

Java内存泄露问题分析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_644797.htm 很多人在谈论内存泄露问题，当然对于c/c++来说，这个应该是老掉牙的问题，但是很多Java人员也越来越多得讨论这个问题，我这里写个小结，希望对大家有一定的参考价值。

内存泄漏的概念

- 1.c/c++是程序员自己治理内存，Java内存是由GC自动回收的。我虽然不是很熟悉C，不过这个应该没有犯常识性错误吧。
- 2.什么是内存泄露? 内存泄露是指系统中存在无法回收的内存，有时候会造成内存不足或系统崩溃。在C/C++中分配了内存不释放的情况就是内存泄露。
- 3.Java存在内存泄露 我们必须先承认这个，才可以接着讨论。虽然Java存在内存泄露，但是基本上不用很关心它，非凡是那些对代码本身就不讲究的就更不要去关心这个了。Java中的内存泄露当然是指：存在无用但是垃圾回收器无法回收的对象。而且即使有内存泄露问题存在，也不一定会表现出来。
- 4.Java中参数都是传值的。对于基本类型，大家基本上没有异议，但是对于引用类型我们也不能有异议。

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，程序员需要通过关键字new为每个对象申请内存空间(基本类型除外)，所有的对象都在堆(Heap)中分配空间。另外，对象的释放是由GC决定和执行的。在Java中，内存的分配是由程序完成的，而内存的释放是有GC完成的，这种收支两条线的方法确实简化了程序员的工作。但同时，它也加重了JVM的工作。这也是Java程序运行速

度较慢的原因之一。因为，GC为了能够正确释放对象，GC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。为了更好地理解GC的工作原理，我们可以将对象考虑为有向图的顶点，将引用关系考虑为图的有向边，有向边从引用者指向被引对象。另外，每个线程对象可以作为一个图的起始顶点，例如大多程序从main进程开始执行，那么该图就是以main进程顶点开始的一棵根树。在这个有向图中，根顶点可达的对象都是有效对象，GC将不回收这些对象。如果某个对象(连通子图)与这个根顶点不可达(注意，该图为有向图)，那么我们认为这个(这些)对象不再被引用，可以被GC回收。以下，我们举一个例子说明如何用有向图表示内存管理。对于程序的每一个时刻，我们都有一个有向图表示JVM的内存分配情况。以下右图，就是左边程序运行到第6行的示意图。Java使用有向图的方式进行内存管理，可以消除引用循环的问题，例如有三个对象，相互引用，只要它们和根进程不可达的，那么GC也是可以回收它们的。这种方式的优点是管理内存的精度很高，但是效率较低。另外一种常用的内存管理技术是使用计数器，例如COM模型采用计数器方式管理构件，它与有向图相比，精度行低(很难处理循环引用的问题)，但执行效率很高。Java内存泄露情况 JVM回收算法是很复杂的，我也不知道他们怎么实现的，但是我只知道他们要实现的就是：对于没有被引用的对象是可以回收的。所以你要造成内存泄露就要做到：持有对无用对象的引用!不要以为这个很轻易做到，既然无用，你怎么还会持有它的

引用? 既然你还持有它, 它怎么会是无用的呢? 我实在想不到比那个堆栈更经典的例子了, 以致于我还要引用别人的例子, 下面的例子不是我想到的, 是书上看到的, 当然假如没有在书上看到, 可能过一段时间我自己也想的到, 可是那时我说是我自己想到的也没有人相信的。

```
public class Stack { private Object[] elements=new Object[10]. private int size = 0. public void push(Object e){ ensureCapacity(). elements[size ] = e. }
```

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