类加载机制？

**加载：将.class文件载入到内存。**

**连接：**

**验证：验证类加载正确性。**

**准备：分配内存，初始化默认值。**

**解析：把类中的符号转化为直接引用。**

**初始化：常量初始化。**

Java引用的类型有哪些？

**强引用：垃圾回收时不会进行回收（new实例化的引用）**

**软引用：内存不足时会进行回收（SoftReference）**

**虚引用：每次垃圾回收时都会进行回收（WeakReference）**

**弱引用：永远不可达（返回null），每次垃圾回收都会被回收（PhantomReference**

**）。**

JVM对内存结构有哪些？

**方法区（jdk1.8之前是永久代，之后是元空间）、堆、栈（JVM栈、本地方法栈、程序计数器）。堆与方法区是线程共享的，栈是线程独有的。**

**堆：主要存放实例化的对象。**

**栈：主要存放线程相关的私有信息。**

**方法区：1.8之前版本为永久代，在JVM中，与堆位置连续，主要存放类信息、静态变量、常量，Full GC时会对其进行垃圾回收；1.8及之后版本为元空间，在本地内存中，主要存放类信息，将静态变量、常量移至堆。**

为什么要把永久代替换为元空间？

1. **永久代大小启动时固定，容易出现OOM，不易调优。**
2. **Full GC会对永久代进行垃圾回收，提升了GC的复杂度。**
3. **常量、静态变量存在永久代中，更容易出现性能问题和内存溢出。**
4. **HotSpot（只有HotSpot存在永久代） 与 JRockit 合二为一。**

堆中是怎样分区的？

**新生代(Eden、FromSuvivor、ToSuvivor)：新建的对象存入新生代的Eden中，Eden不足时会进行垃圾回收把存活的对象存入FromSuvivor，FromSuvivor内存不足时触发垃圾回收，把存活的对象存入ToSuvivor中，ToSuvivor、FromSuvivor交换名称。**

**老年代：长期存或的对象存入老年代，老年代内存不足时会触发Full GC。**

垃圾回收主要发生在何处？栈会进行垃圾回收么？

**垃圾回收GC主要发生在堆内存中，栈内存不会进行垃圾回收。**

通过什么算法确定对象需要被回收？

**引用计数法：对应用次数进行计数，若技术为0则可能垃圾回收标记为可回收。存在循环应用导致对象无法被回收的情况：**

*MyObject myObject1 =newMyObject();*

*MyObject myObject2 =newMyObject();*

*myObject1.ref = myObject2;*

*myObject2.ref = myObject1;*

*myObject1 =null;*

*myObject2 =null;*

**可达性算法：通过由GCRoot为起点开始搜索，搜索到的对象不进行回收，未搜索到的对象进行回收。GC Root包括：虚拟机栈中引用的对象、本地方法栈中引用的对象、方法区中类静态属性引用的对象、方法区中常量引用的对象**

常用的垃圾回收算法有哪些？

**标记-清除、复制、标记-整理、分代回收。**

**常见的垃圾收集器有哪些？**

**如何进行生成dump文件？**

1. **通过命令生成：jmap –dump:format=n,file=/home/dump.hprof pid**
2. **jvm配置：XX:+HeapDumpOnOutOfMemoryError、-XX:+HeapDumpBeforeFullGC 、-XX:+HeapDumpAfterFullGC，配合-XX:HeapDumpPath=/home/app/dumps/**