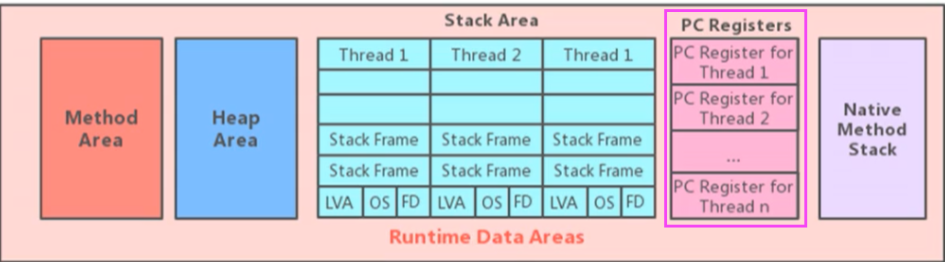
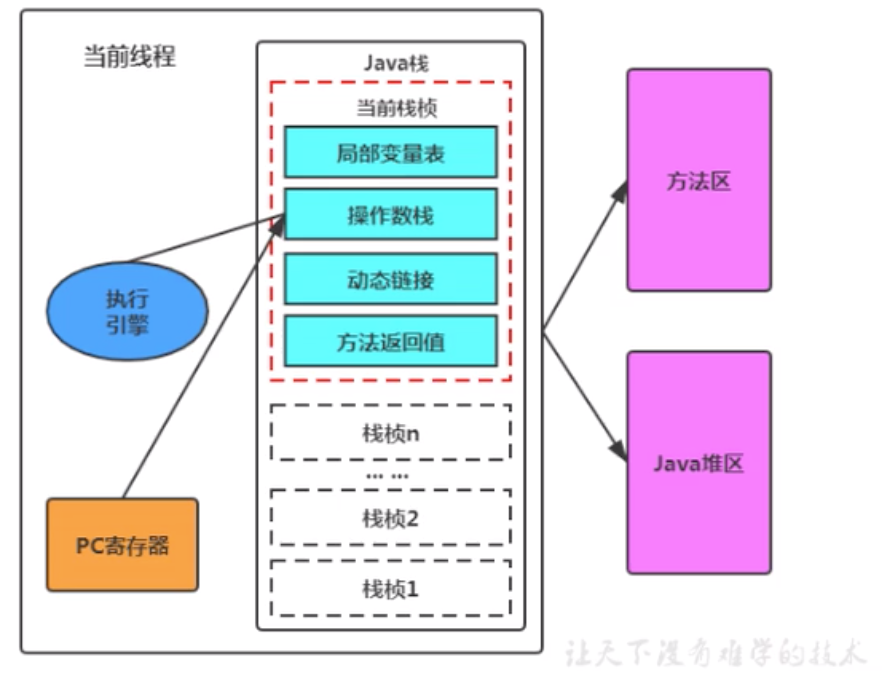
# 寄存器介绍

1. JVM中的程序计数寄存器（Program Counter Register）中，Register的命名源于CPU的寄存器，寄存器存储指令相关的现场信息。CPU只有把数据装载到寄存器才能够运行。
2. 这里，并非是广义上所指的物理寄存器，或许将其翻译为PC计数器（或指令计数器）会更加贴切（也称为程序钩子），并且也不容易引起一些不必要的误会。JVM中的PC寄存器是对物理PC寄存器的一种抽象模拟。
3. 它是一块很小的内存空间，几乎可以忽略不记。也是运行速度最快的存储区域。
4. 在JVM规范中，每个线程都有它自己的程序计数器，是线程私有的，生命周期与线程的生命周期保持一致。
5. 任何时间一个线程都只有一个方法在执行，也就是所谓的当前方法(也就是说在活动线程中，只有位于栈顶的栈帧才是有效的，称为当前栈帧，与这个栈帧相关联的方法称为当前方法。)。程序计数器会存储当前线程**正在执行的Java方法的JVM指令地址**；或者，如果是在执行native方法，则是未指定值（undefned），因为程序计数器不负责本地方法栈
6. 它是程序控制流的指示器，分支、循环、跳转、异常处理、线程恢复等基础功能都需要依赖这个计数器来完成。字节码解释器工作时就是通过改变这个计数器的值来选取下一条需要执行的字节码指令。
7. 它是唯一一个在Java虚拟机规范中没有规定任何OutofMemoryError情况的区域。而且没有垃圾回收 无OOM 无GC



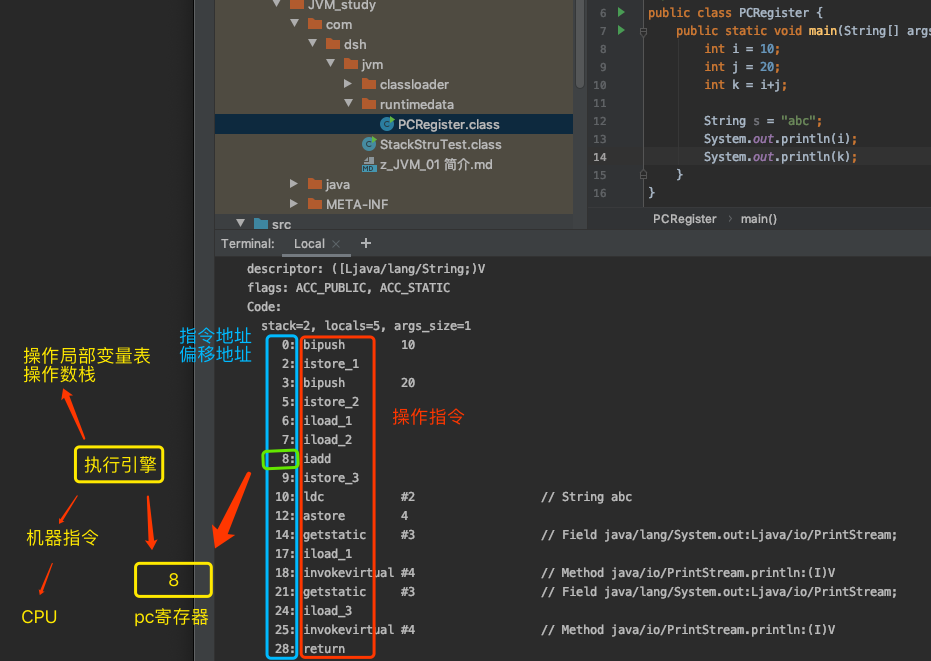
# 程序计数寄存器作用

PC寄存器是用来**存储指向下一条指令的地址**，即将要执行的指令代码。由执行引擎读取下一条指令。

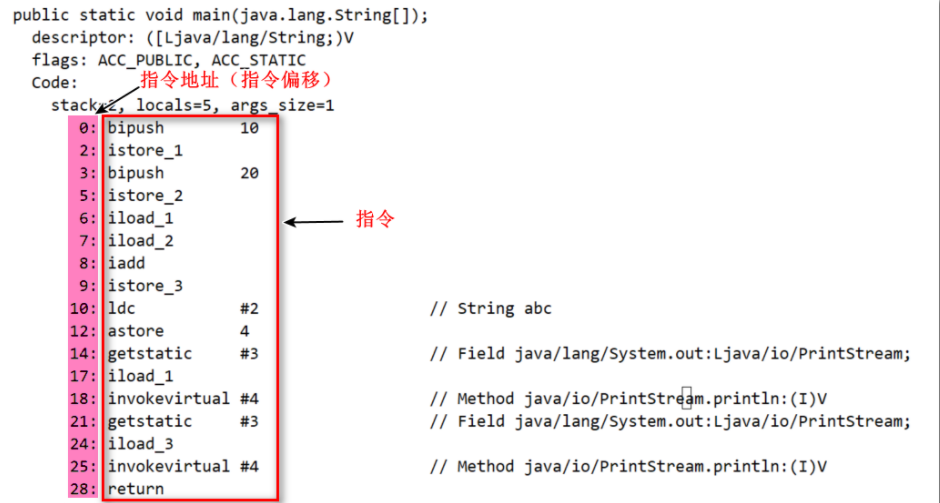


# 代码示例

利用**javap -v xxx.class**反编译字节码文件，查看指令等信息



左边的数字代表指令地址（指令偏移），即 PC 寄存器中可能存储的值，然后执行引擎读取 PC 寄存器中的值，并执行该指令

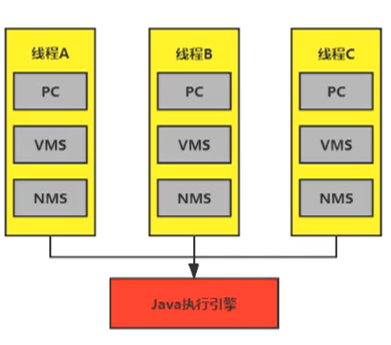


# PC寄存器面试题

**1.使用PC寄存器存储字节码指令地址有什么用呢（为什么使用PC寄存器记录当前线程的执行地址呢）**

多线程实际上是并发交替执行的，CPU需要不停的切换各个线程，当线程切换回来以后，就得知道从哪开始继续执行。

JVM的字节码解释器就需要通过改变PC寄存器的值来明确下一条应该执行什么样的字节码指令，由于PC寄存器线程私有，因此线程彼此互相不会打扰。



**简而言之：**线程并发时，记录每个线程当前执行指令的位置，防止时间片切换后，线程不知道自己执行到哪了。

**2.PC寄存器为什么会设定为线程私有？**

并发场景下，一个CPU或者多核CPU中的一个内核，任一时刻只会执行某一个线程中的指令。

CPU会不停地做任务切换，这样必然导致经常中断或恢复，而为了能够准确地记录各个线程正在执行的当前字节码指令地址，最好的办法自然是为每一个线程都分配一个PC寄存器,这样一来各个线程之间便可以进行独立计算，从而不会出现相互干扰的情况。