##### 一．Jvm内存模型

###### 1. java虚拟机内存模型由下面几部分组成：

**堆：(线程共享)**

**线程栈：(线程私有)**由一个或多个栈帧构成，每个栈帧由：局部变量表、操作数栈、动态链接、方法出口。

局部变量表：程序运行的局部变量信息。

操作数栈：程序计算临时用到内存。

动态链接：存储方法名称对应的具体代码的内存地址。

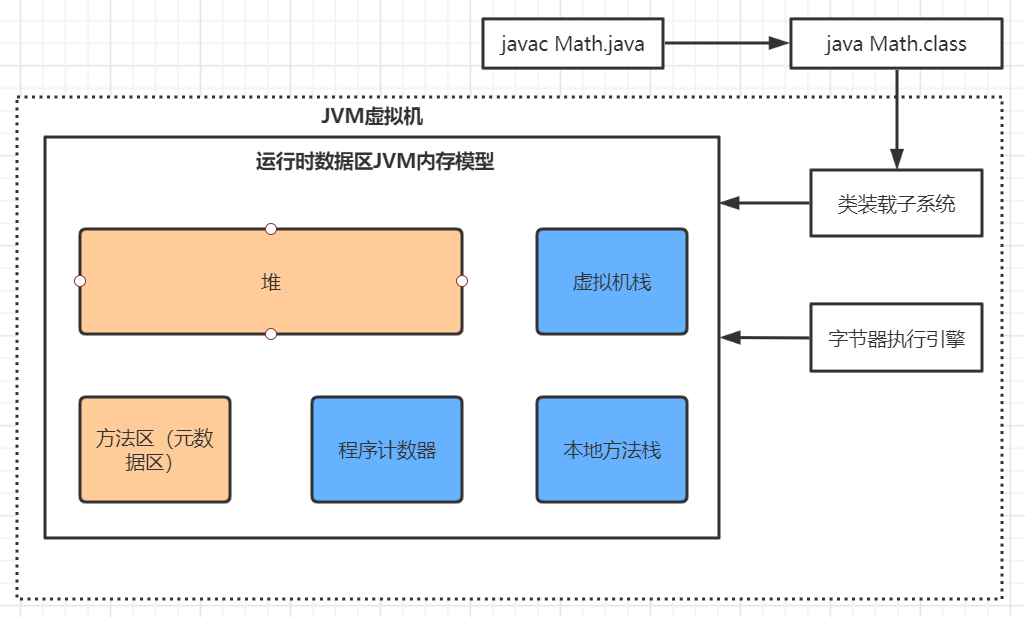
方法出口：存储调用外部方法后接着要执行代码的位置。

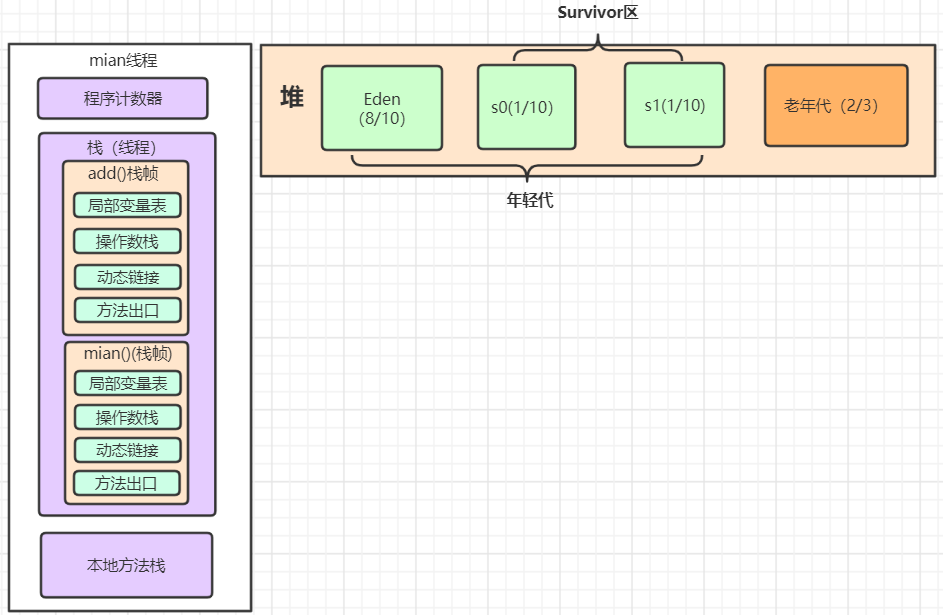
**计数器：(线程私有)**存储程序运行的位置。在多线程Cpu时间分片的情况下执行其他线后，再次执行能找到上次执行的位置接着执行。

**本地方法栈：(线程私有)：**执行本地方法的一块内存区域。本地方法就是用native修饰的方法，它会找到底层c语言类库对应的方法执行。

**方法区**（元数据区）**(线程共享)**jdk1.8用的是直接物理内存，是用来存储：静态变量、常量、类型信息。

**内存模型图：**





##### 二．垃圾回收器

1. **Gc-Roots 可达性分析算法**Minor GC将“**Gc-Roots**”对象作为起点，从这些节点开始向下搜索引用的对象，找到的对象都标记为非垃圾对象，其余未标记的都是垃圾对象**GC Roots**根节点：**线程栈的本地变量、静态变量、本地方法栈的变量**等等。

垃圾回收器会回收Eden区和非空的survivor区，将标记为非垃圾对象的一次全部移动到survivor区并将这些对象的年龄加1，并将未标记垃圾对象全陪清理，这个过程会eden区中反复发生，当对象年龄达到15后，如果对象依然存活会将这些对象移到老年代中，当老年代要满了就会发生Full GC 会对整个堆全局回收，如果由于程序bug导致回不了，最后满了就会发生OOM 内存回收异常。

StopWord (STW) 在Full GC垃圾回收器回收时发生，会停止应用线程。