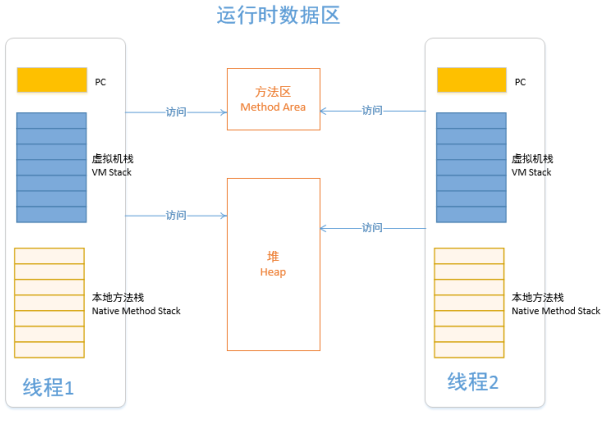
1-1 JVM 运行时数据区:

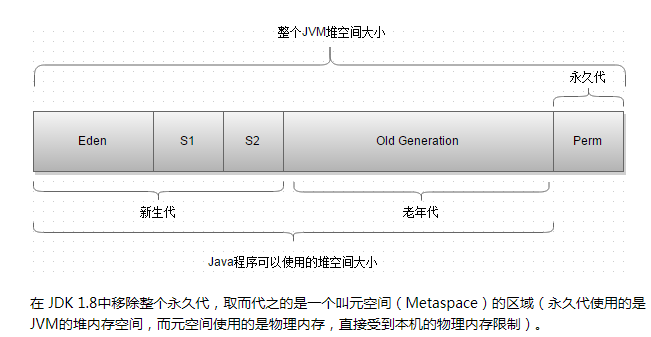


线程私有的: PC(程序计数器):存一些指令; 虚拟机栈: 里面含有栈帧(局部变量表/操作数栈/动态链接/方法出口)

本地方法栈: 存本地方法; 线程共享的: 堆: 存放对象的实例;,垃圾回收器主要管理的区域 方法区: jdk 1.8以后取消; 主要存放: 类的信息/常量/静态变量/即时编译的代码;

1-2 jdk1.8之后为啥取消方法区? 使用元空间代替;

取消原因:永久代内存经常不够用/发生内存泄露,爆出OutOfMemoryError;元空间不在虚拟机中,而是使用本地内存,不局限于JVM可以使用系统的内存;



1-3: java垃圾回收机制? 常用垃圾回收器?

1-3-1对哪些对象回收?

引用计数法 和 可达性分析;

1-3-2 在什么时候回收?

1 cpu空闲时自动回收 2 在堆内存存储满之后 3 主动调用System.gc()尝试进行回收;

1-3-3 如何回收?

标记-清除: 容易产生大量不连续的内存碎片;

复制回收: 效率低, 浪费内存空间;

标记-整理:

分代收集算法: jvm使用最多的一种算法;

说明:

新生代:尽可能快速手机生命周期短的对象; 触发: MinorGC----选用 复制算法;

老年代: 存放生命周期较长的对象;内存比新生代大很多(1:2)触发:Major GC---选用标记-清除/整理算法;



2 垃圾回收器:

串行垃圾回收器:

并行垃圾回收器:

并发标记扫描回收器(CMS):获取最短回收停顿时间为目标的收集器;

G1垃圾回收器:对于heap分配很大的中大型应用,使用G1垃圾回收期比较好;

调优:就是尽量减少Full-GC的次数;

3 java类加载机制: 双亲委派机制;

包含: 启动类加载器/扩展类加载器/应用类加载器;

寻找父类加载器 是否加载过?