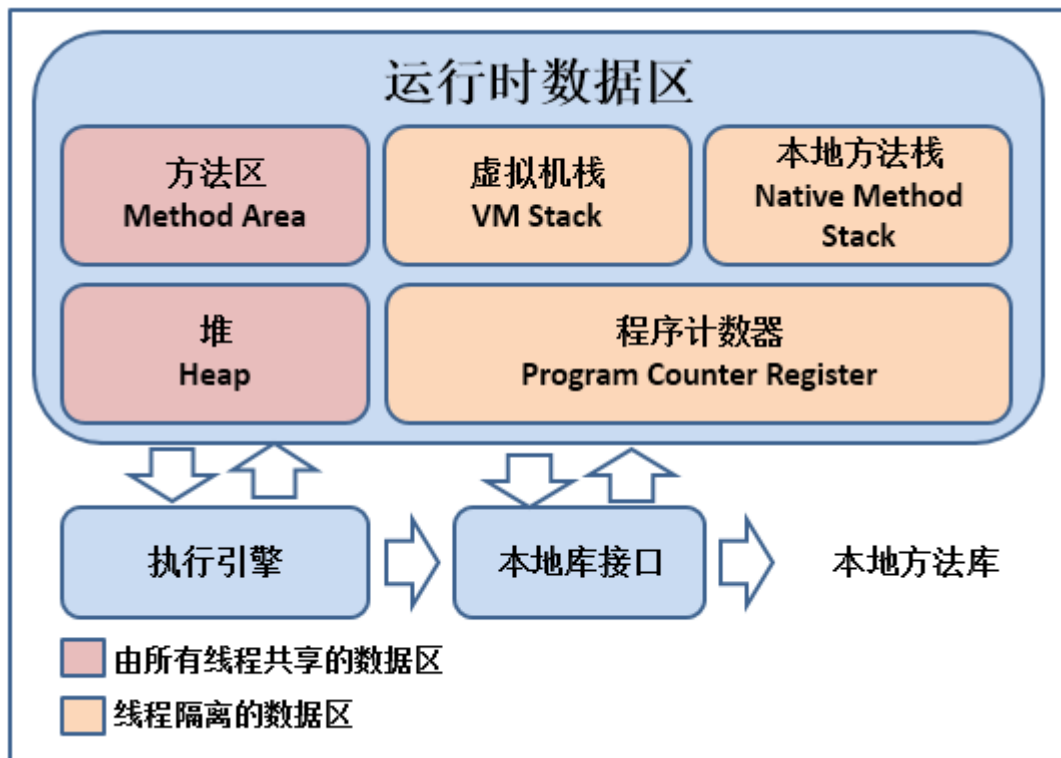


浅谈对 java 内存区域和内存异常的理解

● 运行时的数据区域



1: 程序计数器: 可以看着它为一小块内存区域, 可以看着当前线程所执行的字节码行号计数器。字节码解释器的工作就是改变该计数器的值来选取下一条需要执行的字节码指令, 分支循环, 跳转, 异常处理, 线程恢复等基础工程都有需要依赖该计数器来完成。这个计数器记录的是当前 java 方法执行时, 虚拟机字节码指令的地址。这个内存区域是唯一没有规定 `OutMemoryError` 情况的区域。线程私有, 生命周期和线程相同。

2: 虚拟机栈: 线程私有和线程生命周期相同。它描述的是 java

方法执行的内存模型。如果线程请求栈的深度大于虚拟机栈值的深度就会报错：**StackOverflowError**，如果虚拟机栈支持动态可扩展，而扩展是还是无法申请到足够的内存，就会报错：**OutMemoryError**。

3：本地方法栈：作用和虚拟机栈相同。区别：虚拟机栈执行的是 **java** 方法服务（字节码），而本地方法栈执行的是 **native** 方法服务（**native** 是一个计算机函数，一个 **Native Method** 就是一个 **Java** 调用非 [Java](#) 代码的接口。方法的实现由非 **Java** 语言实现，比如 **C** 或 **C++**。）异常和 **java** 虚拟机报的错相同。

4：Java 堆：它是 **java** 虚拟机中内存分配最大的一块区域，在 **java** 虚拟机启动时就被创建，是被所有线程共享，主要目的是几乎所有对象实例都被存放在这里分配内存，它也是 **GC**（垃圾回收器管理）的主要区域。异常:如果堆中没有完成实例分配，而且堆也无法扩展时火抛出：**OutMemoryError**。

5：方法区：它也是被所有线程共享的一块内存区域，它包含所有的 **class** 和 **static** 变量，而 **class** 和 **static** 是整个程序中唯一的元素，运行时常量池都分配在方法区中，因为有这些特性，它也被称为“永久区”，**GC** 行为在这个区很少出现，它们能被回收，只是回收的方法十分苛刻。异常：当方法区的内存无法满足时，则有：**OutMemoryError** 抛出。

