## 经过这次面试，我总结一下几点：

**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。

**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**

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。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

**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。  
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

**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。

**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。

**10、Java虚拟机**  
Java虚拟机应该是很重要的一块内容，谈谈Java虚拟机中比较重要的内容

1. ava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

也许这些内容面试无用，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。

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）
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。

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。