

【第19话：当年选择学习Java就是因为Java有GC，现在面试次次被问到】

这节课给小伙伴们讲解的是GC是什么，以及Java如何判断对象可以进行回收和常见的GC算法有哪些。

这个面试题可能出现在面试笔试部分，也可能出现在面试中面试官直接问的部分。

1.GC是什么

GC 全称 garbage collection，翻译过来叫做垃圾回收器。

GC的主要作用就是回收JVM中不再使用的对象所占内存。

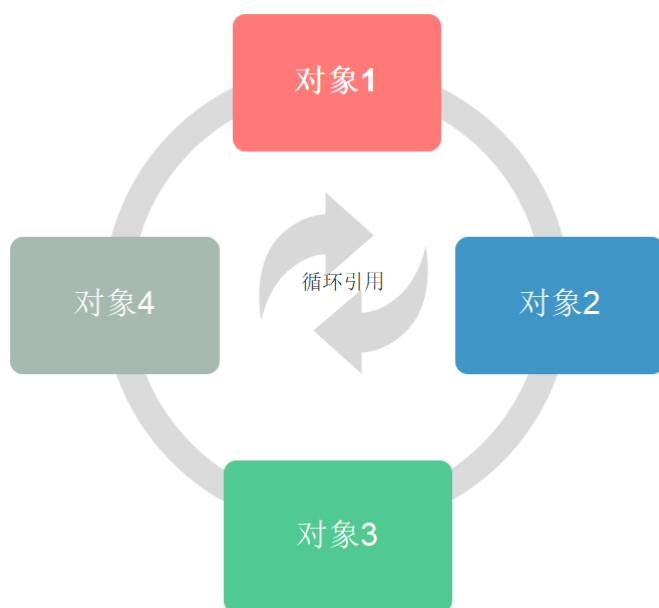
JVM运行时堆内存和方法区中数据。而虚拟机栈、程序计数器、本地方法栈都是根据线程创建而创建，随着线程销毁而销毁，所以不需要进行回收。



2. GC如何判断对象是否可以回收

2.1 引用计数（已淘汰）

引用计数算法就是看对象是否被引用。如果引用则对象计数器加一。如果释放引用计数器减一。但是引用计数算法最大的问题就是循环引用问题。当出现循环引用时对象计数器至少为1.这时候对象可能已经是垃圾了，但是无法被回收。



1. 例如:

```
public class Test {  
    public static void main(String[] args) {  
        MyObject object1 = new MyObject();  
        MyObject object2 = new MyObject();  
  
        object1.object = object2;  
        object2.object = object1;  
  
        object1 = null;  
        object2 = null;  
    }  
}  
  
class MyObject{  
    public Object object = null;  
}
```

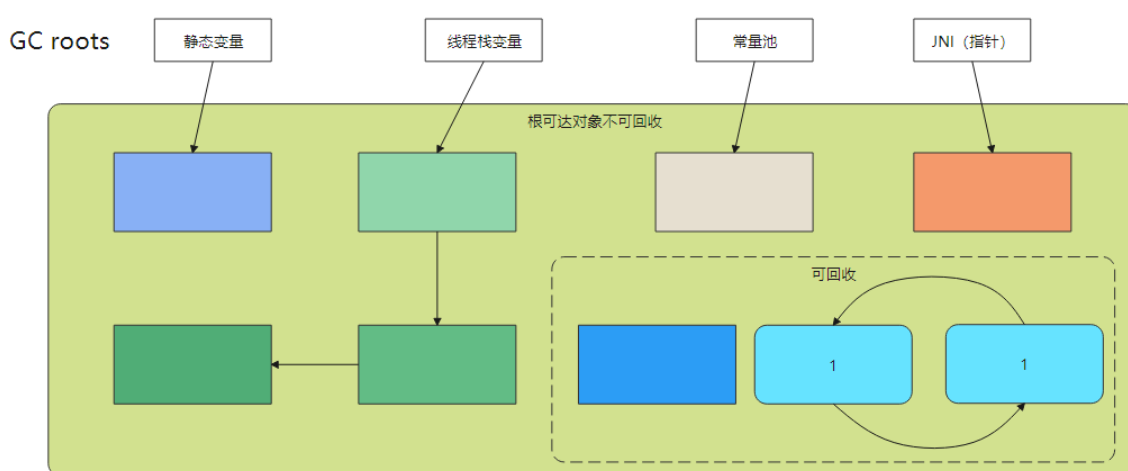
最后面两句将object1和object2赋值为null，也就是说object1和object2指向的对象已经不可能再被访问，但是由于它们互相引用对方，导致它们的引用计数都不为0，那么垃圾收集器就永远不会回收它们。

2.2 可达性分析：

根可达算法没有引用计数算法中循环引用无法被回收的问题。

其主要思路是通过一系列名为GC Roots的对象作为根，从根开始往下搜索，搜索过程经过的路径称为引用链（Reference Chain），当一个对象到达GC Root时表示当前对象还在使用，如果没有引用的或者和其他非GC Roots循环引用的内容都是垃圾。静态变量、线程变量、常量池、JNI（指针）都是GC Roots

JVM内存对象的根可达算法



3.GC如何判断对象是否可以进行回收

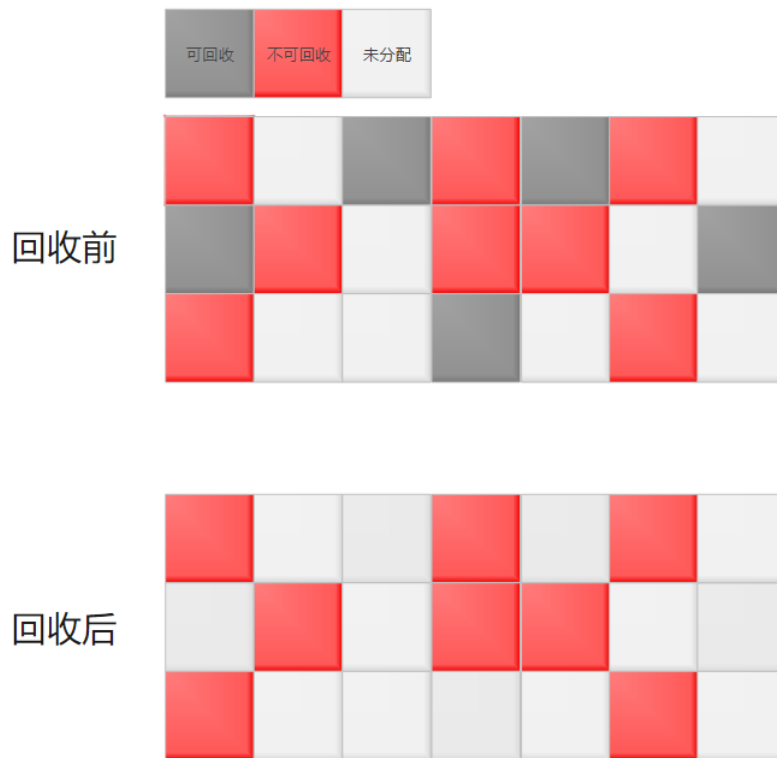
1.标记清除算法（Mark-sweep）

首先标记出所有需要回收的对象。标记完成后统一回收所有标记的对象。

缺点：

效率低。碎片多。

JVM垃圾回收标记清除算法



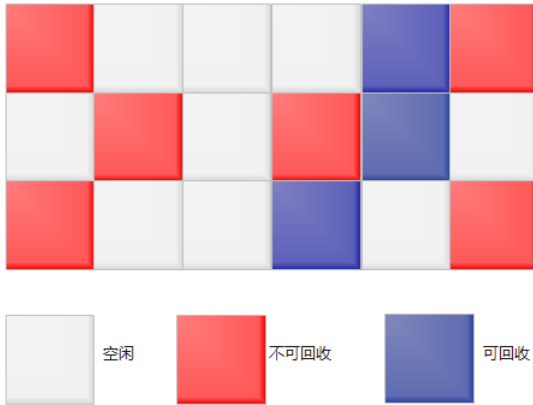
2. 标记压缩算法 (Mark-Compact)

又叫标记整理算法。

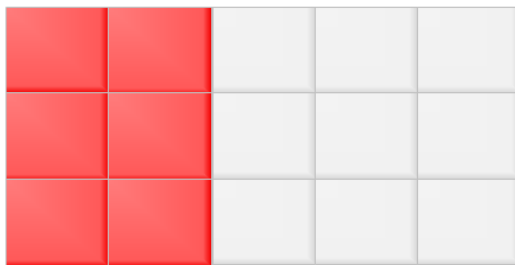
和标记清除算法有点类似。主要区别是标记完成后并不会直接清除，而是把所有不回收对象先向一端移动，然后在清除掉边界外面的对象。这样就不会产生内存碎片。

标记压缩算法

回收前



回收后



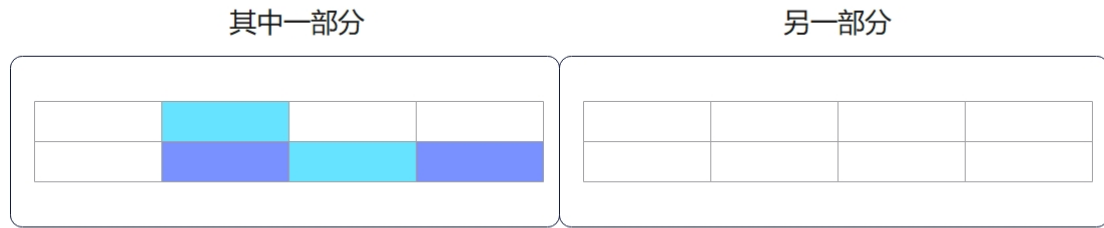
3.复制算法 (copying)

目的：主要是为了解决标记清除算法碎片问题。

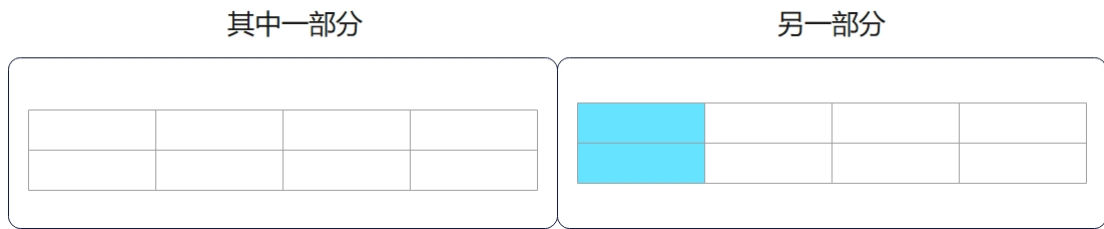
步骤：内存按照容量分为大小相等的两块。每次只使用一块。当一块使用完成后，把存活的对象复制到另一个空间，然后把空间一次清除掉。

缺点：可用内存减少。

回收前



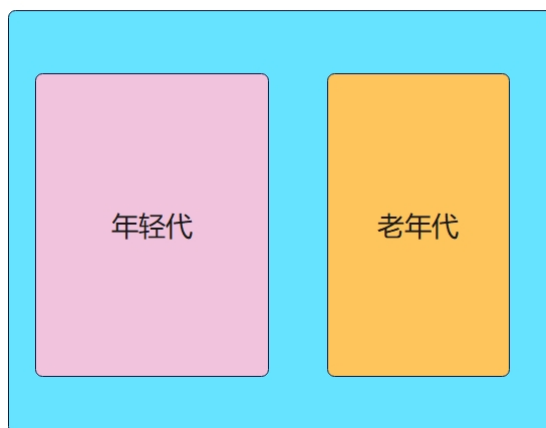
复制后



4.分代收集算法

把堆分为新生代和老年代。新生代采用一种算法，老年代采用另一种算法。具体新生代和老年代采用的算法需要看使用哪种垃圾回收器。

堆内存



GC中几个问题的回答示范

