当对象不被引用的时候，这个对象就是死亡的，等待GC进行回收。

## ****一. 引用计数算法：****

**1. 概念：**

　　给对象中添加一个引用计数器，每当有一个地方引用它时，计数器值就加1；当引用失效时，计数器值就减1；任何时刻计数器为0的对象就是不可能再被使用的。

**2. 但是：**

　　主流的java虚拟机并没有选用引用计数算法来管理内存，其中最主要的原因是：**它很难解决对象之间相互循环引用的问题**。

**3. 优点：**

　　算法的实现简单，判定效率也高，大部分情况下是一个不错的算法。很多地方应用到它

**4. 缺点：**

引用和去引用伴随加法和减法，影响性能

致命的缺陷：对于**循环引用的对象**无法进行回收

## ****二. 根搜索算法：****（jvm采用的算法）

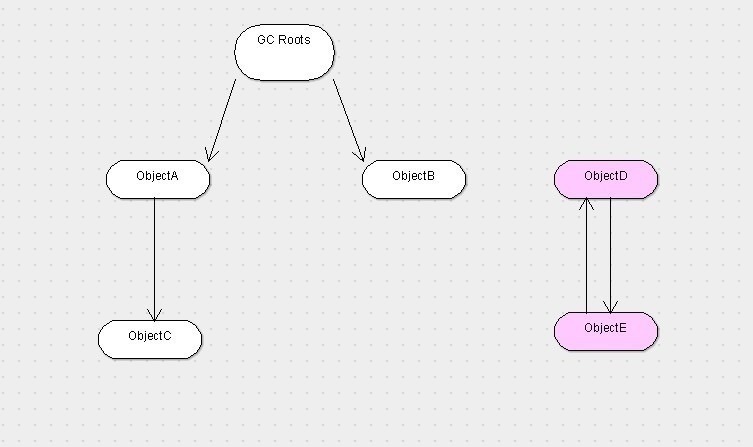
**1. 概念：**

　　设立若干种根对象，当任何一个根对象（GC Root）到某一个对象均不可达时，则认为这个对象是可以被回收的。

**注：**这里提到，设立若干种根对象，当任何一个根对象到某一个对象均不可达时，则认为这个对象是可以被回收的。我们在后面介绍标记-清理算法/标记整理算法时，也会一直强调从根节点开始，对所有可达对象做一次标记，那什么叫做可达呢？

**2. 可达性分析：**

　　从根（GC Roots）的对象作为起始点，开始向下搜索，搜索所走过的路径称为“**引用链**”，当一个对象到GC Roots没有任何引用链相连（用图论的概念来讲，就是从GC Roots到这个对象不可达）时，则证明此对象是不可用的。

[](http://images2015.cnblogs.com/blog/641601/201509/641601-20150915141611679-1989614075.jpg)

如上图所示，ObjectD和ObjectE是互相关联的，但是由于GC roots到这两个对象不可达，所以最终D和E还是会被当做GC的对象，上图若是采用引用计数法，则A-E五个对象都不会被回收。

**3. 根（GC Roots）：**

说到GC roots（GC根），在JAVA语言中，可以当做GC roots的对象有以下几种：

（1）**栈**（栈帧中的本地变量表）**中引用的对象**。

（2）方法区中的静态成员。

（3）方法区中的常量引用的对象（全局变量）

（4）本地方法栈中JNI（一般说的Native方法）引用的对象。

注：第一和第四种都是指的方法的本地变量表，第二种表达的意思比较清晰，第三种主要指的是声明为final的常量值。