

拉勾教育

— 互联网人实战大学 —

《Java 性能优化实战 21 讲》

李国 前京东、陌陌高级架构师

— 拉勾教育出品 —

17 | 高级进阶：JVM 如何完成的垃圾回收？

高级进阶：JVM 如何完成的垃圾回收？

拉勾教育

— 互联网人实战大学 —

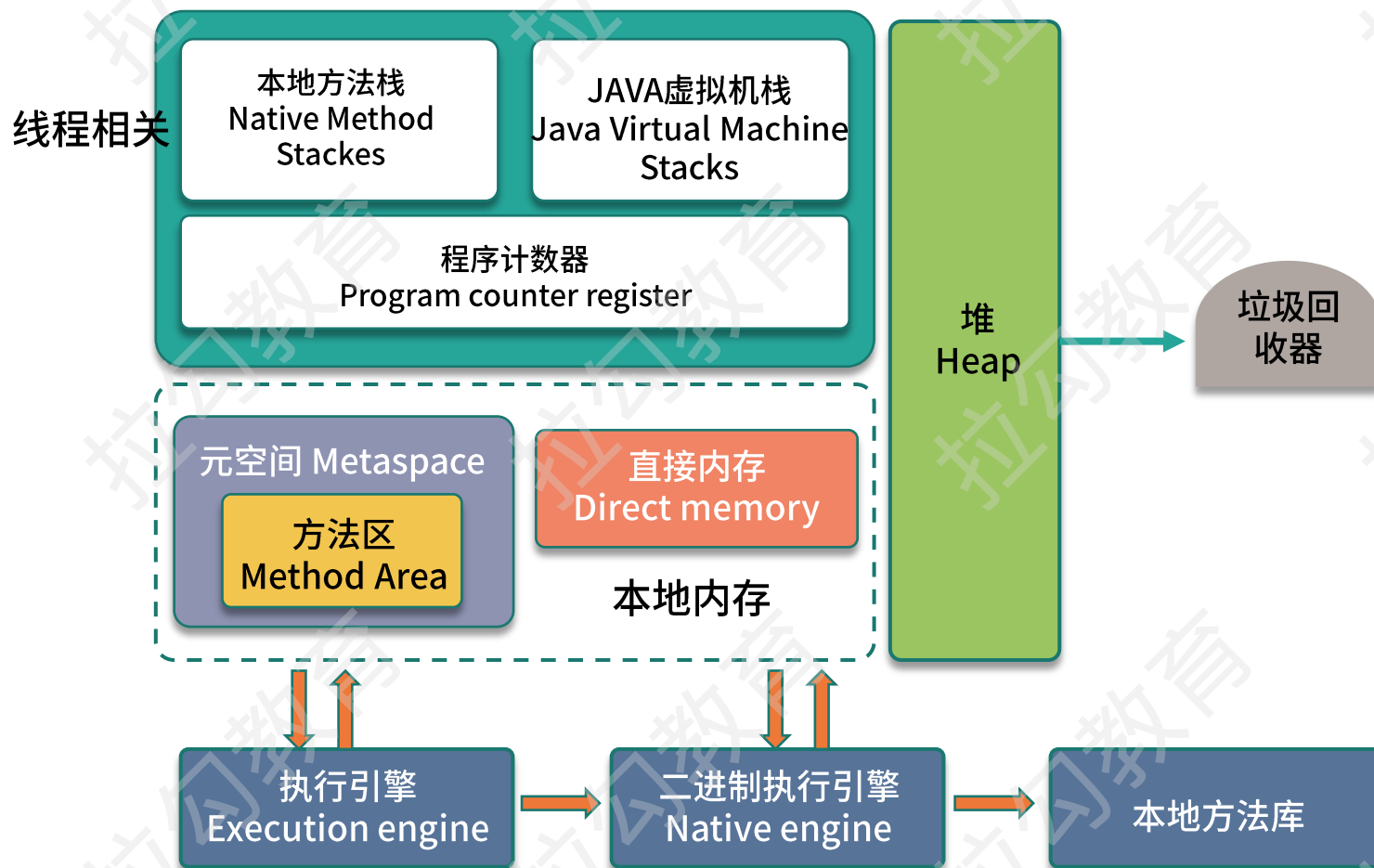
本课时

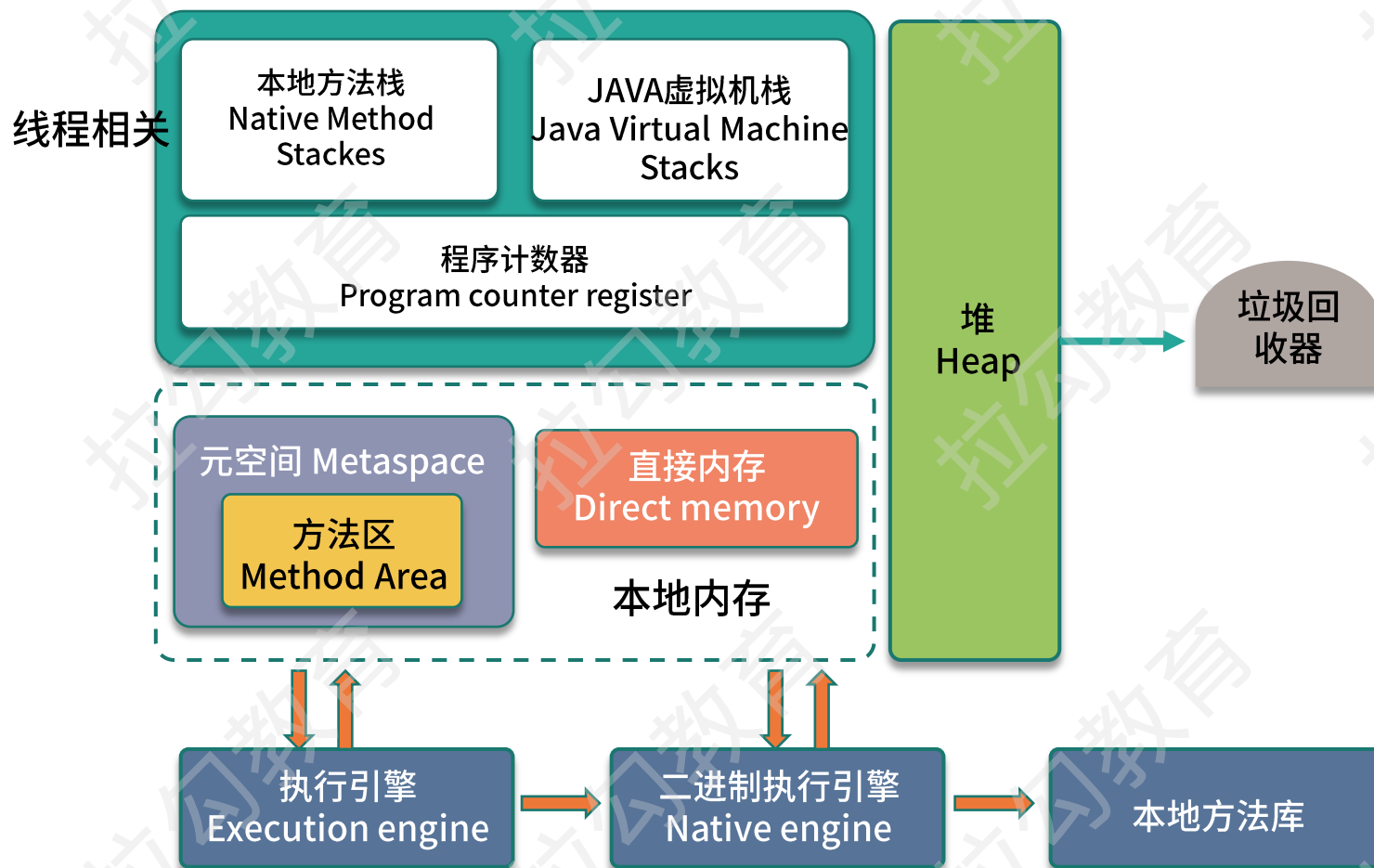
集中讲解 JVM
(Java Virtual Machine)
的基本知识点

JVM 内存区域划分

拉勾教育

— 互联网人实战大学 —

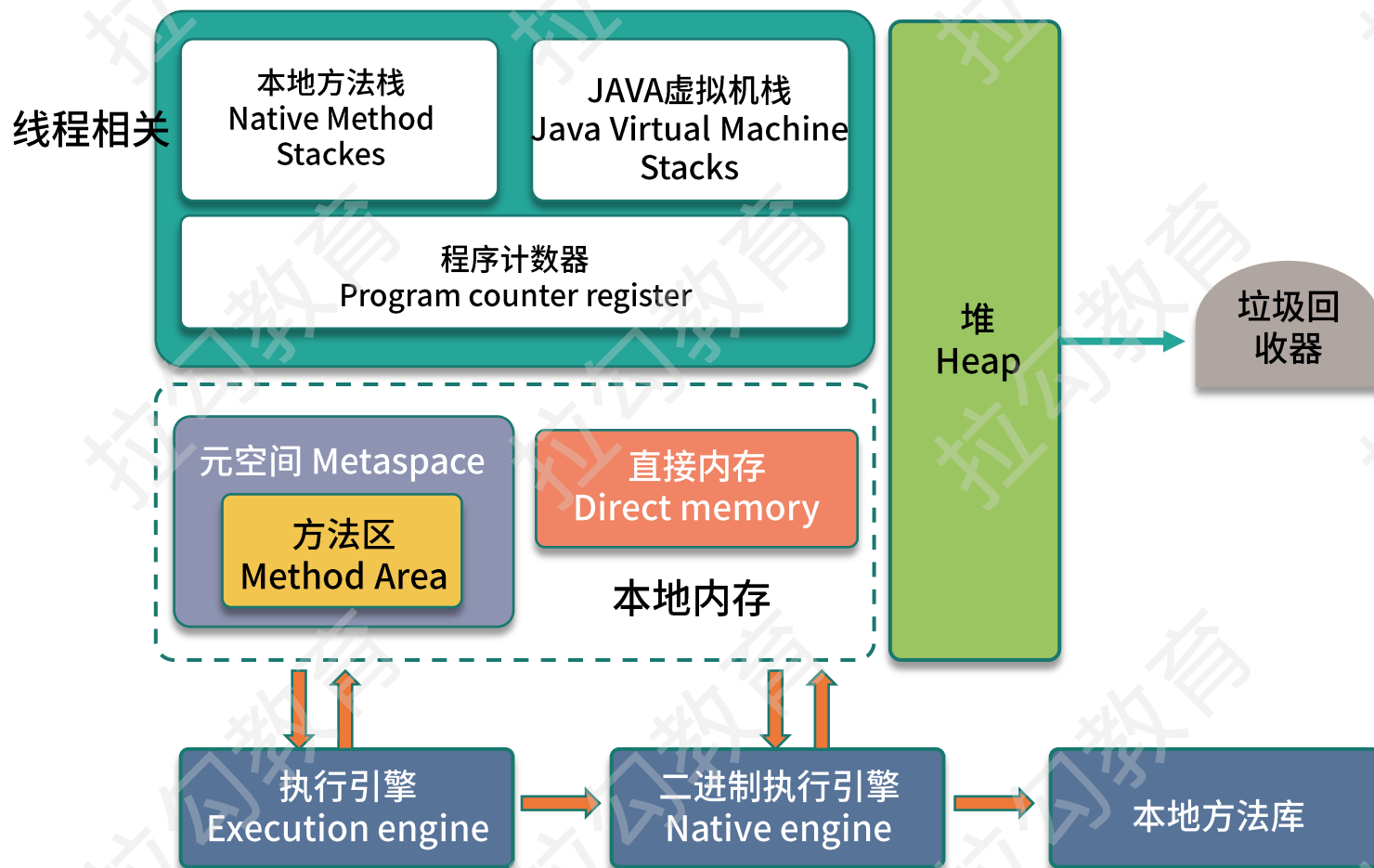


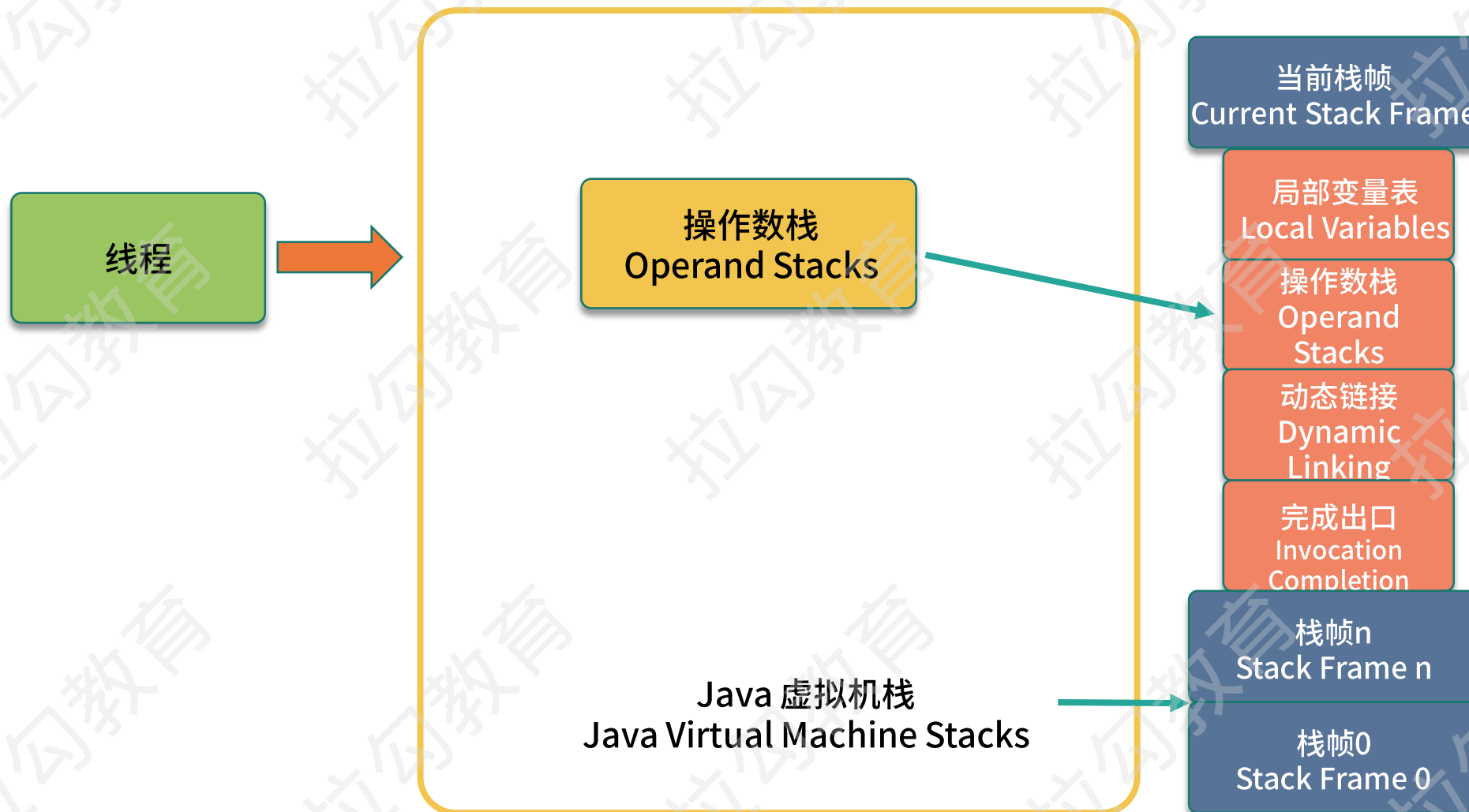


Java 虚拟机栈

拉勾教育

— 互联网人实战大学 —





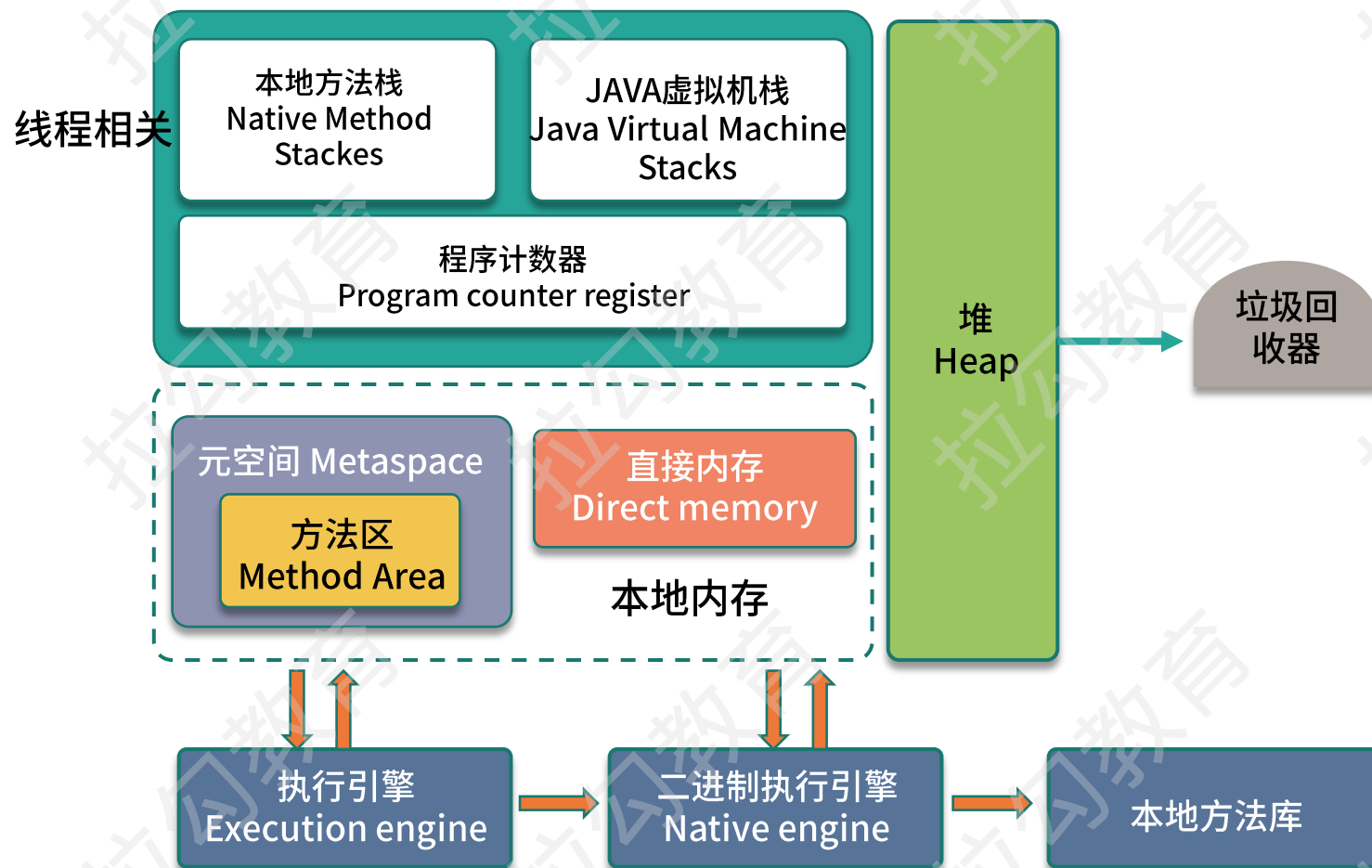
操作数栈是具体的字节码指令所操作的栈区域

```
public void test(){  
    int a = 1;  
    a++;  
}
```


程序计数器

拉勾教育

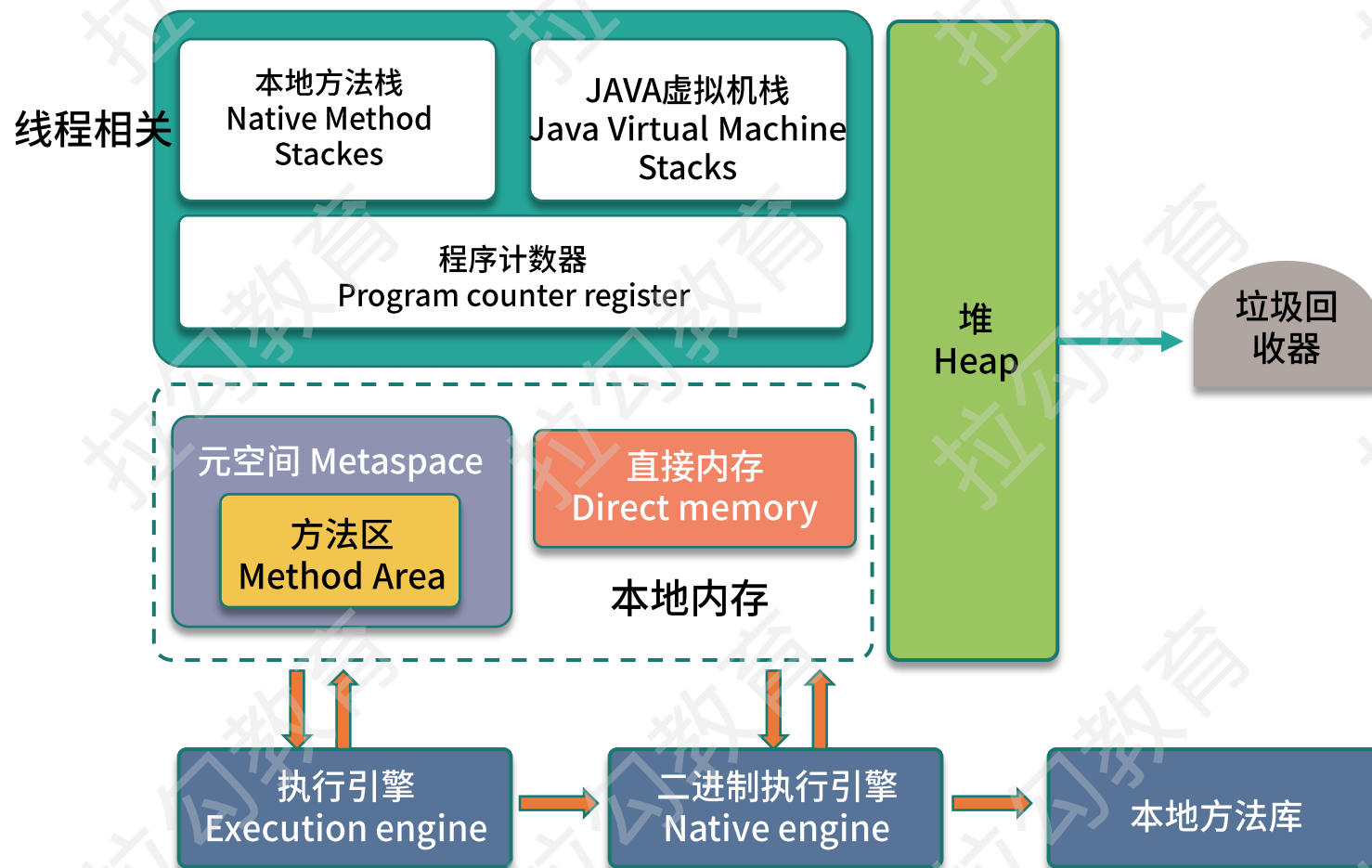
— 互联网人实战大学 —

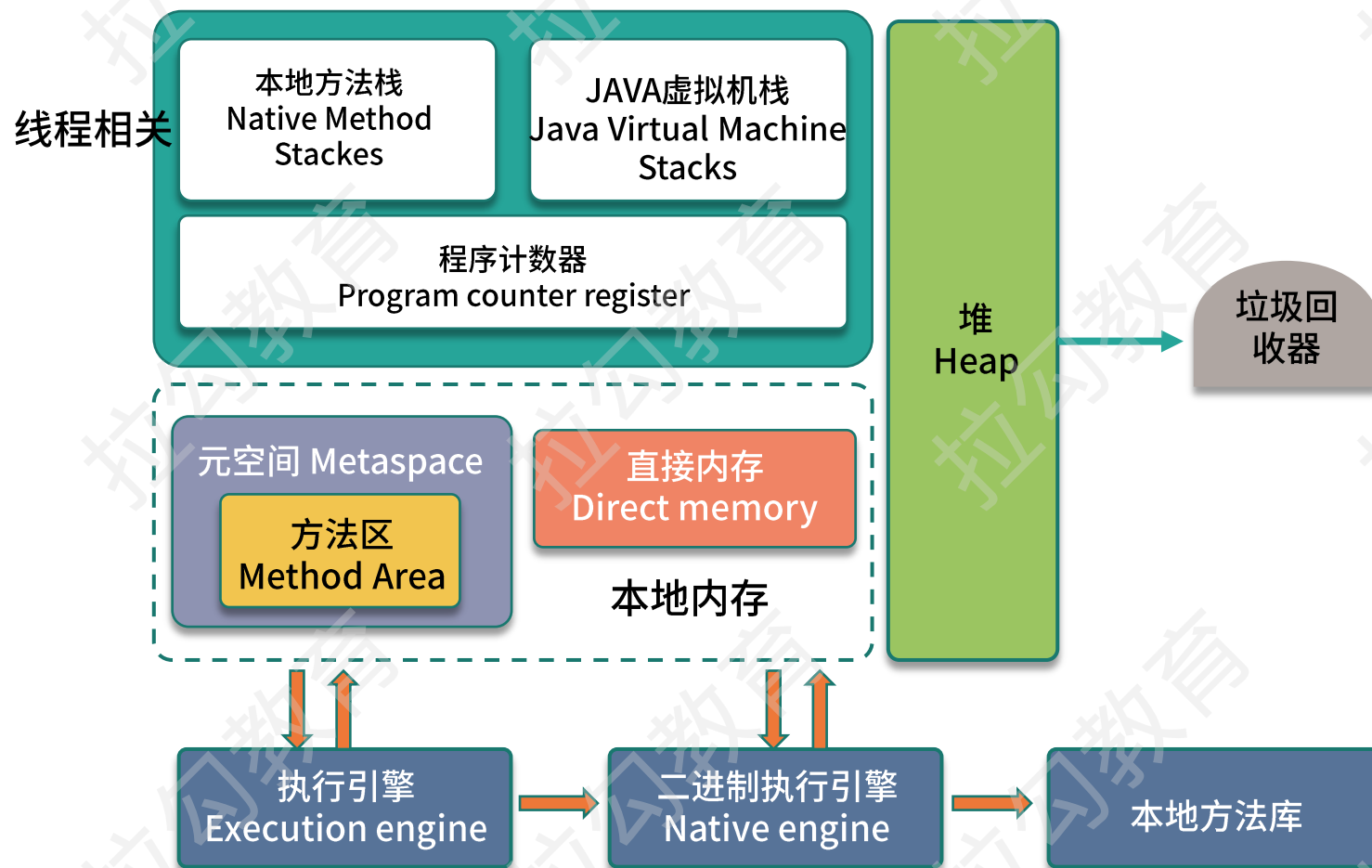


本地方法栈

拉勾教育

— 互联网人实战大学 —





元空间是在堆上么？

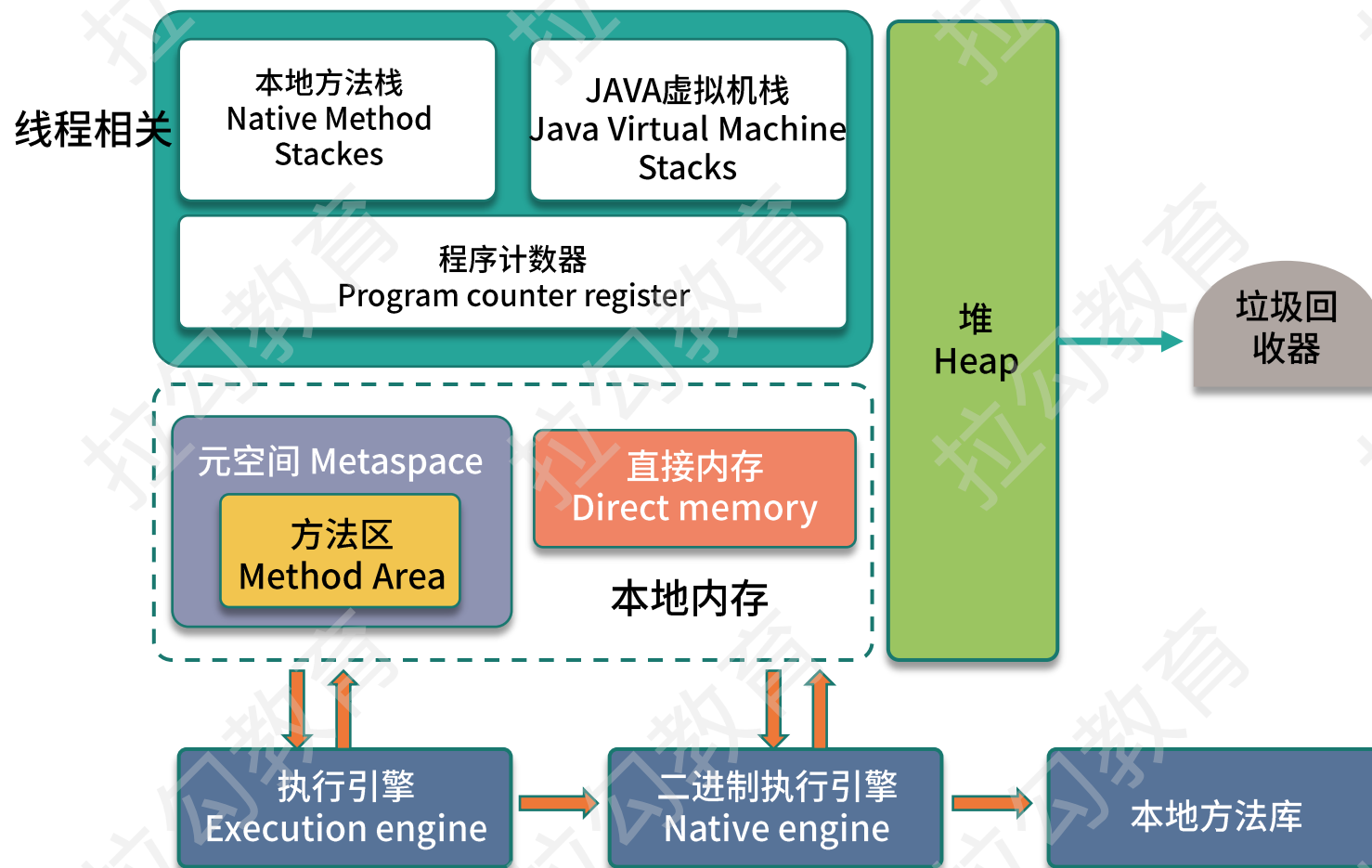
答案：元空间并不是在堆上分配的，而是在堆外空间进行分配的
它的大小默认没有上限，我们常说的方法区，就在元空间中



字符串常量池在那个区域中 ?

- JDK 1.8 之前，没有元空间这个概念，当时的方法区是放在一个叫做永久代的空间中
- JDK 1.7 之前，字符串常量池也放在这个叫做永久带的空间中
- JDK 1.7 版本，已经将字符串常量池从永久带移动到了堆上







直接内存

有具体的 API (ByteBuffer)
也可以使用 -
XX:MaxDirectMemorySize
参数控制大小



本地内存

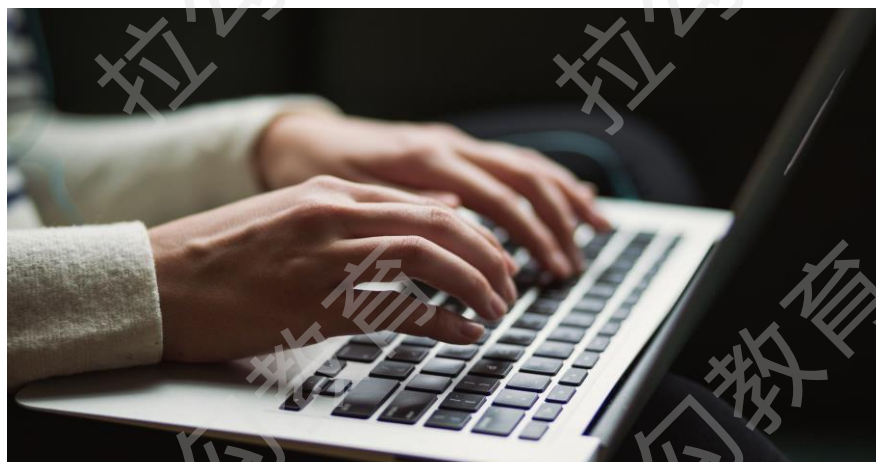
一个统称
使用 native 函数操作的内存
就是本地内存

对象主要是在堆上分配的，把它想象成一个池子，对象不停地创建，后台的垃圾回收进程不断地清理不再使用的对象。当内存回收的速度，赶不上对象创建的速度，这个对象池子就会产生溢出，也就是 OOM

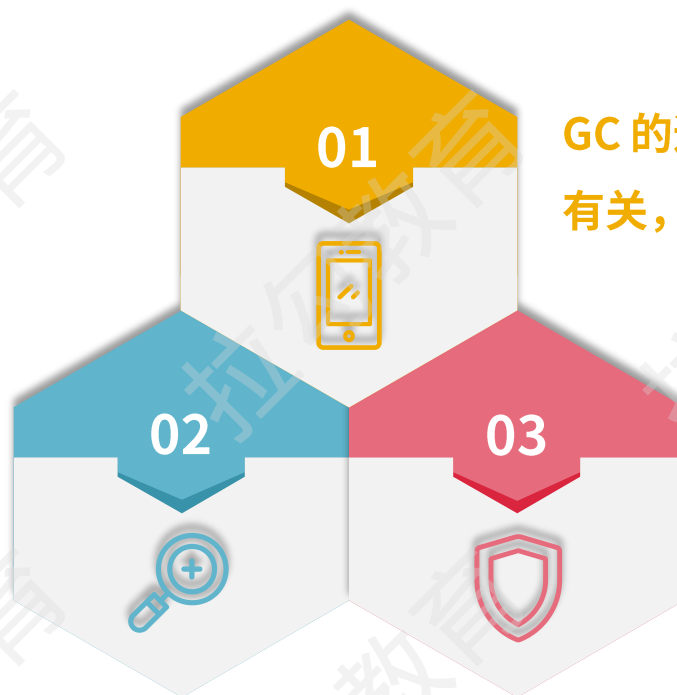


垃圾回收

JVM 会找到正在使用的对象，对这些使用的对象进行标记和追溯，然后一股脑地把剩下的对象判定为垃圾，进行清理



GC 的速度与堆的大小无关，32GB 的堆和 4GB 的堆，只要存活对象是一样的，垃圾回收速度也会差不多



01 GC 的速度，和堆内存活对象的多少有关，与堆内所有对象的数量无关

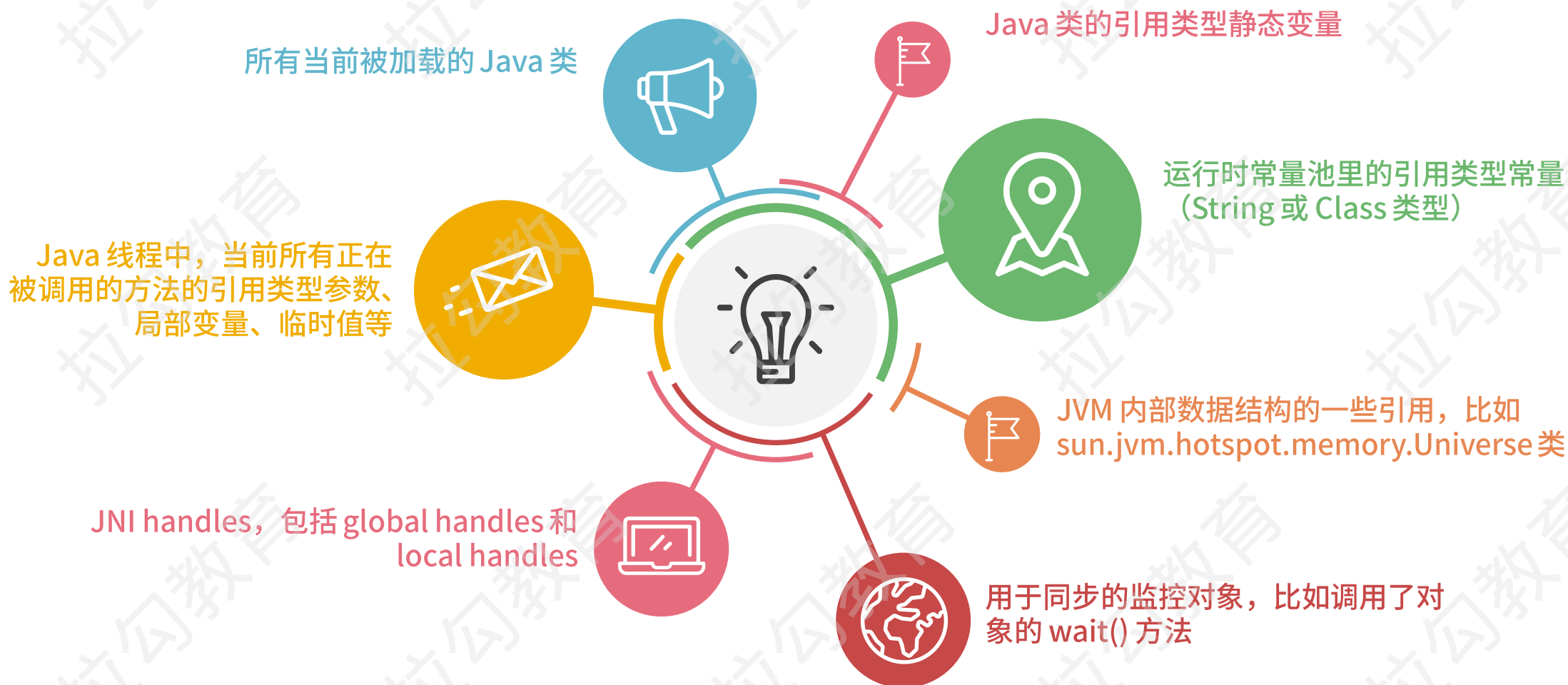
03 垃圾回收不必每次都把垃圾清理得干干净净，最重要的是不要把正在使用的对象判定为垃圾

GC Roots

正在使用的引用的入口

可达性分析法

使用 tracing 方式寻找存活对象的方法



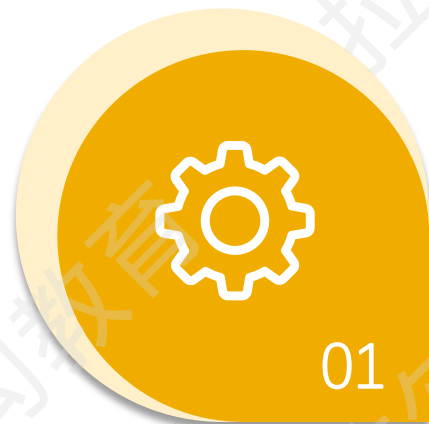
强、软、弱、虚引用

拉勾教育

— 互联网人实战大学 —

强应用

默认的对象关系是强引用。只有在和 GC Roots 断绝关系时，才会被消灭掉



软引用

用于维护一些可有可无的对象



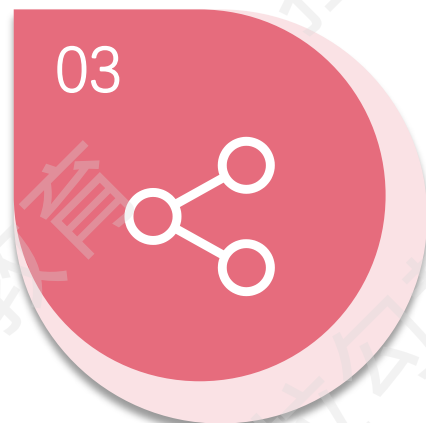
虚引用

Java 9.0 以后新加入了 Cleaner 类，用来替代 Object 类的 finalizer 方法



弱引用

当 JVM 进行垃圾回收时，无论内存是否充足，都会回收被弱引用关联的对象



Stop the World

JVM 在做标记和追溯的时候，会停止业务线程的所有工作
进入 SafePoint 状态

分代垃圾回收

01

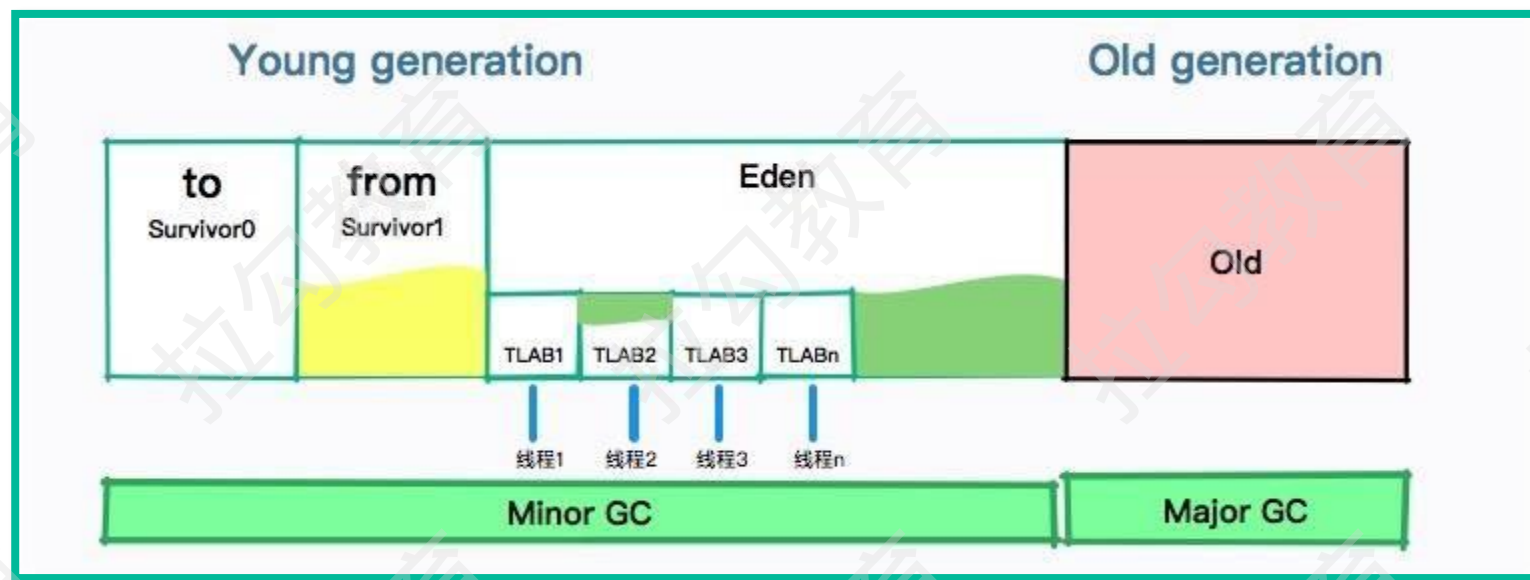


大部分对象的生命周期都很短

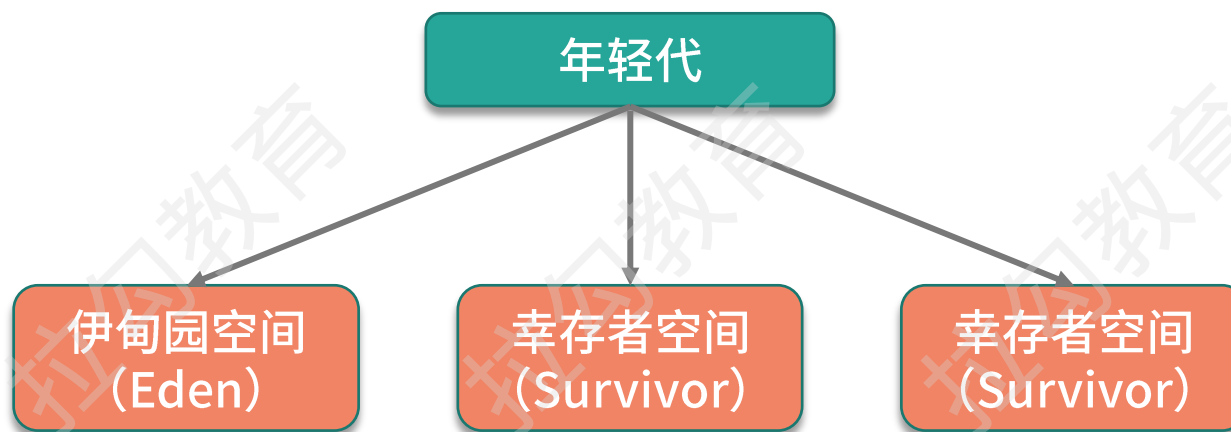
02



其他对象则很可能会存活很长时间



堆空间划分图：年轻代和老年代



Eden、from、to 的默认比例是 8:1:1

这个比例，是由参数 `-XX:SurvivorRatio` 进行配置的（默认为8）

正常提升 (Promotion)

如果对象能够熬过年轻代垃圾回收，它的年龄 (age) 就会加一，当对象的年龄达到一定阈值，就会被移动到老年代中

分配担保

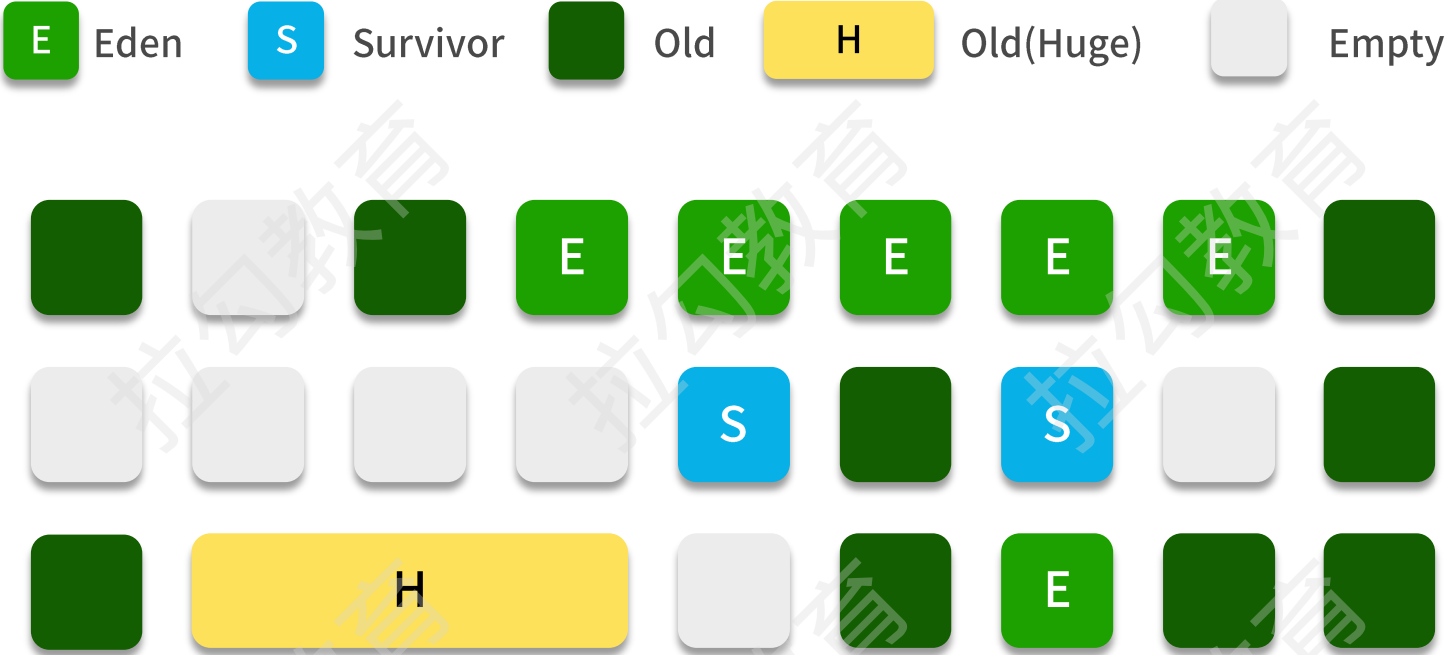
如果年轻代的空间不足，又有新的对象需要分配空间，需要依赖其他内存进行分配担保，对象将直接在老年代创建

大对象直接在老年代分配

超出某个阈值大小的对象，直接在老年代分配，可以通过 -XX:PretenureSizeThreshold 配置阈值

动态对象年龄判定

有的垃圾回收算法，并不要求 age 必须达到 15 才能晋升到老年代，它会使用一些动态的计算方法



G1 垃圾回收器概念图

G1 配置三个参数，获取优异的性能

MaxGCPauseMillis 设置最大停顿的预定目标



G1HeapRegionSize 设置小堆区的大小



InitiatingHeapOccupancyPercent 当整个堆内存使用达到一定比例（默认是45%），并发标记阶段就会被启动



小结

拉勾教育

— 互联网人实战大学 —

- 主要介绍了 JVM 的内存区域划分
堆、Java 虚拟机栈、程序计数器、本地方法栈、元空间、直接内存
- GC Roots 是使用 tracing 方式的可达性分析法来实现的
在对象的引用关系上，还会有强、软、弱、虚的差别
- 分代垃圾回收的概念，了解了年轻代和老年代的一些回收策略

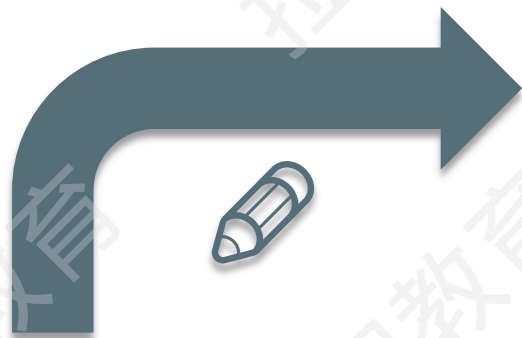


小结

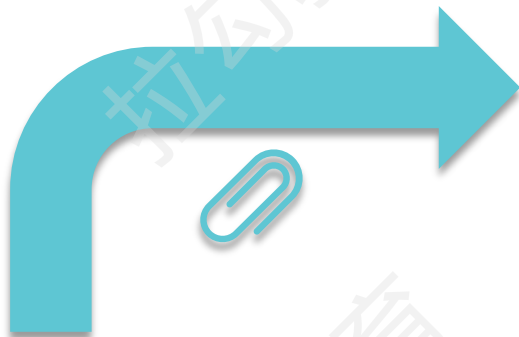
拉勾教育

— 互联网人实战大学 —

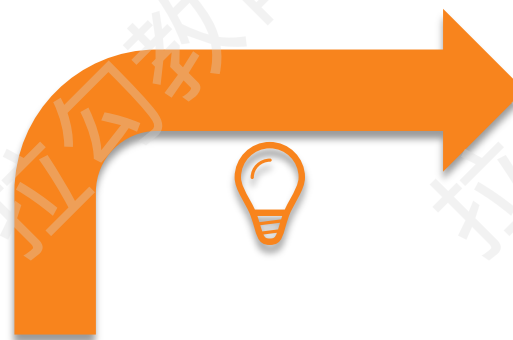
CMS 垃圾回收器



G1 垃圾回收器



支持 16TB 大小的 ZGC



我们常说的对象，除了基本数据类型
一定是在堆上分配的吗？



Next: 18 | 《高级进阶：JIT 如何影响 JVM 的性能？》

拉勾教育

— 互联网人实战大学 —



关注拉勾「教育公众号」
获取更多课程信息