深入JVM垃圾回收

# JVM执行分析

## Java运行流程

所有的Java程序代码必须保存在\*.java的文件之中，这些称为源代码。而这些源代码并不能够直接执行，必须使用javac.exe命令将其编译为\*.class文件，而后利用java.exe命令在JVM进程之中解释此程序。

实际上JVM将所需要的\*.class文件加载到JVM进程之中，那么这个过程就需要有一个类加载器（ClassLoader），有了类加载器的好处在于：可以随意指定程序\*.class文件的所在路径。

JVM：Java虚拟机，所有的程序都要求运行在JVM上，是因为考虑到了可移植性问题，可是如果要想真正去执行程序，你绝对不可能离开操作系统的支持。

在Java里面可以使用native实现本地C函数的调用。但是这些都属于程序运行的辅助手段，而真正的程序运行都在“运行时数据区”之中。

在整个的运行实数区之中，分为如下几块内存空间：

* 堆内存：保存所有引用数据类型的真实信息；
* 栈内存：基本类型、运算、指向堆内存的指针；
* 方法区：所有定义的方法的信息都保存在方法区之中，此区属于共享区；
* 程序计数器：是一个非常小的内存空间，小得可以忽略；
* 本地方法栈：每一次执行递归的方法处理的时候实际上都会将上一个方法入栈；

**Java虚拟机栈（Java Virtual Machine Stacks）**

* 栈内存是线程私有的，其生命周期和线程相同；
* 虚拟机栈描述的是Java方法执行的内存模型：

# 模板

## 模板