链滴

Jvm——垃圾回收

作者：ljxlwyq
原文链接：https://ld246.com/article/1663158927713
来源网站：链滴

<img src="https://ld246.com/images/img-loading.svg" alt="" data-src="https://b3logfile
com/bing/20220630.jpg?imageView2/1/w/960/h/540/interlace/1/q/100">
@<a href="https://ld246.com/forward?goto=%25E5%259E%2583%25E5%259C%25BE%
5E5%259B%259E%25E6%2594%25B6" target="_blank" rel="nofollow ugc">toc
<h2 id="一-如何判断垃圾可以回收">一、如何判断垃圾可以回收</h2>
<h2 id="1-引用计数法">1、引用计数法</h2>

当对象被引用计数 +1，当计数为 0 时被垃圾回收
当出现循环引用时该方法无法处理

<img src="https://ld246.com/images/img-loading.svg" alt="image.png" data-src="https:
/b3logfile.com/file/2022/09/image-OFKb91y.png?imageView2/2/interlace/1/format/jpg"></
>
<h2 id="2-可达性分析算法">2、可达性分析算法</h2>

Java 虚拟机中的垃圾回收器采用可达性分析来探索所有存活的对象
扫描堆中的对象，看是否能够沿着 GC Root 对象 为起点的引用链找到该对象，找不到，表示可
回收

<h2 id="3-四种引用">3、四种引用</h2>

强引用

只有所有 GC Roots 对象都不通过【强引用】引用该对象，该对象才能被垃圾回收
<ol start="2">
软引用（SoftReference）

仅有软引用引用该对象时，在垃圾回收后，内存仍不足时会再次出发垃圾回收，回收软引用对象
以配合引用队列来释放软引用自身

<ol start="3">
弱引用（WeakReference）

仅有弱引用引用该对象时，在垃圾回收时，无论内存是否充足，都会回收弱引用对象可以配合引
队列来释放弱引用自身

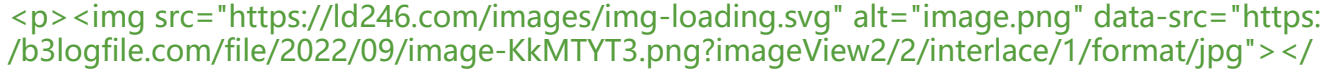
<ol start="4">
虚引用（PhantomReference）

必须配合引用队列使用，主要配合 ByteBuffffer 使用，被引用对象回收时，会将虚引用入队，由
eference Handler 线程调用虚引用相关方法释放直接内存

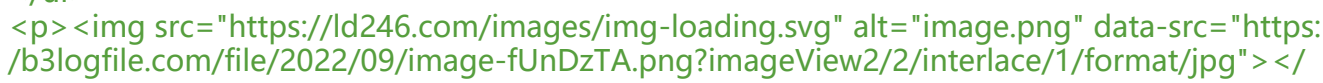
<ol start="5">
终结器引用（FinalReference）

无需手动编码，但其内部配合引用队列使用，在垃圾回收时，终结器引用入队（被引用对象暂时
有被回收），再由 Finalizer 线程通过终结器引用找到被引用对象并调用它的 fifinalize 方法，第二次
C 时才能回收被引用对象

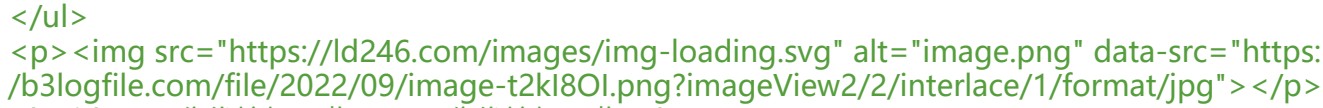
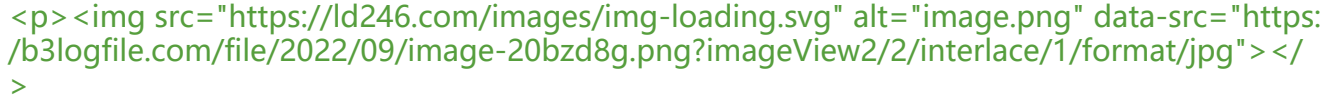
<h2 id="二-垃圾回收算法">二、垃圾回收算法</h2>

<h2 id="1-标记清除">1、标记清除</h2>
定义： Mark Sweep
优点：速度快
缺点：空间不连续

<h2 id="2-标记整理">2、标记整理</h2>

定义：Mark Compact
优点：空间连续
缺点：速度慢


<h2 id="3-复制">3、复制</h2>

定义：Copy
优点：不会有内存碎片
缺点：需要占用双倍内存空间


<h2 id="三-分代垃圾回收">三、分代垃圾回收</h2>


对象首先分配在伊甸园区域
新生代空间不足时，触发 minor gc，伊甸园和 from 存活的对象使用 copy 复制到 to 中，存活对象年龄加 1 并且交换 from to
minor gc 会引发 stop the world，暂停其它用户的线程，等垃圾回收结束，用户线程才恢复运
当对象寿命超过阈值时，会晋升至老年代，最大寿命是 15（4bit）
当老年代空间不足，会先尝试触发 minor gc，如果之后空间仍不足，那么触发 full gc，STW 的间更长

<h2 id="1-相关-VM-参数">1、相关 VM 参数</h2>
| ------ | ------ |
| 堆初始大小 | -Xms |
| 堆最大大小 | -Xmx 或 -XX:MaxHeapSize=size |
| 新生代大小 | -Xmn 或 (-XX:NewSize=size + -XX:MaxNewSize=size) |
| 幸存区比例（动态） | -XX:InitialSurvivorRatio=ratio 和 -XX:+UseAdaptiveSizePolicy |
| 幸存区比例 | -XX:SurvivorRatio=ratio |
| 晋升阈值 | -XX:MaxTenuringThreshold=threshold |
| 晋升详情 | -XX:+PrintTenuringDistribution |
| GC 详情 | -XX:+PrintGCDetails -verbose:gc |
| FullGC 前 MinorGC | -XX:+ScavengeBeforeFullGC |