理解 JVM 笔记系列（二）：内存模型 (附脑图)

作者：hudk
原文链接：https://ld246.com/article/1590577990315
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)

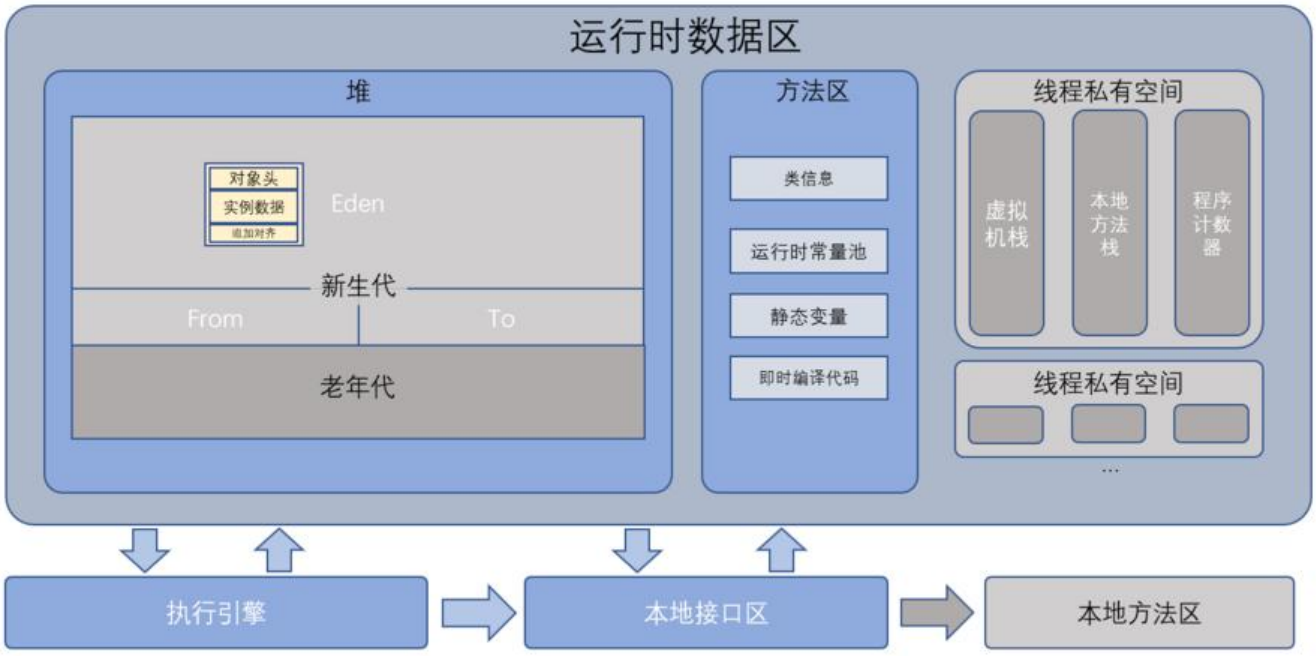
概述

由于Java程序运行过程中，对内存的管理和控制，全权交给了Java虚拟机来管理，虽然可以避免绝大数的内存溢出问题，但也不是绝对的，特殊的代码编写也会让程序出现一定程度上的内存溢出情况，旦出现这种问题，程序员不了解虚拟机的内存模型和管理方式，要找寻和解决问题，将会异常的困难所以需要了解Java虚拟机的内存模型。

总体上虚拟机的运行时数据区域结构分为线程私有和公共区域：

- 线程私有
  - 程序计数器
  - 虚拟机栈
  - 本地方法栈
- 公共区域
  - 堆
  - 方法区

以上全部是逻辑上的概念，具体的虚拟机实现，在物理内存如何映射，这里先不做考虑。主要是认识们在概念上是什么含义，以及在虚拟机规范中，他们分别应该承担什么样的“职责”。



程序计数器

这是一个在虚拟机中空间占用较小的一块区域，它专门用来保存当前线程执行的下一条字节码指令“号”，这样程序运行时，才知道下一条指令应该去哪里找到并运行。当然从多线程的角度考虑，由于序计数器是线程私有的，在线程切换过程中，比如线程等待或堵塞被唤醒后，可以依据程序计数器中内容，继续执行刚才将要执行的指令。当正在执行本地方法时，这个计数器的值应该为空；另外，此存区域是规范中唯一没有规定OutOfMemoryError情况的区域。

Java虚拟机栈、

在线程运行到每一个新方法时，虚拟机栈都会创建一个新的栈帧作为栈顶，每一个栈帧中，包含了当执行方法的局部变量，操作数栈，方法引用（动态链接），方法的出口这些信息，当前方法运行结束，该栈帧随之被弹出（删除）。

栈帧结构图：

本地方法栈

与虚拟机栈很类似，不同的是，本地方法栈是提供虚拟机调用本地方法时专门为运行本地方法而准备内存区域。

Java 堆

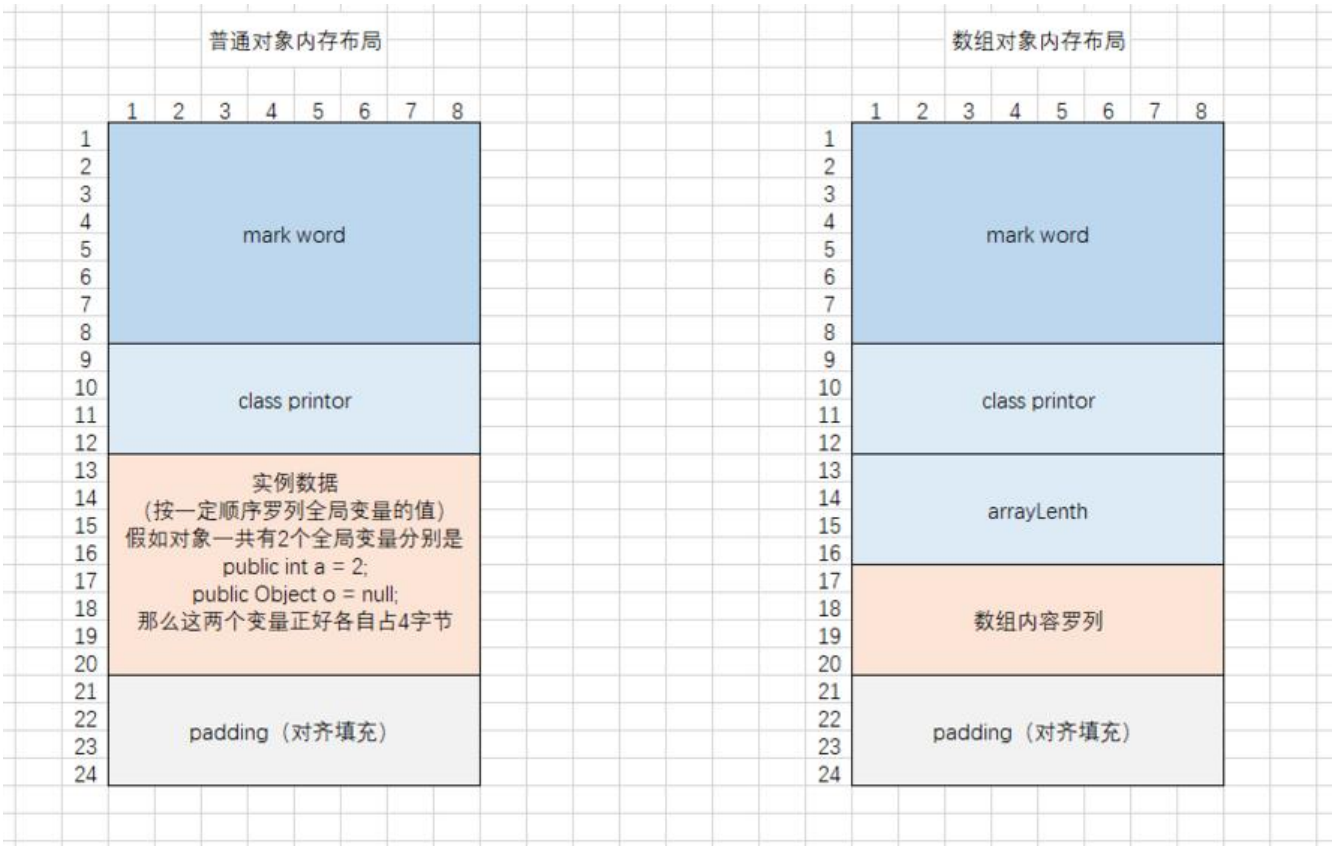
这是Java 虚拟机内存管理的最大的一块区域，这块区域的主要职责是存放Java世界中几乎所有的对象例数据，是虚拟机中所有线程可以共同享有访问权力的内存区域。并且这块内存区域是虚拟机垃圾收器管理的区域，也被成为“GC堆”。从垃圾收集的角度来考虑，Java堆有可以分为以下更细粒度的区域。

- 新生代
  - Eden
  - From Survivor
  - To Survivor

- 老年代
- 永久代

这样来划分堆，单纯是为了垃圾回收算法来服务的，换句话说，高效的垃圾回收算法，决定了堆需要何种方式来存储对象实例。从数据结构角度考虑，这样划分堆，本质上就是为垃圾回收算法提供更高的数据结构。

对象在堆中的结构：



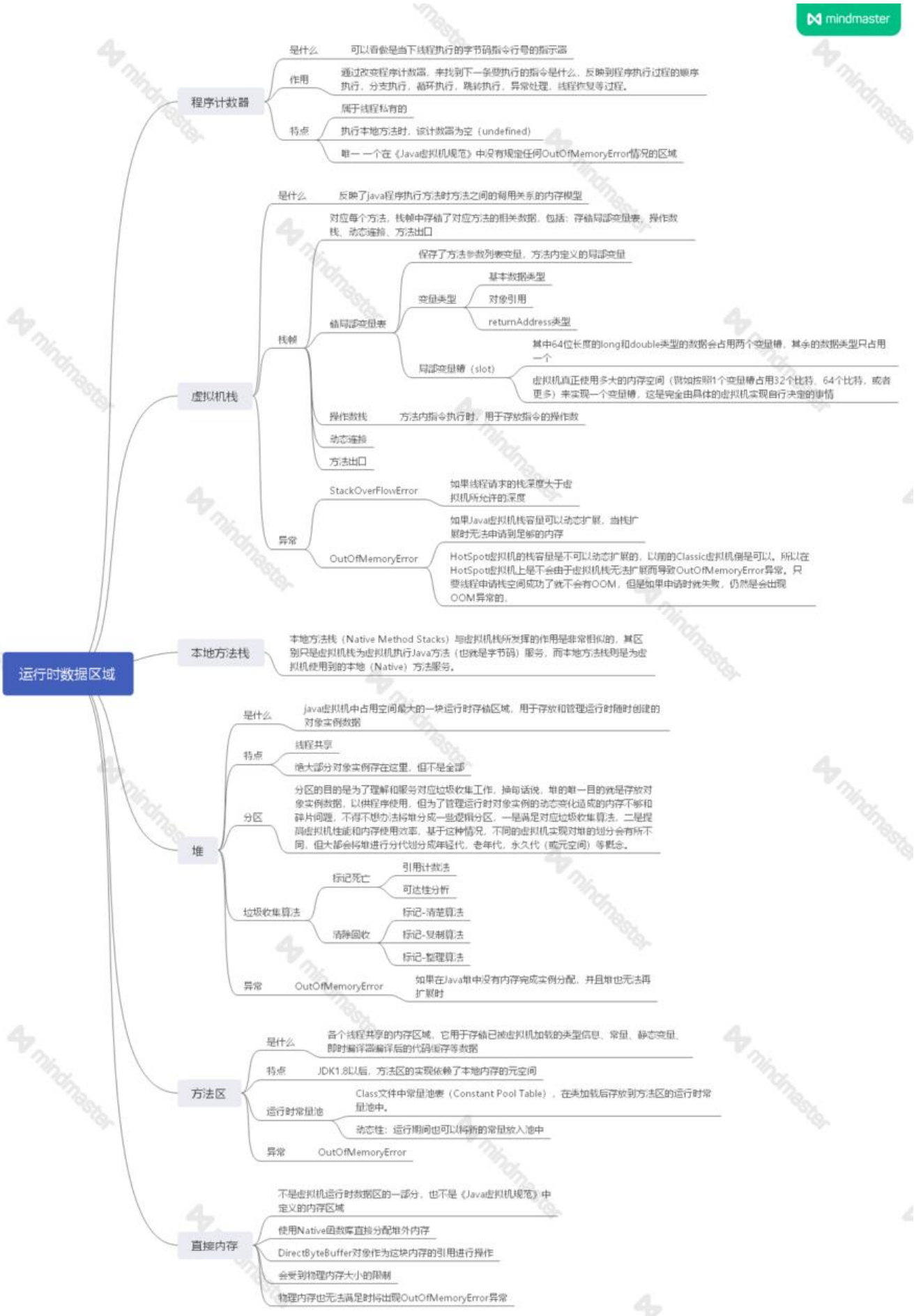
方法区

方法区也是虚拟机中的公共存储区域，主要保存了被虚拟机加载的类型信息，常量，静态变量，即时译后的代码缓存等。

运行时常量池

这是属于方法区的一部分，主要用来保存类型信息中的常量信息。并且支持将运行时产生的新常量加其中。

根据《深入理解Java虚拟机》第3版相关章节总结整理的脑图



原文链接：理解 JVM 笔记系列（二）：内存模型 (附脑图)