



图 4-10 JVM 内存的不同部分

相比之下，PC 则保存的是字节地址，对 PC 进行加减运算改变的地址都是按照字节计算的，而不再是字。对 PC 寻址和其他寄存器的寻址不同，显然 Mic-1 为 PC 提供了特殊的内存端口。请记住 PC 只有一个字节宽。对 PC 加 1 并启动读操作将得到下一个字节的内容，而对 SP 加 1 并启动读操作将得到下一个字的内容。

261

4.2.3 JVM 指令集

JVM 指令集如图 4-11 所示。每条指令都有操作码，有的还有操作数，比如内存偏移量或者常量。表中的第一列是指令的 16 进制编码。第二列是汇编语言助记符。第三列是指令功能的简单描述。

十六进制操作码	助 记 符	操作含义
0x10	BIPUSH byte	把 byte 压入栈
0x59	DUP	拷贝栈顶字节并压入栈
0xA7	GOTO offset	无条件转移
0x60	IADD	从栈顶弹出两个字，把它们的和压入栈
0x7E	IAND	从栈顶弹出两个字，把它们的逻辑“与”结果压入栈
0x99	IFEQ offset	从栈顶弹出一个字，如果是 0 则转移
0x9B	IFLT offset	从栈顶弹出一个字，如果小于 0 则转移
0x9F	IF_ICMPEQ offset	从栈顶弹出两个字，如果相等则转移
0x84	IINC varnum const	把常数加到局部变量中
0x15	ILOAD varnum	把局部变量压入栈
0xB6	INVOKEVIRTUAL disp	调用一个方法
0x80	IOR	从栈顶弹出两个字，把它们 OR 结果压入栈
0xAC	IRETURN	从过程中返回并返回一个整数值
0x36	ISTORE varnum	从栈顶弹出一个字存入局部变量
0x64	ISUB	从栈顶弹出两个字，把它们的差压入栈
0x13	LDC_W index	把常量池中的常量压入栈
0x00	NOP	什么都不做
0x57	POP	从栈顶删除一个字
0x5F	SWAP	交换栈顶的两个字
0xC4	WIDE	前缀指令，表示下一条指令带有 16 位的索引

图 4-11 JVM 指令集。操作数中的 byte、const 和 varnum 都是一个字节。

操作数中的 disp、index 和 offset 都是两个字节