**作业题：JVM 虚拟机论述题**

**题目 01- 请你用自己的语言向我介绍 Java 运行时数据区（内存区域）**

* 堆、虚拟机栈、本地方法栈、方法区（永久代、元空间）、运行时常量池（字符串常量池）、直接内存

堆：是虚拟机中管理的最大一块内存，被所有线程共享的一块内存区域，在虚拟机启动时创建，用于存放对象实例。分为新生代和老年代。

方法区：各个线程共享的内存区域，主要是用来存放已被虚拟机加载的类相关信息，包括类信息、运行时常量池、字符串常量池。

虚拟机栈：是线程私有的，是存放线程栈的内存空间，一个线程对应一个线程栈，一个线程栈里包含了栈帧，每个方法在执行的时候，Java虚拟机都会同步创建一个栈帧用于存储局部变量表，操作数帧，动态链接，方法出口等信息。

本地方法栈：是线程私有的，服务的对象是JVM执行的native方法。

运行时常量池：是方法区的一部分。用于存放编译期生成的各种字面量与符号引用。

直接内存：是jvm定义的空间，跟堆内存是平级的。它不是堆外内存，本身不受gc的影响，但是由于有对象在堆引用这这块内存，那么受到gc的间接影响。相对堆内存，直接内存申请空间更耗时。直接内存IO读写的性能要优于普通的堆内存。

* 为什么堆内存要分年轻代和老年代？

提高垃圾回收效率，合理使用内存空间。对象的寿命有长有短，寿命长的放在一个区，寿命短的放在另一个区。不同的区采用不同的垃圾收集算法。大部分的对象都是朝生夕灭的。

**题目 02- 描述一个 Java 对象的生命周期**

* 解释一个对象的创建过程

检查类是否已经被加载，为对象分配内存空间，将分配到的内存空间初始化零值，为对象进行设置对象头，执行构造方法。

* 解释一个对象的内存分配

1、指针碰撞，新生代的分配方式，内存地址是连续的。

2、空闲列表，老年代的分配方式，内存地址不连续。

* 解释一个对象的销毁过程

当对象不再被引用时，且内存空间需要腾出空间放新对象时，jvm会通过根可达算法标记出哪些对象是垃圾对象，jvm会执行垃圾回收器对垃圾对象进行回收，对象就是这样被销毁的。

* 对象的 2 种访问方式是什么？

1. 句柄访问：对象被移动只要修改句柄中的地址，所以比较稳定。

2. 直接指针访问：这种方式访问速度快，节省了一次指针定位的开销。

* 为什么需要内存担保？

新对象在创建的时候，如果当新生代无法分配内存，会把新生代的老对象转移到老年代，然后把新对象放入腾空的新生代中，这是内存担保机制。为什么需要，是因为新生代采用复制收集算法，假如大量对象在Minor GC后仍然存活，而Survivor空间是比较小的，这时就需要老年代进行分配担保，为新生代腾出空间。

**题目 03- 垃圾收集算法有哪些？垃圾收集器有哪些？他们的特点是什么？**

Serial收集器： 单线程的收集器，收集垃圾时，必须stop the world，使用复制算法。

ParNew收集器： Serial收集器的多线程版本，也需要stop the world，复制算法。

Parallel Scavenge收集器： 新生代收集器，复制算法的收集器，并发的多线程收集器，目标是达到一个可控的吞吐量。Parallel Scavenge 收集器关注点是吞吐量（高效率的利用 CPU）。CMS 等垃圾收集器的关注点更多的是用户线程的停顿时间（提高用户体验）。Parallel Scavenge 收集器提供了很多参数供用户找到最合适的停顿时间或最大吞吐量

Serial Old收集器： 是Serial收集器的老年代版本，单线程收集器，使用标记整理算法。可以作为 CMS 垃圾回收器的备选预案。

Parallel Old收集器： 是Parallel Scavenge收集器的老年代版本，使用多线程，标记-整理算法。

CMS(Concurrent Mark Sweep) 收集器： 是一种以获得最短回收停顿时间为目标的收集器，是并发收集器，标记清除算法，运作过程：初始标记，并发标记，重新标记，并发清除，收集结束会产生大量空间碎片。

G1收集器： 标记整理算法实现，运作流程主要包括以下：初始标记，并发标记，最终标记，筛选标记。一种兼顾吞吐量和停顿时间的 GC 实现，不会产生空间碎片，可以精确地控制停顿。