A

Appendix

Opcode Tables

In Chapter 2, you saw that Java bytecode is found in the code attribute part of the Java class file. Table A-1 lists all the possible Java bytecodes. The hex value for each opcode is shown along with the assembler-like opcode mnemonic.

Table A-1. *Java Bytecode-to-Opcode Mapping*

Opcode	Hex Value	Opcode Mnemonic
0	(0x00)	Nop
1	(0x01)	aconst_null
2	(0x02)	iconst_m1
3	(0x03)	iconst_0
4	(0x04)	iconst_1
5	(0x05)	iconst_2
6	(0x06)	iconst_3
7	(0x07)	iconst_4
8	(0x08)	iconst_5
9	(0x09)	lconst_0

Opcode	Hex Value	Opcode Mnemonic
10	(0x0a)	lconst_1
11	(0x0b)	fconst_0
12	(0x0c)	fconst_1
13	(0x0d)	fconst_2
14	(0x0e)	dconst_0
15	(0x0f)	dconst_1
16	(0x10)	bipush
17	(0x11)	sipush
18	(0x12)	ldc
19	(0x13)	ldc_w
20	(0x14)	ldc2_w
21	(0x15)	iload
22	(0x16)	lload
23	(0x17)	fload
24	(0x18)	dload
25	(0x19)	aload
26	(0x1a)	iload_0
27	(0x1b)	iload_1
28	(0x1c)	iload_2
29	(0x1d)	iload_3
30	(0x1e)	lload_0

31	(0x1f)	lload_1
32	(0x20)	lload_2
33	(0x21)	lload_3
34	(0x22)	fload_0
35	(0x23)	fload_1
36	(0x24)	fload_2
37	(0x25)	fload_3
38	(0x26)	dload_0
39	(0x27)	dload_1
40	(0x28)	dload_2
41	(0x29)	dload_3
42	(0x2a)	aload_0
43	(0x2b)	aload_1
44	(0x2c)	aload_2
45	(0x2d)	aload_3
46	(0x2e)	iaload
47	(0x2f)	laload
48	(0x30)	faload
49	(0x31)	daload
50	(0x32)	aaload
51	(0x33)	baload
52	(0x34)	caload

Opcode	Hex Value	Opcode Mnemonic
53	(0x35)	saload
54	(0x36)	istore
55	(0x37)	lstore
56	(0x38)	fstore
57	(0x39)	dstore
58	(0x3a)	astore
59	(0x3b)	istore_0
60	(0x3c)	istore_1
61	(0x3d)	istore_2
62	(0x3e)	istore_3
63	(0x3f)	lstore_0
64	(0x40)	lstore_1
65	(0x41)	lstore_2
66	(0x42)	lstore_3
67	(0x43)	fstore_0
68	(0x44)	fstore_1
69	(0x45)	fstore_2
70	(0x46)	fstore_3
71	(0x47)	dstore_0
72	(0x48)	dstore_1
73	(0x49)	dstore_2

74	(0x4a)	dstore_3
75	(0x4b)	astore_0
76	(0x4c)	astore_1
77	(0x4d)	astore_2
78	(0x4e)	astore_3
79	(0x4f)	iastore
80	(0x50)	lastore
81	(0x51)	fastore
82	(0x52)	dastore
83	(0x53)	aastore
84	(0x54)	bastore
85	(0x55)	castore
86	(0x56)	sastore
87	(0x57)	pop
88	(0x58)	pop2
89	(0x59)	dup
90	(0x5a)	dup_x1
91	(0x5b)	dup_x2
92	(0x5c)	dup2
93	(0x5d)	dup2_x1
94	(0x5e)	dup2_x2
95	(0x5f)	swap

Opcode	Hex Value	Opcode Mnemonic
96	(0x60)	iadd
97	(0x61)	ladd
98	(0x62)	fadd
99	(0x63)	dadd
100	(0x64)	isub
101	(0x65)	lsub
102	(0x66)	fsub
103	(0x67)	dsub
104	(0x68)	imul
105	(0x69)	lmul
106	(0x6a)	fmul
107	(0x6b)	dmul
108	(0x6c)	idiv
109	(0x6d)	ldiv
110	(0x6e)	fdiv
111	(0x6f)	ddiv
112	(0x70)	irem
113	(0x71)	lrem
114	(0x72)	frem
115	(0x73)	drem
116	(0x74)	ineg

117	(0x75)	lneg
118	(0x76)	fneg
119	(0x77)	dneg
120	(0x78)	ishl
121	(0x79)	lshl
122	(0x7a)	ishr
123	(0x7b)	lshr
124	(0x7c)	iushr
125	(0x7d)	lushr
126	(0x7e)	iand
127	(0x7f)	land
128	(0x80)	ior
129	(0x81)	lor
130	(0x82)	ixor
131	(0x83)	lxor
132	(0x84)	iinc
133	(0x85)	i2l
134	(0x86)	i2f
135	(0x87)	i2d
136	(0x88)	l2i
137	(0x89)	l2f
138	(0x8a)	l2d

Opcode	Hex Value	Opcode Mnemonic
139	(0x8b)	f2i
140	(0x8c)	f2l
141	(0x8d)	f2d
142	(0x8e)	d2i
143	(0x8f)	d2l
144	(0x90)	d2f
145	(0x91)	i2b
146	(0x92)	i2c
147	(0x93)	i2s
148	(0x94)	lcmp
149	(0x95)	fcmpl
150	(0x96)	fcmpg
151	(0x97)	dcmpl
152	(0x98)	dcmpg
153	(0x99)	ifeq
154	(0x9a)	ifne
155	(0x9b)	iflt
156	(0x9c)	ifge
157	(0x9d)	ifgt
158	(0x9e)	ifle
159	(0x9f)	if_icmpeq

160	(0xa0)	if_icmpne
161	(0xa1)	if_icmplt
162	(0xa2)	if_icmpge
163	(0xa3)	if_icmpgt
164	(0xa4)	if_icmple
165	(0xa5)	if_acmpeq
166	(0xa6)	if_acmpne
167	(0xa7)	goto
168	(0xa8)	jsr
169	(0xa9)	ret
170	(0xaa)	tableswitch
171	(0xab)	lookupswitch
172	(0xac)	ireturn
173	(0xad)	lreturn
174	(0xae)	freturn
175	(0xaf)	dreturn
176	(0xb0)	areturn
177	(0xb1)	return
178	(0xb2)	getstatic
179	(0xb3)	putstatic
180	(0xb4)	getfield
181	(0xb5)	putfield

Opcode	Hex Value	Opcode Mnemonic
182	(0xb6)	invokevirtual
183	(0xb7)	invokespecial
184	(0xb8)	invokestatic
185	(0xb9)	invokeinterface
186	(0xba)	invokedynamic
187	(0xbb)	new
188	(0xbc)	newarray
189	(0xbd)	anewarray
190	(0xbe)	arraylength
191	(0xbf)	athrow
192	(0xc0)	checkcast
193	(0xc1)	instanceof
194	(0xc2)	monitorenter
195	(0xc3)	monitorexit
196	(0xc4)	wide
197	(0xc5)	multianewarray
198	(0xc6)	ifnull
199	(0xc7)	ifnonnull
200	(0xc8)	goto_w
201	(0xc9)	jsr_w

Table A-2 lists all the possible Dalvik bytecodes encountered first in Chapter 3 and then throughput the book. The hex value for each opcode is shown along with the assembler-like opcode mnemonic.

Table A-2. *Dalvik Bytecode-to-Opcode Mapping*

Opcode	Hex Value	Opcode Mnemonic
1	(0x00)	Nop
2	(0x01)	move vx, vy
3	(0x02)	move/from16 vx, vy
4	(0x03)	move/16
5	(0x04)	move-wide
6	(0x05)	move-wide/from16 vx, vy
7	(0x06)	move-wide/16
8	(0x07)	move-object vx, vy
9	(0x08)	move-object/from16 vx, vy
10	(0x09)	move-object/16
11	(0x0A)	move-result vx
12	(0x0B)	move-result-wide vx
13	(0x0C)	move-result-object vx
14	(0x0D)	move-exception vx
15	(0x0E)	return-void
16	(0x0F)	return vx
17	(0x10)	return-wide vx
18	(0x11)	return-object vx
19	(0x12)	const/4 vx, lit4

Opcode	Hex Value	Opcode Mnemonic
20	(0x13)	const/16 vx, lit16
21	(0x14)	const vx, lit32
22	(0x15)	const/high16 v0, lit16
23	(0x16)	const-wide/16 vx, lit16
24	(0x17)	const-wide/32 vx, lit32
25	(0x18)	const-wide vx, lit64
26	(0x19)	const-wide/high16 vx, lit16
27	(0x1A)	const-string vx, string_id
28	(0x1B)	const-string-jumbo
29	(0x1C)	const-class vx, type_id
30	(0x1D)	monitor-enter vx
31	(0x1E)	monitor-exit
32	(0x1F)	check-cast vx, type_id
33	(0x20)	instance-of vx, vy, type_id
34	(0x21)	array-length vx, vy
35	(0x22)	new-instance vx, type
36	(0x23)	new-array vx, vy, type_id
37	(0x24)	filled-new-array {parameters}, type_id
38	(0x25)	filled-new-array-range {vx..vy}, type_id
39	(0x26)	fill-array-data vx, array_data_offset
40	(0x27)	throw vx

41	(0x28)	goto target
42	(0x29)	goto/16 target
43	(0x2A)	goto/32 target
44	(0x2B)	packed-switch vx, table
45	(0x2C)	sparse-switch vx, table
46	(0x2D)	cmpl-float
47	(0x2E)	cmpg-float vx, vy, vz
48	(0x2F)	cmpl-double vx, vy, vz
49	(0x30)	cmpg-double vx, vy, vz
50	(0x31)	cmp-long vx, vy, vz
51	(0x32)	if-eq vx, vy, target
52	(0x33)	if-ne vx, vy, target
53	(0x34)	if-lt vx, vy, target
54	(0x35)	if-ge vx, vy, target
55	(0x36)	if-gt vx, vy, target
56	(0x37)	if-le vx, vy, target
57	(0x38)	if-eqz vx, target
58	(0x39)	if-nez vx, target
59	(0x3A)	if-ltz vx, target
60	(0x3B)	if-gez vx, target
61	(0x3C)	if-gtz vx, target
62	(0x3D)	if-lez vx, target

Opcode	Hex Value	Opcode Mnemonic
63	(0x3E)	unused_3E
64	(0x3F)	unused_3F
65	(0x40)	unused_40
66	(0x41)	unused_41
67	(0x42)	unused_42
68	(0x43)	unused_43
69	(0x44)	aget vx, vy, vz
70	(0x45)	aget-wide vx, vy, vz
71	(0x46)	aget-object vx, vy, vz
72	(0x47)	aget-boolean vx, vy, vz
73	(0x48)	aget-byte vx, vy, vz
74	(0x49)	aget-char vx, vy, vz
75	(0x4A)	aget-short vx, vy, vz
76	(0x4B)	aput vx, vy, vz
77	(0x4C)	aput-wide vx, vy, vz
78	(0x4D)	aput-object vx, vy, vz
79	(0x4E)	aput-boolean vx, vy, vz
80	(0x4F)	aput-byte vx, vy, vz
81	(0x50)	aput-char vx, vy, vz
82	(0x51)	aput-short vx, vy, vz
83	(0x52)	iget vx, vy, field_id

84	(0x53)	iget-wide vx, vy, field_id
85	(0x54)	iget-object vx, vy, field_id
86	(0x55)	iget-boolean vx, vy, field_id
87	(0x56)	iget-byte vx, vy, field_id
88	(0x57)	iget-char vx, vy, field_id
89	(0x58)	iget-short vx, vy, field_id
90	(0x59)	iput vx, vy, field_id
91	(0x5A)	iput-wide vx, vy, field_id
92	(0x5B)	iput-object vx, vy,field_id
93	(0x5C)	iput-boolean vx, vy, field_id
94	(0x5D)	iput-byte vx, vy, field_id
95	(0x5E)	iput-char vx, vy, field_id
96	(0x5F)	iput-short vx, vy, field_id
97	(0x60)	sget vx, field_id
98	(0x61)	sget-wide vx, field_id
99	(0x62)	sget-object vx, field_id
100	(0x63)	sget-boolean vx, field_id
101	(0x64)	sget-byte vx, field_id
102	(0x65)	sget-char vx, field_id
103	(0x66)	sget-short vx, field_id
104	(0x67)	sput vx, field_id
105	(0x68)	sput-wide vx, field_id

Opcode	Hex Value	Opcode Mnemonic
106	(0x69)	sput-object vx, field_id
107	(0x6A)	sput-boolean vx, field_id
108	(0x6B)	sput-byte vx, field_id
109	(0x6C)	sput-char vx, field_id
110	(0x6D)	sput-short vx, field_id
111	(0x6E)	invoke-virtual { parameters }, methodtocall
112	(0x6F)	invoke-super {parameter}, methodtocall
113	(0x70)	invoke-direct { parameters }, methodtocall
114	(0x71)	invoke-static {parameters}, methodtocall
115	(0x72)	invoke-interface {parameters}, methodtocall
116	(0x73)	unused_73
117	(0x74)	invoke-virtual/range {vx..vy}, methodtocall
118	(0x75)	invoke-super/range
119	(0x76)	invoke-direct/range {vx..vy}, methodtocall
120	(0x77)	invoke-static/range {vx..vy}, methodtocall
121	(0x78)	invoke-interface-range
122	(0x79)	unused_79
123	(0x7A)	unused_7A
124	(0x7B)	neg-int vx, vy
125	(0x7C)	not-int vx, vy
126	(0x7D)	neg-long vx, vy

127	(0x7E)	not-long vx, vy
128	(0x7F)	neg-float vx, vy
129	(0x80)	neg-double vx, vy
130	(0x81)	int-to-long vx, vy
131	(0x82)	int-to-float vx, vy
132	(0x83)	int-to-double vx, vy
133	(0x84)	long-to-int vx, vy
134	(0x85)	long-to-float vx, vy
135	(0x86)	long-to-double vx, vy
136	(0x87)	float-to-int vx, vy
137	(0x88)	float-to-long vx, vy
138	(0x89)	float-to-double vx, vy
139	(0x8A)	double-to-int vx, vy
140	(0x8B)	double-to-long vx, vy
141	(0x8C)	double-to-float vx, vy
142	(0x8D)	int-to-byte vx, vy
143	(0x8E)	int-to-char vx, vy
144	(0x8F)	int-to-short vx, vy
145	(0x90)	add-int vx, vy, vz
146	(0x91)	sub-int vx, vy, vz
147	(0x92)	mul-int vx, vy, vz
148	(0x93)	div-int vx, vy, vz

Opcode	Hex Value	Opcode Mnemonic
149	(0x94)	rem-int vx, vy, vz
150	(0x95)	and-int vx, vy, vz
151	(0x96)	or-int vx, vy, vz
152	(0x97)	xor-int vx, vy, vz
153	(0x98)	shl-int vx, vy, vz
154	(0x99)	shr-int vx, vy, vz
155	(0x9A)	ushr-int vx, vy, vz
156	(0x9B)	add-long vx, vy, vz
157	(0x9C)	sub-long vx, vy, vz
158	(0x9D)	mul-long vx, vy, vz
159	(0x9E)	div-long vx, vy, vz
160	(0x9F)	rem-long vx, vy, vz
161	(0xA0)	and-long vx, vy, vz
162	(0xA1)	or-long vx, vy, vz
163	(0xA2)	xor-long vx, vy, vz
164	(0xA3)	shl-long vx, vy, vz
165	(0xA4)	shr-long vx, vy, vz
166	(0xA5)	ushr-long vx, vy, vz
167	(0xA6)	add-float vx, vy, vz
168	(0xA7)	sub-float vx, vy,vz
169	(0xA8)	mul-float vx, vy, vz

170	(0xA9)	div-float vx, vy, vz
171	(0xAA)	rem-float vx,vy,vz
172	(0xAB)	add-double vx, vy, vz
173	(0xAC)	sub-double vx, vy, vz
174	(0xAD)	mul-double vx, vy, vz
175	(0xAE)	div-double vx, vy, vz
176	(0xAF)	rem-double vx, vy, vz
177	(0xB0)	add-int/2addr vx, vy
178	(0xB1)	sub-int/2addr vx, vy
179	(0xB2)	mul-int/2addr vx, vy
180	(0xB3)	div-int/2addr vx, vy
181	(0xB4)	rem-int/2addr vx, vy
182	(0xB5)	and-int/2addr vx, vy
183	(0xB6)	or-int/2addr vx, vy
184	(0xB7)	xor-int/2addr vx, vy
185	(0xB8)	shl-int/2addr vx, vy
186	(0xB9)	shr-int/2addr vx, vy
187	(0xBA)	ushr-int/2addr vx, vy
188	(0xBB)	add-long/2addr vx, vy
189	(0xBC)	sub-long/2addr vx, vy
190	(0xBD)	mul-long/2addr vx, vy
191	(0xBE)	div-long/2addr vx, vy

Opcode	Hex Value	Opcode Mnemonic
192	(0xBF)	rem-long/2addr vx, vy
193	(0xC0)	and-long/2addr vx, vy
194	(0xC1)	or-long/2addr vx, vy
195	(0xC2)	xor-long/2addr vx, vy
196	(0xC3)	shl-long/2addr vx, vy
197	(0xC4)	shr-long/2addr vx, vy
198	(0xC5)	ushr-long/2addr vx, vy
199	(0xC6)	add-float/2addr vx, vy
200	(0xC7)	sub-float/2addr vx, vy
201	(0xC8)	mul-float/2addr vx, vy
202	(0xC9)	div-float/2addr vx, vy
203	(0xCA)	rem-float/2addr vx, vy
204	(0xCB)	add-double/2addr vx, vy
205	(0xCC)	sub-double/2addr vx, vy
206	(0xCD)	mul-double/2addr vx, vy
207	(0xCE)	div-double/2addr vx, vy
208	(0xCF)	rem-double/2addr vx, vy
209	(0xD0)	add-int/lit16 vx, vy, lit16
210	(0xD1)	sub-int/lit16 vx, vy, lit16
211	(0xD2)	mul-int/lit16 vx, vy, lit16
212	(0xD3)	div-int/lit16 vx, vy, lit16

213	(0xD4)	rem-int/lit16 vx, vy, lit16
214	(0xD5)	and-int/lit16 vx, vy, lit16
215	(0xD6)	or-int/lit16 vx, vy, lit16
216	(0xD7)	xor-int/lit16 vx, vy, lit16
217	(0xD8)	add-int/lit8 vx, vy, lit8
218	(0xD9)	sub-int/lit8 vx, vy, lit8
219	(0xDA)	mul-int/lit8 vx, vy, lit8
220	(0xDB)	div-int/lit8 vx, vy, lit8
221	(0xDC)	rem-int/lit8 vx, vy, lit8
222	(0xDD)	and-int/lit8 vx, vy, lit8
223	(0xDE)	or-int/lit8 vx, vy, lit8
224	(0xDF)	xor-int/lit8 vx, vy, lit8
225	(0xE0)	shl-int/lit8 vx, vy, lit8
226	(0xE1)	shr-int/lit8 vx, vy, lit8
227	(0xE2)	ushr-int/lit8 vx, vy, lit8
228	(0xE3)	unused_E3
229	(0xE4)	unused_E4
230	(0xE5)	unused_E5
231	(0xE6)	unused_E6
232	(0xE7)	unused_E7
233	(0xE8)	unused_E8
234	(0xE9)	unused_E9

Opcode	Hex Value	Opcode Mnemonic
235	(0xEA)	unused_EA
236	(0xEB)	unused_EB
237	(0xEC)	unused_EC
238	(0xED)	unused_ED
239	(0xEE)	execute-inline {parameters}, inline ID
240	(0xEF)	unused_EF
241	(0xF0)	invoke-direct-empty
242	(0xF1)	unused_F1
243	(0xF2)	iget-quick vx, vy, offset
244	(0xF3)	iget-wide-quick vx, vy, offset
245	(0xF4)	iget-object-quick vx, vy, offset
246	(0xF5)	iput-quick vx, vy, offset
247	(0xF6)	iput-wide-quick vx, vy, offset
248	(0xF7)	iput-object-quick vx, vy, offset
249	(0xF8)	invoke-virtual-quick {parameters}, vtable offset
250	(0xF9)	invoke-virtual-quick/range {parameter range}, vtable offset
251	(0xFA)	invoke-super-quick {parameters}, vtable offset
252	(0xFB)	invoke-super-quick/range {register range}, vtable offset
253	(0xFC)	unused_FC
254	(0xFD)	unused_FD

255	(0xFE)	unused_FE
256	(0xFF)	unused_FF

Index

A

B

C

D, E

L

M, N

O

P, Q

R

S

T

U

V

W, X

Y

Z