

Java内存泄漏分析与解决方案Java认证考试 PDF转换可能丢失
图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/644/2021_2022_Java_E5_86_85_E5_AD_98_c104_644893.htm Java内存泄漏是每个Java程序员都会遇到的问题，程序在本地运行一切正常，可是布署到远端就会出现内存无限制的增长，最后系统瘫痪，那么如何最快最好的检测程序的稳定性，防止系统崩盘，作者用自己的亲身经历与各位网友分享解决这些问题的办法。作为Internet最流行的编程语言之一,Java现正非常流行。我们的网络应用程序就主要采用Java语言开发，大体上分为客户端、服务器和数据库三个层次。在进入测试过程中，我们发现有一个程序模块系统内存和CPU资源消耗急剧增加，持续增长到出现java.lang.OutOfMemoryError为止。经过分析Java内存泄漏是破坏系统的主要因素。这里与大家分享我们在开发过程中遇到的Java内存泄漏的检测和处理解决过程. 一. Java是如何管理内存 为了判断Java中是否有内存泄露，我们首先必须了解Java是如何管理内存的。Java的内存管理就是对象的分配和释放问题。在Java中，内存的分配是由程序完成的，而内存的释放是由垃圾收集器(Garbage Collection，GC)完成的，程序员不需要通过调用函数来释放内存，但它只能回收无用并且不再被其它对象引用的那些对象所占用的空间。Java的内存垃圾回收机制是从程序的主要运行对象开始检查引用链，当遍历一遍后发现没有被引用的孤立对象就作为垃圾回收。GC为了能够正确释放对象，必须监控每一个对象的运行状态，包括对象的申请、引用、被引用、赋值等，GC都需要进行监控。监视对象状态是为了更加准确地、及时地释放对象，而释放对

象的根本原则就是该对象不再被引用。在Java中，这些无用的对象都由GC负责回收，因此程序员不需要考虑这部分的内存泄露。虽然，我们有几个函数可以访问GC，例如运行GC的函数System.gc()，但是根据Java语言规范定义，该函数不保证JVM的垃圾收集器一定会执行。因为不同的JVM实现者可能使用不同的算法管理GC。通常GC的线程的优先级别较低。JVM调用GC的策略也有很多种，有的是内存使用到达一定程度时，GC才开始工作，也有定时执行的，有的是平缓执行GC，有的是中断式执行GC。但通常来说，我们不需要关心这些。

二. 什么是Java中的内存泄露 导致内存泄漏主要的原因是，先前申请了内存空间而忘记了释放。如果程序中存在对无用对象的引用，那么这些对象就会驻留内存，消耗内存，因为无法让垃圾回收器GC验证这些对象是否不再需要。如果存在对象的引用，这个对象就被定义为"有效的活动"，同时不会被释放。要确定对象所占内存将被回收，我们就要务必确认该对象不再会被使用。典型的做法就是把对象数据成员设为null或者从集合中移除该对象。但当局部变量不需要时，不需明显的设为null，因为一个方法执行完毕时，这些引用会自动被清理。

在Java中，内存泄漏就是存在一些被分配的对象，这些对象有下面两个特点，首先，这些对象是有被引用的，即在有向树形图中，存在树枝通路可以与其相连；其次，这些对象是无用的，即程序以后不会再使用这些对象。如果对象满足这两个条件，这些对象就可以判定为Java中的内存泄漏，这些对象不会被GC所回收，然而它却占用内存。这里引用一个常看到的例子，在下面的代码中，循环申请Object对象，并将所申请的对象放入一个Vector中，如果仅仅释放

对象本身，但因为Vector仍然引用该对象，所以这个对象对GC来说是不可回收的。因此，如果对象加入到Vector后，还必须从Vector中删除，最简单的方法就是将Vector对象设置为null。 Vector v = new Vector(10).for (int i = 1.i 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