# Day：JVM相关

## Java垃圾回收机制分析

1、JVM栈记录了当前线程的方法调用，栈帧存储有方法参数、局部变量、返回地址。但引用的对象存储在堆中，所以方法调用结束后，对象会依旧存在于堆中，易造成内存泄漏。

2、Java的垃圾回收机制可以自动回收不再使用的对象占用的内存空间。

3、垃圾回收算法

有：引用计数算法、可达性分析算法、标记清除算法、复制算法、标记整理算法、分代回收算法。

a、引用计数算法中，每个对象包含一个计数器，有新的引用指向该对象时计数器加1，引用移除时计数器减1，计数器为0时回收对象，引用计数法有个缺陷是无法检测出循环引用。

b、可达性分析算法中，每个对象都有一个用于标示该对象是否可到达的标记信息。从根出发跟随所有的引用，就可以找到所有的可到达对象，不可到达对象就是需要垃圾回收的对象。

c、标记清除算法流程为，垃圾回收启动时，Java程序暂停运行，JVM从根出发找到所有的可达对象并标记，然后扫描整个堆找到不可达对象并清空。缺点是标记、清除的效率不高；会产生内存碎片。

d、复制算法为，将内存分为大小相等的两块，每次使用一块，使用完时将存活的对象复制到另一块，清空原来那块。在对象存活率较低时回收效率高，缺点是内存空间使用率只有一半。

e、标记整理算法下，找到所有的可达对象并标记后，将所有存活对象都向内存区域的一端移动，然后清理掉边界以外的内存区域。

d-1、分代回收算法下，堆分为三代：永久代、老年代、新生代，新生代又分为三个区域：eden区、from区和to区。

d-3、永久世代主要存放静态文件如Java静态类、方法等，对于垃圾回收没有显著影响。

d-4、分代回收流程为，上次垃圾回收后创建的对象叫新生对象，存放于eden区，eden区没有空间存放新生对象时，触发一次MinorGC，采用标记整理算法将eden区和from区的可达对象复制到to区，清空eden区和from区空间，此时from区和to区已交换；当to区也放不下eden和from区的可达对象时，将部分对象放到成熟世代；当成熟世代也已满时触发MajorGC，采用标记清除算法进行垃圾回收。

d-5、其中，即使to区没有满，JVM也会移动生命周期足够久远的对象到成熟世代；MinorGC发送频率较高，MajorGC发生频率较低。

## Java内存结构

1、堆是用来存储对象本身和数组的，在JVM中只有一个堆，因此堆是被所有线程共享的。

2、栈中存放着栈帧，每个栈帧分别对应一个被调用的方法，栈是线程私有的。

3、方法区存放着已经加载的类信息、常量池、静态变量以及方法代码的内存区域

4、方法的调用过程对应栈帧在虚拟机中入栈到出栈的过程。

5、栈帧中存放着局部变量表、指向运行时常量池的引用、方法返回地址等。