关于JVM——简答题

**-什么是JVM**

答：JVM（Java Virtual Machine）是JAVA虚拟机，是一个虚构出来的计算机，通过在实际的计算机上仿真模拟各种计算机的功能来实现的。如果当前线程执行的是native方法，则其值为null。

**-JVM内存分哪几个区，每个区的作用是什么？**

答： **程序计数器(Program Counter Register)：**内存空间小，记录代码执行地址。

**Java栈(VM Stack)**：Java栈也称作虚拟机栈。线程私有，每个线程对应一个JAVA栈，其生命周期与线程同进同退。每个Java方法再被调用的时候都会创建一个栈帧，并入栈。一旦完成调用，则出栈。所有的栈帧都出栈后，线程也就结束。

**本地方法栈(Native Method Stack)**：作用与Java栈的作用和原理相似。区别在于Java栈是为执行Java方法服务的，而本地方法栈则是为执行本地方法服务的。

**方法区(Method Area)**：线程共享，存储每个类的信息、静态变量、以及编译器编译后的代码等。方法区里有一个运行时常量池，用于存放静态编译产生的字面量和符号引用。该常量池具有动态性，也就是说常量并不一定是编译时确定，运行时生成的常量也会存在这个常量池中。

**堆(Heap)**：存放对象实例，被所有线程共享，JVM只有一个堆。

**-简述JAVA垃圾回收机制**

答：JVM中有一个垃圾回收线程，它是低优先级的，在正常情况下是不会执行的，只有当虚拟机空闲或当前堆内存不足时，才会触发执行，回收没有被引用的对象。

**-如何判断一个对象是否存活**

答： **引用计数器法**——给每一个对象设置一个引用计数器，每当有一个地方引用这个对象时，计数器加一，反之。当一个对象的引用计数器为零时，说明此对象没有被引用，也就是“死对象”，将被回收。

**可迭性算法（引用链算法）**——从一个称为GC ROOTS的对象开始向下搜索如果一个对象到GC ROOTS没有任何引用链相连时，则说明此对象不可用。若要真正的回收需要经历两次判定标记。GC ROOTS：虚拟机栈中引用的对象；方法区类静态属性引用的对象；方法区常量池引用对象；本地方法栈JNI引用的对象。

**-JVM在执行一个JAVA类时所经历的过程有哪些?请按顺序写出！**

答： **装载**：查找和导入Class文件；

**链接**：执行校验、准备和解析步骤，其中解析步骤是可以选择的：

**校验**：检查载入Class文件数据的正确性；

**准备**：给类的静态变量分配存储空间；

**解析**：将符号引用转成直接引用；

**初始化**：对类的静态变量、静态代码块执行初始化工作。