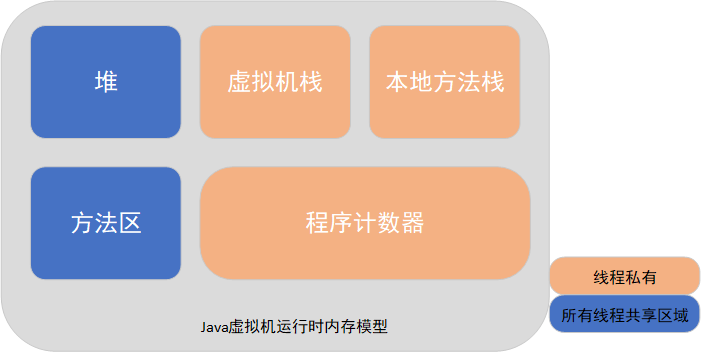
# Java内存区域



## 线程私有区域

**虚拟机栈**：Java虚拟机栈是线程私有的，生命周期与线程相同。虚拟机栈描述的是java方法执行的内存模型，与数据结构的栈结构相同，每个方法被执行的时候都会同时创建一个栈帧，并压入栈中**（这就是被称为虚拟机栈的原因）**，用于存储局部变量表，操作数栈，动态链接，方法出口等信息。

***局部变量表***：存放变量，对于对象，存的是对象实例在堆中的指针

***操作数栈***：用于计算表达式

***动态链接***：

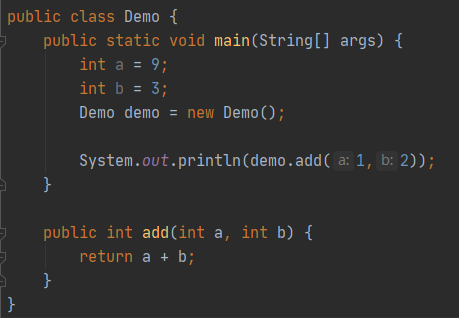
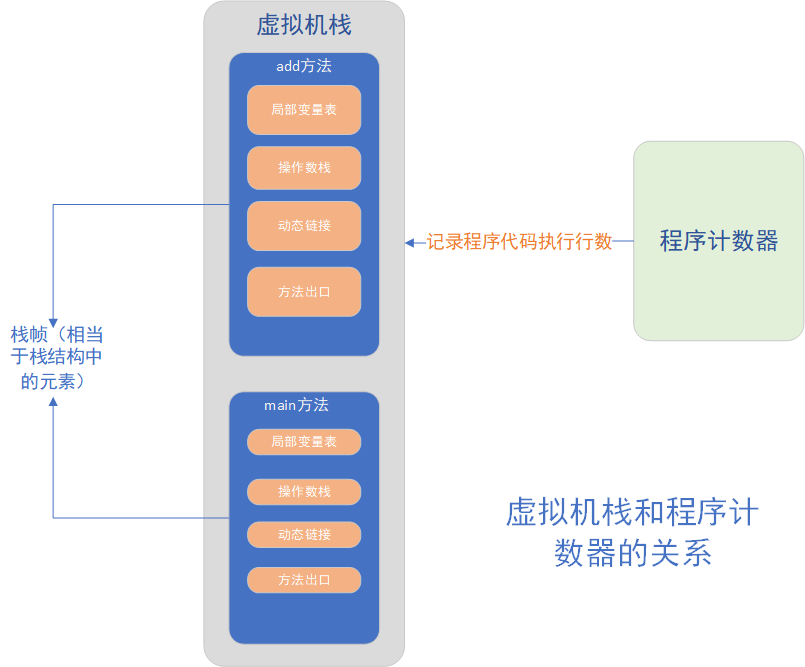
***方法出口***：如main调用add方法，方法出口中保存了main的执行位置。add方法计算执行完成后，返回main方法的某一行代码

**StackOverflowError异常**：如果线程请求的栈深度大于虚拟机所允许的深度，将抛出StackOverflowError异常。

**OutOfMemoryError异常**：如果虚拟机可以动态扩展，在扩展时无法申请到足够的内存，则抛出OutOfMemoryError异常。

**程序计数器：**程序计数器是一块较小的内存空间，作用是当前线程所执行的字节码的行号指示器。在任何时刻，一个处理器只会执行一条线程中的指令。为了线程切换后能恢复到正确的执行位置，每条线程都需要有一个独立的程序计数器。

**本地方法栈：**本地方法栈与虚拟机栈类似，区别不过是虚拟机栈为虚拟机执行Java方法服务，而本地方法栈则为虚拟机使用到的Native方法服务。



## 线程共享区域

**Java堆**：java堆是java虚拟机所管理的内存中最大的一块，java堆是被所有线程共享的一块区域，在虚拟机启动时创建，此区域的唯一目的就是存放对象实例。

Java堆是垃圾收集器管理的主要区域，因此也被称为GC堆。从内存回收角度看，由于现在收集器基本都采用了分代收集算法，所以java堆中还可以细分为新生代和老年代，在细致一点可以分为Eden空间，From Survivor空间，To Survivor空间等

**方法区：**方法区与java堆一样，是线程共享的内存区域，用于存储已经被虚拟机加载的类信息，常量，静态变量，即时编译器编译后的代码等数据。

***运行时常量池***：运行时常量池是方法区的一部分。Class文件中除了有类的版本，字段，方法，接口等描述信息外，还有一项信息是常量池，用于存放编译器生成的各种字面量和符号引用，这部分内容将在类加载后进入方法区的运行时常量池中存放。