操作码助记符

本章给出了从 Java 虚拟机指令操作码(包括保留操作码(§6.2))到由这些操作码表示的指令的助记符的映射。

在 Java SE 7 之前没有使用操作码值 186。



常量			加雪	散		存储	
(0x0)	nop	21	(0x15)	iload	54	(0x36)	istore
(0x01)	aconst_null	22	(0x16)	lload	55	(0x37)	lstore
(0x02)	iconst_m1	23	(0x17)	fload	56	(0x38)	fstore
(0x03)	iconst_0	24	(0x18)	dload	57	(0x39)	dstore
(0x04)	iconst_1	25	(0x19)	aload	58	(0x3a)	astore
(0x05)	iconst_2	26	(0x1a)	$iload_0$	59	(0x3b)	$istore_0$
(0x06)	iconst_3	27	(0x1b)	$iload_1$	60	(0x3c)	istore_1
(0x07)	iconst_4	28	(0x1c)	iload_2	61	(0x3d)	$istore_2$
(0x08)	iconst_5	29	(0x1d)	iload_3	62	(0x3e)	istore_3
(0x09)	$lconst_0$	30	(0x1e)	$lload_0$	63	(0x3f)	$lstore_0$
(0x0a)	lconst_1	31	(0x1f)	lload_1	64	(0x40)	$lstore_1$
(0x0b)	fconst_0	32	(0x20)	lload_2	65	(0x41)	$lstore_2$
(0x0c)	fconst_1	33	(0x21)	lload_3	66	(0x42)	lstore_3
(0x0d)	fconst_2	34	(0x22)	$fload_0$	67	(0x43)	fstore_0
(0x0e)	$dconst_0$	35	(0x23)	fload_1	68	(0x44)	fstore_1
(0x0f)	dconst_1	36	(0x24)	fload_2	69	(0x45)	fstore_2
(0x10)	bipush	37	(0x25)	fload_3	70	(0x46)	fstore_3
(0x11)	sipush	38	(0x26)	$dload_0$	71	(0x47)	$dstore_0$
(0x12)	ldc	39	(0x27)	$dload_1$	72	(0x48)	$dstore_1$
(0x13)	ldc_w	40	(0x28)	$dload_2$	73	(0x49)	$dstore_2$
(0x14)	ldc2_w	41	(0x29)	dload_3	74	(0x4a)	$dstore_3$
		42	(0x2a)	$aload_0$	75	(0x4b)	$astore_0$
		43	(0x2b)	aload_1	76	(0x4c)	astore_1
		44	(0x2c)	aload_2	77	(0x4d)	astore_2
		45	(0x2d)	aload_3	78	(0x4e)	astore_3
		46	(0x2e)	iaload	79	(0x4f)	iastore
		47	(0x2f)	laload	80	(0x50)	lastore
		48	(0x30)	faload	81	(0x51)	fastore
		49	(0x31)	daload	82	(0x52)	dastore
		50	(0x32)	aaload	83	(0x53)	aastore
		51	(0x33)	baload	84	(0x54)	bastore
		52	(0x34)	caload	85	(0x55)	castore
		53	(0x35)	saload	86	(0x56)	sastore
	(0x00) (0x01) (0x02) (0x03) (0x04) (0x05) (0x06) (0x07) (0x08) (0x00) (0x0b) (0x0c) (0x0d) (0x0e) (0x0f) (0x10) (0x11) (0x12) (0x13)	(0x00) nop (0x01) aconst_null (0x02) iconst_ml (0x03) iconst_0 (0x04) iconst_1 (0x05) iconst_2 (0x06) iconst_3 (0x07) iconst_4 (0x08) iconst_5 (0x09) lconst_0 (0x0a) lconst_1 (0x0b) fconst_0 (0x0c) fconst_1 (0x0d) fconst_2 (0x0e) dconst_1 (0x0f) dconst_1 (0x10) bipush (0x11) sipush (0x12) ldc (0x13) ldc_w	(0x00) nop 21 (0x01) aconst_null 22 (0x02) iconst_ml 23 (0x03) iconst_0 24 (0x04) iconst_1 25 (0x05) iconst_2 26 (0x06) iconst_3 27 (0x07) iconst_4 28 (0x08) iconst_5 29 (0x09) lconst_0 30 (0x0a) lconst_1 31 (0x0b) fconst_0 32 (0x0c) fconst_1 33 (0x0d) fconst_2 34 (0x0e) dconst_0 35 (0x10) bipush 37 (0x11) sipush 38 (0x12) ldc 39 (0x13) ldc_w 40 (0x14) ldc2_w 41 45 46 47 48 49 50 51 52	(0x00) nop 21 (0x15) (0x01) aconst_null 22 (0x16) (0x02) iconst_ml 23 (0x17) (0x03) iconst_0 24 (0x18) (0x04) iconst_1 25 (0x19) (0x05) iconst_2 26 (0x1a) (0x06) iconst_3 27 (0x1b) (0x07) iconst_4 28 (0x1c) (0x08) iconst_5 29 (0x1d) (0x09) lconst_0 30 (0x1e) (0x0a) lconst_1 31 (0x1f) (0x0b) fconst_0 32 (0x20) (0x0c) fconst_1 33 (0x21) (0x0d) fconst_2 34 (0x22) (0x0e) dconst_0 35 (0x23) (0x10) bipush 37 (0x25) (0x11) sipush 38 (0x26) (0x12) ldc 39 (0x27) (0x13) ldc_w 40 (0x28) (0x14) ldc2_w 41 (0x29) 42 (0x2a) 43 (0x2b) 44 (0x2c) 45 (0x2d) 46 (0x2e) 47 (0x2f) 48 (0x30) 49 (0x31) 50 (0x32) 51 (0x33) 52 (0x34)	(0x00) nop	(0x00) nop	(0x00) nop 21 (0x15) iload 54 (0x36) (0x01) aconst_null 22 (0x16) lload 55 (0x37) (0x02) iconst_ml 23 (0x17) fload 56 (0x38) (0x03) iconst_0 24 (0x18) dload 57 (0x39) (0x04) iconst_1 25 (0x19) aload 58 (0x3a) (0x05) iconst_2 26 (0x1a) iload_0 59 (0x3b) (0x06) iconst_3 27 (0x1b) iload_1 60 (0x3c) (0x07) iconst_4 28 (0x1c) iload_2 61 (0x3d) (0x08) iconst_5 29 (0x1d) iload_3 62 (0x3e) (0x09) lconst_0 30 (0x1e) lload_0 63 (0x3f) (0x0a) lconst_1 31 (0x1f) lload_1 64 (0x40) (0x0a) lconst_0 32 (0x20) lload_2 65 (0x41) (0x0b) fconst_0 32 (0x20) lload_2 65 (0x41) (0x0c) fconst_1 33 (0x21) lload_3 66 (0x42) (0x0d) fconst_2 34 (0x22) fload_0 67 (0x43) (0x0e) dconst_0 35 (0x23) fload_1 68 (0x44) (0x0f) dconst_1 36 (0x24) fload_2 69 (0x45) (0x10) bipush 37 (0x25) fload_3 70 (0x46) <

	栈			数学		转换	
87	(0x57)	pop	96	(0x60)	iadd	133 (0x85)	i2l
88	(0x58)	pop2	97	(0x61)	ladd	134 (0x86)	i2f
89	(0x59)	dup	98	(0x62)	fadd	135 (0x87)	i2d
90	(0x5a)	dup_x1	99	(0x63)	dadd	136 (0x88)	l2i
91 92	(0x5b) (0x5c)	dup_x2 dup2	100 101	(0x64) (0x65)		137 (0x89) 138 (0x8a)	12f 12d
93	(0x5d)	$dup2_x1$	102	(0x66)	fsub	139 (0x8b)	f2i
94	(0x5e)	$dup2_x2$	103	(0x67)	dsub	140 (0x8c)	f2l
95	(0x5f)	swap	104	(0x68)	imul	141 (0x8d)	f2d
			105	(0x69)	lmul	142 (0x8e)	d2i
			106	(0x6a)	fmul	143 (0x8f)	d2l
			107	(0x6b)	dmul	144 (0x90)	d2f
			108	(0x6c)	idiv	145 (0x91)	i2b
			109	(0x6d)	ldiv	146 (0x92)	i2c
			110	(0x6e)	fdiv	147 (0x93)	i2s
			111	(0x6f)	ddiv		
			112	(0x70)	irem		
			113	(0x71)	lrem		
			114	(0x72)	frem		
			115	(0x73)	drem	A	
			116		ineg		
			117		lneg	(O)	
			118			9//	
				(0x77)		(/	
				(0x78) (0x79)			
				(0x73)			
			123	(0x7a)			
				(0x7c)			
			125	(0x7d)			
			126	(0x7e)	*****		
			127				
			128	(0x80)	ior		
			129	(0x81)	lor		
			130	(0x82)	ixor		
			131	(0x83)	lxor		
			132	(0x84)	iinc		

比较		引用		
148 (0x94)	lcmp	178	(0xb2)	getstatic
149 (0x95)	fcmpl	179	(0xb3)	putstatic
150 (0x96)	fcmpg	180	(0xb4)	getfield
151 (0x97)	dcmpl	181	(0xb5)	putfield
152 (0x98)	dcmpg	182	(0xb6)	invokevirtual
153 (0x99)	ifeq	183	(0xb7)	invokespecial
154 (0x9a)	ifne	184	(0xb8)	invokestatic
155 (0x9b)	iflt		(0xb9)	invokeinterface
156 (0x9c)	ifge	186	(0xba)	invokedynamic
157 (0x9d)	ifgt	187	(0xbb)	new
158 (0x9e)	ifle	188	(0xbc)	newarray
159 (0x9f)	if_icmpeq	189	(0xbd)	anewarray
160 (0xa0)	if_icmpne	190	(0xbe)	arraylength
161 (0xa1)	if_icmplt	191	(0xbf)	athrow
162 (0xa2)	if_icmpge	192	(0xc0)	checkcast
163 (0xa3)	if_icmpgt	193	(0xc1)	instanceof
164 (0xa4)	if_icmple	194	(0xc2)	monitorenter
165 (0xa5)	if_acmpeq	195	(0xc3)	monitorexit
166 (0xa6)	if_acmpne		ž	广展
ł	空制	196	(0xc4)	wide
167 (0xa7)	goto		(0xc5)	multianewarray
168 (0xa8)	jsr	198	(0xc6)	ifnull
169 (0xa9)	ret	199	(0xc7)	ifnonnull
170 (0xaa)	tableswitch	200	(0xc8)	goto_w
171 (0xab)	lookupswitch	201	(0xc9)	jsr_w
172 (0xac)	ireturn		f	呆留
173 (0xad)	lreturn	202	(0)	1 1
174 (0xae)	freturn		(0xca)	breakpoint
175 (0xaf)	dreturn	254	(0xfe)	impdep1
176 (0xb0)	areturn	255	(0xff)	impdep2
177 (0xb1)	return			