1、虚拟机的前世今生和Java内存区域

T A H N K Y O U F O R W A T C H I N G

主讲老师Mark：446106311

课程咨询安生老师：669100976

课程安排

享学课堂-JVM和编写高效优雅Java程序 课程表	
课次序号	章节名称
1	虚拟机的前世今生和Java内存区域
2	垃圾回收器和内存分配策略
3	JVM的执行子系统
4	编写高效优雅Java程序
5	总览性能优化

注意：为了保证学员的学习效果以及内容的深度，上课进度会根据实际情况有所变动

上课说明：

1、首次出现的知识如需要进行编码，会进行手写，以后再出现则可能会事先准备好或者进行拷贝。

2、一个知识点如果大部分同学明白，不会重复讲解，未明白的同学请看视频或加老师QQ。

3、以上为JVM和编写高效优雅Java程序的章节安排，如果一章内容在一次课内未讲完，则会顺延到下次课继续讲解。

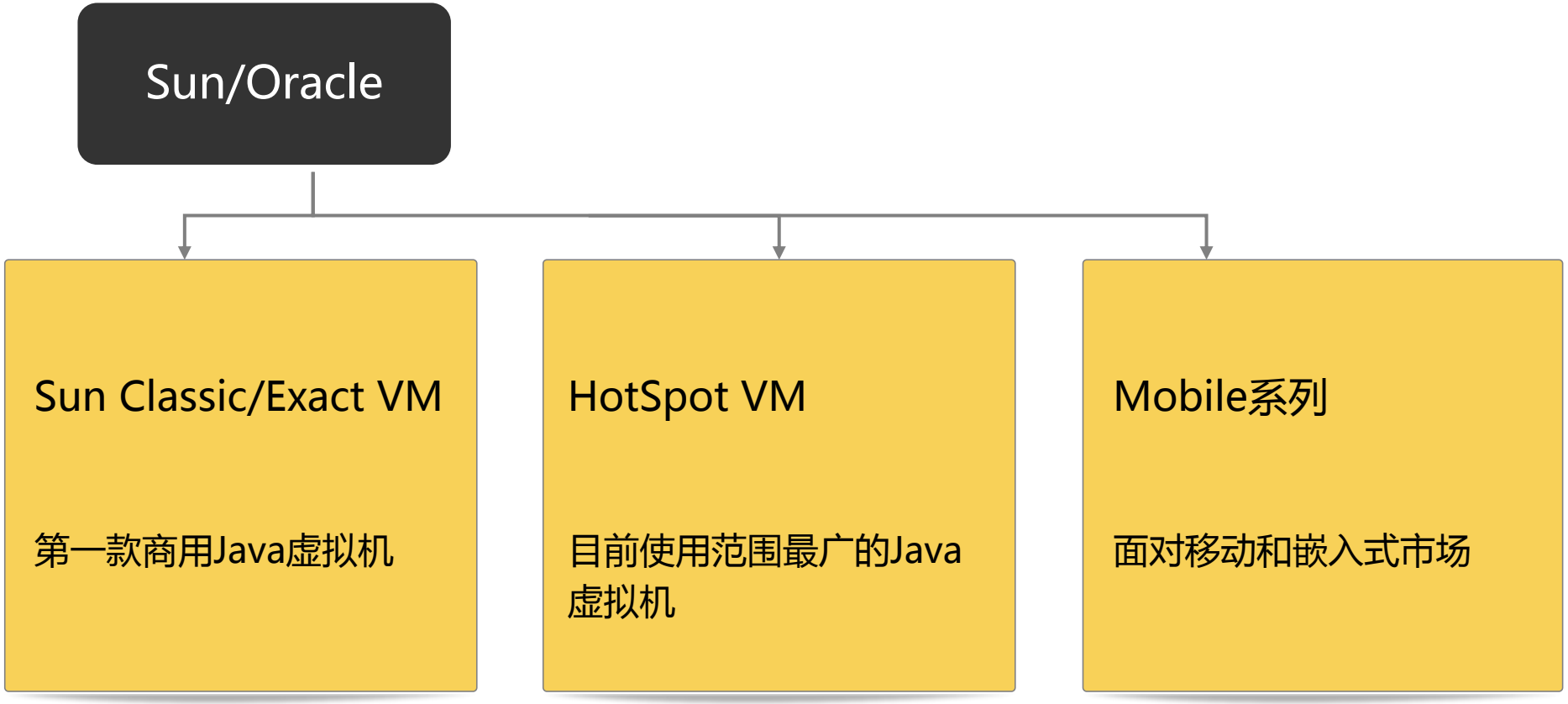
为什么要了解虚拟机

写出更好、更健壮的Java程序

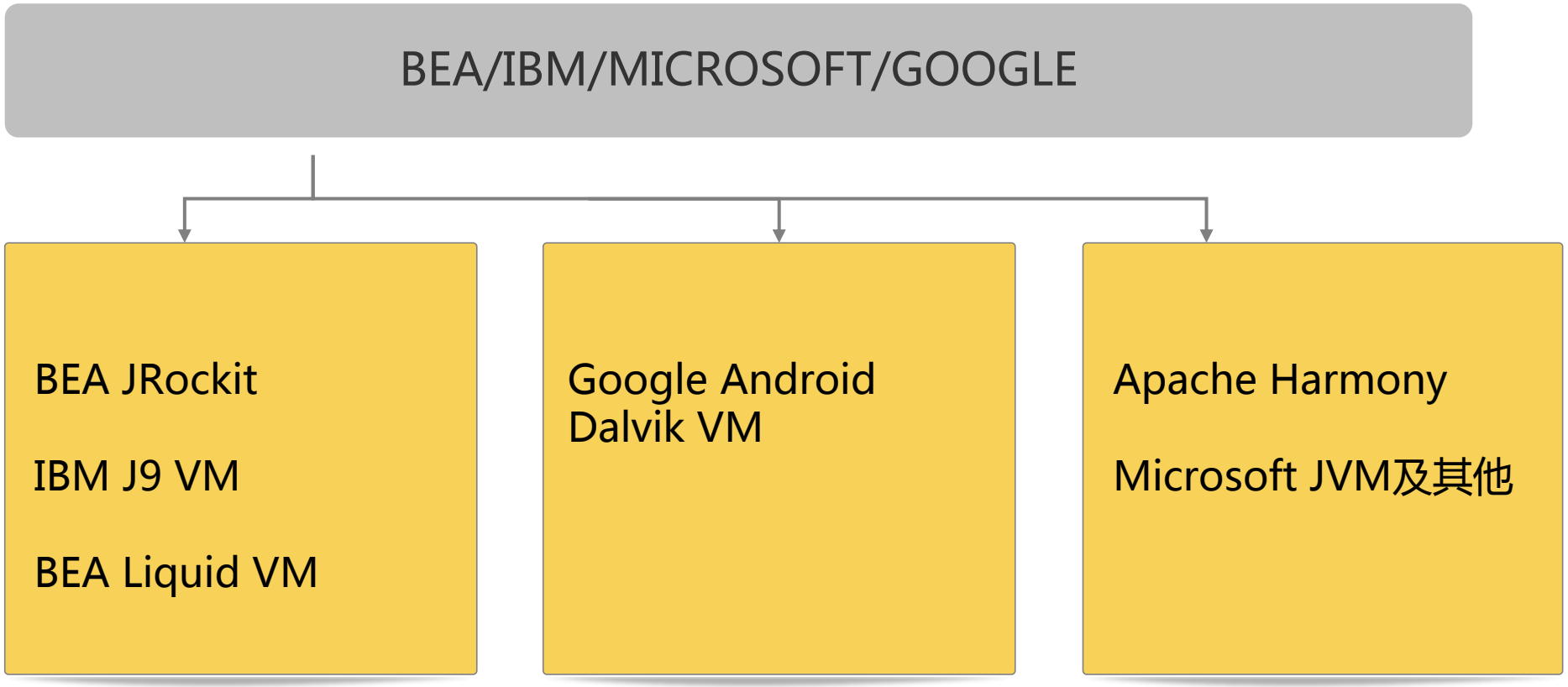
提高Java应用的性能，排除问题

面试必问

Sun/Oracle系列的虚拟机



其他公司的虚拟机


BEA/IBM/MICROSOFT/GOOGLE
BEA JRockit
IBM J9 VM
BEA Liquid VM
Google Android
Dalvik VM
Apache Harmony
Microsoft JVM及其他

未来的Java技术

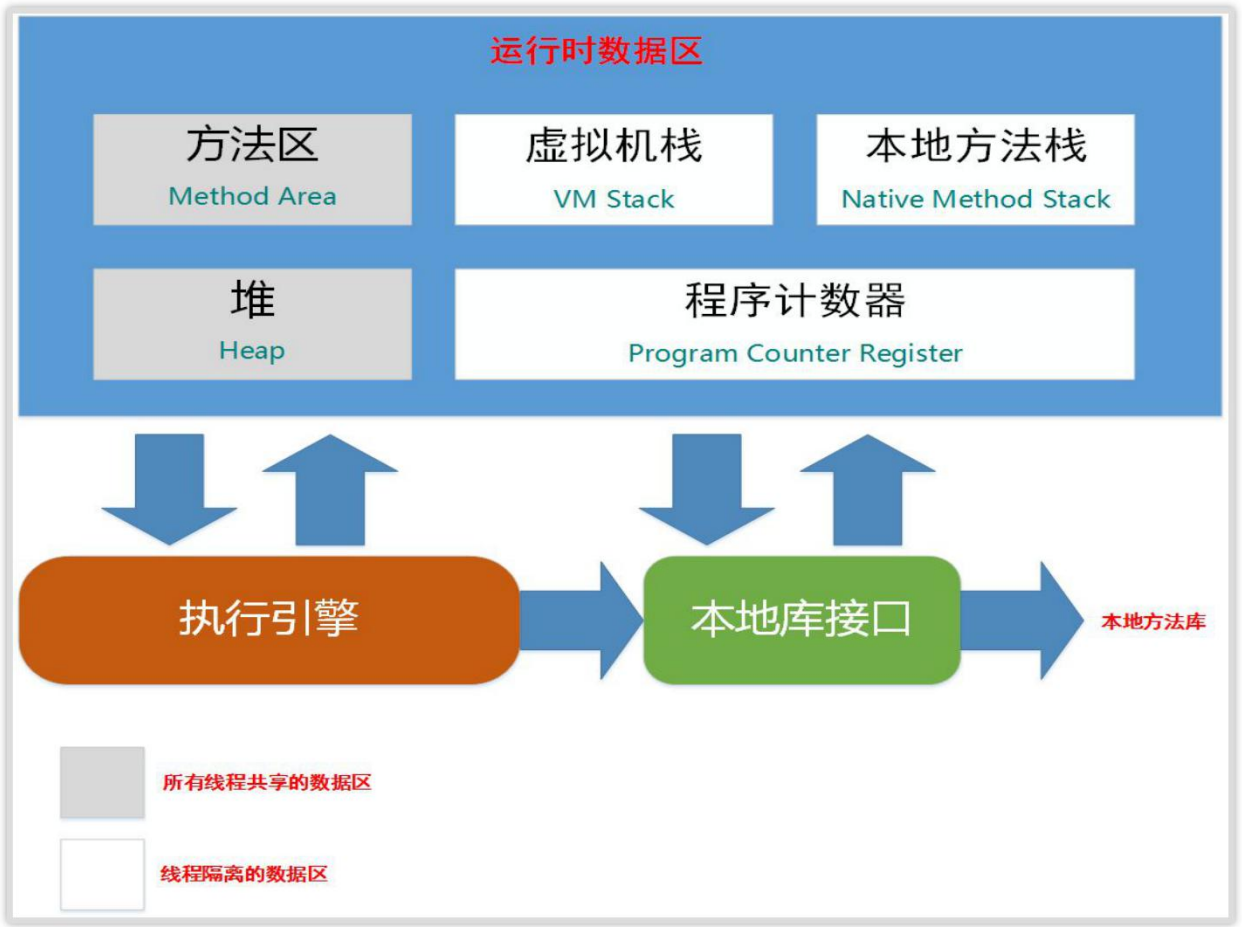
运行时数据区域

定义

Java虚拟机在执行Java程序的过程中会把它所管理的内存划分为若干个不同的数据区域

类型

程序计数器、虚拟机栈、本地方法栈、Java堆、方法区（运行时常量池）、直接内存



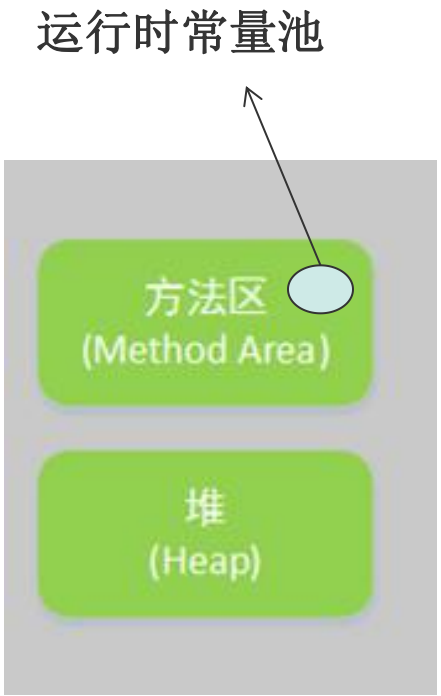
各个区域的作用

程序计数器：较小的内存空间，当前线程执行的字节码的行号指示器；各线程之间独立存储，互不影响；

java栈：线程私有，生命周期和线程，每个方法在执行的同时都会创建一个**栈帧**用于存储局部变量表，操作数栈，动态链接，方法出口等信息。方法的执行就对应着栈帧在虚拟机栈中入栈和出栈的过程；栈里面存放着各种基本数据类型和对象的引用（-Xss）；

本地方法栈：本地方法栈保存的是native方法的信息，当一个JVM创建的线程调用native方法后，JVM不再为其在虚拟机栈中创建栈帧，JVM只是简单地动态链接并直接调用native方法；

各个区域的作用



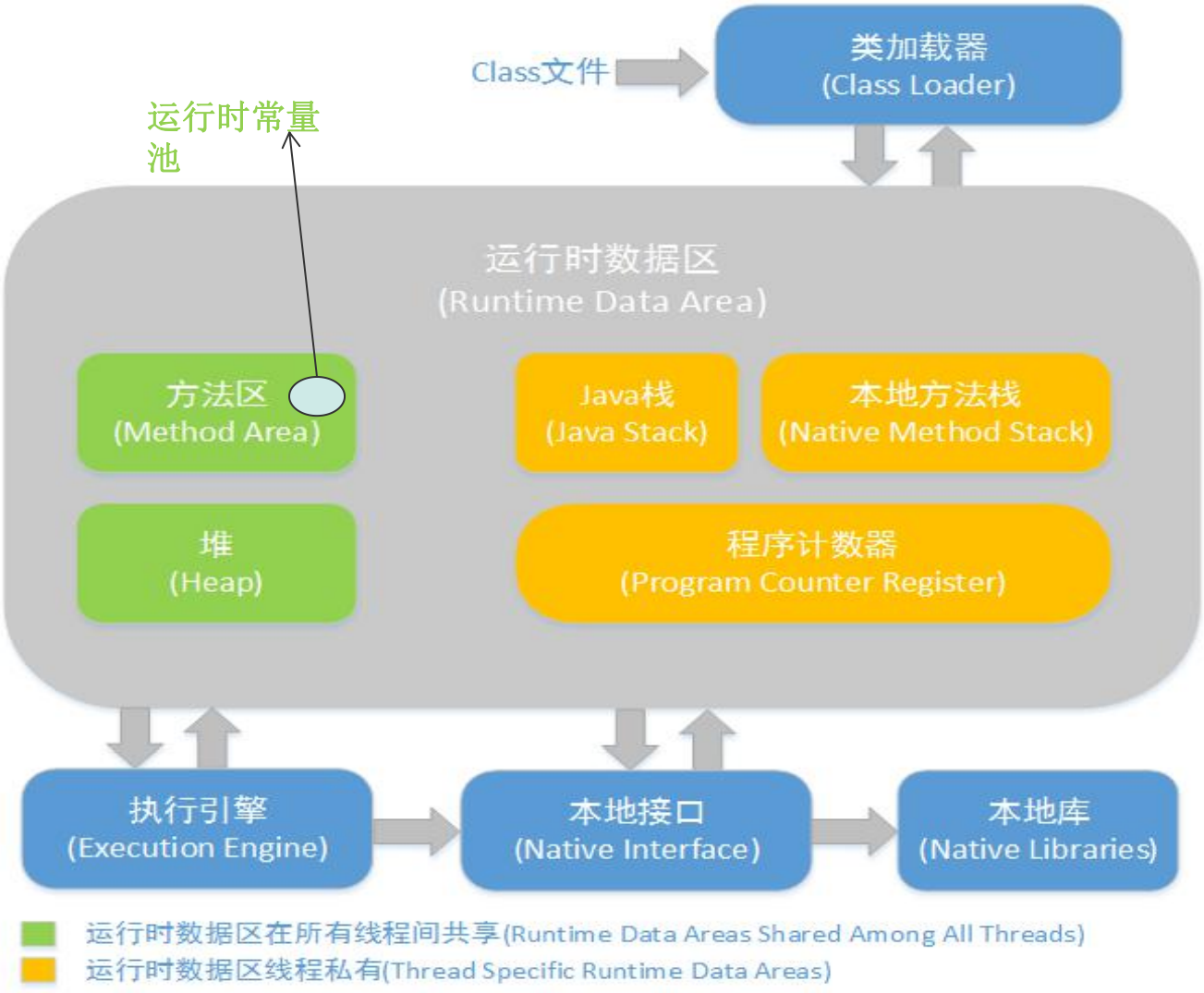
堆：Java堆是Javaer需要重点关注的一块区域，因为涉及到内存的分配(new关键字，反射等)与回收(回收算法，收集器等)（-Xms；-Xmx；-Xmn；-XX:NewSize；-XX:MaxNewSize）；

方法区：也叫永久区，用于存储已经被虚拟机加载的类信息，常量("zdy","123"等)，静态变量(static变量)等数据（-XX:PermSize；-XX:MaxPermSize；-XX:MetaspaceSize；-XX:MaxMetaspaceSize）。

运行时常量池：运行时常量池是方法区的一部分，用于存放编译期生成的各种字面量("zdy","123"等)和符号引用。

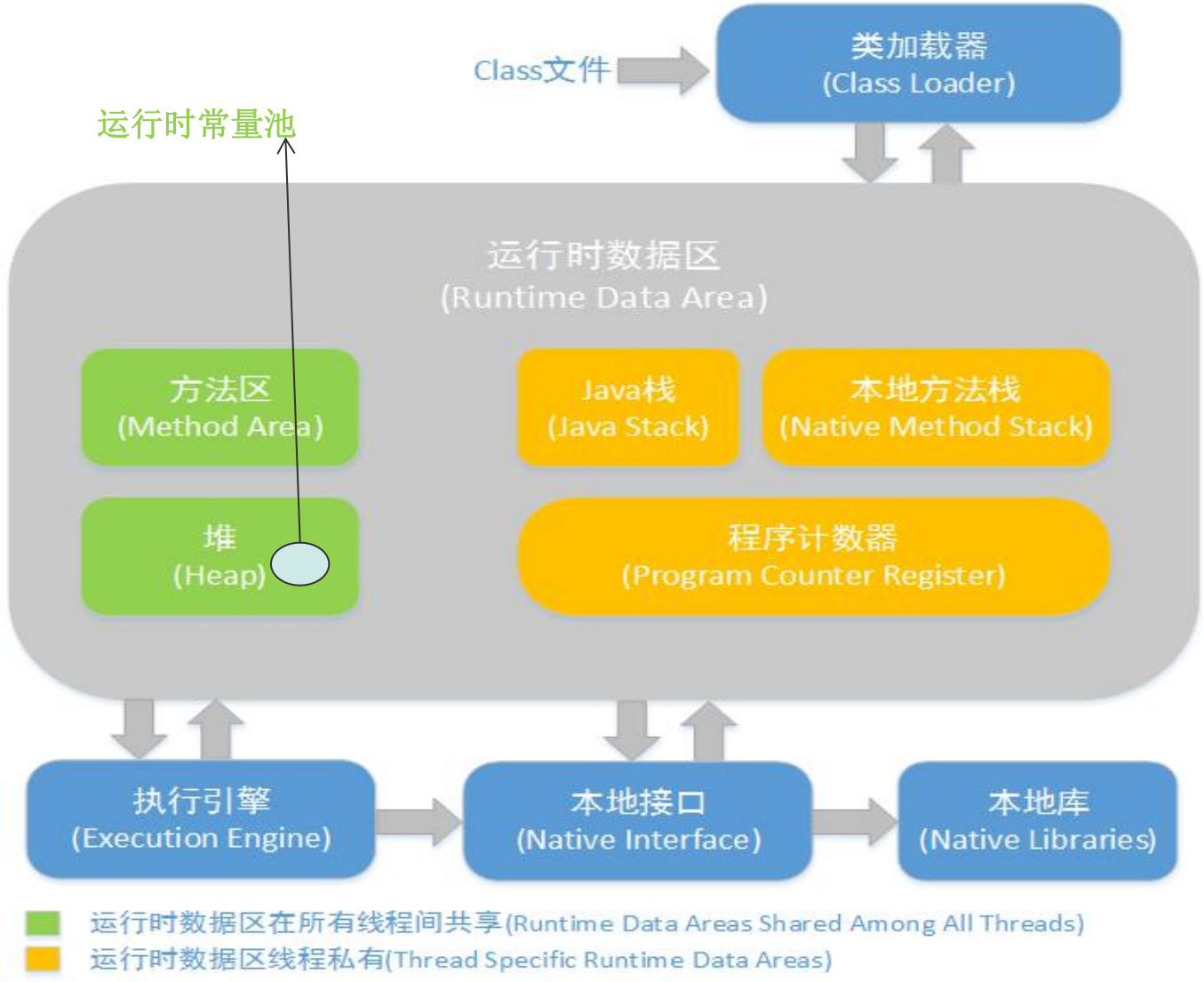
各个版本内存区域的变化

1.6



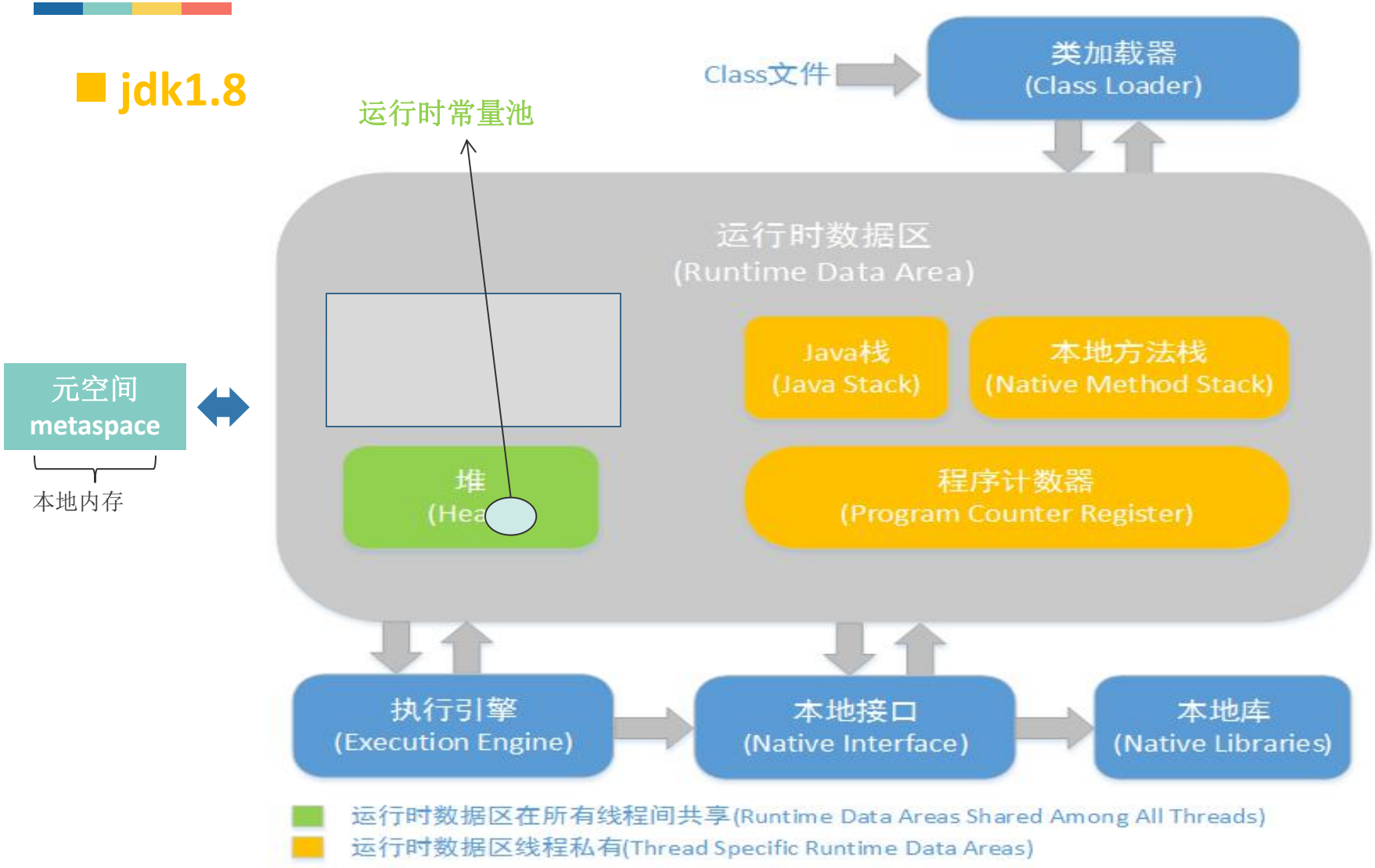
各个版本内存区域的变化

jdk1.7



各个版本内存区域的变化

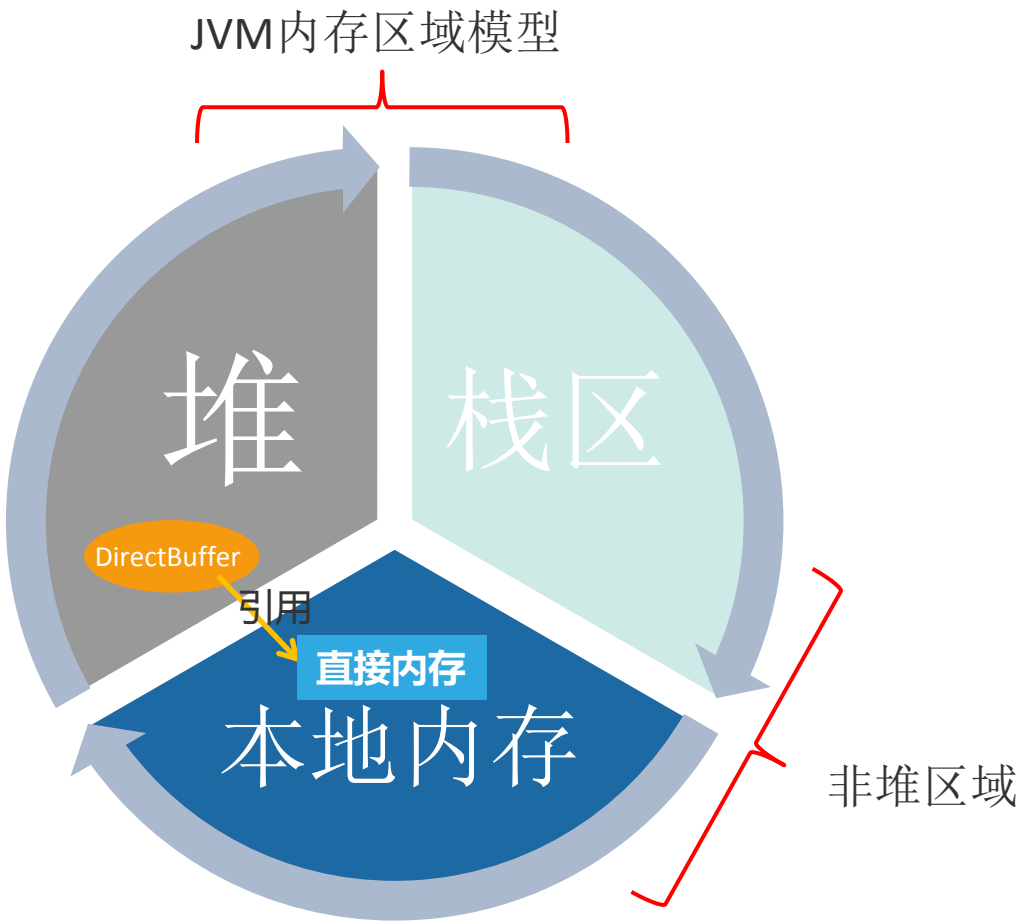
jdk1.8



永久代来存储类信息、常量、静态变量等数据不是个好主意，很容易遇到内存溢出的问题。

对永久代进行调优是很困难的，同时将元空间与堆的垃圾回收进行了隔离，避免永久代引发的Full GC和OOM等问题；

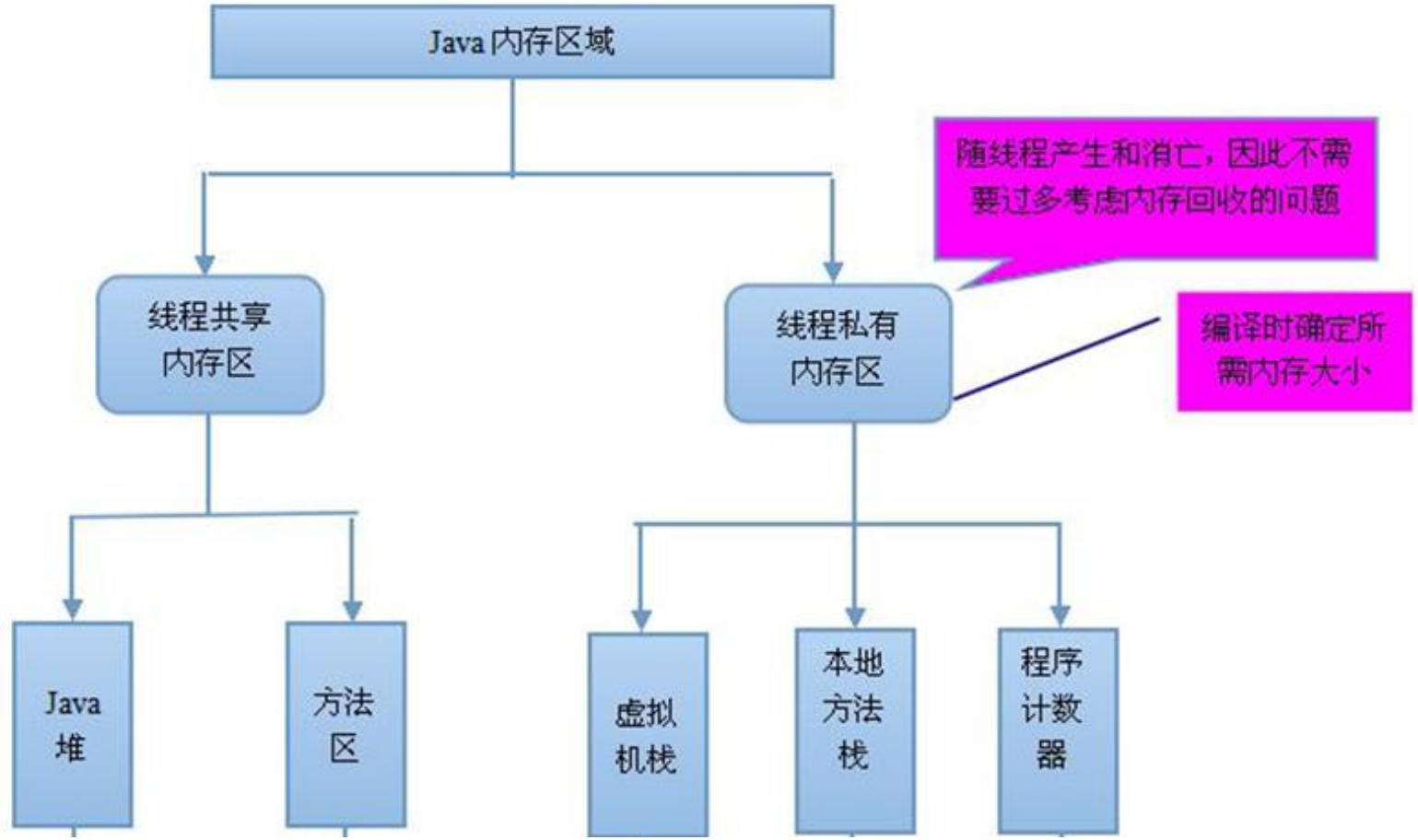
直接内存



直接内存：不是虚拟机运行时数据区的一部分，也不是java虚拟机规范中定义的内存区域；

- ✓ 如果使用了NIO,这块区域会被频繁使用，在java堆内可以用directByteBuffer对象直接引用并操作；
- ✓ 这块内存不受java堆大小限制，但受本机总内存的限制，可以通过MaxDirectMemorySize来设置（默认与堆内存最大值一样），所以也会出现OOM异常；

站在线程角度来看


Java 内存区域
线程共享
内存区
线程私有
内存区
随线程产生和消亡，因此不需
要过多考虑内存回收的问题
编译时确定所
需内存大小
Java
堆
方法
区
虚拟
机栈
本地
方法
栈
程序
计数
器

深入辨析堆和栈

功能

- 以栈帧的方式存储方法调用的过程，并存储方法调用过程中基本数据类型的变量（int、short、long、byte、float、double、boolean、char等）以及对象的引用变量，其内存分配在栈上，变量出了作用域就会自动释放；
- 而堆内存用来存储Java中的对象。无论是成员变量，局部变量，还是类变量，它们指向的对象都存储在堆内存中；

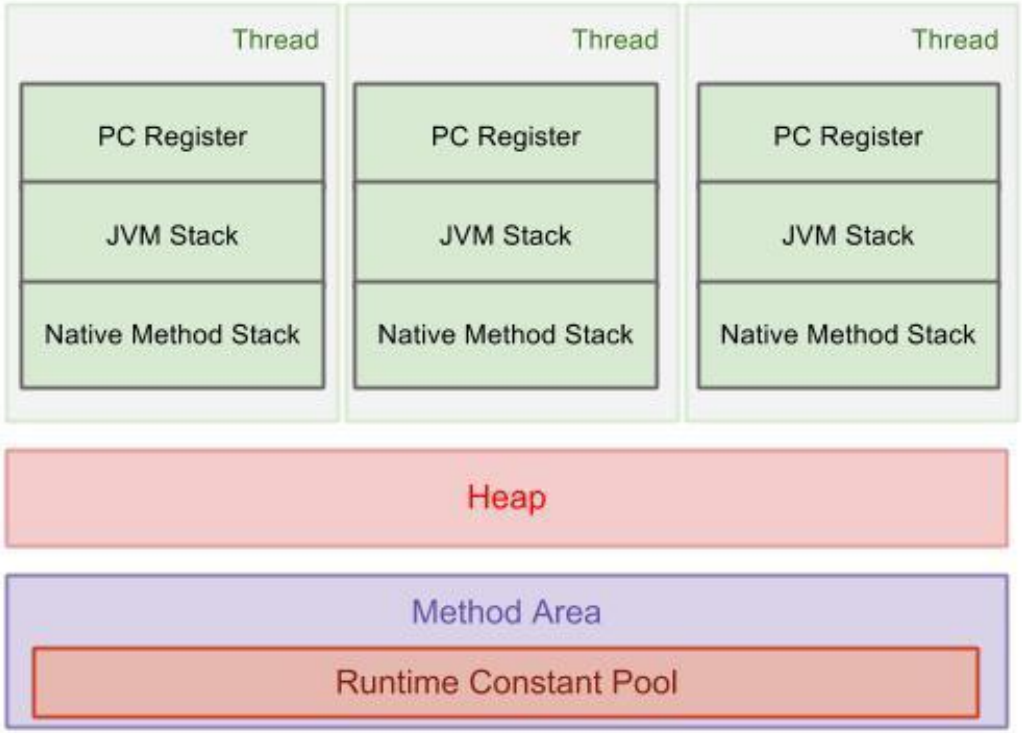
线程独享还是共享

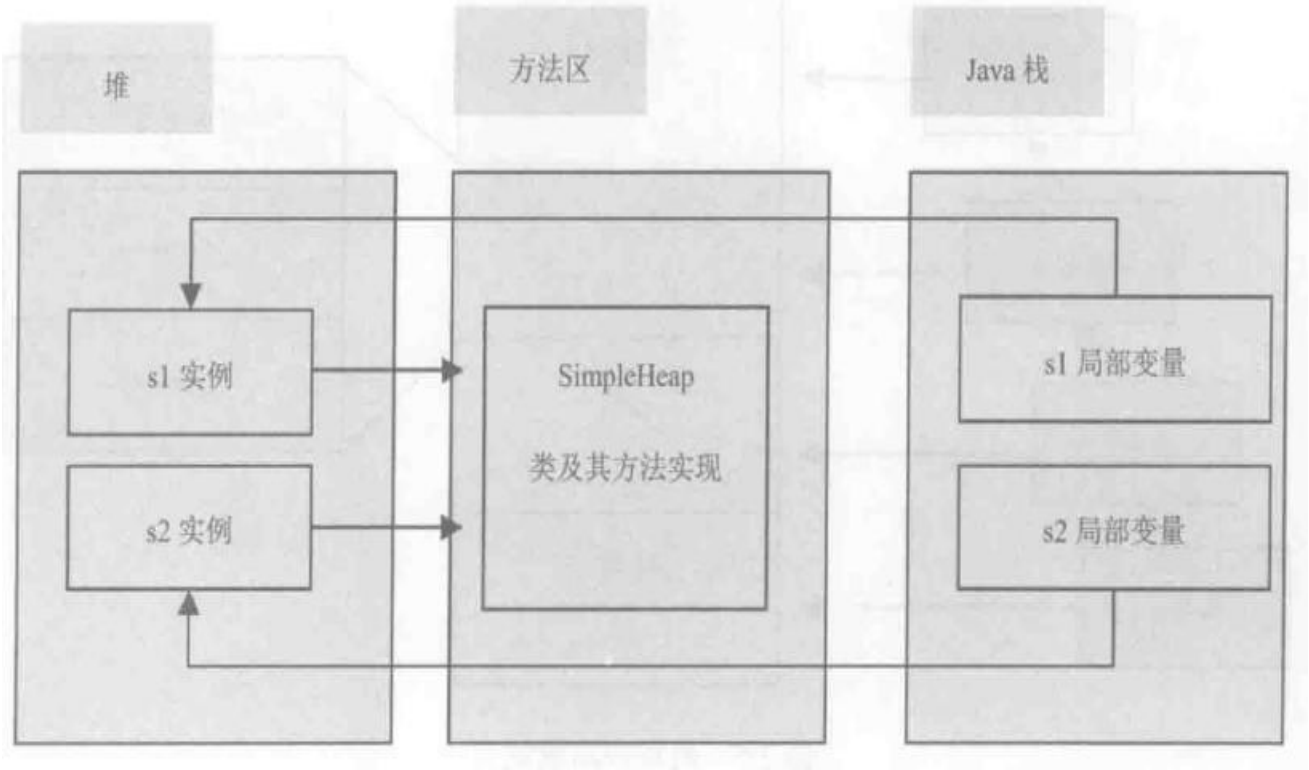
- 栈内存归属于单个线程，每个线程都会有一个栈内存，其存储的变量只能在其所属线程中可见，即栈内存可以理解成线程的私有内存。
- 堆内存中的对象对所有线程可见。堆内存中的对象可以被所有线程访问。

空间大小

- 栈的内存要远远小于堆内存，栈的深度是有限制的，可能发生StackOverFlowError问题。

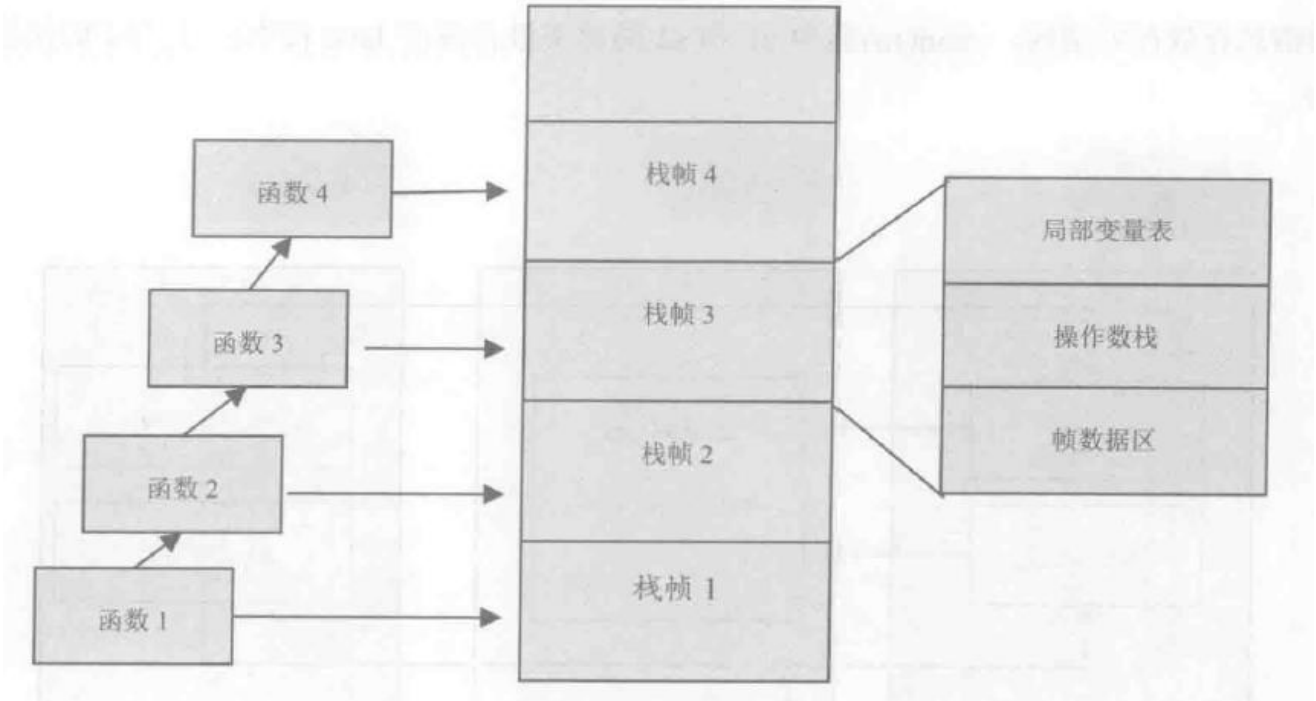
深入辨析堆和栈


Thread
PC Register
JVM Stack
Native Method Stack
Thread
PC Register
JVM Stack
Native Method Stack
Thread
PC Register
JVM Stack
Native Method Stack
Heap
Method Area
Runtime Constant Pool


堆
方法区
Java 栈
s1 实例
s2 实例
SimpleHeap
类及其方法实现
s1 局部变量
s2 局部变量

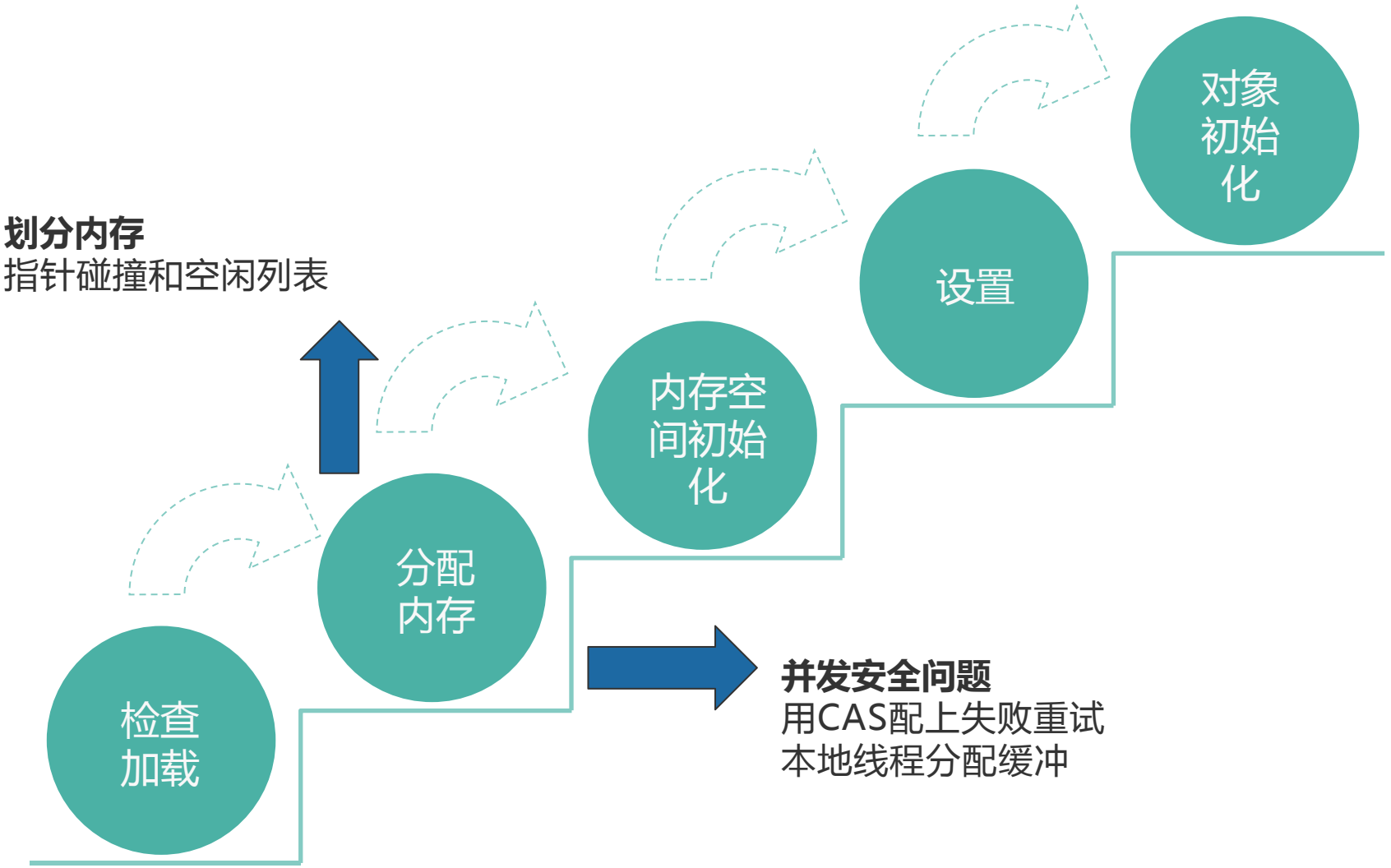
方法的出入栈

- 方法会打包成栈桢，一个栈桢至少要包含局部变量表，操作数栈和帧数据区



- 栈上分配

虚拟机中的对象


检查加载
分配内存
内存空间初始化
设置
对象初始化
划分内存
指针碰撞和空闲列表
并发安全问题
用CAS配上失败重试
本地线程分配缓冲

对象的内存布局

对象头（Header）

对象自身的运行时数据

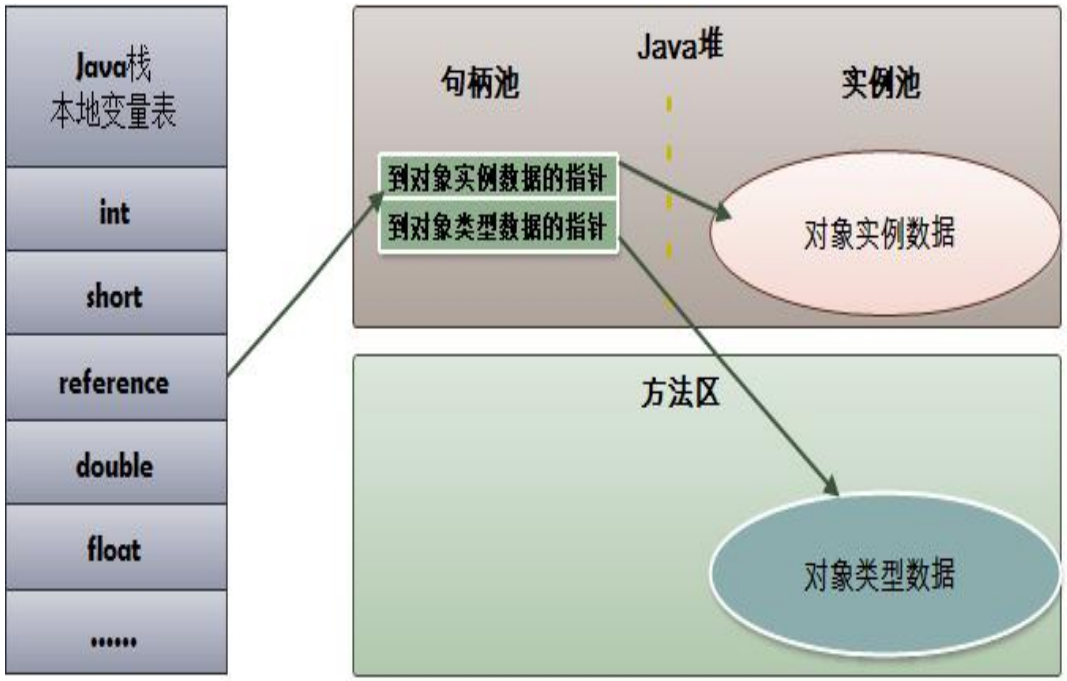
类型指针

实例数据（Instance Data）

程序代码中所定义的各种类型的字段内容

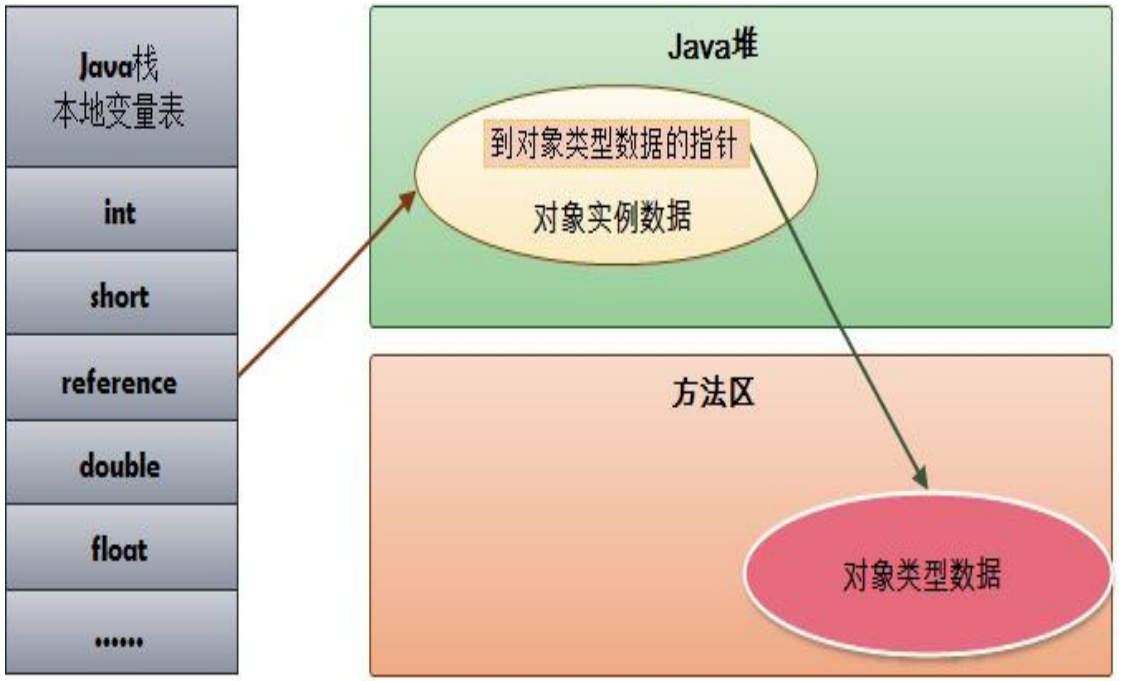
对齐填充

对象的访问定位



句柄方式访问对象

使用句柄



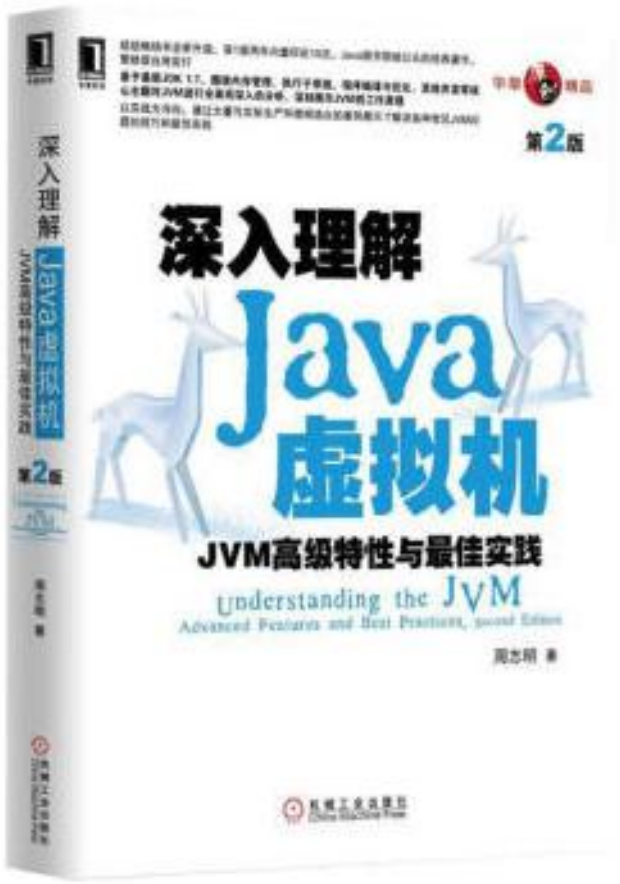
直接指针方式访问对象

直接指针

堆参数设置、对性能的影响和内存溢出实战

1. Java堆溢出
2. 新生代配置
3. 方法区和运行时常量池溢出
4. 虚拟机栈和本地方法栈溢出
5. 本机直接内存溢出

推荐书籍


深入理解Java虚拟机
JVM高级特性与最佳实践
第2版
周志明 著


知名技术专家 蔡学镛 力荐
Broadview
Java虚拟机精讲
中国工信出版集团