

java初学者实践教程14 - 垃圾收集器 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/269/2021_2022_java_E5_88_9D_E5_AD_A6_c67_269940.htm 用过C编程的人知道，编的时候总是要跟踪所创建的对象，并且需要显示地删除不用的对象。这种方式太麻烦了，容易出错。写了那么多代码，能记住吗，要是把有用的给删了怎么办，要是有没有用的忘删了怎么办？这些问题是很严重的。在java语言中采用的垃圾收集器这种方式管理内存，就很方便也很安全了。垃圾收集器，可以自动确定哪个对象不再被利用，它可以自动将它删除。这也是java语言的一大优势。我们要想显示的删除一个对象的引用也很简单，将该引用的变量赋值为null不就行了吗？对于垃圾收集器来说，当程序员创建对象时，垃圾收集器就开始监控这个对象的地址、大小以及使用情况。通常，垃圾收集器采用有向图的方式记录和管理堆（heap）中的所有对象。通过这种方式确定哪些对象是“可用的”，哪些对象是“不可用的”。当垃圾收集器确定一些对象为“不可用”时，垃圾收集器就回收这些内存空间。可是垃圾收集器却以较低的优先级在系统空闲周期中执行，通俗一点说就是它级别低，别人不运行时候才轮到它，因此垃圾收集器的速度比较慢。有些时候我们会使用System.gc（）。手动回收。这样来提高性能。对于垃圾收集器来说还有一个值得一提的是finalize（）这个方法，每一个对象都有一个finalize（）方法，这个方法是从Object类继承来的。它用来回收内存以外的系统资源，就像是文件处理器和网络连接器。该方法的调用顺序和用来调用该方法的对象的创建顺序是无关的。换句话说，书写程

序时该方法的顺序和方法的实际调用顺序是不相干的。这只是finalize () 方法的特点。还有，每个对象只能调用finalize () 方法一次。如果在finalize () 方法执行时产生异常 (exception) ，则该对象仍可以被垃圾收集器收集。那是一定了，不能说用到finalize () 了。垃圾收集器就什么也不做了啊。finalize () 的工作量是很大的哦 总结：Java用了垃圾收集器的内存管理方式，并不是说它完全的好。有的时候会影响它的性能，我们还是要手动来收集的。但是要是像C那样完全手动来收集的话，那也实在是太麻烦了而且不是很安全。根据垃圾收集器的工作原理，我们可以通过一些技巧和方式，让垃圾收集器运行更加有效率。

- 1.最基本的建议就是尽早释放无用对象的引用。大多数程序员在使用临时变量的时候，都是让引用变量在退出活动域 (scope) 后，自动设置为null.
- 2.尽量少用finalize函数。finalize函数是Java提供给程序员一个释放对象或资源的机会。但是，它会加大垃圾收集器的工作量，因此尽量少采用finalize方式回收资源。
- 3.当程序有一定的等待时间，程序员可以手动执行System.gc () ，通知垃圾收集器运行，但是Java语言规范并不保证垃圾收集器一定会执行。

100Test 下载频道开通，各类考试题目直接下载。
详细请访问 www.100test.com