堆空间存储New对象的，但是只有初始化的值。在栈中有实际new出来的对象赋值，地址指向堆中的空间。

GCRoot：栈上面的直接或者间接引用的对象，方法区静态的常量或者方法是不能被删除的，本地方法栈是不能被删除的。

标记删除：标记删除，再标记一遍就删除。删除的两个垃圾的内存不是连续的。

标记整理：标记删除之后，后面的对象前移，整体代价大

复制算法：将内存一分为二，内存快满的时候将将内存中有用的对象进行迁移。没有内存碎片

分带算法：根据Java内存对象存在的周期不同，分为Yong和Old。Yong使用复制算法（清楚大量对象），Old使用标记清理或者标记整理（清理对象少）

堆：Yong （1/3） Old（2/3）

Yong：伊甸园区 Survivol0（幸存区） Survivol1（8：1：1） old（老年代）

实际GC：伊甸园中满使用YongGC，复制算法，将伊甸园区的内容标记删除，存活对象存储到S0，后续GC将S0和伊甸园区中内容标记删除存放到S2，如此往复这个过程。 比直接将内存一分为二利用率更高

OldGC：每一次YongGC，age+1，age=6的时候直接存放到old，很有可能是存在很长时间。还存储大对象（大对象存储很大）。OLD满了的时候回个YONGGC一起，是FULLGC引起stop the word

·Serial：单线程的收集器，收集垃圾时，必须stop the world，使用复制算法。它的最大特点是在进 行垃圾回收时，需要对所有正在执行的线程暂停（stop the world），对于有些应用是难以接受 的，但是如果应用的实时性要求不是那么高，只要停顿的时间控制在N毫秒之内，大多数应用还是。可以接受的，是client级别的默认GC方式。

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

·Parallel Scaverge：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，和ParNew的最大区别是GC自动调节策略；虚拟机会根据系统的运行状态收集性能监控信息，动态设置这些参数，以提供最优停顿时间和最高的吞吐量；

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

·CMS：是一种以获得最短回收停顿时间为目标的收集器，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：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，筛选回收。不 会产生空间碎片，可以精确地控制停顿；G1将整个堆分为大小相等的多个Region（区域），G1跟 踪每个区域的垃圾大小，在后台维护一个优先级列表，每次根据允许的收集时间，优先回收价值最大的区域，已达到在有限时间内获取尽可能高的回收效率；

类加载器：启动类加载器，扩展类加载器，系统类加载器，自定义类加载器



