

Symbols

- (minus sign)
 - in regular expressions, 409, 411
 - operator, 55–58, 203
- operator, 55, 58, 67
- _ (underscore)
 - in constant names, 45, 115
 - in package names, 187
 - in variable names, 44
- , (comma)
 - in numbers, 200, 203, 271–272
 - in regular expressions, 419
- ;(semicolon)
 - in class paths, 193, 458
 - in statements, 28, 67
- : (colon), in class paths, 193
- ! operator, 58
- !/ separator, 516
- != operator, 55, 59–60
- ? (question mark)
 - in regular expressions, 409, 416–419, 423, 427–428
 - in wildcards, 177
- ?: operator, 55, 60
- / (forward slash)
 - file name separator, 191, 494
 - operator, 55–57, 203
- // in comments, 25
- /* in comments, 24
- ** in comments, 25
- /= operator, 55
- . (dot)
 - in class paths, 460, 503
 - in method invocations, 105
 - in numbers, 200, 203, 271–272
 - . (dot) (*continued*)
 - in regular expressions, 408, 414, 416, 426
 - in variable names, 104
 - ... (ellipsis), 95
 - ^ (caret)
 - in regular expressions, 409–410, 424, 426
 - operator, 55, 62
 - ^= operator, 55
 - ~ operator, 62
 - ' (single quote), escape sequence for, 49, 212
 - " (double quote)
 - escape sequence for, 49, 212
 - in literals, 213
 - () (parentheses)
 - in annotations, 133
 - in declarations, 90, 105
 - in expressions, 66–67
 - in regular expressions, 409, 420, 422
 - [] (square brackets)
 - in arrays, 51–52
 - in regular expressions, 409–410
 - { } (braces)
 - in arrays, 52
 - in blocks, 68, 70, 117
 - in declarations, 87–88, 90
 - in regular expressions, 409, 419
 - @ (at)
 - in annotations, 133
 - in Javadoc, 135
 - \$ (dollar sign)
 - in `DecimalFormat` patterns, 203
 - in regular expressions, 409, 424, 426
 - in variable names, 44
 - * (asterisk)
 - in `import` statements, 188–189
 - in command line, 492

- * (asterisk) (*continued*)
 - in regular expressions, 409, 416–419
 - operator, 55–57, 203
- * / in comments, 24
- *= operator, 55
- \ (backslash)
 - file name separator, 191
 - in escape sequences, 49, 212, 409, 414
 - in regular expressions, 409, 414, 423
- & (ampersand)
 - in generic types, 174
 - operator, 55, 62
- &&
 - in regular expressions, 409, 412–413
 - operator, 55, 60–61
- &= operator, 55
- # (pound sign)
 - in `DecimalFormat` patterns, 203
 - in regular expressions, 426
- % (percent sign)
 - format specifier, 199, 274
 - operator, 55–57, 203
- %= operator, 55
- + (plus sign)
 - in regular expressions, 409, 416–419
 - operator, 55–58, 203, 215
- ++ operator, 55, 58, 67
- += operator, 55, 57
- < operator, 55, 59–60
- << operator, 55, 62
- <<= operator, 55
- <= operator, 55, 59–60
- <> (angle brackets), 169, 296
- = operator, 48, 51, 56
 - return values of, 66
- = operator, 55
- == operator, 55, 59–60, 158
- > operator, 55, 59–60
- >= operator, 55, 59–60
- >> operator, 55, 62
- >>= operator, 55
- >>> operator, 55, 62
- >>>= operator, 55
- | (vertical bar)
 - in regular expressions, 409
 - operator, 55, 62, 427
- |= operator, 55
- || operator, 55, 60–61

Numbers

- 0 (zero), in octal numbers, 48
- 0x in hexadecimal numbers, 48

A

- abs method, 204
- abstract keyword, 161–163, 603
- Abstract Window Toolkit (AWT), 189, 547, 552, 581
- AbstractXXX implementations, 363
- AccessControlException, 457
- access modifiers. *See also* by individual type
 - classes and, 87–89, 110–112, 127, 151
 - constants and, 143
 - fields and, 151
 - interfaces and, 142–143
 - methods and, 90, 151, 154
- acos method, 207
- add method
 - for applets, 555
 - for collections, 298–299, 303, 307–309, 317, 345, 353, 363
- addAll method, 300, 304, 307–309, 353, 359
- addFirst method, 346
- addLast method, 346
- algorithms (collections), 300, 309, 314, 316, 355–360
 - composition, 360
 - finding extreme values, 360
 - polymorphic, defined, 294
 - routine data manipulation, 358–359
 - searching, 359
 - shuffling, 358
 - sorting, 356–358
- Alloy look and feel, 477, 479
- alt attribute, 575
- ampersand. *See* &
- Anagrams example, 326–328
- Anagrams2 example, 357
- angle brackets. *See* <>
- AnimatorApplet applet, 587
- annotations, 132–137
 - comments and, 133
 - naming, 135
 - types in, 133
- Apache, 525

- APIs (Application Programming Interfaces), 141
 - array-based vs. collection-based, 353
 - compatibility of, 364–366
 - design of, 366–367
 - interoperability of, 294–295, 364–367
 - JAR-related, 514–520
 - legacy, 367
- append method, 227–228
- appendReplacement method, 432, 436
- appendTail method, 432, 436
- applet tag, 498–499, 544, 564, 572, 574–578, 595–596
 - alternate HTML code in, 575–576
 - closing, 600
- Applet class, 543, 546
 - extending, 546–547
- AppletButton applet, 575, 582
- AppletContext interface, 559
- applets, 543–602
 - accessibility of, 594
 - API of, 559–578
 - background color of, 581, 601
 - capabilities of, 559
 - common problems with, 600–602
 - debugging, 583, 593
 - deploying, 594–600
 - directories of, 576–577
 - displaying documents in, 562–564
 - download time of, 488
 - event-handling methods in, 552–554
 - final cleanup, 550
 - finding all on a page, 567–568
 - finding by name, 564–567
 - finishing, 593–594
 - flexibility of, 594
 - GUIs in, 554, 581–583
 - leaving and returning to Web pages, 549
 - life cycle of, 543–544, 547–550
 - loading, 544–545, 549
 - loading data files, 560
 - loops in, 587
 - networking, 589–593
 - non-editable text fields in, 555–558
 - overriding methods in, 551
 - packed in JAR files, 498–499, 577–578
 - parameters in, 570–574
 - playing sounds, 568–570, 588, 593
 - quitting the browser, 550
 - reloading, 549, 600
 - representation of, 552
- applets (*continued*)
 - security and, 444, 508, 558–559, 564, 578–581, 585, 591–593
 - sending messages to other applets, 564–568
 - status line and, 561, 581
 - threads in, 585–589
 - vs. applications, 544, 546, 580
- appletviewer application, 544
- applet viewers. *See also* Web browsers
 - reloading applets, 549
 - security and, 567, 578–581, 585
- applications
 - accessibility of, 465, 482
 - command-line arguments in, 448–449
 - concurrent, 369
 - efficiency of, 269
 - entry points of, 502–503, 518
 - exiting, 445–446
 - launching, 521
 - life cycle of, 444–445
 - liveness of, 384–386
 - localizing, 481
 - look and feel of, 464, 476–479
 - measuring time intervals in, 457
 - packed in JAR files, 499–500
 - performance of, 294, 361, 363
 - pluggability of, 35
 - portability of, 5
 - reusing, 295
 - running, 7, 12–13, 15, 18, 20, 23, 445
 - security and, 455–457, 535–536
 - starting up, 445–446
 - vs. applets, 544, 546, 580
- archive attribute, 574
- args variable, 44
- arguments, 93–97
 - command-line. *See* command-line arguments
 - number of, 94–95
 - of reference data types, 96–97
 - primitive, 96
 - vs. parameters, 94
- ArithmeticDemo example, 56–57
- arithmetic operators, 55–57, 203
- ArrayBlockingQueue class, 349
- arraycopy method, 53, 346, 457
- ArrayCopyDemo example, 53
- ArrayDemo example, 49, 51
- ArrayIndexOutOfBoundsException, 237, 240–246

ArrayList implementation, 307, 309, 314,
342–343, 346
initial capacity of, 346
performance of, 346
synchronization and, 346
arrays, 49–53, 293
 assigning values to, 52
 collections and, 301
 copying, 53
 creating, 51–52
 declaring, 50–51
 elements of, 49, 52
 indexes in, 49, 52
 initializing, 51–52
 length of, 49, 52–53
 lists and, 352
 looping through, 76
 multidimensional, 52, 78
 naming, 51
 of characters, 213
 of strings, 220
Arrays class, 310
asin method, 207
asList method, 310, 347, 352–353, 364
assert keyword, 603
assignments, 48, 51, 55–56, 67
 compound, 55, 57
 return values of, 66
asterisk. *See* *
at. *See* @
atan method, 207
atan2 method, 207
atomic actions, 383–384
AtomicCounter example, 402
AtomicInteger class, 196
AtomicLong class, 196
Attributes class, 517
AudioClip interface, 568–570
audiotool application, 568
autoflush, 269

B

backslash. *See* \
backspace, 49
BadThreads example, 403
BasicMathDemo example, 204–205
BasicService interface, 536
Bicycle class, 35–116, 148–151, 155
BicycleDemo example, 36–37
bidi texts, 481

BigDecimal class, 46, 196, 271, 282, 284
 natural ordering, 329
BigInteger class, 196, 271
 natural ordering, 329
binarySearch method, 316, 359
BitDemo example, 62–63
bit shift operators, 55, 62–63
bitwise operators, 55, 62–63
BlockDemo example, 68
BlockingQueue implementation, 318, 343,
349–350, 400
blocks, 68
boolean data type, 46, 603
 default value of, 47
Boolean class, 329
BorderLayout class, 582
boxing, 196, 211
braces. *See* {}
break statement, 71–74, 77–79, 240, 603
 labeled, 78
 unlabeled, 77–78
BreakDemo example, 77–78
BreakWithLabelDemo example, 78
browsers. *See* Web browsers
BufferedInputStream class, 269
BufferedOutputStream class, 269
BufferedReader class, 267, 269
BufferedWriter class, 269
buffers, 269
bugs. *See* errors
byte data type, 45, 603
 default value of, 47
 expressing, 48
 switch statement and, 71
Byte class, 216
 natural ordering, 329
 switch statement and, 71
bytecodes, 2, 7, 14, 20
byteValue method, 197

C

C++ programming language, 5, 26
 Standard Template Library (STL), 294
cache implementation, 347
Calendar class, 284, 457
Callable interface, 398
call stacks, 233, 243, 255–257
Canvas Swing component, 555
capacity method, 226

- capturing groups, 422–423
 - backreferences in, 423
 - quantifiers and, 420–421
- carriage return, 49, 212, 267
- case statement, 71–74, 603
- catch block, 239–245, 250, 257, 374, 603
- Catch or Specify Requirement, 235–236
 - bypassing, 236
- cd command, 17, 21–22, 29
- CDE/Motif look and feel, 477–478
- ceil method, 204
- change method, 279
- char data type, 46, 210–212, 603
 - converting to strings, 211
 - default value of, 47
 - escape sequences in, 49
 - expressing, 48
 - getting by index, 218–219
 - switch statement and, 71
- Character class, 210–212
 - natural ordering, 329
 - switch statement and, 71
- character classes, 409–413
 - intersections of, 412–413
 - negation of, 410
 - predefined, 414–416
 - quantifiers and, 420–421
 - ranges of, 411
 - simple, 409–413
 - subtractions of, 413
 - unions of, 411–412
- charAt method, 213, 218–219
- CharSequence interface, 220
- checkXXX methods, 456
- class keyword, 25, 603
 - definitions of, 25
- Class class, 159
- ClassCastException, 169, 329, 333, 352
- classes, 35–37, 85–97. *See also* inheritance;
 - nested classes
 - abstract, 161–163
 - adapter, 367
 - base or parent. *See* superclasses
 - child, derived, or extended. *See* subclasses
 - constructors for, 92–93, 103
 - controlling access to, 110–112
 - declaring, 87–88
 - deprecated, 134
 - grouping, 122
 - hierarchy of, 108, 142, 148
 - classes (*continued*)
 - instantiating, 43, 100–101
 - interfaces implemented by, 143–144, 163
 - naming, 186
 - static initialization blocks in, 117
 - wrapper, 196
 - class files, 191–192
 - classid attribute, 596–597
 - class library. *See* Java API
 - ClassNotFoundException, 284
 - CLASSPATH variable, 30, 192–193, 457–460, 601
 - class paths, 192–193
 - class variables. *See* fields, static
 - clear method, 300, 322
 - ClipboardService interface, 536
 - clone method, 156–158, 343, 346
 - Cloneable interface, 157, 161
 - CloneNotSupportedException, 157
 - close method, 271
 - cmd command, 16
 - code
 - case sensitivity in, 11, 15, 21
 - error-prone, 70, 241, 258
 - platform-independent, 286
 - readability of, 45, 95, 122–123, 191, 241, 252–255
 - codebase attribute, 574, 576–577, 597
 - CollationKey class
 - natural ordering, 329
 - Collection interface, 176, 295–296, 298–301
 - array operations of, 301
 - bulk operations of, 300
 - implementations of, 363
 - standard conversion constructor of, 298
 - view methods of, 322–326
 - wrappers for, 350, 352
 - collections, 293–367
 - benefits of, 294–295
 - concurrent, 400–401
 - elements of. *See* elements
 - filtering, 300
 - frameworks of, 294
 - hierarchy of, 176, 366
 - immutable, 352
 - older APIs and, 301
 - ordered, 297
 - read-only access to, 352
 - synchronized, 343, 350–351

- collections (*continued*)
 - traversing, 76, 299–300
 - with no duplicate elements, 297, 301–306
- Collections class
 - empty constants in, 354
 - methods in, 350–353
 - polymorphic algorithms in, 355–360
- colon. *See* :
- comma. *See* ,
- command line, 276
- command-line arguments, 26, 448–449
 - echoing, 448–449
 - numeric, 449
- comments, 24–25, 168
 - annotations and, 133
- Comparable interface, 161, 328–335
- Comparator interface, 332–339, 349, 357, 359–360
- compare method, 332–335
- compareAndSet method, 401
- compareTo method
 - for files, 287
 - for objects, 329–332
 - for primitive data types, 197
 - for strings, 225
- compareToIgnoreCase method, 225
- ComparisonDemo example, 59
- comparisons, 55
 - interclass, 329
 - of numbers, 59–60, 204
 - of object, 61–62
- compatibility, 364–366
 - backward, 365–366
 - binary, 178
 - upward, 364–365
- compile method, 426–428
- compilers
 - error handling. *See* errors, compile-time
 - information for, 132, 152
 - verifying types of objects, 296
- ComputeResult class, 231
- concat method, 214–215
- ConcatDemo example, 57
- concurrency, 369–404
- ConcurrentHashMap implementation, 348–349, 400
- ConcurrentMap interface, 343, 348–349, 400
- ConcurrentModificationException, 346
- ConcurrentNavigableMap interface, 400
- ConcurrentSkipListMap interface, 400
- ConditionalDemo1 example, 60
- ConditionalDemo2 example, 60–61
- conditionals, 55, 60–61, 69
- Console, 277–279
- const keyword, 603
- constants, 112, 115, 128
 - compile-time, 115
 - for upper and lower bounds, 197
 - importing, 190, 203
 - in interfaces, 140, 143
 - naming, 45, 115
- constructors, 99–103
 - calling methods from, 161
 - chaining, 156
 - conversion, 298
 - declaring, 92–99
 - default, 103
 - for enum types, 130
 - generic, 172–173
 - inheritance and, 155–156
 - invoking, 109–110, 148
 - no-argument, 93, 103, 110, 156
 - synchronization and, 381
- Consumer example, 389–390
- Container class, 552, 555
- containers. *See* collections
- contains method, 298, 345, 447
- containsAll method, 300, 304, 323–326, 347
- containsKey method, 320–322, 447
- containsValue method, 320–322
- continue statement, 79–81, 240, 603
 - labeled, 80
 - unlabeled, 79
- ContinueDemo example, 79–80
- ContinueWithLabelDemo example, 80
- control flow statements, 68–82
 - branching, 77–82
 - decision-making, 69–71
- controlling access. *See* access modifiers
- conversions, 199–201, 274–275
- copy method, 316, 359
- CopyBytes example, 263–266
- CopyCharacters example, 265–266
- CopyLines example, 267
- CopyOnWriteArrayList implementation, 346
- CopyOnWriteArraySet implementation, 345
- core collection interfaces, 294–342. *See also* by
 - individual type
 - compatibility of, 364–366
 - hierarchy of, 295–297

core collection interfaces (*continued*)

- implementations of, 366
- modification operations in, 296

cos method, 207

Countdown example, 318

Counter example, 377, 401

CreateObjectDemo example, 98–106

createTempFile method, 289

currentTimeMillis method, 457

D

d in 64-bit double literals, 48

D in 64-bit double literals, 48

data encapsulation, 34

DataInput interface, 279, 290

DataInputStream class, 279–282

DataOutput interface, 279, 290

DataOutputStream class, 279–282

DataStreams example, 279–282

data types, 89. *See also* by individual type
reference, 96–97, 145

- returned by expressions, 66
- switch statement and, 71

Date class, 328–329

dates, 200, 284

Deadlock example, 384–385

Deal example, 314–315

debugging, 35, 160, 167, 249
output, 593

DecimalFormat class, 202–203

declarations. *See* by individual type

declaration statements, 67

decode method, 198

default statement, 71–74, 603

DelayQueue class, 350

delete method, 228, 289

deleteOnExit method, 289

@deprecated Javadoc tag, 134

@deprecated annotation type, 134

Desktop API, 483–484

- adding Java Web Start applications to, 524

destroy method, 544, 550, 552, 585

dir command, 17, 29

directories, 289

- changing, 17, 21–22, 29

- creating, 21

- current, 16, 21–22, 29

- home, 16, 21

- in JAR files, 492–493

disjoint method, 360

do keyword, 603

documentation, 133–134, 168, 296

dollar sign. *See* \$

dot. *See* .

double data type, 46, 603

- default value of, 47

- expressing, 48

Double class, 216

- natural ordering, 329

double quote. *See* "

doubleValue method, 197

do-while statement, 74–75

- skipping the current iteration, 79
- terminating, 77

DoWhileDemo example, 75

DownloadService interface, 536

Drop example, 388, 390

E

e in scientific notation, 48

E

- as type parameter, 171

- constant, 204

- in scientific notation, 48

Echo example, 448–449

element method, 317

elementAt method, 307, 309

elements (in collections), 296, 447

- adding, 298, 303, 323

- checking, 298

- counting, 298, 303

- manipulating, 358–359

- minimum and maximum, 360

- not duplicated, 301–306

- null, 301, 318

- ordering, 328–335

- randomization of, 358

- removing, 297–298, 303, 323

- searching, 359

- sorting, 318, 338, 356–358

ellipsis. *See* . . .

else statement, 603

emacs text editor, 19

embed tag, 595, 598–600

emptyList method, 354

emptyMap method, 354

emptySet method, 354, 363

EmptyStackException, 247

encapsulation, 122–123

end method, 423, 431–433

- endsWith method, 225
- EnhancedForDemo example, 77
- ensureCapacity method, 227, 351
- entrySet method, 322–323
- enum keyword, 128, 603
- Enum class, 130
- Enumeration collection, 567
 - compatibility and, 364–366
- EnumMap implementation, 348
- EnumSet implementation, 345
- EnumTest example, 129
- enum types, 128–132, 345
 - constructors for, 130
 - naming, 128
- Env example, 450
- environment, 443–461
 - properties of, 443–448, 583–585
 - restricted, 538, 544
- environment variables, 449–451
 - passing to new processes, 450
 - platforms and, 451
 - querying, 449–450
- EnvMap example, 449–450
- EOFException, 282
- equality operators. *See* comparisons
- equals method, 156, 158–160, 197, 225, 301, 309, 318, 322, 331, 334
- equalsIgnoreCase method, 225
- Error class, 235–236, 248, 252
- error messages, 27–30, 51, 167, 239, 277
- errors, 233, 235
 - compile-time, 27–29, 104, 107, 135, 152, 154, 156, 167, 176–178, 188
 - memory consistency, 379–381, 384, 401
 - runtime, 29–30, 167, 296
 - semantic, 28–29
 - syntax, 12, 27–28, 437–439
- escape sequences, 211–212
 - in regular expressions, 414
 - in Unicode, 48
- Exception class, 248, 251, 258
- exception handlers, 233, 236–245, 257–258
 - associating with try blocks, 238–240
 - catching multiple exceptions, 251
- exception objects, 233
- exceptions, 152, 233–260
 - advantages of, 253–258
 - catching, 234, 236–245, 257
 - chained, 249–250
 - checked, 235–236, 253
 - exceptions (*continued*)
 - class hierarchy of, 251
 - external. *See* errors
 - grouping, 257–258
 - runtime, 236, 248, 252
 - throwing, 233, 235, 243, 246–252
 - unchecked, 236, 252–253
 - user-defined, 240, 246, 250–252
 - exclamation sign. *See* !
 - execute method, 398
 - Executor interface, 397–398
 - executors, 371, 397–400
 - Executors class, 399
 - ExecutorService interface, 397–398
 - exit method, 457
 - exp method, 205
 - ExponentialDemo example, 206
 - exponents, 205
 - expressions, 66–67
 - compound, 66
 - expression statements, 67
 - ExtendedService interface, 537
 - extends keyword, 38, 87, 143, 173–174, 177, 603
 - extensions, 451, 488, 498
- F**
 - f in 32-bit float literals, 48
 - F in 32-bit float literals, 48
 - fields, 43, 88. *See also* variables
 - controlling access to, 110–112
 - declaring, 47, 150
 - default values of, 47–48
 - deprecated, 134
 - final, 381, 393
 - hiding, 150, 154
 - inherited, 150
 - initializing, 47, 116–118
 - non-static, 43, 112, 115, 118
 - private, 393
 - referencing, 103–105
 - reusing, 148
 - shadowing, 95, 109
 - static, 44, 112–115, 163
 - static final. *See* constants
 - FIFO (first-in, first-out), 297, 317, 343, 349
 - File class, 39, 286–289
 - natural ordering, 329
 - FileContents interface, 537
 - file descriptors, 159

FileInputStream class, 263, 266–267
 Filename class, 221–224
 FilenameDemo example, 223
 FileNotFoundException, 235, 240, 257
 FileOpenService interface, 537
 FileOutputStream class, 263, 266–267
 FileReader class, 265–267
files
 creating, 289
 I/O and, 286–291
 manipulating, 288–289
 modification data/time of, 289
 names of, 286–288
 random access to, 289–291
 FileSaveService interface, 537
 FileStuff example, 288
 FileWriter class, 265–267, 286
 fill method, 316, 359
 final modifier, 603
 constants and, 115, 143
 fields and, 44
 methods and, 118, 160–161
 finalize method, 156, 159
 finally block, 240–245, 264, 603
 find method, 431
 FindDups example, 303
 FindDups2 example, 305
 first method, 338
 float data type, 46, 282, 603
 default value of, 47
 expressing, 48
 Float class, 216
 natural ordering, 329
 floatValue method, 197
 floor method, 204
 flush method, 269
 for statement, 75–77, 299–300, 603
 enhanced, 76
 nested, 78
 skipping the current iteration, 79
 terminating, 77
 ForDemo example, 75–76
 format method, 198–200, 215–216, 269,
 273–276
 Format example, 275
 format specifiers, 199–201, 274–276
 format strings, 199, 274
 Formatter class, 199
 formatting, 272–276
 form feed, 49, 212

forward slash. *See* /
 Frame class, 546
 Freq example, 320–321
 frequency method, 360
functions. *See* methods
 Future interface, 398

G

garbage collection, 5, 106, 159
 weak references and, 348
 generic types, 169–172, 296
 declaring, 169
 instantiating, 178
 invoking, 170
 get method, 307, 309, 320–322, 363, 401
 getAbsolutePath method, 287
 getApplet method, 564–567, 579
 getAppletInfo method, 594
 getApplets method, 567–568, 579
 getAudioClip method, 568
 getCanonicalPath method, 287
 getCause method, 249
 getChars method, 214
 getClass method, 156, 159
 getCodeBase method, 560, 582, 589
 getDescription method, 437
 getDocumentBase method, 560, 582
 getEnv method, 449–450
 getFields method, 159
 getFirst method, 346
 getHost method, 589
 getImage method, 582, 601
 getIndex method, 437
 getInterfaces method, 159
 getLast method, 346
 getMainAttributes method, 517
 getMainClassName method, 515–517, 519
 getMessage method, 437
 getMethods method, 159
 getName method, 287
 GetOpenProperties applet, 584
 getParameter method, 573
 getParameterInfo method, 573–574, 594
 getParent method, 287
 getPattern method, 437
 getProperty method, 447, 452–454
 getResource method, 535
 getSecurityManager method, 456
 getSimpleName method, 159
 getStackTrace method, 250

getSuperClass method, 159
 getValue method, 517
 goto method, 603
 graphical user interfaces (GUIs), 4, 40, 463. *See also* Swing
 creating by applets, 581–583
 GridBagLayout class, 582
 group method, 423
 groupCount method, 423
 GTK+, look and feel, 464
 GTK+ look and feel, 477
 guarded blocks, 386–390

H

hashCode method, 156, 160, 301, 309, 318, 322, 331
 HashMap implementation, 320, 322, 342–343, 347, 361
 HashSet implementation, 302, 304–305, 342–345
 initial capacity of, 344
 load factor of, 345
 Hashtable collection, 293, 297, 320, 343, 348–349, 444, 448, 570
 compatibility and, 364–366
 hasNext method, 299, 311
 headSet method, 337
 HelloRunnable example, 371
 HelloThread example, 371–372
 HelloWorld, 6–26
 applet, 545, 551
 for Microsoft Windows, 14–19
 for Solaris OS, 19–23
 for UNIX/Linux, 19–23
 HelloWorldApp example, 25
 HostnameVerifier class, 539
 HTML Converter tool, 600
 HTML specification, 595
 HTTP requests, 399
 HTTPS Certificates, 539

I

I/O, 261–293
 atomic, 383–384
 from the command line, 276–279, 452
 line-oriented, 267–268
 non-stream file, 286–291
 of complex objects, 284–285

I/O (*continued*)
 of objects, 282–285
 of primitive data type values, 279–282
 scalable, 291
 I/O streams. *See* streams, I/O
 IdentityHashMap implementation, 348
 IDE Project, 7–9
 if statement, 603
 IfElseDemo example, 70
 if-then statement, 69–70
 if-then-else statement, 60, 70–71
 comparing to switch, 72
 IllegalArgumentException, 248
 IllegalStateException, 317
 images, in Swing applets. *See* icons
 ImmutableRGB example, 394
 implementations, 294, 342–355
 abstract, 342, 361–364
 adapter, 361
 anonymous, 350
 concurrent, 342, 348–350
 convenience, 342, 352–354
 custom, 360–364, 367
 documenting, 296
 general-purpose, 342–347, 349, 352
 performance of, 344
 special-purpose, 342, 345–346, 348
 wrapper, 342–343, 350–352
 implements keyword, 39, 87–88, 143, 603
 import statement, 188–190, 203, 603
 indexOf method, 307–309
 indexOfSubList method, 316
 information-hiding, 34
 inheritance, 37–38, 147–165
 multiple, 141–142, 145
 init method, 543, 546, 550–551, 570, 585, 601
 initCause method, 249
 initializer blocks, 118
 inner classes, 122–127
 anonymous, 127
 controlling access to, 127
 instantiating, 124
 local, 127
 InputStream class, 263
 InputStreamReader class, 267, 277
 insert method, 227–228
 insertElementAt method, 308–309
 instanceof operator, 55, 61–62, 152, 603
 InstanceofDemo example, 61

instances, 35, 43, 100–101, 113
 of inner classes, 123
 testing, 61–62

instance variables. *See* fields, non-static

int data type, 45–46, 603
 default value of, 47
 expressing, 48
 switch statement and, 71

Integer class, 198, 216
 natural ordering, 329
 switch statement and, 71

interface keyword, 142, 603

interfaces, 38–39, 139–147
 abstract classes and, 161–162
 access control of, 142
 body of, 142
 collection. *See* core collection interfaces
 defining, 140, 142–143, 161
 extending other interfaces, 142
 implementing, 88, 140–144, 163
 rewriting, 145–146
 using as types, 144–145

internationalization, 265
 in Swing, 465, 480–482

Internet domain names, 186, 191

Internet Explorer, 540, 545, 595–599

Inter Process Communication (IPC), 370

interrupt method, 373

interrupted method, 374–375

InterruptedException, 373–375, 387

intValue method, 197

invokeClass method, 517–518

IOException, 236

IOException, 237, 239–240, 243–246, 257

isAnnotation method, 159

isEmpty method, 298, 303, 320–322

isEnum method, 159

isInterface method, 159

isInterrupted method, 375

isLetter method, 211

isLowerCase method, 211

isUpperCase method, 211

isWhitespace method, 211, 270

iterator method, 298–300, 303, 323, 335, 339, 351, 363

Iterator class, 303, 310–313, 335, 339

iterators, 298–300
 fail-fast, 343

J

JApplet Swing component, 474, 543, 546, 559
 extending, 546–547

Jar tool, 489
 setting entry points with, 503

JAR files, 451, 487–520
 benefits of, 488
 creating, 489–493, 534
 extracting, 495–496
 paths in, 494
 reading, 535
 referencing classes in other JAR files from, 503–504
 running, 498–500
 sealing, 507
 signing, 507–513, 538
 uncompressed, 492
 updating, 496–498
 verifying, 508, 513
 viewing, 493–495

JarClassLoader example, 514–519

JarRunner example, 514–515, 518–520

Jarsigner tool, 511–513

JarURLConnection class, 514–517

java launcher, 2, 4, 7, 15, 20, 23, 29–30, 499
 arguments, 30
 case sensitivity in, 11, 15, 21

Java programming language, 1–5

java.awt package, 189

java.io package, 263–291

java.nio.* packages, 291

java.util.concurrent package, 342, 348–350, 369–404

java.util.concurrent.atomic package, 401–402

java.util.concurrent.locks package, 395–397

java.util.jar package, 514–515

java.util.regex package, 405–441

Java 2D, 4, 464

Java 3D, 488

Java API (Application Programming Interface), 3–4, 26, 488

Java Archive Tool. *See* Jar tool

JavaBeans, 5

javac compiler, 2, 4, 7, 14, 20, 27
 case sensitivity in, 11, 15, 21

- Java Cache Viewer, 523, 540
 - Java Collections Framework. *See* collections
 - javadoc tool, 4
 - doc comments and, 25
 - tags in, 135
 - Java Foundation Classes (JFC), 464
 - Java HotSpot virtual machine, 2
 - Java IDL API, 5
 - Java look and feel, 470–477
 - JavaMail, 488
 - Java Naming and Directory Interface API, 5
 - Java Network Launching Protocol. *See* JNLP files
 - Java platform, 3–6
 - API Specification, 40
 - earlier versions of, 293, 522
 - multithreaded execution and, 370
 - supported encodings on, 533
 - Java Plug-in, 451, 465, 487, 545
 - versions of, 596
 - Java Remote Method Invocation over Internet Inter-ORB Protocol (Java RMI-IIOP) Technology, 5
 - Java RMI, 5
 - JavaScript, 599–600
 - Java SE Development Kit 6. *See* JDK 6
 - Java SE Runtime Environment. *See* JRE
 - Java Virtual Machine (Java VM), 2–4, 7
 - failures in, 248, 252
 - terminating, 289
 - Java Web Start, 5, 451, 466, 487, 521–541
 - benefits of, 521
 - common problems with, 539–540
 - deploying applications, 524–534
 - importing certificates, 539
 - running applications, 522–524
 - security and, 538–539, 544
 - signing applications, 538
 - javax.jnlp package, 536–537
 - JButton Swing component, 471, 555
 - JCheckBox Swing component, 471, 555
 - JCheckBoxMenuItem Swing component, 555
 - JColorChooser Swing component, 472
 - JComboBox Swing component, 471
 - JDBC API, 5
 - JDialog Swing component, 474, 547
 - JDK 6 (Java SE Development Kit 6), 4, 6, 14, 19, 27, 136, 463, 483, 485, 601
 - JDK Applet Viewer, 579, 600
 - JEditorPane Swing component, 472
 - JFileChooser Swing component, 473
 - JFrame Swing component, 474, 546–547
 - JInternalFrame Swing component, 476
 - JLabel Swing component, 474, 555
 - JLayeredPane Swing component, 476
 - JList Swing component, 471, 555
 - JMenu Swing component, 471, 555
 - JMenuBar Swing component, 555
 - JMenuItem Swing component, 555
 - JNLP files, 451, 523, 525
 - contents of, 526–533
 - creating, 525–533
 - encoding, 533
 - links to, 534
 - security and, 538
 - JNLP API, 536–537
 - join method, 375
 - JPanel Swing component, 475, 555
 - JPasswordField Swing component, 471
 - JProgressBar Swing component, 474
 - JRadioButton Swing component, 471
 - JRE (Java SE Runtime Environment), 523, 545, 596–597, 601
 - JRootPane Swing component, 476
 - JScrollBar Swing component, 555
 - JScrollPane Swing component, 475
 - JSeparator Swing component, 474
 - JSlider Swing component, 471, 555
 - JSpinner Swing component, 471
 - JSplitPane Swing component, 475
 - JTabbedPane Swing component, 475
 - JTable Swing component, 473
 - JTextArea Swing component, 473, 555
 - JTextField Swing component, 555–556
 - JTextPane Swing component, 472
 - JToolBar Swing component, 475
 - JToolTip Swing component, 474
 - JTree Swing component, 473
 - JWindow Swing component, 555
- ## K
- K type parameter, 172
 - keys method, 447
 - keySet method, 322–323
 - keywords, 603. *See also* by individual type
 - variable names and, 45
- ## L
- last method, 338
 - lastIndexOf method, 221–224, 307–309

- lastIndexOfSubList method, 316
 - length method, 53, 213–214, 219, 226
 - line feed, 49, 212, 215, 267
 - line terminators, 199–200, 267–268, 274
 - in regular expressions, 427
 - LinkedBlockingQueue class, 349
 - LinkedHashMap implementation, 320–321, 342, 347
 - LinkedHashSet implementation, 302, 342, 344–345
 - LinkedList implementation, 307, 318, 342–343, 346, 349
 - performance of, 346
 - Linux. *See* UNIX/Linux
 - list method, 289, 447
 - List interface, 297, 306–316
 - algorithms of, 316
 - arrays and, 352
 - basic operations of, 308–309
 - comparing to Vector, 307–309
 - empty constants of, 354
 - implementations of, 307, 342–343, 345–347, 353, 363
 - order of elements in, 356–359
 - positional access and search operations of, 309–310
 - range-view operation in, 313–315
 - wrappers for, 350, 352
 - listeners. *See* event listeners
 - listFiles method, 289
 - listIterator method, 307, 311, 363
 - ListOfNumbers example, 236–246
 - listRoots method, 289
 - lists
 - cursor positions in, 312
 - iterating backward, 311
 - literals, 48–409
 - class, 49
 - concatenating with strings, 215
 - locales, 200, 271–272, 274, 480
 - character sets in, 265
 - overriding, 272
 - Lock interface, 395–397
 - lockInterruptibly method, 395
 - locks
 - deadlocks, 384–385, 395
 - intrinsic, 381–383, 387
 - livelocks, 386
 - starvation, 386
 - log method, 205
 - logarithms, 205
 - logging, 250
 - logical operators, 55, 60–61
 - bitwise, 62
 - long data type, 46, 603
 - default value of, 47
 - expressing, 48
 - Long class, 216
 - natural ordering, 329
 - longValue method, 197
 - lookingAt method, 431, 434
 - loop method, 568
 - loops, 50, 74–77
 - in applets, 587
 - infinite, 74, 76
 - nested, 78, 80
 - skipping the current iteration, 79
 - terminating, 77
 - ls command, 22, 30
- ## M
- Mac OS, 2–3, 6–7
 - main method, 25–26, 30, 36, 44, 546
 - manifest files, 492, 494, 500–507
 - default, 500
 - modifying, 500–502
 - package sealing in, 506–507
 - package versions in, 504–506
 - referencing classes in, 503–504
 - Map interface, 295–297, 319–328, 449
 - basic operations of, 320–322
 - bulk operations in, 322
 - comparing to Hashtable, 320
 - empty constants of, 354
 - implementations of, 320, 342–343, 347–349, 363
 - standard conversion constructor of, 322
 - viewing as a Collection, 322–326
 - wrappers for, 350, 352
 - Matcher class, 406
 - methods of, 431–437
 - MatcherDemo example, 432–434
 - matches method, 225, 428, 430, 432, 434
 - MatchesLooking example, 434
 - Math class, 190, 203–208
 - max method, 204, 360
 - MAX_VALUE constant, 197
 - members, 44
 - controlling access to, 151

- memory
 - freeing, 106, 159, 240–241
 - leaks, 5, 241, 264
 - reserving for a variable, 99
 - saving in large arrays, 45
 - methods, 34. *See also* by individual type
 - abstract, 161–163
 - accessor, 213
 - as members, 44
 - atomic, 384
 - body of, 90, 94, 140
 - called from constructors, 161
 - class, 115, 152–153
 - controlling access to, 110–112
 - declarations of, 90–97, 246
 - deprecated, 134–135
 - exiting from, 81
 - final, 118, 160–161, 393
 - generic, 172–173
 - hiding, 152
 - importing, 190, 203
 - inherited, 150
 - instance, 115, 150, 152
 - invoking, 67, 105–106
 - naming, 90–91
 - overloading, 154
 - overriding, 108, 135, 152–154
 - package-private, 111
 - passing to methods, 94
 - returning values from, 81, 90, 105–106–109
 - reusing, 117, 148
 - specifying exceptions in, 245–246
 - static, 114–115, 150, 163
 - static private, 117
 - synchronized, 380–382, 387
 - method signatures, 91
 - inheritance and, 150, 152, 154
 - in interface declarations, 140, 142
 - in method declarations, 91
 - Microsoft Windows, 2–3, 6–7, 14–19
 - common error messages on, 27, 29
 - environment variables on, 193, 451, 458, 460
 - file name separators on, 191
 - file names on, 286–287
 - path separators on, 193
 - root directories on, 289
 - system tray on, 484
 - MIME types, 525, 539
 - min method, 204, 360
 - MIN_VALUE constant, 197
 - mkdir method, 21, 289
 - modifiers. *See* access modifiers
 - modularity, 34
 - monitor locks. *See* locks, intrinsic
 - mouseDown method, 593
 - Mozilla, 545, 595–596, 598–599
 - MultiDimArrayDemo example, 52
 - multimaps, 326–328
 - multisets, 361
- ## N
- N type parameter, 172
 - name attribute, 564
 - NameSort example, 331–332
 - nanoTime method, 457
 - native keyword, 603
 - nCopies method, 353
 - NegativeArraySizeException, 248
 - nested classes, 122–128
 - as members, 44
 - controlling access and, 151
 - importing, 189
 - non-static. *See* inner classes
 - static, 122–123, 127
 - NetBeans IDE, 6–14
 - new keyword, 48, 51, 92, 99–101, 104–105, 170, 213, 603
 - newCachedThreadPool method, 399
 - newFixedThreadPool method, 399
 - newline. *See* line terminators
 - newSingleThreadExecutor method, 400
 - next method, 299
 - nextIndex method, 312
 - NoSuchElementException, 317
 - NoSuchMethodError, 30
 - Notepad text editor, 14–16
 - Notepad demo, 525–526
 - notify method, 157
 - notifyAll method, 157
 - null value, 47, 49, 62, 100, 106, 197, 249, 252
 - NullPointerException, 236, 248, 253, 331
 - Number class, 195–198, 216–218
 - NumberFormatException, 449
 - numbers, 195–210
 - comparing, 204
 - converting to strings, 197–198, 217–218
 - currency values and, 282
 - formatting, 198–203, 274
 - random, 208
 - separators for, 200–203, 271–272

number systems, 48, 197
 converting between, 197–198

O

object tag, 595–596–600
 Object class, 93, 103, 148, 156–160
 ObjectInput interface, 282
 ObjectInputStream class, 282
 object-oriented programming, 33–41
 ObjectOutput interface, 282
 ObjectOutputStream class, 282
 object references, 56, 158
 I/O and, 284–285
 immutability and, 393
 objects, 33–35, 97–106
 assigning, 175
 casting, 151
 comparing, 158, 197
 converting to primitive types, 197
 converting to strings, 160, 197, 215
 copying, 156–158
 creating, 67, 99–103
 hash codes of, 160
 immutable, 47, 331, 391–394
 initializing, 99, 101–103
 instantiating, 99
 passing to methods, 94, 96–97
 referencing, 105–106, 114, 123
 reusing, 35
 serialization of, 282
 testing, 61–62
 using, 103–106
 verifying types of, 296
 ObjectStreams example, 282–284
 offer method, 317, 363
 operators, 55–65. *See also* by individual type
 precedence of, 55, 67
 prefix/postfix, 55, 58
 OutputStream class, 263
 OutputStreamWriter class, 267
 @Override annotation type, 135, 152

P

package statement, 184–195, 603
 package members, 187–191
 importing, 188
 public, 187
 referring to, 187–188

packages, 3, 39–40, 183–195
 apparent hierarchy of, 189
 controlling access to, 110–112
 creating, 184–195
 exceptions in, 245–246, 251
 importing, 188–189
 naming, 186–187, 191
 sealing, 488, 506–507
 unnamed, 186
 versioning, 488, 504–506
 pages. *See* Web pages
 paint method, 551–552
 palindromes, 213–214, 219, 228–230
 Panel class, 581
 param tag, 565, 574–575
 parameterized types, 170
 parameters, 44, 88, 90, 366
 data types of, 94
 different lists of, 91
 naming, 95–96
 of reference data types, 96–97
 primitive, 96
 shadowing, 95, 109
 vs. arguments, 94
 parentheses. *See* ()
 parseXXX methods, 198, 217, 449
 PassPrimitiveByValue class, 96
 Password example, 277–279
 passwords, 277–279
 PasswordStore demo, 466–470
 accessibility of, 482–483
 data transfer support in, 480
 integrating with the Desktop, 483–484
 localized, 481–482
 look and feel, 479
 look and feels of, 477–
 PATH variable, 27, 457–459
 Pattern class, 406. *See also* regular
 expressions
 boundary matchers in, 424–425
 capturing groups in, 422–423
 character classes in, 409–422
 methods of, 425–431
 PatternSyntaxException, 406
 methods of, 437–439
 peek method, 317, 363
 percent sign. *See* %
 performance, 294, 361, 363
 boosting, 2
 implementations and, 343–344, 346

Perl, 405
 PI constant, 190, 204
 Pico text editor, 19
 Planet class, 130
 play method, 568
 poll method, 297, 317, 363
 Popup Swing component, 555
 pound sigh. *See* #
 pow method, 205
 Preferences API, 451
 PrePostDemo example, 58
 primitive data types, 45–49, 96, 195. *See also*
 numbers
 assigning literals to, 48
 comparing, 158
 converting to objects, 197
 reserving memory for, 99
 tokens and, 271
 wrappers for, 196
 print method, 198, 273–274, 583
 printf method, 95, 198–200
 println method, 160, 198, 215, 267, 269,
 273–274, 583
 PrintStream class, 198–200, 272–277
 PrintThread applet, 585
 PrintWriter class, 240–241, 267, 269,
 272–276
 PriorityBlockingQueue class, 349
 PriorityQueue implementation, 343, 349
 private modifier, 123, 603
 classes and, 87–89, 110–112, 122, 127
 inheritance and, 151
 methods and, 90
 problems. *See* errors
 ProcessBuilder, 370, 450
 processes, 370
 lightweight. *See* threads
 processEvent method, 552
 processKeyEvent method, 552
 processMouseEvent method, 552
 Producer example, 389
 ProducerConsumerExample example, 390
 programs. *See* applications
 properties
 managing, 443
 saving, 446
 setting, 446–448
 system, 452–455, 583–585
 Properties class, 443–448
 PropertiesTest example, 454

propertyNames method, 447
 protected modifier, 603
 classes and, 110–112, 122, 127
 inheritance and, 150–151
 proxy servers, 539
 public modifier, 603
 classes and, 87–89, 110–112, 122, 127
 constants and, 143
 inheritance and, 150–151
 interfaces and, 142
 methods and, 26, 90, 142
 package members and, 187
 types and, 185
 put method, 320–322, 363
 putAll method, 322
 pwd command, 22, 30

Q

qualified names, 187–188
 quantifiers, 416–422
 types of, 421–422
 question mark. *See* ?
 Queue interface, 297, 316–319
 implementations of, 342–343, 349–350, 363
 queues
 bounded, 317
 order of elements in, 317
 quote method, 430
 QuoteClientApplet applet, 590–591
 quoteReplacement method, 432
 QuoteServer applet, 590
 QuoteServerThread applet, 590

R

radians, 206
 random method, 208
 Random class, 208, 358
 RandomAccessFile class, 290–291
 raw types, 178
 readDouble method, 280
 reader method, 277
 Reader class, 265
 readInt method, 280
 readLine method, 267, 279
 readObject method, 284–285
 readPassword method, 279
 readUTF method, 280
 Receiver applet, 565–567
 Rectangle class, 101–103, 143–144
 Reflection API, 518

- RegexTestHarness *example*, 406–407, 427
 - RegexTestHarness2 *example*, 437–439
 - regionMatches *method*, 224–225
 - RegionMatchesDemo *example*, 224
 - regular expressions**, 221, 225, 271, 405–441
 - backreferences in, 423
 - boundary matchers in, 424–425
 - capturing groups in, 422–423
 - case sensitivity in, 426
 - character classes in, 409–413
 - commonly used, 414–416
 - embedded flag expressions in, 427–428
 - metacharacters in, 408–409
 - non-capturing groups in, 423
 - quantifiers in, 416–422
 - syntax errors in, 437–439
 - zero-length matches in, 417–420
 - Relatable *interface*, 143–145
 - relational operators**. *See* comparisons
 - remove *method*
 - for applets, 555
 - for collections, 297–300, 303, 307–309, 311, 317, 345, 353, 363
 - for properties, 448
 - removeAll *method*, 300, 304, 323–326, 353
 - removeDups *method*, 303
 - removeEldestEntry *method*, 347
 - removeElementAt *method*, 309
 - removeFirst *method*, 346
 - removeLast *method*, 346
 - renameTo *method*, 289
 - replace *method*, 221, 228, 314, 431
 - replaceAll *method*, 221, 316, 432, 434–437
 - ReplaceDemo *example*, 434–436
 - ReplaceDemo2 *example*, 435
 - replaceFirst *method*, 221, 432, 434–436
 - reserved words**. *See* keywords
 - retainAll *method*, 300, 304, 323–326
 - return statement, 81–82, 106–107, 240, 603
 - return types, 81, 90, 106–109, 366–367
 - by values, 105
 - class names as, 107–109
 - covariant, 108, 152
 - interface names as, 109
 - reverse *method*, 228–229, 316, 359
 - rint *method*, 204
 - Root *example*, 273–274
 - Root2 *example*, 274
 - rotate *method*, 316
 - round *method*, 204
 - run *method*, 371
 - Runnable *interface*, 371–372, 398, 587
 - runtime**
 - checks at, 152
 - examining annotations at, 132, 136
 - handling errors, 233–235, 241–246
 - performance boosting at, 2
 - terminating, 234
 - RuntimeException, 235–236, 248, 252–253
- ## S
- S *type* parameter, 172
 - Safelock *example*, 395–397
 - sandboxes, 538, 544
 - Scanner *class*, 270–272
 - scanning, 270–272
 - locales and, 271–272
 - ScanSum *example*, 271–272
 - ScanXan *example*, 270–271
 - schedule *method*, 399
 - ScheduledExecutorService *interface*, 398–399
 - ScheduledThreadPoolExecutor *class*, 400
 - ScrollingSimple *applet*, 555–558
 - security**, 261
 - applets and, 558–560, 564, 567, 578–581, 585, 591–593
 - certificates and, 508
 - digitally signed files and, 488, 507–513
 - Java Web Start and, 538–539
 - keystores and, 511
 - password entry and, 277
 - public and private keys, 508
 - violation of, 456–457
 - SecurityException, 455–456, 559, 579
 - SecurityManager *class*, 456, 559, 579
 - security managers, 444, 455–457
 - accessing to system properties, 452
 - semicolon. *See* ;
 - Sender *applet*, 565–567
 - sequences. *See* collections, ordered
 - Serializable *interface*, 282, 343
 - servers. *See* Web servers
 - ServiceLoader *class*, 451
 - services, 451
 - set *method*, 307–309, 345, 363, 401
 - Set *interface*, 295–297, 301–306
 - array operations of, 306
 - basic operations of, 303

- Set interface (*continued*)
 - bulk operations of, 304–306
 - empty constants of, 354
 - implementations of, 302, 342–345, 353, 363
 - wrappers for, 350, 352
- setDefaultHostnameVerifier method, 539
- setDefaultSSLSocketFactory method, 539
- setElementAt method, 307–309
- setLastModified method, 289
- setLayout method, 555
- setLength method, 227
- setProperty method, 454–455
- setProperty method, 448
- setSize method, 308
- short data type, 46, 603
 - default value of, 47
 - expressing, 48
 - switch statement and, 71
- Short class, 216
 - natural ordering, 329
 - switch statement and, 71
- shortValue method, 197
- showDocument method, 562–564
- ShowDocument applet, 562–564
- showStatus method, 561
- shuffle method, 310, 314, 316, 358
- Shuffle example, 310
- signature block files, 510
- signature files, 509
- Simple applet, 547–550, 552
- SimpleClick applet, 552–554
- SimpleThreads example, 372, 375–377
- sin method, 207
- single quote. *See* '
 - singleton method, 301, 353
- size method, 303, 320–322, 363, 447
 - for collections, 298
- slash. *See* /
- sleep method, 372–373, 375
- SleepMessages example, 373
- Smalltalk's collection hierarchy, 294
- Socket class, 39
- sockets, 267
- software. *See* applications
- Solaris OS, 2–3, 6–7, 19–23
 - environment variables on, 458–460
 - file names on, 286–287
- sort method, 316, 328–335, 356–359
- Sort example, 356
- SortedMap interface, 297, 338–340
 - operations in, 338–339
 - standard conversion constructor in, 339
 - wrappers for, 350, 352
- SortedSet interface, 295–297, 335–338
 - Comparator access operations in, 335, 338
 - endpoint operations in, 335, 338
 - range-view operations in, 335–338
 - standard conversion constructor of, 336
 - wrappers for, 350, 352
- SoundExample classapplet, 569–570, 589
- SoundList class, 570, 589
- SoundLoader class, 570, 589
- source files, 191–193
 - adding code to, 7, 9–11
 - compiling, 7, 11–12, 14, 16–18, 20
 - creating, 14–16, 20–22
 - naming, 11, 15, 21
 - renaming nodes in, 13
 - skeleton, 7, 9
 - package statement in, 184–195
- split method, 220, 428–430
- SplitDemo example, 428
- SplitDemo2 example, 429
- sqrt method, 205
- square brackets. *See* []
- square root, 205
- SSLSocketFactory class, 539
- StackOfInts class, 125–127
- stacks, 124
- stack traces, 236, 249–250
 - printing, 257
- Standard Error, 276–277, 583
- Standard Input, 276–277
- Standard Output, 276–277, 583
- start method, 372, 423, 431–433, 543, 546, 550–551, 570, 585, 587
- startsWith method, 225
- statements, 67–68. *See also* by individual type
 - synchronized, 382–383
- static modifier, 603
 - classes and, 117
 - constants and, 115, 143
 - fields and, 44, 112–114
 - methods and, 26
- static initialization blocks, 117
- stop method, 544, 550–551, 568, 585, 587, 593

- streams, I/O, 261–285. *See also* by individual type
 - buffered, 268–269, 291
 - byte, 263–267, 269, 277
 - character, 265–269
 - closing, 264–265
 - data, 279–282
 - object, 282–285
 - standard, 276–277
 - unbuffered, 269
- strictfp method, 603
- String class, 39, 47, 160, 212–232
 - constructors for, 213
 - default value of, 47
 - escape sequences in, 49
 - expressing, 48
 - natural ordering, 328–329
 - regular expressions in, 430–431, 436–437
- StringBuilder class, 224–230
- string builders, 224–230
 - capacity of, 226–227
 - length of, 226–227
- StringDemo example, 213–214, 228–230
- StringIndexOutOfBoundsException, 223
- stringPropertyNames method, 447
- strings, 212–232
 - comparing, 224
 - concatenating, 57, 214–215
 - converting to numbers, 197–198, 216–217
 - converting to objects, 197
 - creating, 212–213
 - immutability of, 213
 - length of, 213–214
 - multiline, 215
 - replacing, 221
 - sorting, 356
 - substrings from, 219
- subclasses, 37–165
 - constructors for, 155–156
 - controlling access to, 111, 148, 150–151, 154
 - final methods and, 118
- subList method, 307–308, 313–315
- submit method, 398
- subSequence method, 220
- subSet method, 336–338
- substring method, 219, 221–224
- subtyping, 175–177
- Sun Certified Programmer for the Java Platform, 605–615
- super keyword, 151, 154–156, 177, 603
- superclasses, 37, 87, 135, 145, 148
 - accessing, 154–155
 - constructors for, 155–156
 - private members in, 151
- @SuppressWarnings annotation type, 135
- swap method, 309, 316, 359
- Swing, 4, 463–485, 547
 - basic controls, 470–471, 555
 - cut, copy, and paste, 480
 - drag-and-drop support, 480
 - general-purpose containers, 474–475
 - interactive displays, 472
 - internationalization, 465, 480–482
 - list component, 466
 - look and feel, 479
 - look and feels, 464, 470–
 - special-purpose containers, 475–476
 - system tray support, 484
 - tabbed pane, 468–469
 - table component, 467
 - top-level containers, 474
 - undo framework, 465, 470
 - uneditable information displays, 473–474
- switch block, 71–74, 603
 - terminating, 77
- SwitchDemo example, 72
- SwitchDemo2 example, 73
- synchronization
 - intrinsic locks and, 381–383
 - of collections, 343, 346, 349–350–351
 - of threads, 377–384
 - reentrant, 383
- synchronized keyword, 380, 603
- synchronizedCollection method, 350–351
- SynchronizedCounter example, 380–381, 401
- synchronizedList method, 346, 350–351
- synchronizedMap method, 350–351
- SynchronizedRGB example, 391–393
- synchronizedSet method, 350–351
- synchronizedSortedMap method, 350–351
- synchronizedSortedSet method, 350–351
- SynchronousQueue class, 350
- Synthetica look and feel, 477, 479
- Synth package, 464, 477
- System class, 26, 444, 452–457
- System.console, 277
- System.err, 273, 276–277, 583
- System.in, 276–277
- System.out, 198, 273, 276–277, 583

T

T type parameter, 172
tab, 49, 212
tailSet method, 337
TalkClientApplet applet, 591
TalkServer applet, 592
TalkServerThread applet, 592
tan method, 207
ternary operator, 55, 60
TestFormat example, 201
this keyword, 109–110, 603
 using with constructors, 109–110
Thread class, 371–377, 570
 subclasses of, 371–372
ThreadPoolExecutor class, 400
thread pools, 399–400
threads, 369–404
 blocked, 384–385
 coordinating, 386–390
 defining, 371–372
 execution history of, 249
 in applets, 585–589
 interference of, 377–379
 locks and, 381–383
 main, 370
 memory consistency errors and, 379–381, 384, 401
 suspending, 372–373, 386
 synchronization of, 377–384
 system, 370
 using to perform initialization tasks, 588
throw statement, 246–247, 603
Throwable class, 239, 246–249, 258
 constructors for, 249
throws keyword, 235, 246, 257, 603
TicTacToe example, 491–498–499
toArray method, 301, 335, 339, 362
toDegrees method, 207
tokens
 breaking input into, 270–271
 for primitive types, 271
 separators for, 270–271
 translating, 271–272
toLowerCase method, 211, 220
toRadians method, 206–207
toString method, 156, 160, 198, 211, 215, 218, 228, 273–274, 287, 331, 336, 339, 430
ToStringDemo example, 218
toUpperCase method, 211, 220
transient keyword, 603

TreeMap implementation, 320–321, 339, 342–343, 347
TreeSet implementation, 302, 304, 334, 336, 342–345
TrigonometricDemo example, 206–207
trigonometric functions, 206–208
trim method, 220
troubleshooting. *See* errors
try block, 235, 238–245, 603
tryLock method, 395
type erasure, 178–179
type inference, 173
type parameters
 bounded, 173–174
 formal, 170
 multiple, 171
 naming, 171–172
types, 183. *See also* by individual type
 in interfaces, 140
 multiple in one source file, 185
 package-private, 185
 public, 185
type variables, 169–171

U

U type parameter, 172
UnaryDemo example, 58
unary operators, 55, 57–59, 62
unboxing, 196, 211
underscore. *See* `_`
Unicode encoding, 46, 265
 escape sequences in, 48
Uniform Resource Identifiers (URIs), 483
UNIX/Linux, 2–3, 6–7, 19–23
 common error messages on, 27, 29–30
 environment variables on, 193, 451, 458–460
 file name separators on, 191
 file names on, 287
 path separators on, 193
 root directories on, 289
unmodifiableXXX methods, 352
unset command, 30
UnsupportedClassVersionError, 601
UnsupportedOperationException, 296, 352–353
URL class, 515, 518
URLClassLoader implementation, 514–515
useDelimiter method, 271
UTF-8 encoding, 281, 502, 533

utilities

- configuration, 443–451
- system, 452–457

V

- V type parameter, 172
- valueOf method, 198, 216
- ValueOfDemo example, 216–217
- values method, 130, 322–323
- varargs, 94–95
- variables, 34, 43–55. *See also* by individual
 - type; fields
 - as members, 44
 - assigning literals to, 48
 - atomic, 401–402
 - declaring, 45, 67, 99–100
 - initializing, 28
 - local, 44, 47, 88
 - naming, 44–45, 89–90
 - reference, 100
 - referring to arrays, 50–51
 - volatile, 384
- Vector collection, 293, 297, 307–309, 343, 346, 351
 - compatibility and, 364–366
- verify method, 279
- vertical bar. *See* |
- vi text editor, 19
- void keyword, 39, 81, 107, 603
- volatile keyword, 384, 603

W

- wait method, 157, 386–390
- WarningDemo example, 179
- warning messages, 134, 179
 - suppressing, 132, 135
- WeakHashMap implementation, 348
- Web browsers, 543
 - displaying documents in, 562–564
 - frames in, 562
 - quitting, 550
 - reloading applets, 549
 - running applets, 544, 559–578
 - security and, 558–560, 564, 567, 578–581, 585
 - status lines of, 561
 - that don't understand the <APPLET> tag, 575–576

Web pages

- adding applets to, 594–600
- creating, 533–534

Web servers

- applets and, 589–593
- placing applications on, 533
- setting up, 525

- while statement, 74–75, 603
 - skipping the current iteration, 79
 - terminating, 77

WhileDemo example, 74

- white space, 211
 - leading and trailing, 220
 - tokens and, 270

wildcards, 177–178, 188–189

- bounded, 177
- unbounded, 177

Windows. *See* Microsoft Windows

Windows look and feel, 470–478

wrappers, 342–343, 350–352

- checked interface, 352
- synchronization, 343, 350–351
- unmodifiable, 351–352

write method, 273

writeDouble method, 280

writeInt method, 280

writeObject method, 284–285

writer method, 277

Writer class, 265

writeUTF method, 280–281

Z

- ZIP archives, 289–291, 488