

一. 常量入栈:

0x01	aconst_null	null值入栈。
0x02	iconst_m1	-1(int)值入栈。
0x03	iconst_0	0(int)值入栈。
0x04	iconst_1	1(int)值入栈。
0x05	iconst_2	2(int)值入栈。
0x06	iconst_3	3(int)值入栈。
0x07	iconst_4	4(int)值入栈。
0x08	iconst_5	5(int)值入栈。
0x09	lconst_0	0(long)值入栈。
0x0a	lconst_1	1(long)值入栈。
0x0b	fconst_0	0(float)值入栈。
0x0c	fconst_1	1(float)值入栈。
0x0d	fconst_2	2(float)值入栈。
0x0e	dconst_0	0(double)值入栈。
0x0f	dconst_1	1(double)值入栈。
0x10	bipush valuebyte	valuebyte值带符号扩展成int值入栈。
0x11	sipush valuebyte1	(valuebyte1 << 8) valuebyte2 值带符号扩展成int值
	valuebyte2	入栈
0x12	ldc indexbyte1	常量池中的常量值 (int, float, string reference, object reference) 入栈。
0x13	ldc_w indexbyte1	常量池中常量 (int, float, string reference, object reference) 入栈。(带索引)
	indexbyte2	
0x14	ldc2_w indexbyte1	常量池中常量 (long, double) 入栈。
	indexbyte2	

助记符:

ldc表示将int, float或是String类型的常量值从常量池中推送至栈顶

bipush表示将单字节 (-128 ~ 127) 的常量值推送至栈顶

sipush表示将一个短整型常量值 (-32768 ~ 32767) 推送至栈顶

iconst_1表示将int类型1推送至栈顶 (iconst_1 ~ iconst_5)

二. 局部变量值转载到栈中指令

指令码	操作码 (助记符)	操作数	描述 (栈指操作数栈)
-----	-----------	-----	-------------

0x19	(wide) aload	indexbyte	从局部变量indexbyte中装载引用类型值入栈。
0x2a	aload_0		从局部变量0中装载引用类型值入栈。
0x2b	aload_1		从局部变量1中装载引用类型值入栈。
0x2c	aload_2		从局部变量2中装载引用类型值入栈。
0x2d	aload_3		从局部变量3中装载引用类型值入栈。
0x15	(wide) iload	indexbyte	从局部变量indexbyte中装载int类型值入栈。
0x1a	iload_0		从局部变量0中装载int类型值入栈。
0x1b	iload_1		从局部变量1中装载int类型值入栈。
0x1c	iload_2		从局部变量2中装载int类型值入栈。
0x1d	iload_3		从局部变量3中装载int类型值入栈。
0x16	(wide) lload	indexbyte	从局部变量indexbyte中装载long类型值入栈。
0x1e	lload_0		从局部变量0中装载int类型值入栈。
0x1f	lload_1		从局部变量1中装载int类型值入栈。
0x20	lload_2		从局部变量2中装载int类型值入栈。
0x21	lload_3		从局部变量3中装载int类型值入栈。
0x17	(wide) fload	indexbyte	从局部变量indexbyte中装载float类型值入栈。
0x22	fload_0		从局部变量0中装载float类型值入栈。
0x23	fload_1		从局部变量1中装载float类型值入栈。
0x24	fload_2		从局部变量2中装载float类型值入栈。
0x25	fload_3		从局部变量3中装载float类型值入栈。
0x18	(wide) dload	indexbyte	从局部变量indexbyte中装载double类型值入栈。
0x26	dload_0		从局部变量0中装载double类型值入栈。
0x27	dload_1		从局部变量1中装载double类型值入栈。
0x28	dload_2		从局部变量2中装载double类型值入栈。
0x29	dload_3		从局部变量3中装载double类型值入栈。
0x32	aaload		从引用类型数组中装载指定项的值。
0x2e	iaload		从int类型数组中装载指定项的值。
0x2f	laload		从long类型数组中装载指定项的值。
0x30	faload		从float类型数组中装载指定项的值。
0x31	daload		从double类型数组中装载指定项的值。

0x33	baload	从boolean类型数组或byte类型数组中装载指定项的值（先转换为int类型值，后压栈）。
0x34	caload	从char类型数组中装载指定项的值（先转换为int类型值，后压栈）。
0x35	saload	从short类型数组中装载指定项的值（先转换为int类型值，后压栈）。

三. 将栈顶值保存到局部变量中指令

指令码	操作码（助记符）	操作数	描述（栈指操作数栈）
0x3a	(wide)astore	indexbyte	将栈顶引用类型值保存到局部变量indexbyte中。

0x4b **astore_0** 将栈顶引用类型值保存到局部变量0中。

0x4c **astore_1** 将栈顶引用类型值保存到局部变量1中。

0x4d **astore_2** 将栈顶引用类型值保存到局部变量2中。

0x4e **astore_3** 将栈顶引用类型值保存到局部变量3中。

0x36	(wide)istore	indexbyte	将栈顶int类型值保存到局部变量indexbyte中。
------	---------------------	------------------	-----------------------------

0x3b **istore_0** 将栈顶int类型值保存到局部变量0中。

0x3c **istore_1** 将栈顶int类型值保存到局部变量1中。

0x3d **istore_2** 将栈顶int类型值保存到局部变量2中。

0x3e **istore_3** 将栈顶int类型值保存到局部变量3中。

0x37	(wide)lstore	indexbyte	将栈顶long类型值保存到局部变量indexbyte中。
------	---------------------	------------------	------------------------------

0x3f **lstore_0** 将栈顶long类型值保存到局部变量0中。

0x40 **lstore_1** 将栈顶long类型值保存到局部变量1中。

0x41 **lstore_2** 将栈顶long类型值保存到局部变量2中。

0x42 **lstore_3** 将栈顶long类型值保存到局部变量3中。

0x38	(wide)fstore	indexbyte	将栈顶float类型值保存到局部变量indexbyte中。
------	---------------------	------------------	-------------------------------

0x43 **fstore_0** 将栈顶float类型值保存到局部变量0中。

0x44 **fstore_1** 将栈顶float类型值保存到局部变量1中。

0x45 **fstore_2** 将栈顶float类型值保存到局部变量2中。

0x46 **fstore_3** 将栈顶float类型值保存到局部变量3中。

0x39	(wide)dstore	indexbyte	将栈顶double类型值保存到局部变量indexbyte中。
0x47	dstore_0		将栈顶double类型值保存到局部变量0中。
0x48	dstore_1		将栈顶double类型值保存到局部变量1中。
0x49	dstore_2		将栈顶double类型值保存到局部变量2中。
0x4a	dstore_3		将栈顶double类型值保存到局部变量3中。
0x53	aastore		将栈顶引用类型值保存到指定引用类型数组的指定项。
0x4f	iastore		将栈顶int类型值保存到指定int类型数组的指定项。
0x50	lastore		将栈顶long类型值保存到指定long类型数组的指定项。
0x51	fastore		将栈顶float类型值保存到指定float类型数组的指定项。
0x52	dastore		将栈顶double类型值保存到指定double类型数组的指定项。
0x54	bastore		将栈顶boolean类型值或byte类型值保存到指定boolean类型数组或byte类型数组的指定项。
0x55	castore		将栈顶char类型值保存到指定char类型数组的指定项。
0x56	sastore		将栈顶short类型值保存到指定short类型数组的指定项。

四. wide指令

指令码	操作码（助记符）	操作数	描述（栈指操作数栈）
0xc4	wide		使用附加字节扩展局部变量索引（iinc指令特殊）。

五. 通用（无类型）栈操作指令

指令码	操作码（助记符）	操作数	描述（栈指操作数栈）
0x00	nop		空操作。
0x57	pop		从栈顶弹出一个字长的数据。
0x58	pop2		从栈顶弹出两个字长的数据。
0x59	dup		复制栈顶一个字长的数据，将复制后的数据压栈。

0x5a	dup_x1	复制栈顶一个字长的数据，弹出栈顶两个字长数据，先将复制后的数据压栈，再将弹出的两个字长数据压栈。
0x5b	dup_x2	复制栈顶一个字长的数据，弹出栈顶三个字长的数据，将复制后的数据压栈，再将弹出的三个字长的数据压栈。
0x5c	dup2	复制栈顶两个字长的数据，将复制后的两个字长的数据压栈。
0x5d	dup2_x1	复制栈顶两个字长的数据，弹出栈顶三个字长的数据，将复制后的两个字长的数据压栈，再将弹出的三个字长的数据压栈。
0x5e	dup2_x2	复制栈顶两个字长的数据，弹出栈顶四个字长的数据，将复制后的两个字长的数据压栈，再将弹出的四个字长的数据压栈。
0x5f	swap	交换栈顶两个字长的数据的位置。Java指令中没有提供以两个字长为单位的交换指令。

六. 类型转换指令

指令码	操作码 (助记符)	操作数	描述 (栈指操作数栈)
0x86	i2f		将栈顶int类型值转换为float类型值。
0x85	i2l		将栈顶int类型值转换为long类型值。
0x87	i2d		将栈顶int类型值转换为double类型值。
0x8b	f2i		将栈顶float类型值转换为int类型值。
0x8c	f2l		将栈顶float类型值转换为long类型值。
0x8d	f2d		将栈顶float类型值转换为double类型值。
0x88	l2i		将栈顶long类型值转换为int类型值。
0x89	l2f		将栈顶long类型值转换为float类型值。
0x8a	l2d		将栈顶long类型值转换为double类型值。
0x8e	d2i		将栈顶double类型值转换为int类型值。
0x90	d2f		将栈顶double类型值转换为float类型值。
0x8f	d2l		将栈顶double类型值转换为long类型值。
0x91	i2b		将栈顶int类型值截断成byte类型，后带符号扩展成int类型值入栈。
0x92	i2c		将栈顶int类型值截断成char类型值，后带符号扩展成int类型值入栈。

0x93 i2s 将栈顶int类型值截断成short类型值，后带符号扩展成int类型值入栈。

七. 整数运算

指令码	操作码 (助记符)	操作数	描述 (栈指操作数栈)
0x60	iadd		将栈顶两int类型数相加，结果入栈。
0x64	isub		将栈顶两int类型数相减，结果入栈。
0x68	imul		将栈顶两int类型数相乘，结果入栈。
0x6c	idiv		将栈顶两int类型数相除，结果入栈。
0x70	irem		将栈顶两int类型数取模，结果入栈。
0x74	ineg		将栈顶int类型值取负，结果入栈。
0x61	ladd		将栈顶两long类型数相加，结果入栈。
0x65	lsub		将栈顶两long类型数相减，结果入栈。
0x69	lmul		将栈顶两long类型数相乘，结果入栈。
0x6d	ldiv		将栈顶两long类型数相除，结果入栈。
0x71	lrem		将栈顶两long类型数取模，结果入栈。
0x75	lneg		将栈顶long类型值取负，结果入栈。
0x84	(wide) iinc	indexbyte	将整数值constbyte加到indexbyte指定的int类型constbyte 的局部变量中。

八. 浮点运算

指令码	操作码 (助记符)	操作数	描述 (栈指操作数栈)
0x62	fadd		将栈顶两float类型数相加，结果入栈。
0x66	fsub		将栈顶两float类型数相减，结果入栈。
0x6a	fmul		将栈顶两float类型数相乘，结果入栈。
0x6e	fdiv		将栈顶两float类型数相除，结果入栈。
0x72	frem		将栈顶两float类型数取模，结果入栈。
0x76	fneg		将栈顶float类型值取反，结果入栈。
0x63	dadd		将栈顶两double类型数相加，结果入栈。
0x67	dsub		将栈顶两double类型数相减，结果入栈。
0x6b	dmul		将栈顶两double类型数相乘，结果入栈。

0x6f	ddiv	将栈顶两double类型数相除，结果入栈。
0x73	drem	将栈顶两double类型数取模，结果入栈。
0x77	dneg	将栈顶double类型值取负，结果入栈。

九. 逻辑运算——移位运算

指令码	操作码（助记符）	操作数	描述（栈指操作数栈）
0x78	ishl		左移int类型值。
0x79	lshl		左移long类型值。
0x7a	ishr		算术右移int类型值。
0x7b	lshr		算术右移long类型值。
0x7c	iushr		逻辑右移int类型值。
0x7d	lushr		逻辑右移long类型值。

十. 逻辑运算——按位布尔运算

指令码	操作码（助记符）	操作数	描述（栈指操作数栈）
0x73	iand		对int类型按位与运算。
0x7f	land		对long类型的按位与运算。
0x80	ior		对int类型的按位或运算。
0x81	lor		对long类型的按位或运算。
0x82	ixor		对int类型的按位异或运算。
0x83	lxor		对long类型的按位异或运算。

十一. 控制流指令——条件跳转指令

指令码	操作码（助记符）	操作数	描述（栈指操作数栈）
0x99	ifeq	branchbyte1 branchbyte2	若栈顶int类型值为0则跳转。
0x9a	ifne	branchbyte1 branchbyte2	若栈顶int类型值不为0则跳转。
0x9b	iflt	branchbyte1 branchbyte2	若栈顶int类型值小于0则跳转。
0x9e	ifle	branchbyte1 branchbyte2	若栈顶int类型值小于等于0则跳转。
0x9d	ifgt	branchbyte1	若栈顶int类型值大于0则跳转。

		branchbyte2	
0x9c	ifge	branchbyte1	若栈顶int类型值大于等于0则跳转。
		branchbyte2	
0x9f	if_icmpeq	branchbyte1	若栈顶两int类型值相等则跳转。
		branchbyte2	
0xa0	if_icmpne	branchbyte1	若栈顶两int类型值不相等则跳转。
		branchbyte2	
0xa1	if_icmplt	branchbyte1	若栈顶两int类型值前小于后则跳转。
		branchbyte2	
0xa4	if_icmple	branchbyte1	若栈顶两int类型值前小于等于后则跳
转。			
		branchbyte2	
0xa3	if_icmpgt	branchbyte1	若栈顶两int类型值前大于后则跳转。
		branchbyte2	
0xa2	if_icmpge	branchbyte1	若栈顶两int类型值前大于等于后则跳
转。			
		branchbyte2	
0xc6	ifnull	branchbyte1	若栈顶引用值为null则跳转。
		branchbyte2	
0xc7	ifnonnull	branchbyte1	若栈顶引用值不为null则跳转。
		branchbyte2	
0xa5	if_acmpeq	branchbyte1	若栈顶两引用类型值相等则跳转。
		branchbyte2	
0xa6	if_acmpne	branchbyte1	若栈顶两引用类型值不相等则跳转。
		branchbyte2	

十二. 控制流指令——比较指令

指令码	操作码（助记符）	操作数	描述（栈指操作数栈）
0x94	lcmp		比较栈顶两long类型值，前者大，1入栈；相等，0入栈；后者大，-1入栈。
0x95	fcmpl		比较栈顶两float类型值，前者大，1入栈；相等，0入栈；后者大，-1入栈；有NaN存在，-1入栈。

0x96	fcmpg	比较栈顶两float类型值，前者大，1入栈；相等，0入栈；后者
		大，-1入栈；有NaN存在，-1入栈。
0x97	dcmpl	比较栈顶两double类型值，前者大，1入栈；相等，0入栈；后者
		大，-1入栈；有NaN存在，-1入栈。
0x98	dcmpg	比较栈顶两double类型值，前者大，1入栈；相等，0入栈；后者
		大，-1入栈；有NaN存在，-1入栈。

十三. 控制流指令——无条件跳转指令

指令码	操作码（助记符）	操作数	描述（栈指操作数栈）
0xa7	goto	branchbyte1 branchbyte2	无条件跳转到指定位置。
0xc8	goto_w	branchbyte1 branchbyte2 branchbyte3 branchbyte4	无条件跳转到指定位置（宽索引）。

十四. 控制流指令——表跳转指令

指令码	操作码（助记符）	操作数	描述（栈指操作数栈）
0xaa	tableswitch	<0-3bytepad> defaultbyte1 defaultbyte2 defaultbyte3 defaultbyte4 lowbyte1 lowbyte2 lowbyte3 lowbyte4 highbyte1 highbyte2 highbyte3 highbyte4	通过索引访问跳转表，并跳转。

		jump offsets...	
0xab	lookupswitch	<0-3bytepad>	通过键值访问跳转表，并跳转。
		defaultbyte1	
		defaultbyte2	
		defaultbyte3	
		defaultbyte4	
		npairs1	
		npairs2	
		npairs3	
		npairs4	
		match offsets	

十五. 控制流指令——异常和finally

指令码	操作码（助记符）	操作数	描述（栈指操作数栈）
0xbf	athrow		抛出异常。
0xa8	jsr	branchbyte1 branchbyte2	跳转到子例程序。
0xc9	jsr_w	branchbyte1 branchbyte2 branchbyte3 branchbyte4	跳转到子例程序（宽索引）。
0xa9	(wide)ret	indexbyte	返回子例程序。

十六. 对象操作指令

指令码	操作码（助记符）	操作数	描述（栈指操作数栈）
0xbb	new	indexbyte1 indexbyte2	创建新的对象实例。
0xc0	checkcast	indexbyte1 indexbyte	类型强转。
0xc1	instanceof	indexbyte1 indexbyte2	判断类型。
0xb4	getfield	indexbyte1 indexbyte2	获取对象字段的值。
0xb5	putfield	indexbyte1	给对象字段赋值。

		indexbyte2	
0xb2	getstatic	indexbyte1	获取静态字段的值。
		indexbyte2	
0xb3	putstatic	indexbyte1	给静态字段赋值。
		indexbyte2	

十七. 数组操作指令

指令码	操作码（助记符）	操作数	描述（栈指操作数栈）
0xbc	newarray	atype	创建type类型的数组。
0xbd	anewarray	indexbyte1	创建引用类型的数组。
		indexbyte2	
0xbe	arraylength		获取一维数组的长度。
0xc5	multianewarray	indexbyte1	创建dimension维度的数组。
		indexbyte2	
		dimension	

十八. 方法调用指令

指令码	操作码（助记符）	操作数	描述（栈指操作数栈）
0xb7	invokespecial	indexbyte1	编译时方法绑定调用方法。
		indexbyte2	
0xb6	invokevirtual	indexbyte1	运行时方法绑定调用方法。
		indexbyte2	
0xb8	invokestatic	indexbyte1	调用静态方法。
		indexbyte2	
0xb9	invokeinterface	indexbyte1	调用接口方法。
		indexbyte2	
		count	
		0	

十九. 方法返回指令

指令码	操作码（助记符）	操作数	描述（栈指操作数栈）
0xac	ireturn		返回int类型值。
0xad	lreturn		返回long类型值。

0xae	freturn	返回float类型值。
0xaf	dreturn	返回double类型值。
0xb0	areturn	返回引用类型值。
0xb1	return	void函数返回。

二十. 线程同步指令

指令码	操作码 (助记符)	操作数	描述 (栈指操作数栈)
0xc2	monitorenter		进入并获得对象监视器。
0xc3	monitorexit		释放并退出对象监视器。