Index

A

^ (exclusive-or operator), 312–313
^ (xor (exclusive-or) operator, 362
| (bit-or operator), 362, 364
| (non-short-circuited or operator), 313
|| (logical or operator), 312
~ (logical negation operator), 362, 365
! (not operator), 312
% (modulus operator), 355
& (bit-and operator), 362, 364
&& (and operator), 312
+ (add operator), 67
+ (String concatenation), 106, 344
++ (increment operator), 144–145
— (decrement operator), 144
; (statement terminator), 39
<< (bit shift left operator), 367
= (assignment operator), 42–43
== (reference comparison operator), 346
>> (bit shift right operator), 367
>>> (unsigned bit shift right ope- rator), 367
? (wildcard character), 519
?: (ternary operator), 245–246
\n (line feed escape sequence), 105, 109
^ (bit-xor operator), 362, 365–367
^ (logical exclusive or operator), 312
@deprecated, 537–538
@Override annotation 216, 217,
325, 345, 538
@Retention annotation, 545
@return, 99
@Target annotation, 546–547
abbreviations, 60
abstract, 204–205
abstract classes, 204–205
abstract test pattern, 221–223
abstraction, 13, 188
AbstractTableModel
(javax.swing.table), 637
access modifiers, 122–124,
212–213
action methods, 133
ActionListener (java.awt.event), 587
actionPerformed (java.awt.event.Ac-
tionListener method), 587
activation, 476
active object pattern, 462
actual value, 48
adapter, 435–438, 590–591
additional bounds, parameterized
types, 525–526
advanced streams, 399
agile, 9–11
algorithms, hash code, 333–336
American Standard Code for In-
formation Interchange
(ASCII), 105
and operator (&&), 312
annotations, 537–538
@Retention, 545
@Target, 546–547
@TestMethod, 543–545, 548
array parameters, 553–554
compatibility, 561
INDEX

annotations (cont.)
  complex, 558
  default values, 557
  member-value pairs, 555
  multiple parameter support, 555–556
  package, 559–560
  single-value, 550–552
  types, 558–559
anonymous inner classes, 433–440, 471
Ant, 25, 41, 128–131, 527, 542–543
API (application programming interface), 3, 11, 426
documentation, 72
Java API Library, 368
application, 11
exceptions 276
application programming interface.
  See API
argument, 36, 38
argument list, 39
arithmetic
  BigDecimal (java.math), 350–353
  bit manipulation, 360–367
  expression evaluation order, 356–357
  functions, 368–369
  infinity, 358–359
  integers, 354
  NaN, 357–358
  numeric casting, 355–356
  numeric overflow, 359
  random numbers, 371–374
  ArithmeticException, 358
array initialization, 258–259, 262
ArrayList (java.util), 71–73, 189, 246, 510–511
arrays, 255–262, 264–266
  annotations, 553–554
  parameterized types, 530
  references, 263
Arrays.equals, 263
ASCII, 105
asList (java.util.Arrays method), 266
Assert (junit.framework), 421, 535, 537
  assert keyword, 537
  assertEquals (junit.framework.Assert method), 46–48, 175, 324, 537
  assertFalse (junit.framework.Assert method), 150
  AssertionError, 536
  assertions, 46–48, 535–536
  failure messages, 152
  JUnit, 537
  assertTrue (junit.framework.Assert method), 336
assignment, 42–43
atomic variables, 501–502
atomic wrappers, 506
attributes, 15, 51, 65
autoboxing, 254
autounboxing, 255
avoiding numeric overflow, 359
await (java.util.concurrent.locks.Condition method), 493
AWT, 35, 566, 572

B

background threads, 474
base class, 203
base class references, 220
batch files, 40
beans. See JavaBeans
Beck, Kent, 19
BigDecimal (java.math), 350–353
binary numbers, bit manipulation, 360–361
binding parameterized types, 72
  additional bounds, 525–526
  arrays, 530
checked collections, 528, 530
generic methods, 522
limitations of, 531
raw types, 526–527
reflection, 532
support, 514–515
upper bound constraints, 516–518
wildcards, 518–524
bit manipulation, 360–367
bit shift left operator. See <<
bit shift right operator. See >>
bit shifting, 367–368
bit-and operator. See &
bit-or operator. See |
bit-xor operator. See ^
BitSet (java.util), 246, 368
bitwise multiplication, 362
Bloch, Joshua, 505
BlockingQueue (java.util.concurrent), 483–484
blocks
  catch
    exceptions, 271
    rethrowing exceptions, 282–284
  finally, 285–287
  try-catch exceptions, 271
Boolean, 252
boolean expressions, 312. See also logical operators
boolean type, 150–154, 312, 362
border, 608
BorderFactory (javax.swing), 602, 609
BorderLayout (java.awt), 597–599
bounds
  arrays, 530
  parameterized types, 525–526
BoxLayout (javax.swing), 601, 681
branch, 172
break statements, 197–198, 242, 243, 244–245
BufferedReader (java.io), 387, 392, 394, 691
BufferedWriter (java.io), 387, 394
build targets, 131
build.xml, 128–131
byte type, 354
Byte, 252
byte codes, 12
byte streams, 380
ByteArrayInputStream, 391, 416
ByteArrayOutputStream, 392, 394, 416
_byte type, 354
吡_BYTE type, 354
byte type, 354
Bytes, 252
byte codes, 12
byte streams, 380
ByteArrayInputStream, 391, 416
ByteArrayOutputStream, 392, 394, 416
ByteArrayOutputStream, 392, 394, 416
C
C language, 2, 232, 471
Calendar (java.util), 87–90, 681
call by reference, 681–683
call by value, 681–683
callback, 471–472, 587
calling superclass constructors, 211–215
camel case, 59
case labels, 196–198
INDEX

casting
  numeric, 355–356
  references, 230–231
catch blocks, 271, 282–284
catches, exceptions, 270, 280–282
causes, exception, 283
char type, 103–106, 353–354
character, 103–105, 252
  literals, 104–105
  streams, 380–385
  strings, 106. See also strings
system properties, 109–110
checked collections, parameterized
types, 528–530
checked exceptions, 273–278
checked wrappers, 528
chess, 62
ChoiceFormat (java.text), 679–690
class, 14, 73
  constants, 84–86, 284
  diagram, 16
  file, 12, 27
  library, 11–12
  loader, 456, 693–694
  method, 133–136, 138–141
  variables, 137, 138–141
Class class, 442–444, 453, 667
  modifiers, 447
class keyword, 33, 167
ClassCastException, 327–328, 530, 584
ClassNotFoundException, 401, 459
classpath attribute, 131, 441–448
ClassPathTestCollector (junit.runner), 441–442
client, 83
cloning, 662–664
closing a solution. See open-closed
  principle
code smells, 94
coding standards, 61
Collection, 339, 482, 509–510
Collections Framework, 73, 510
Collections.sort, 164–166
collective code ownership, 98
collisions, 332–333
command objects, 462
comment, 96–100
comment, multiline, 97
comment, single-line, 96
Comparable, 166–168, 169–171
Comparator objects, sorting, 342
compareTo (Comparable method), 167, 169–171
compilation, 81
  Ant, 128–131
compile errors, 35, 40, 41, 45, 76, 117
compiler, 11, 26
compiler warnings, 91–92
complex annotations, 558
Component (java.awt), 574, 578, 617, 698
ComponentOrientation (java.awt), 681
composition, 445
compound assignment, 144, 362
concatenation of strings, 106–107
Condition (java.util.concurrent.
  locks), 493
conditional branching, 172. See also
  if statement
conditional expressions, 245
conditionals, boolean values, 153
Connection (java.sql), 668
connection pool, 668
console, 118
ConsoleHandler (java.util.logging), 294, 298
class variables, 137–141
classes, 84–86	enum, 364–365
constraints, upper bound, 515–518
constructors, 41–42, 576
assertions, 46–48
chaining, 170
default, 69–70
enums, 209
exceptions, 300. See also exceptions
fields (initializing), 68–69
inheritance, 218–219
overloaded, 86–87
superclasses, 211–214
Container (java.awt), 573
continue statement, 243–244
contract, design by, 220–221
contract, equality, 326
control statements, 242
break statement, 243
continue statement, 243–244
labeled break statement, 244–245
labeled continue statement, 244–245
return statement, 45–46, 173, 178
controller, 571
conventions used in book, 6
conventions, naming, 59–60
cooperative multitasking, 477
covariance, 663
creation method, 146
creation test, 64
critical section, 481
currentThread (Thread method), 494, 502
custom exceptions, 276–278

D
daemon thread, 502
data streams (IO), 395–399
character streams, 380–385
management of, 379–380
writing to files, 385–387
DataInputStream (java.io), 398
DataOutputStream (java.io), 395–397
Date (java.util), 86–91
DateFormat (java.text), 681
dates, 86–91
deadlocks, 495
debugger, 22
debugging (toString method), 343–344
decode (Integer method), 371
decrement operator, 144
deep clone, 664
default access, 122
default constructor, 69–70, 218–219
default package, 79–80
default value, annotations, 557
DefaultListModel (javax.swing), 590
DefaultTableModel (javax.swing.table), 637
dependency, 16, 188, 426
dependency inversion principle, 188
deployment (Ant), 128–131
deprecated, 87–88
deprecation warnings, 91–92
design, 10, 32, 53, 115, 147
synchronization, 506
INDEX

Design Patterns, 268, 448
do keyword, 239
do loop, 239–240, 241
doClick (javax.swing.JButton), 588
document (javax.swing.text), 620
DocumentFilter (javax.swing.text),
    620–621
domain map, 639
double type, 173–174, 177,
    355–356
Double, 252, 357, 358
downloading code, 6
driver, JDBC, 665
DriverManager (java.sql), 667
duplication, 53, 85, 111, 113–118,
    147, 168, 185, 195–196, 440,
    673
dynamic proxy class, 448–449,
    455–458
equal (java.util.Arrays method),
    263
equalities (Object method), 324–330,
    335, 346
erasure, 514–515, 531
Error, 276
escape sequence, 105
Exception, 276
exceptions, 73, 158–159, 270–287
causes, 283
checked, 273–278
finally block, 285–287
hierarchies, 275–276
logging, 290–292, 294–305
messages, 279–280
multiple, 280–282
refactoring, 287–289
rethrowing, 282–284
stack traces, 285
types, 276–278
exclusive-or (xor) operator (^),
    312–313
exercises, 6, 62, 100, 131, 160
exit (System method), 503–504,
    552
expected value, 48
expression evaluation order,
    356–357
expressions, boolean, 150–154,
    312
extending methods, 206–207
extends keyword, 33, 202, 516
extreme programming, 9

F

factory, 450
factory method, 93–94, 145–146
Feathers, Michael, 576
Fibonacci sequence, 239, 242
field, 51
  access modifiers, 122–127
  edits, 619–620
  initializers, 67
  primitive type initialization, 159–160
FIFO (first-in, first-out), 474
File (java.io), 388–389
files
  logging, 298–299
  random access, 407–410
  redirecting, 118–119
  writing to (IO), 385–387
FileHandler (java.util.logging), 294, 298–299, 302
FileInputStream (java.io), 398
FileOutputStream (java.io), 397
FileReader (java.io), 387
FileWriter (java.io), 387
filter, field, 620–626
filtered streams, 395
final keyword, 54, 84, 438–439
finalize method, 694–695
finally blocks, 285–287, 399
Float, 252, 357, 358
floating point numbers. See floats
floats, 173–175, 350
FlowLayout (java.awt), 595
flyweight pattern, 345
for loop, 236–239, 241
for-each loop, 111
  iterators, 249–250
testing, 340
formal parameter, 56–57
format (String method), 287–289
format specifier, 287–288
Formatter (java.util), 679
Formatter (java.util.logging), 299
forName (Class method), 667
Fowler, Martin, 16–17, 94, 322
Frame (java.awt), 610
frame, 567
fully qualified class name, 78
G
garbage collection, 149, 694–695
generic methods, 522
generics. See parameterized types
getClass (Object method), 284, 328
getEnv (System method), 689
getProperties (System method),
  683–684
getRuntime (Runtime method),
  689
getter method, 65
global, 134
Goldeberg, Adele, 2
green bar, 38–39
GregorianCalendar (java.util), 86,
  95
GridBagConstraints (java.awt),
  603–605
GridLayout (java.awt), 595–596
GridLayout (java.awt), 595–596
guard clause, 177, 328
H
hacking, 10
Handler (java.util.logging), 294–296
hash code, 330–337
hash tables, 313–319, 321–331
  algorithms, 333–336
  collisions, 332–333
HashMap class, 337–341
  implementation, 341–343
hashCode (Object method),
  331–337
HashMap (java.util), 314–315, 330, 332, 333, 339, 511
HashSet (java.util), 335, 339
Hashtable (java.util), 246–247, 482, 683
heavyweight process, 10
hello world, 2, 26–27, 718
hexadecimal, 43, 353
Hibernate, 665
hourglass, 647

I
i18n. See internationalization
icon, 610
IDE, 21–22
IDEA. See IntelliJ IDEA
identifier, 59
IdentityHashMap (java.util), 343
if statement, 172–173, 176–177, 311–312
ignoring test methods, 557
IllegalArgumentException, 277
Image (java.awt), 610
Icon (javax.swing), 612
immutable, 58, 107
implementing interfaces, 167, 169–171, 184, 205
implements keyword, 168, 202
import statement, 78–81, 93
static, 141–143
increment operator. See ++
incremental refactoring, 75
incrementing, 67–68, 144–145
infinite loop, 239
infinity, 358–359
inherit, 33
inheritance, 17–18, 74, 201–204
initial value, 67
initialization, 68–69, 157–160, 434, 468
inlining code, 186
inner classes, 412–413, 438
anonymous, 433–435, 440
InputStream (java.io), 389, 391
InputStreamReader (java.io), 389–390, 392, 394
instance initializers, 434
instance variable. See field
instanceof operator, 328–329, 574–575, 600
instantiate, 16
instrumentation, 695
int type, 65–68, 354
Integer, 252, 359, 370
integers
bit manipulation, 360–367
math, 354–355
integrated development environment. See IDE
IntelliJ IDEA, 21, 22, 717–732
interface keyword, 167
interface references, 187–188
interfaces, 73, 166–169, 184
InternalError, 276
internationalization, 85, 673–681
interpreter, 12
interrupt (Thread method), 486
InterruptedException, 472, 486
inverting dependences, 188
InvocationHandler (java.lang.reflect), 451, 456, 458
InvocationTargetException (java.lang.reflect), 454
invokeAndWait (javax.swing.SwingUtilities method), 649
invokeLater (javax.swing.SwingUtilities method), 649
IO (input-output), 379
  advanced streams, 399
  application testing, 393–395
  byte streams, 389
  character streams, 380–385
  data streams, 395–399
  files
    File (java.io), 388–389
    writing to, 385–387
  functionality, 422
  interfaces, 390–393
  management, 379–380
  object streams, 399–406
  random access files, 407–410
IOException (java.io), 382
isAlive (Thread method), 485
Iterable (java.util), 249–250
iteration, map, 337–341
Iterator (java.util), 247–249
iterator, 247–249

J

J2EE, 3, 698
J2ME, 3
J2SE, 3
jar command, 654–655
JAR, 34, 653–656
java command, 23–24, 27–28, 655, 656
Java, 2, 11–13
  API documentation, 72–73
  language specification, 4
  platform, 11
  SDK, 23
  version, 12, 23–24
  virtual machine. See virtual machine
Java Archive. See JAR
java.awt package, 572
java.io package, 379–380, 389
java.lang.instrument package, 695
java.lang.management package, 696
java.lang.ref package, 694
java.net package, 696
java.security package, 698
java.util package, 73
java.util.concurrent package, 483
java.util.concurrent.atomic package, 506
JAVA_HOME, 23
JavaBeans, 698
javac command, 26–27, 35, 41, 91, 527
javadoc command, 99
javadoc comment, 97–100
JavaRanch, 29
JavaServer Faces, 699
javax.swing package, 572
JAX_RPC, 699
JAXP, 699
JAXR, 699
JButton (javax.swing), 581, 587
JComponent (javax.swing), 573
JDBC, 3, 495, 664–673
JDO, 665
Jeff's Rule of Statics, 149–150
Jeffries, Ron, 576
JetBrains, 21, 717
JFormattedTextField (javax.swing), 620, 626–628
JFrame (javax.swing), 567–570, 572, 594, 610
JLabel, 573
JList (javax.swing), 581–582, 589–590, 635
JLS. See Java language specification
JMS, 3, 699
JNI, 697
INDEX

join (Thread method), 485
JPanel, 572–574
JRE, 23
JScrollPane (javax.swing), 608
JSPs, 699
JSTL, 699
JTable (javax.swing), 635, 640
JUnit, 25, 33–38, 47–49, 50, 329, 336, 343, 440–446, 537, 548
JUnit setUp method, 81–82
JUnit suite, 70–71
JUnitPerf, 336
JVM. See VM

K

Kerievsky, Joshua, 146
KeyAdapter (java.awt.event), 616
KeyListener (java.awt.event), 616

L

labeled break, 244–245
labeled continue, 244–245
layout managers, 594
lazy initialization, 201
leaks, memory, 43
length (array field), 262
Level (java.util.logging), 293
lightweight process, 10
line feed. See \n
line.separator system property, 109
LinkedBlockingQueue (java.util.concurrent), 484–486
LinkedHashMap (java.util), 343
LinkedHashSet (java.util), 343
LinkedList (java.util), 474
List (java.util), 189, 249, 510–511
listener, 471
ListModel (javax.swing), 590
literals
  boolean, 150–154
  characters, 105
  classes, 70, 84–86
  numeric, 65
  strings, 39
  local variable, 42–44
Locale (java.util), 676, 678, 681
locale, 673
localization, 673, 676–678
Lock (java.util.concurrent.locks), 492–493
locking monitors, 481–482
Log4J, 291
Logger (java.util.logging), 292–293, 304
logging, 290–305
logging handlers, 294
logging hierarchies, 304
logging levels, 302–304
logical bit operators, 361–367
logical negation. See ~
logical operators, 311–313
long type, 354
Long, 232
loops
  control statements, 242
  break statements, 243
  continue statements, 243–244
  labeled break statements, 244–245
  labeled continue statements, 244–245
  constructs, 232–233
  comparing, 240–241
do loops, 239–240
for loops, 236–239
for-each loops, 249–250
while loop, 234–235
for-each, 111, 340
lower bounds, 523–524

M

main method, 261, 393–395, 568
make command, 41
MalformedURLException (java.net), 273–274
manifest, JAR, 655–656
Map (java.util), 199–201, 341, 343, 511
map, 199–201
Map.Entry (java.util), 340
marker annotation, 550
marker interfaces, 401, 663
masks, defining, 363
Matcher (java.util.regex), 661
Math, 135, 368–370
mathematics
  BigDecimal class, 350–353
  bit manipulation, 360–367
  expression evaluation order, 356–357
  functions, 368–369
  infinity, 358–359
  integers, 354
  NaN, 357–358
  numeric casting, 355–356
  numeric overflow, 359
  primitive numerics, 353–356
  random numbers, 371–374
  wrapper classes, 370–371
member, 140
member-value pairs, 555
memory leak, 43–44
merge sort, 165
MessageFormat (java.text), 679, 680
Method (java.lang.reflect), 453, 454, 458
method, 37–38
  static, 71
methodology, 9–11
mixed case, 59
mnemonic, button, 614–615
mock objects. See mocking
model, 571
model-view-controller, 571
Modifier (java.lang.reflect), 447–448
modifiers, class, 446–448
modulus operator. See %
monitor, 481–482
mouse listener, 642
multi-line comment, 97
multidimensional arrays, 261–262
multiple exceptions, catching, 280–281
multiple return statements, 173
multiplication, bitwise, 362
multithreading, 109, 462
  atomic variables, 501
  atomic wrappers, 506
  BlockingQueue interface, 483–484
  cooperative/preemptive multi-tasking, 477
  deadlocks, 495
  groups, 505
  objects, 492–494
  priority levels, 494
Runnable interface, 480
shutting down, 502–504
stopping, 485–486
synchronization, 478–483, 506
ThreadLocal class, 495–498
Timer class, 499–500
mutual exclusion, 481
MySQL, 665, 667

N

naked type variable, 513–514
namespaces, 78–81
naming conventions, 59–61
NaN, 357–358
natural sort order, 169
navigable association, 16
nested class, 373, 412–413
networking, 696
new operator, 39–40, 42
NIO, 696–697
non-short-circuited logical operators, 313
not operator (!), 312
notify (Object method), 491–492
notifyAll (Object method), 489, 491
NotSerializableException, 402
NullPointerException, 82, 159, 257, 327, 357, 599
NumberFormat (java.text), 681
numbers, 66
  BigDecimal class, 350–353
  characters, 103–105
dates, 86–91
  floating-point, 173–174
  random, 371–374
  sorting, 171–172
  strings, 370
numeric casting, 355–356
numeric literal, 65
numeric overflow, 359
numeric wrapper classes, 370–371
numerics, primitive numeric types, 353–356

O

Object, 73–74
Object Mentor, 1
object streams, 399–407
object-oriented, 2, 12, 13–17
objects, mock, 425–432
ObjectInputStream (java.io), 400, 416
ObjectOutputStream (java.io), 400, 416, 419
octal, 105, 354
one-to-many, 75, 294
open-closed principle, 182–183, 186, 450
operating system, 11
operators, 107
  and (&&), 312
  bit-and (&), 362
  bit-or (|), 362
  exclusive-or (^), 312–313
  increment, 144
  instanceof, 328
  logical, 311–313
  logical negation (~), 362
  new, 39
  not (!), 312
  or (||), 312
  postfix, 145
  prefix, 144
  ternary, 245
  xor (^), 362
or operator (||), 312
OutOfMemoryError, 276
OutputStream (java.io), 389
OutputStreamWriter (java.io), 389–390
OutOfMemoryError, 276
OutputStream (java.io), 389
OutputStreamWriter (java.io), 389–390
outOfMemoryError, 276
OverflowException, 357
OverflowException, 357
overloaded constructor, 86–87
overloaded operator, 107
overriding, 216

P

package (keyword), 72, 78–81
package access, 122–123
package import, 93
package structure, 121
packages
  java.awt, 572
  java.io, 379–380, 389
  java.lang.instrument, 695
  java.lang.management, 696
  java.lang.ref, 694
  java.net, 696
  java.security, 698
  java.util, 73
  java.util.concurrent, 483
  java.util.concurrent.atomic, 506
pair programming, 72
parameter, 38, 56–57
parameterized types, 71–72, 74–75, 509–510
  additional bounds, 525–526
  arrays, 530
  checked collections, 528–530
  Collections Framework, 510–511
  creating, 511–513
  generic methods, 542
  limitations, 531
  raw types, 526–527
  reflection, 532
  specifying, 509
  support, 514–515
  upper bound constraints, 516–518
  wildcards, 518–524
parameters
  annotations, 555–556
  arrays, 553–554
  interface references, 187–189
  parity checking, 365
parseInt (Integer method), 370–371
path, 24
Pattern (java.util.regex), 657, 661–662
performance testing, 336–337
PermissionException (java.lang.reflect), 454, 457
persistence, 664
phantom references, 694
pi, 368
piped streams, 399
platform, 11
Point (java.awt), 617
polymorphism, 14, 181–186, 199, 220
postcondition, 220
postfix, 145
precedence rules, 715
preemptive multitasking, 477
Preference (java.util.prefs), 688
preferences, 685–689
prefix operator, 144–145
PreparedStatement (java.sql), 670–672
primitive types, 66
casting, 251
  field initialization, 159–160
  wrapper classes, 252–255
printStackTrace (Throwable method), 285
PrintStream (java.io), 389, 394
PrintWriter (java.io), 387
priority levels, threads, 494
private keyword, 57–59, 122
private constructor, 135
<table>
<thead>
<tr>
<th>Index</th>
</tr>
</thead>
<tbody>
<tr>
<td>Process, 691</td>
</tr>
<tr>
<td>process framework, 10</td>
</tr>
<tr>
<td>process instance, 10</td>
</tr>
<tr>
<td>ProcessBuilder, 689, 691–693</td>
</tr>
<tr>
<td>processes, creating, 689</td>
</tr>
<tr>
<td>profiling, Java, 345</td>
</tr>
<tr>
<td>program, 11</td>
</tr>
<tr>
<td>programmer’s editor, 21–22</td>
</tr>
<tr>
<td>programming, 11</td>
</tr>
<tr>
<td>programming by intention,</td>
</tr>
<tr>
<td>213–214, 292</td>
</tr>
<tr>
<td>project, Ant, 129</td>
</tr>
<tr>
<td>Properties (java.util), 683</td>
</tr>
<tr>
<td>properties, 667, 683</td>
</tr>
<tr>
<td>Ant, 130</td>
</tr>
<tr>
<td>system, 109–110</td>
</tr>
<tr>
<td>property files, 685</td>
</tr>
<tr>
<td>property, setting on command line,</td>
</tr>
<tr>
<td>684</td>
</tr>
<tr>
<td>protected keyword, 213–215</td>
</tr>
<tr>
<td>proxy, 448–449, 697</td>
</tr>
<tr>
<td>public keyword, 33, 122–124, 184</td>
</tr>
<tr>
<td>public interface, 66</td>
</tr>
<tr>
<td>pushback streams (java.io), 399</td>
</tr>
<tr>
<td>Reader (java.io), 390</td>
</tr>
<tr>
<td>ReadWriteLock (java.util.concurrent.locks), 492</td>
</tr>
<tr>
<td>realizes association, UML, 183</td>
</tr>
<tr>
<td>receiver, 14, 44</td>
</tr>
<tr>
<td>recursion, 229–230, 242</td>
</tr>
<tr>
<td>red bar, 36–37</td>
</tr>
<tr>
<td>ReentrantLock (java.util.concurrent.locks), 493</td>
</tr>
<tr>
<td>refactoring, 53, 94–95. See also duplication</td>
</tr>
<tr>
<td>references, 42</td>
</tr>
<tr>
<td>interface, 187–189</td>
</tr>
<tr>
<td>reflection, 284, 440–459, 532, 542, 667</td>
</tr>
<tr>
<td>regression test suite, 301</td>
</tr>
<tr>
<td>regular expressions, 266, 656–662</td>
</tr>
<tr>
<td>relational database, 664</td>
</tr>
<tr>
<td>remote method invocation. See RMI</td>
</tr>
<tr>
<td>replaceAll (String method), 659</td>
</tr>
<tr>
<td>required fields, 615</td>
</tr>
<tr>
<td>requirements. See stories</td>
</tr>
<tr>
<td>resource bundle, 85, 674–676</td>
</tr>
<tr>
<td>ResourceBundle (java.util), 674, 676, 678, 679</td>
</tr>
<tr>
<td>ResultSet (java.sql), 670, 672</td>
</tr>
<tr>
<td>ResultSetMetaData (java.sql), 672</td>
</tr>
<tr>
<td>rethrowing exceptions, 282–283</td>
</tr>
<tr>
<td>return statement, 45–46, 173, 178, 287</td>
</tr>
<tr>
<td>return type, 38, 45</td>
</tr>
<tr>
<td>annotations, 558–559</td>
</tr>
<tr>
<td>return within finally, 287</td>
</tr>
<tr>
<td>RMI, 400, 449, 697–698</td>
</tr>
<tr>
<td>Robot (java.awt), 617</td>
</tr>
<tr>
<td>rounding, BigDecimal, 352</td>
</tr>
<tr>
<td>RuleBasedCollator (java.text), 681</td>
</tr>
<tr>
<td>run (Thread method), 473, 477, 485, 490, 504, 649</td>
</tr>
<tr>
<td>Runnable, 473, 480, 659</td>
</tr>
</tbody>
</table>
Runtime, 503, 689
RuntimeException, 276
RUP, 9, 10

S

SAAJ, 699
SBCS. See single-byte character set scaling (BigDecimal class), 352
scope, static 137
Scrum, 9
SDK, 11, 23
SecureRandom (java.util), 374
sending messages, 14
sequence diagram, 476
SequenceInputStream (java.io), 399
Serializable (Java.io), 401, 402
serialization, 399–407
serialVersionUID, 404
Set (Java.util), 339
setDaemon (Thread method), 503
setProperty (System method), 694
setUp (junit.framework.TestCase method), 81–82
shallow clone, 664
Short, 252
short type, 354
short-circuited logical operators, 313
shutting down threads, 502–504
signal (java.util.concurrent.locks.Condition method), 493
signalAll (java.util.concurrent.locks.Condition method), 493
signature, 166
simple design, 147, 577
SimpleFormatter (java.util.logging), 299–300
single-byte character set, 105
single-line comment, 96
small steps, 5
soft references, 694
software development kit, 11
sorting, 163–172, 263
source file, 12
special characters, 105
split (String method), 265–266, 347, 657
SpringLayout (javax.swing), 606
spurious wakeup, 491
SQL, 664–665, 668–671, 673
SQLException, 666
SRP. See Single Responsibility Principle
Stack (java.util), 246
stack trace, 48, 159, 285
stack walkback, 159
start (Thread method), 477
state, 67, 133–134
Statement (java.sql), 670
statement, 39
static import, 141–143
static initialization block, 136
static keyword, 109, 135, 137
static method, 71, 127
static nested class, 412–413
static scope, 137
statics, use of, 147–150
stderr, 118, 680, 691
stdin, 389, 690
stdout, 118, 389, 690, 691
stereotype, 136
stop (Thread method), 485
story, 33
INDEX

strategy, 183–184
stream, 379–380
stream unique identifier. See serial
    VersionUID
StreamTokenizer (java.io), 399
StrictMath, 368
String, 39, 168
string concatenation, 106
string constant, 54
string literal, 39, 54
StringBuffer, 109
StringBuilder, 107–109
strings, 103
    concatenation, 106–107
    literals, 39
    splitting, 264–266
StringTokenizer, 264
strongly typed, 74
student information system, 6, 31
student information system stories, 32
style
    naming conventions, 60
    whitespace, 61
subclass, 33, 90, 202
subcontracting, 220–228, 537
subdirectories
    classes, 81
    creating, 121
subscripting, array, 257
suite, 70–71
Sun, 28
super keyword, 207, 212
superclass, 203
Swing, 3, 5, 35, 504, 565–567
Swing layout, 594
SwingUtilities (javax.swing), 649
switch statement, 195–199
synchronization, 478–480, 506
synchronization wrappers, 482–483
synchronized keyword, 481–482,
    488–489, 502
System, 109, 389–390, 503, 689
system environment, 689
system properties, 109
System.err, 118
System.out, 118, 119–120

T
tags. See annotations
target, Ant build, 129
TDD. See test-driven development
TDTT (test-driven truth table), 312
tearDown (junit.framework.TestCase method), 485
temp variable. See local variable
temporary variable. See local variable
ternary operator, 245–246
test class, 32
test package, 126
test suite, 70–71
test-driven development, 1–2, 4–5,
    11, 18–19, 31–32, 154–157,
    651–652
test-driven development cycle, 55
testDriven (junit.framework.TestCase), 33–34,
    46–47, 421
TestRunner (junit.awtui,
    junit.swingui, junit.textui), 35–36
tests as documentation, 154–155
TextPad, 21, 656
this keyword, 55–57, 170
Thread, 472, 473
thread groups, 505–506
thread pool, 491–492, 496
thread priorities, 494
ThreadLocal, 495–499
throw statement, 278
Throwable, 276
throws clause, 274–275
Timer (java.util), 500–501
TimerTask (java.util), 500
timestamps, 86–91, 95
title bar, 609–610
tokens, 264
tolerance, float, 175
toString (Object method), 107, 343–345, 590
trace statement, 119
transient (keyword), 402–403, 405
trapping exceptions, 271
TreeMap (java.util), 342–343
TreeSet (java.util), 342–343
trim (String method), 243
try (keyword), 271
try-catch block, 271–272
two's complement, 361
type parameter list, 511

U

UML, 16–17
abstract test, 223
activation, 476
activity diagram, 476
class box, 15
class dependency, 16
composition, 445
inheritance, 18, 34, 204
interface, 167, 184, 211
message send, 14
multiplicity, 294

V

varargs, 260–261
variables
   atomic, 501
   boolean, 150–154
   classes, 137–143
   enums, 209
variables (cont.)
  incrementing, 67
  instance, 49–52, 187–189
  local, 42–43
  naked type, 513, 531
  numeric overflow, 359
  settings, 364
  statics
    applying, 148
    rules, 149–150
  troubleshooting, 147
  switch statements, 196
Vector (java.util), 246–247, 482
view, 571
virtual machine. See VM
VM, 11, 23
void keyword, 38, 45, 68
volatile keyword, 502

W
wait (Object method), 489, 491–492
wait cursor, 647–648
wait/notify, 472, 486–492
walkback, 48
warnings, deprecation, 91–92
waterfall, 9
weak references, 694
WeakHashMap (java.util), 694
while loops, 234–235, 241
whitespace, 61, 243
wildcard capture, 522–523
wildcards, 518–523
Windows, 11, 23, 40, 50, 109, 299, 477, 689
worker threads, 474
wrapper classes, 252–255
wrapper streams, 380
wrappers, checked, 528
Writer (java.io), 390

X
XML, 319, 719. See also Ant
XMLFormatter (java.util.logging), 300
xor (^) operator, 312–313, 362–365
XP, 9, 10

Y
yield (Thread method), 475, 477

Z
ZIP file, 654