。
- 曾被评选为年度杰出面试官。

自我评价

- 对软件编程行业也很浓厚的兴趣，曾在技术领域获取过国家专利，估值 800 万元。
- 对一些技术点有自己独到的看法。
- 有很强的时间观念，工作中严以律己，宽以待人。
- 有上进心，不安于现状。
- 性格偏内向，但为人真诚。
- 有自己的想法，不喜欢随波逐流。
- 对工作认真负责，有担当。
- 注重过程，不过分在意结果。
- 高调做事，低调做人
- 谦虚但不自卑，自信但不骄傲。

专业技能

语言	java	关系数据库	mysql
缓存数据库	redis	web 框架	struts2、springmvc、springBoot、jfinal
持久层框架	hibernate、mybatis、mybatisPlus	web 容器	tomcat、weblogic
网络传输	tcp、udp、ftp、socket、http、https	定时任务	quartz、xxl-job
json 框架	fastJson、gson、jackson	模板引擎	velocity
日志管理	slf4j、log4j、logback、log4j2	消息队列	rocketMq、kafka

OFFICE 框架	Apache POI	微服务	dubbo、springCloud
网关服务	nginx, keepalived 、 gateway	注册中心	zookeeper、eureka、nacos
服务调用	feign、openFeign	负载均衡	nginx、ribbon、loadbalabcer
限流熔断降级	nginx、hystrix 、sentinel	配置中心	springCloudConfig、nacos、apollo
分布式事务	seata、自研分布式事务解决方案	分布式锁	redission
分布式 session	Redis 统一存储、jwt	数据同步	canal
大数据	elasticsearch	加密算法	md5、rsa、3des 等等
Jvm 监控	arthas	容器管理	docker、k8s
版本控制	svn、git	项目构建	maven

目录:

一、如何成为一个优秀的面试官	5
1.1 技术能力过硬	5
1.2 避免八股文	5
1.3 基础能力的考察	5
1.4 善于发现应聘者的闪光点	5
1.5 不要奢望招一个全才	6
1.6 考察应聘者的上进心	6
1.7 让应聘者有充分发挥自己能力的空间	6
1.8 给应聘者一个提问自己的机会	7
二、各个级别的 JAVA 工程师的区别	7
2.1 初级 java 工程师	7
2.2 中级 java 工程师	7
2.3 高级 java 工程师	7
2.3 架构师	8
三、我做 JAVA 面试官时，常问的问题	8
3.1 介绍自己最近做过的项目	8
3.2 基础知识的考察	8
3.3 解决问题的能力考察	9
3.4 对面向对象思想理解的考察	12
3.5 对代码习惯的考察	12
3.6 给应聘者一个展示自我的空间	12
3.7 考察应聘者的发展潜力	12
3.8 通过一些外部信息来进一步综合考察	12
3.9 给应聘者一个提问自己的机会	12

一、如何成为一个优秀的面试官

1.1 技术能力过硬

首先作为一个面试官，你必须自身技术能力过硬，这样你才有资格去面试别人。虽然说应聘者的技术能力强过面试官的这种现象，也是很正常的现象，但是作为一个面试官来讲，你不能说每来一个应聘者，技术能力都比你强，那你就真的该反思了。麻烦请先提升自身的技术能力，再来做面试官了。不得不说，这种自身技术能力较弱的面试官，还是很常见的。

1.2 避免八股文

现在有很多人反应，面试就是背八股文，有些东西工作时用不到，但是面试时会问，所以面试前需要去背这些八股文。我个人是比较反感这些八股文的，因为它无法考察出应聘者的真实水平。我一个技术能力很强的人，面试前没有准备这些八股文，结果可能答得不好。一个技术能力很差的，就因为面试前背了这些八股文，结果可以答得很好。这有什么意义呢？

我是从来都不会问这些八股文的问题的，我通常会拿一个工作中实际遇到的场景来作为切入点，由面到点，层层递进。比如说我经常问的一个问题：接口变慢的问题你如何去定位问题并且如何优化？你可能会想到由于数据量变大了，那这个问题怎么解决，就会涉及到数据库的优化，比如，索引结构，存储引擎，分库分表，然后在递进，就会引出缓存，那么使用缓存，就会涉及到数据一致性等等的问题。再往下延伸，还会延伸到 es, hbase 的使用。就这样一点一点地延伸，不仅可以考察出应聘者的知识面，也可以考察应聘者自己会的这些知识点，能否在一个合适的场景下，选择去运用。还可以考察出应聘者实际解决问题以及沟通的能力，这些是不是比八股文强多了呢？

1.3 基础能力的考察

记得有位艺人曾经说过这样一句话：艺人拼到最后，拼的是文化。那么我同样想说：技术人员拼到最后，拼的就是基础。一个基础不行的人，也许他的能力可能暂时会强过同龄人，但是他一定走不远，后面，他一定会被那些基础好的同学反超。

所以在面试的过程中，对基础的考察也是很重要的，它可以看出应聘者的上限在哪里。比如 java 中的线程安全的问题，各种集合类的原理，接口和抽象类的使用场景等等。当然了，这些问题你不能直接去问，如果你直接去问，那就又变成八股文的问题了，考察不出来应聘者的真实水平，没啥意义，你还是拿一个实际的场景作为切入点，来考察应聘者的对基础的掌握情况。

1.4 善于发现应聘者的闪光点

有很多面试官，偶然间学会了某个很偏的知识点，然后在后面的面试中，就拿着这个很偏的

知识点去为难别人,这样做没有任何意义,可能唯一的意义就是,让面试官自己得到了满足。有时应聘者在面试的过程中,可能会比较紧张,这时面试官需要去尽量让面试的氛围变得轻松,让应聘者慢慢地放下紧张,这样他才能更好的表达他真实的自己。然后面试官在去慢慢的引导,慢慢的发现应聘者的闪光点。作为面试官,要时刻记得面试的目的,是要招聘人才的,而不是为了秀自己的。

1.5 不要奢望招一个全才

我自己曾经帮助过其他公司做过做过初试,但是这家公司的要求特别高,各种各样的框架,中间件,不仅要会,而且还要精通,而且必须要做过大数据量,高并发的。对这种要求,我是很反感的,按照这样的要求,我面试一个人,没有一个小时的时间,我连问题都不完,有的甚至要问两个小时,才能把问题问完。但是没办法,人家要求这样,我就得按照这样的要求来面。我面试了很长很长时间,面试了很多很多人,都没能找到符合这种要求的人。后来面试了一个人,这个人其他的方面都很符合要求,唯一的不满足的就是没有做过微服务。但是我觉得其实也不是什么大问题,因为他的学习能力挺强的,微服务也不是什么多高深的技术,即使没用过,接手起来也会很快的。结果在复试的时候,人家直接以没做过微服务为由,给 pass 了。

我对这种做法真的很不理解,我觉得,我们不要去奢望招一个全才,什么都懂,什么都精通。这种人可遇而不可求,我们招聘这种人要花费的精力太大了,我在面试时,更注重的是解决问题的能力,学习能力,发展潜力,沟通能力等等。至于他会有一些技术短板,都不是什么大问题。

1.6 考察应聘者的上进心

面试的时候,我通常会问这样一个问题:你对自己的未来有什么规划?并且是如何为了这个目标去努力的?

我问这个问题的意义,其实就是想看应聘者有没有上进心。也许一个有明确职业规划并且为之努力奋斗的人,技术能力也不一定特别强,但是一个没有职业规划的人,它的技术能力一定不怎么样,没有个例,试想,一个连奋斗目标都没有的人,它会努力吗?

记得有个应聘者是这样回答这个问题的:你也知道,做我们这个行业,基本 35 岁就失业了,所以我打算趁年轻就干几年,然后就转行不干这个了。我问他:那你有没有想过,如何让自己在 35 岁以后,不失业呢?

其实我在这里也想说下 35 岁失业这个问题,你年轻的时候没有进取心,享受安逸,然后到了 35 岁,能力匹配不上年龄,那不淘汰你淘汰谁呢?那些 40 岁甚至 50 岁还拿着高薪的人,他们为什么没有被淘汰呢?

1.7 让应聘者有充分发挥自己能力的空间

在前面的面试中,可能都是面试官提问,应聘者来回答,节奏始终掌握在面试官的这一边,应聘者没有发挥自己的空间,为了让应聘者充分的发挥自己的空间,我通常会问几个开放性

的问题，比如：说一下你在实际工作中有没有上遇到过一些难以解决的技术问题，是通过你自己独立解决的？说一下你在平时工作时自己认为一些比较好的工作经验或者工作习惯？我问这些问题的目的是，就是为了让应聘者有可以充分发挥自己的空间，可以更加深入地去了解应聘者的真实情况。

1.8 给应聘者一个提问自己的机会

其实面试本来就是一个双向选择的过程，面试官在考察应聘者的能力，应聘者同样也在考察公司是否值得我去，所以双方是平等的。那么作为面试官，一定要给应聘者一个提问自己的机会，让应聘者也去了解公司。面试官不用觉得自己高高在上，应聘者也不用觉得自己低人一头，双方都是平等的。

有很多时候，应聘者面试通过，进入公司后，发现和自己预期的大相径庭，于是很快选择了离职，这对于一个公司来讲，也是很浪费成本的，而造成这种问题的原因，有时候面试官是有一定的责任的。

二、各个级别的 java 工程师的区别

首先，你要清楚你要招聘什么级别的 java 工程师，然后每个级别都应该掌握什么样的技能，具备什么样的素质，这样，你才能具有针对性的去面试。

2.1 初级 java 工程师

- 1、按照要求实现好代码。
- 2、思维逻辑、理解能力要强。这点很重要，技术能力可以稍差一些，毕竟经验不足，但是思维逻辑和理解能力一定要强。当别人在给你讲解一些复杂的业务或者代码逻辑时，你最起码要听得懂并且理解，减少无效沟通。

2.2 中级 java 工程师

- 1、对于如何写出优秀代码、如何做好一个系统、常见问题的解决，要有自己的想法，可能想法不是很成熟，但是必须要有。因为对于中级的 java，一般都会让你去做一些某个具体功能的代码设计和数据库设计，然后交给架构师来评审。如果你连如何做好的想法都没有，你是做不好设计的。
- 2、基础一定要掌握扎实。

2.3 高级 java 工程师

- 1、当你的对于如何写出优秀代码、如何做好一个系统、常见问题的解决得想法，都成熟以

后，利用这些想法，自己或者带领团队，可以独立维护一个大型的模块或者大型的系统。

2、对于一些突发事件，或者生产问题，可以快速定位并解决问题。

2.3 架构师

1、可以深入的分析业务需求，选择合适的技术架构和方案，并独立设计出可以支持高并发、高可用、高安全、高维护性的系统。

2、对于生成环境的维护、生产问题的追踪，有一套成熟的方案，并且可以实施。

三、我做 java 面试官时，常问的问题

下面来说下我在做面试官的时候，经常问的一些问题。

需要说明的是，在面试这些不同的级别的工程师的时候，我问的问题大体上都是相同的，只不过对于不同的级别，我对应聘者给我的回答的要求，是不同的。

3.1 介绍自己最近做过的项目

这个问题，主要考察应聘者对于自己做过的项目的熟悉程度，以及对项目中的难点做剖析。我面试过很多应聘者，真的就连自己做过的项目都说不明白，那这种人就可以直接 pass 了。

3.2 基础知识的考察

说出如下图所示代码的输出结果及原因：

```
public class Test {  
  
    String str;  
    Integer i;  
    public Test(String str, Integer i) {  
        super();  
        this.str = str;  
        this.i = i;  
    }  
  
    public static void main(String[] args){  
        Test t1 = new Test("abc",1);  
        Test t2 = new Test("abc",1);  
        Map<Test,String> map = new HashMap<>();  
        map.put(t1, "a");  
        map.put(t2, "b");  
        System.out.println(map.size());  
    }  
}
```


此题主要考察的就是最基础的 hashCode 方法以和 equals 方法的使用，以及 HashMap 如何判断 key 是否重复。如果应聘者能够准确的回答，并且把原因都说的很清楚，那最起码可以证明，他的基础是没问题的。

记得曾经面试过一个高级 java 工程师，他的解决问题的能力、沟通表达、思维逻辑都特别的好，唯独就这个问题回答的不是很好，基础有些不太好。当然了，我最后还是让他面试通过了，但是最后我也跟他说了如下这番话：可以看得出来，你的解决问题的能力、沟通表达、思维逻辑都很好，但是到了你这个级别，连 hashCode 方法以和 equals 方法都说不明白的话，我觉得有点不太应该，所以你回去一定要多掌握掌握这些基础的东西，如果这些特别基础的东西都掌握的不太清楚的话，你就没有办法保证你写的代码一定没有问题。我们这些 java 工程师，其实拼到最后，拼的就是基础，如果你基础掌握的不好，你在后面提升自己的过程中，一定会遇到瓶颈的。

其实这个问题，还可以延伸出来以下问题：

- 1、hashmap 的线程安全问题
- 2、如何高效的使用 hashmap
- 3、什么样的数据适合用 hashmap 存储，什么样的数据适合用自定义的对象存储

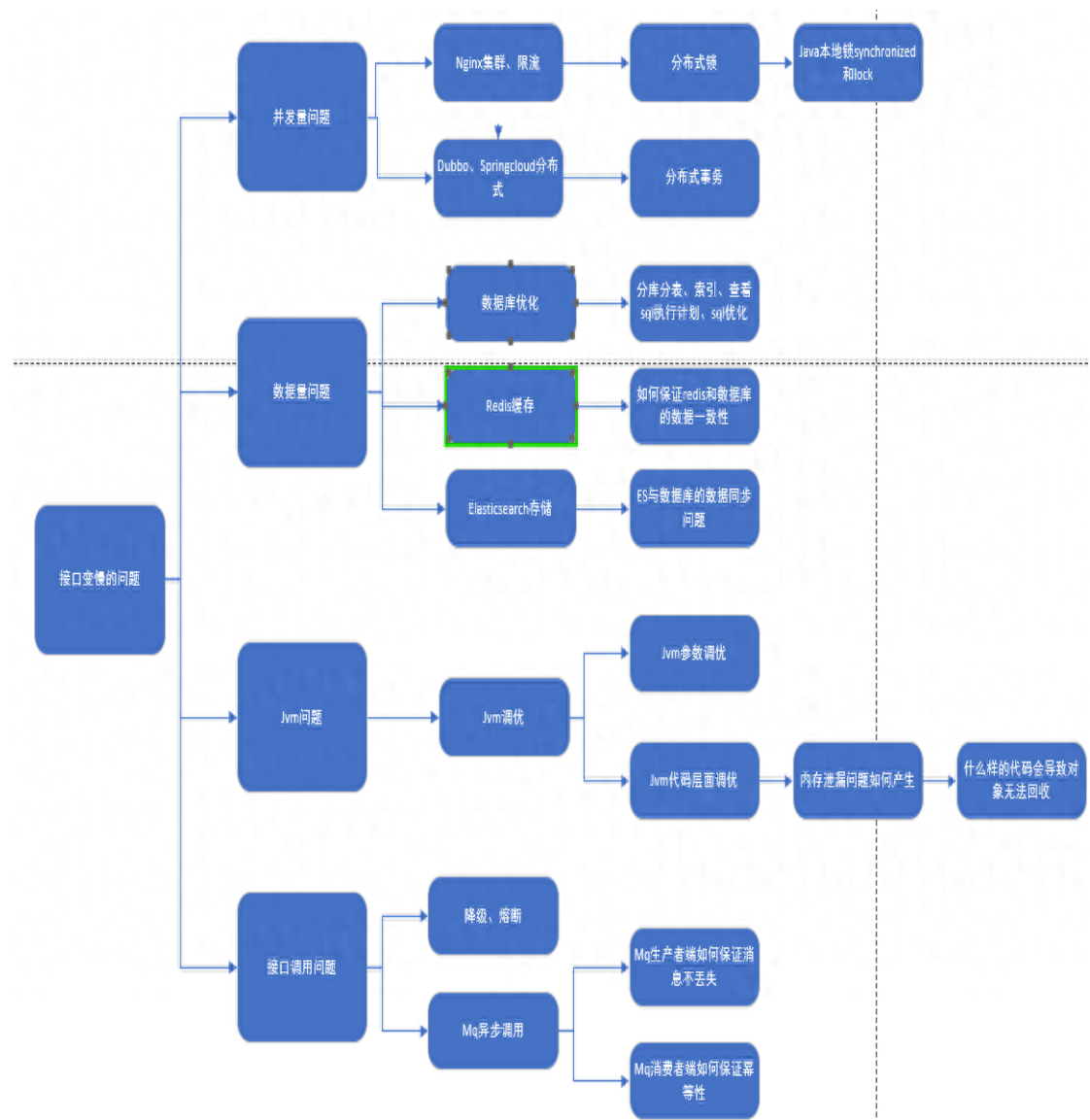
可能后面两个问题，对中级的来说有些超纲了，答不上来也无所谓，但是你一旦答上来了，就是一个加分项。

有很多面试官，都喜欢问 hashmap 的红黑树结构、链表闭环的问题，concurrenthashmap 如何加锁的问题。这些问题我从来不会去问，因为这些问题都是纯纯的八股文的问题，工作中根本用不到，没什么意义。我问的问题，都是可以在工作中用到的东西，这些问题，才能看出一个人的真实水平。

3.3 解决问题的能力考察

问题：如果一个后端的接口，在刚刚投产运行时，执行的效率很快，但是随着用的时间越来越长，它的执行效率变得越来越慢，你能想到有哪些情况会导致这个问题，并且如何解决？

其实这个问题能延伸出来的内容特别多，下图是我列举出来的可以延伸出来的内容，但是这些也并不是全部。



对于上图中提到的 java 本地的 synchronized 锁，我觉得有两点是比较重要的，一个是锁的粒度的问题，一个是可重入的问题，如果这两点了解的不清楚的话，在用的时候可能会出现问题，而对于这两点，我通常会用下面两道编程题来考察应聘者的掌握情况：

```

public class LockTest {

    private static int i;

    public synchronized void method1(){
        i = i + 1;
    }

    public static void main(String[] args) throws InterruptedException{

        Thread t1 = new Thread(new ThreadTest());
        Thread t2 = new Thread(new ThreadTest());
        Thread t3 = new Thread(new ThreadTest());
        Thread t4 = new Thread(new ThreadTest());
        Thread t5 = new Thread(new ThreadTest());
        t1.start();
        t2.start();
        t3.start();
        t4.start();
        t5.start();
        Thread.sleep(1000);
        System.out.println(LockTest.i);
    }
}

class ThreadTest implements Runnable{

    @Override
    public void run() {
        LockTest lock = new LockTest();
        lock.method1();
    }
}

```

```

public class LockTest2 {

    boolean flag = true;
    int i;

    public synchronized void method1(){

        i = i + 1;
        if(flag){
            method2();
        }
    }

    public synchronized void method2(){

        i = i + 5;
        flag = false;
        method1();
    }

    public static void main(String[] args){
        LockTest2 lock = new LockTest2();
        lock.method1();
        System.out.println(lock.i);
    }
}

```

3.4 对面向对象思想理解的考察

问题：如果后端有十个接口，这十个接口都在调用同一个公共方法，这个公共方法做的事情又比较多，现在有一个接口的业务发生了变化，需要调整这个公共接口，但是需要调整的只是其中一小部分。那么如何利用面向对象的思想解决这个问题？使得既能满足业务的需求，又可能保证代码的可维护性和可扩展性？

3.5 对代码习惯的考察

问题：是否有在平时地工作中积累一些自己认为比较好的代码规范？

记得我刚刚毕业进入第一家公司的时候，我就在工作中总结了很多的代码规范，比如：如何做解耦合、枚举的使用，map 与自定义类的适合使用的场景等等。在总结这些规范的过程中，也让自己的编码水平得到了很大的提升。

但是我现在我在面试时问这个问题的時候，很多人都回答，用的是阿里的编码规范，但是我覺得，阿里的规范是很好，但是那终究是别人总结出来的东西，有些事情，只有你真的自己做过了，你才能在过程中受益。而且，我相信，只要你真的用心的做这件事了，你一定可以总结出来一些很好的，但是阿里规范中没有的规范。

3.6 给应聘者一个展示自我的空间

问题：是否有在平时地工作过程中，遇到过一些疑难的，并且是自己独立解决的 bug。

可能在前面的提问中，都是由我来主导这个面试的流程，那现在通过这个问题，把主导权交给应聘者，给应聘者一个展示自我的空间，看看你能发挥成什么样。

3.7 考察应聘者的发展潜力

问题：对自己的未来有什么规划，并且为这个目标去努力？

我始终相信，一个对自己的未来有明确规划的人，他的能力一定比没有规划的人强很多，发展潜力也会很大。因为人需要有梦想，有了梦想，才会努力。

3.8 通过一些外部信息来进一步综合考察

问题：以你现在的技术能力，在你们现在的团队中，可以排到第几？

问题：目前面试了几家，感觉如何，是否有收到 offer？

3.9 给应聘者一个提问自己的机会

尊重每一个应聘者，无论他的技术能力如何。