### JAVA第五阶段—DAY04-JAVA作业答案

1. 简答题：描述一下JVM的结构组成。

答案：

JVM基本可以分成类加载器，内存结构，执行引擎，本地方法接口组成。

其中内存结构又可以分成

java栈、java堆、程序计数器、方法区、本地方法区。

1. 简答题：描述一下java类的生命周期。

答案：

java类的生命周期就是指一个class文件从加载到卸载的全过程。一个java类的完整的生命周期会经历加载、连接、初始化、使用、和卸载五个阶段。其中连接阶段又可以分解为验证、准备、解析三个小阶段。

1. 简答题：java堆和java栈有什么区别？

答案：

Java把内存划分为两种：一种是栈内存，一种是堆内存。在函数中定义的一些基本类型的变量和对象的引用变量都是在函数的栈内存中分配，当在一段代码块定义一个变量时，Java就在栈中为这个变量分配内存空间，当超过变量的作用域后，Java会自动释放掉为该变量分配的内存空间，该内存空间可以立即被另作他用。java栈遵循“先进先出”的规则。

堆内存用来存放由new创建的对象和数组，在堆中分配的内存，由Java虚拟机的自动垃圾回收器来管理。堆内存的使用不遵循进出先后的规则。

1. 简答题：描述一下类加载的双亲委派机制。

答案：

JVM中有三种系统提供的类加载器：启动类加载器，扩展类加载器、应用程序类加载器。

java的三种类加载器存在父子关系，子 加载器保存着附加在其的引用，当一个类加载器需要加载一个目标类时，会先委托父加载器去加载，然后父加载器会在自己的加载路径中搜索目标类，父加载器在自己的加载范围中找不到时，才会交给子加载器加载目标类。

1. 简答题：描述一下jvm垃圾收集的机制。

答案：

jvm的垃圾收集是自动执行的，无需程序员手工操作。

jvm的垃圾标记算法分为2种，引用计数法与可达性分析法：

引用计数法就是给对象中添加一个引用计数器。每当有一个地方引用它，计数器就加 1；当引用失效，计数器就减 1；任何时候计数器为 0 的对象就是不可能再被使用的，可以当做垃圾收集。

可达性分析算法的基本思想就是通过一系列的称为 “GC Roots” 的对象作为起点，从这些节点开始向下搜索，节点所走过的路径称为引用链，当一个对象到 GC Roots 没有任何引用链相连的话，则证明此对象是不可用的。

垃圾回收算法也可以分为3种：

标记-清除（Mark-Sweep） GC分为两个阶段，标记和清除。首先标记所有可回收的对象，在标记完成后统一回收所有被标记的对象。同时会产生不连续的内存碎片。碎片过多会导致以后程序运行时需要分配较大对象时，无法找到足够的连续内存，而不得已再次触发GC。

复制（Copy） 将内存按容量划分为两块，每次只使用其中一块。当这一块内存用完了，就将存活的对象复制到另一块上，然后再把已使用的内存空间一次清理掉。这样使得每次都是对半个内存区回收，也不用考虑内存碎片问题，简单高效。缺点需要两倍的内存空间。

标记-整理（Mark-Compact） 也分为两个阶段，首先标记可回收的对象，再将存活的对象都向一端移动，然后清理掉边界以外的内存。此方法避免标记-清除算法的碎片问题，同时也避免了复制算法的空间问题。 一般年轻代中执行GC后，会有少量的对象存活，就会选用复制算法，只要付出少量的存活对象复制成本就可以完成收集。而老年代中因为对象存活率高，没有额外过多内存空间分配，就需要使用标记-清理或者标记-整理算法来进行回收。

而目前JVM使用的分代处理法，实际是对以上垃圾标记回收算法的综合使用。

根据对象存活周期不同，分为三个年代：年青代、老年代、持久代。这是因为不同对象存活时间不一致，有些可能只使用一次，使用后就需要回收，而有些对象却会伴随整个程序的生命周期。分代有利于堆不同生命周期的对象进行管理，减少GC次数，提高运行效率。

1、年青代

      年青代包含一个eden区，两个survivor区，默认比例为8:1:1。

新创建的对象基本都会存放在EDEN区（大对象直接放在老年代），而这部分对象大部分都会“朝生暮死”，使用后被快速回收。常规一次回收可回收70%-95%的空间，效率非常高。

年青代采用复制算法进行回收，假如当前正在使用的是eden及survivor1区，大部分新生对象都会放在eden区（大对象会直接放在老年代），当eden区内存满了以后，会将存活对象保存到survivor2区，eden进行Minor GC。survivor1区的存活对象根据年龄（默认是15，每经历一轮GC，年龄加1）决定去向，年龄达到阀值，复制到老年代，若年龄未到，复制到survivor2区，survivor1区清空。如果survivor2区内存不足以存放所有的存活对象，则需要以来老年代的担保机制将部分对象复制到老年代。

2、老年代

新生代与老年代默认比例为1：2，老年代用来存放存活时间较长，但还是会死的对象信息（如缓存对象、单例对象等）。老年代对象来源有一下几个方向：

大部分来自于年青代，对象在年青代存活时间过阀值，就会被复制到老年代。

年青代中部分对象虽然为过阀值，但是因为survivor区已满，由担保机制复制到老年代。

部分大对象直接在老年代创建，不经历年青代，如长字符串、长数组等需要大量连续空间的对象。

老年代一版采用标记-整理算法或标记-清除算法，如果采用年青代的复制算法，效率较低。老年代对象存活时间较久，回收频率较低。

3、持久代

保存在方法区，保存类的信息（构造器），常量，静态变量等，一般不进行垃圾回收，但是不是不会进行垃圾回收。