首先JVM面试一般逃不过：GC算法，类加载机制，调优（堆内存设置，算法GC算法配置）

我们同时站在面试者与被面试者去阅读深入java虚拟机，去思考自己会去考察什么，通过搜罗面试题以及有经验的大神们比较关注什么

第一章：

相当于引言，如果是我我是不会浪费时间在这个上面

第二章：Java内存区域与内存溢出异常

1作者以运行时数据区域来对标题呼应，java内存区域和运行时数据区域是一个意思么？（方法区、栈，程序计数器，堆，栈+直接内存）

2.程序计数器为什么没有规范OOM？？OOM是在什么情况下发生的？或者问程序计数器在什么情况下会抛出OOM异常？（好像有点误导的感觉）

3.java规范对虚拟机栈规定了两种内存异常？他们在什么情况下发生?

4.堆主要存放的是什么数据？（很多人可以答出所有的对象的实例，但是会漏掉数组不是原始类型的数组）

5.JIT编译器有没有？逃逸分析技术呢？栈上分配，标量替换？所有的对象一定都在堆上么？

6为什么要优化呢？在堆上不好么？堆的空间那么大，栈那么小？

7**栈上分配技术有没有了解过**

8**标量替换有没有了解过**

9你用过哪些虚拟机（HotSpot VM、J9 VM、Zing VM，BEA Jrockit）

10方法区主要放置什么（那些不怎么变的，比如类信息，常量，jit编译的代码（马长累）），常被人成为permanent，注意姿势大家这么说，并非官方说法。因为GC有的时候还是会拜访这个区域的，只是一般都很难访问，甚至有的虚拟机都没有！

11除了刚刚说的这些内存区域，还有没有其他的？（直接内存，）

12直接内存是一个容易被忽略的boy，一般虚拟机配置内存的时候都不会将其包含进去，然而其确实又占内存，很容易出现个个内存区域总和大于物理内存限制，导致OOM

13.直接内存一般用来（NIO）

14句柄和指针（句柄的本质仍然是指针，但是指针的权限因为太大，所以就进行了封装，就像身份证证和学生证的区别吧）

15引用是用什么实现的（或者说有那两种实现方式）：使用句柄的好处在于，当对象被移动的时候，引用本身不会修改，缺点就是慢，使用指针的方式就反着看。

**16有没有遇到OOM或Stack Overflow的的情况，**