



链滴

# JVM 运行时数据区

作者: [AlanSune](#)

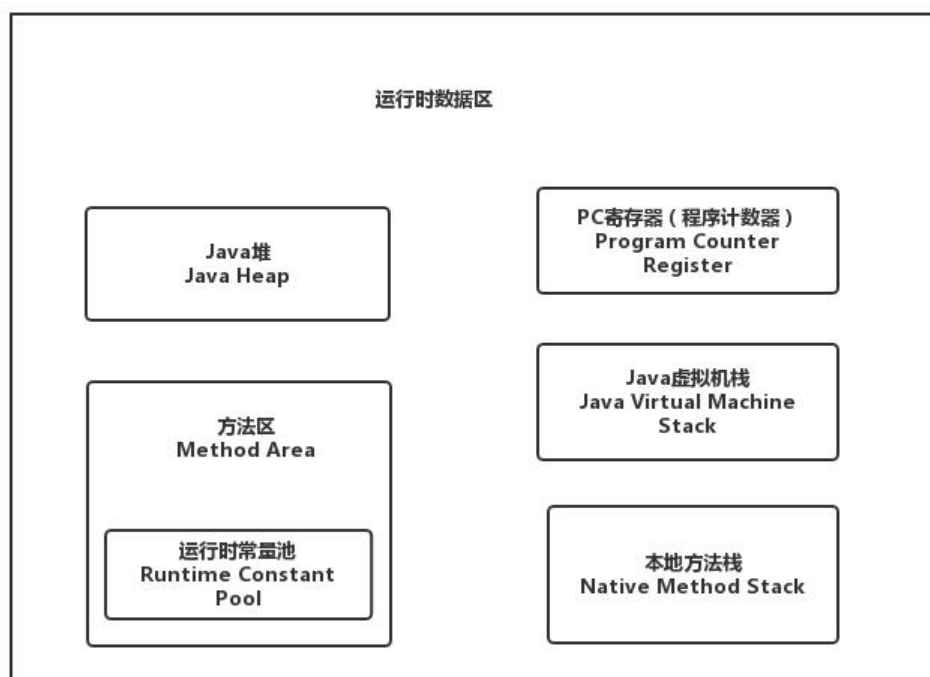
原文链接: <https://ld246.com/article/1591524422936>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)



Java虚拟机在运行Java程序的过程中，会将管理的内存分为若干区域，每个区域拥有各自的用途。Java运行时数据区可以由下图表示：



运行时数据区有一些会随着虚拟机的创建而创建，随着虚拟机的退出而销毁。另外一些则与线程对应，随着线程的开始和结束而创建和销毁。上图中左侧的Java堆和方法区与虚拟机对应，而右侧的程序计数器、虚拟机栈和本地方法区与线程对应。

## Java堆

java堆在虚拟机启动时即被创建，是供各个线程共享的运行时内存区域，也是所有实例对象和数组对分配内存的区域。它在虚拟机启动时创建，存储了自动内存管理系统(也就是Garbage Collector 垃圾集器)所管理的对象，所使用的内存不要求连续，容量也可以动态扩展。如果实际所需内存超过GC所提供的最大容量，那么虚拟机将会抛出OutOfMemoryError异常。

## 方法区

方法区也是可供各个线程共享的运行时内存区域，它存储了每一个类的结构信息，如运行时常量池、段数据、方法数据、构造方法和普通方法的字节码内容。它在虚拟机启动时创建，实际内存可以不连。虚拟机可以选择不进行垃圾收集，如果方法区的内存空间不能满足内存分配请求，那么虚拟机将抛出一个OutOfMemoryError异常。

## 运行时常量池

运行时常量池是class文件中每一个类或接口的常量池表的运行时表示形式，具备动态性，既包含编译可知的常量，也包含运行期解析的常量。每一个运行时常量池都在Java虚拟机的方法区中分配，在加类或接口到虚拟机时，就创建对应的运行时常量池。如果创建时需要的内存超过方法区所能提供的最大值，那么虚拟机将抛出一个OutOfMemoryError异常。

## 程序计数器

Java虚拟机支持多线程，每个线程都拥有自己的程序计数器。程序计数器可以理解为保存正在执行的字节码指令的地址，通过修改该值来选取下一条需要执行的指令。在任意时刻，一条Java虚拟机线程只执行一个方法，该方法称为当前方法(Current Method)，如果这个方法不是本地方法(Native Method)，那么程序计数器就保存正在执行的指令地址，否则为undefined。

## Java虚拟机栈

虚拟机栈和程序计数器一样，是线程私有的，生命周期与线程相同。它与线程同时创建，用于存储栈帧(Stack Frame)，该结构包含局部变量表、操作数栈、动态链接、方法出口等信息。局部变量表存放编译器可知的各种**基本类型**、对象引用和returnAddress类型。如果线程请求分配的栈容量超过虚拟机栈容器的最大容量，虚拟机将抛出一个StackOverflowError异常；如果虚拟机栈可以动态扩展，在扩展时无法申请到足够的内存，将抛出OutOfMemoryError异常。

## 本地方法栈

本地方法栈与虚拟机栈相似，虚拟机栈为虚拟机执行Java方法提供服务，而本地方法栈为虚拟机使用本地方法服务。同样，本地方法栈也会抛出StackOverflowError和OutOfMemoryError异常。

本文参考《深入理解Java虚拟机(第二版)》和《Java虚拟机规范(Java SE 8版)》，只是对Java运行时数据区进行了简单的介绍，其中的细节还未进行展开，如栈帧的结构、Java堆的细分等。在对Java内存有了总体认识后，对后面的学习可以有一定的结构支撑。

## 补充

### 虚拟机配置的参数：

- -Xms Java堆的最小值，也是初始化值
- -Xmx Java堆的最大值，当-Xms和-Xmx参数相同时，Java堆不可扩展

- -Xss 每个虚拟机栈的容量
- -Xoss 每个本地方法栈的容量，HotSpot虚拟机不区分虚拟机栈和本地方法栈，故该参数无效
- -XX:PermSize 持久代初始容量
- -XX:MaxPermSize 持久代最大容量

## jdk6和jdk7下String#intern方法的区别：

jdk6中调用String#intern方法，虚拟机首先会在字符串常量池中查找该字符串，找到后返回常量池中地址，没有查找到会将字符串复制到常量池，然后返回地址。

jdk7中调用String#intern方法，虚拟机首先会在字符串常量池中查找该字符串，找到后返回常量池中地址，没有查找到会将字符串的\*\*地址\*\*保存到常量池，然后返回保存的地址。