

1. 栈：存放基本类型的数据和对象的引用，但对象本身不存放在栈中，而是存放在堆中
2. 堆：存放用new产生的数据，new出的实例
3. 静态域：存放在对象中用static定义的静态成员
4. 常量池：存放常量

## Java内存分配中的栈

在函数中定义的一些基本类型的变量数据和对象的引用变量都在函数的栈内存中分配。当在一段代码块定义一个变量时，Java就在栈中为这个变量分配内存空间，当该变量退出该作用域后，Java会自动释放掉为该变量所分配的内存空间，该内存空间可以立即被另作他用。

## Java内存分配中的堆

堆内存用来存放由new创建的对象和数组。在堆中分配的内存，由Java虚拟机的自动垃圾回收器来管理。

实际上，栈中的变量指向堆内存中的变量，这就是Java中的指针！

## 堆与栈

Java的堆是一个运行时数据区，类的（对象从中分配空间。这些对象通过new、newarray、anewarray和multianewarray等指令建立，它们不需要程序代码来显式的释放。堆是由垃圾回收来负责的，堆的优势是可以动态地分配内存大小，生存期也不必事先告诉编译器，因为它是在运行时动态分配内存的，Java的垃圾收集器会自动收走这些不再使用的数据。但缺点是，由于要在运行时动态分配内存，存取速度较慢。

栈的优势是，存取速度比堆要快，仅次于寄存器，栈数据可以共享。但缺点是，存在栈中的数据大小与生存期必须是确定的，缺乏灵活性。栈中主要存放一些基本类型的变量数据（int, short, long, byte, float, double, boolean, char）和对象句柄（引用）。

## 二、如何定位内存泄漏的问题

要定位问题，首先你需要获取服务器jvm某刻内存快照。jdk自带的jmap可以获取内存某一时刻的快照，导出为dmp文件后，就可以用Eclipse MAT来分析了，找出是那个对象使用内存过多。

