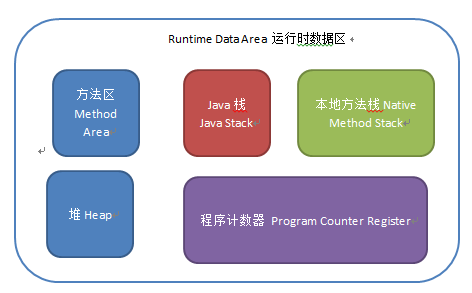
## Java堆

Java堆是被所有内存共享的一块内存区域

几乎所有的对象实例和数组都在堆上分配内存

Java堆是垃圾收集器管理的主要区域

如果在堆中没有内存完成实例分配，并且堆也无法再扩展，将会抛出OOM

## 方法区

和java堆一样，是各个内存共享的区域

用于存储被虚拟机加载的类信息，常量，静态变量，即时编译器编译的代码等

当方法区无法满足内存分配时，将抛出OOM

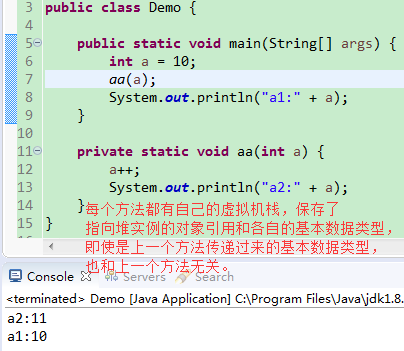
## Java虚拟机栈

线程私有，声明周期与线程相同

虚拟机栈描述的是java方法执行的内存模型：每个方法在执行的同时会创建一个栈帧，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：存放了编译器可知的各种基本数据类型和对象引用。

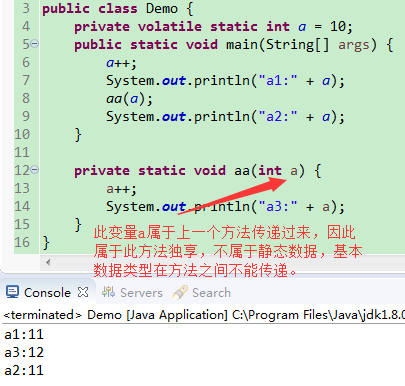
每个方法创建的基本类型都属于线程私有，用完即扔掉。如果一个基本类型int a是由上一个方法传递过来的，那么int a 的操作不会影响上一个方法。



每个方法创建的对象都在堆申请内存User user = new User()，虚拟机栈存放的是对象引用，将此对象引用user传递到下一个方法进行操作，影响的也是堆里面的对象。如果此方法结束，堆里面的实例也会被销毁。



基本数据类型在方法之间不能传递



静态数据全局共享

