filters for, 904–918
stream model for, 904
Imaging, 893
implements clause, 210
and generic interfaces, 369, 370
import statement, 206–207, 1163
and static import, 339–341
in, 318, 501, 505, 657, 718. *See also* System.in
Increment operator (++), 34, 67, 70–72
indexOf( ), 460–462, 474, 475, 543, 544, 604
IndexOutOfBoundsException, 240, 543
Inet4Address class, 763
Inet6Address class, 763
InetAddress class, 761–763, 774
InetSocketAddress class, 774
infinity (IEEE floating-point specification value), 482
inForkJoinPool( ), 966
INHERIT, 505
InheritableThreadLocal class, 527
Inheritance, 5, 22, 23–24, 26–27, 150, 152, 171–195
and annotations, 314
and enumerations, 283
final and, 194–195
and interfaces, 199, 208, 218–219, 222–223, 224
and local variable type inference, 195–196
multilevel, 181–184
and multiple superclasses, 173, 199
@Inherited built-in annotation, 305, 306
init( ) and servlets, 1122, 1125, 1127
initCause( ), 242, 244
Inline method calls, 194
Inner classes, 157–159, 810–813
anonymous, 159, 812–813, 850, 851, 940, 1037, 1054, 1075, 1089–1090
InputEvent class, 787, 790–791, 792, 793, 1083
InputMismatchException, 661
InputStream class, 316, 317, 319, 657, 688, 689, 690, 695, 698, 699, 701, 702,

706, 723, 726, 748, 780, 781, 1130
methods, table of, 690
objects, concatenating, 702
InputStreamReader class, 318, 319
insert( ), 472, 867, 1076
insertSeparator( ), 1076
Insets
definition of the term, 827–828
example program, 872–873
Insets class, 817, 828, 872–873, 1039
Instance of a class, 23, 117, 119, 122. *See also* Object(s)
Instance variables
accessing, 119, 124, 125, 128
default values of, 131
definition of the term, 23, 118
hiding, 133
and interfaces, 220
static, 153–154
transient, 333
as unique to their object, 119, 120–121
using super to access hidden, 180–181
instanceof operator, 67, 333–335, 570
and generic classes, 376–378
Instant class, 627, 1021
Instrumentation interface, 535
int, 31, 39, 40, 41
and automatic type conversion, 52
and automatic type promotion, 54–55, 75–76, 78–79
and integer literals, 45
IntBuffer class, 729
Integer class, 287, 288, 289, 292, 477, 483, 490–491, 975
constructors, 288
methods, table of, 486–488
Integer(s), 39, 40–42, 72–73
literals, 45–46, 52
interface keyword, 199, 208

and annotations, 295
Interface methods
default, 18, 19, 209, 219–223, 389, 391
extension, 219
private, 19, 209, 224–225
static, 209, 223–224
traditional, 208, 210, 391
Interface(s), 199, 208–225
functional. *See* Functional interfaces
general form of, 209
generic. *See* Generic interfaces
implementing, 210–212
inheritance hierarchy, 208
inheritance of, 218–219, 223
and JShell, 1160–1161
member, 212
methods. *See* Interface methods
nested, 212–213
reference variables, 210–212, 216
and state information, 220, 223
types for bounded types, using, 357
variables, 209, 216–218
interfaceModifiers( ), 1007
interim( ), 502–503
Internet, 3, 6, 7, 8, 9, 15, 20, 759
addresses, obtaining, 762–763
addressing scheme, 760
and portability, 7, 8, 9–10
and security, 8, 9
Internet Engineering Task Force (IETF) BCP 49, 636
Internet of Things (IoT), 419
Internet Protocol (IP)
addresses, 760
definition of the term, 759
InterNIC, 764, 765
InterruptedException, 241, 251–252, 902

Introspection, 1110–1113, 1116, 1118–1119

Introspector class, 1115, 1116

ints( ), 638

IntStream interface, 972, 973, 985

IntSummaryStatistics class, 672

intValue( ), 288, 289, 478, 480, 481, 484, 485, 486, 489

InvalidPathException, 735

invoke( ), 953, 955, 959, 965, 967

invokeAll( ), 953, 959, 963, 966

invokeAndWait( ), 1034, 1035

invokeLater( ), 1034

I/O, 30, 315–332, 679–726

- channel-based, 15, 316, 727. *See also* NIO; NIO and channel-based I/O
- classes, list of, 680
- console, 30, 99, 315, 319–323, 718–720
- error handling, 326–329
- exceptions, 687
- file, 323–332, 681–686
- formatted. *See* I/O, formatted
- interfaces, list of, 680
- and Java’s GUI frameworks, 315
- new. *See* NIO
- redirection, 505
- streams. *See* Streams, I/O

I/O, formatted, 16

- format specifiers. *See* Format specifiers
- using Formatter, 642–657. *See also* Formatter class
- using printf( ), 163, 704–706
- using Scanner, 657–667. *See also* Scanner class

io package. *See* java.io package

IOError, 719

IOException, 99, 319, 324, 327, 328, 687, 689, 690, 694, 701, 709, 710, 711, 717, 722, 723, 733, 752, 756, 764, 768, 773

ipadx constraint field, 879, 880

ipady constraint field, 879, 880

IPv4 (Internet Protocol, version 4), 760, 761, 762, 763

IPv6 (Internet Protocol, version 6), 760, 761, 762, 763
isAbsolute( ), 682, 733
isAlive( ), 250, 258–260, 496, 498, 521
isAltDown( ), 791
isAltGraphDown( ), 791
isAnnotationPresent( ), 301, 303, 528
isBound( ), 764, 774, 1116
isCancelled( ), 965, 966
isClosed( ), 764
isCompletedAbnormally( ), 966
isCompletedNormally( ), 966
isConnected( ), 764, 774
isConstrained( ), 1116
isControlDown( ), 791
isDigit( ), 492, 493, 495
isDirectory( ), 683–684, 734, 737
isEditable( ), 865, 867
isEmpty( ), 467, 541, 542, 572, 605, 609, 610, 611, 622, 625
isEnabled( ), 883, 1095
isExecutable( ), 734, 750
isExported( ), 529
isFile( ), 682
isHidden( ), 683, 734, 736, 738, 750
isInfinite( ), 480, 482–483
isLeapYear( ), 632
isLetter( ), 492, 493, 495
isLightweight( ), 892
isLowercase( ), 492, 493
isMetaDown( ), 791
isMulticastAddress( ), 763
isNamed( ), 529
isNaN( ), 480, 482–483
isNestMateOf( ), 515
ISO-Latin-1 character set, 43, 47
isOpen( ), 529
isPopupTrigger( ), 793, 1088, 1090

isPresent( ), 625, 975
isPropertyName( ), 1111
isPublic( ), 1006
isQuiescent( ), 967
isReadable( ), 734, 750
isSelected( ), 1050, 1052, 1054, 1076
isSet array (Calendar class), 628
isShiftDown( ), 791
isShutdown( ), 968
isTemporary( ), 790
isTerminated( ), 968
isTimeSet (Calendar class instance variable), 628
isUppercase( ), 492, 493
isWhitespace( ), 492, 493
isWritable( ), 734, 738, 750
ItemEvent class, 787, 791–792, 852, 856, 859, 884, 1050, 1052
ItemListener interface, 797, 798, 851, 852, 856, 884, 1050, 1052
ItemSelectable interface, 792
itemStateChanged( ), 798, 852, 856, 1050, 1052
Iterable interface, 467, 533, 540, 565, 570, 603
Iterable<Path> interface, 732, 753
Iteration statements, 87, 95–106
Iterator interface, 539, 540, 561, 563–565, 566, 602, 603, 990, 991
    methods, table of, 562
iterator( ), 533, 541, 543, 563, 753, 970, 990
Iterator(s), 539, 543, 561–565, 566–568
    and maps, 570
    obtaining an, 563
    and primitive types, 539
    and PriorityQueue, 560
    and stream API streams, 990–993
    and synchronized collections, 595

J

J2SE 5, features added by, 16

JApplet class, 1029
JAR files. *See* Java Archive (JAR) files
jar tool, 445
Java
  and C, 3, 5, 7, 13
  and C++, 3, 7–8, 13
  and C#, 8
  design features (buzzwords), 12–15
  history of, 3, 6–8, 15–20
  and the Internet, 3, 6, 7–10, 14, 20, 759
  as an interpreted language, 10, 14
  keywords. *See* Keywords
  look and feels, 1027
  open-source implementations of, 20, 27
  release schedule, 11–12, 19
  as a strongly typed language, 13, 39, 45
  versions of, 15–20, 502–503
  and the World Wide Web, 7, 13
Java Archive (JAR) files, 446, 673
  modular, 445
Java Beans, 514, 536, 995, 1004, 1109–1119
  advantages of, 1110
  API, 1114–1116
  customizers, 1113
  definition of the term, 1109
  demonstration program, 1116–1119
  events, design pattern for, 1112
  introspection, 1110–1113, 1116, 1118–1119
  persistence, 1113
  properties. *See* Property, Java Bean
  serialization, 1113
Java Community Process (JCP), 20
Java EE SDK, 1122, 1126
.java filename extension, 27, 1165
Java Foundation Classes (JFC), 1026
java (Java application launcher), 28, 29, 201, 324, 446, 1165, 1166

and main( ), 29
and modules, 419, 425, 446
Java Native Interface (JNI), 336
java package, 201, 207
Java Runtime Environment (JRE), 10, 11, 427
Java SE 7, 17–18
Java SE 8, 18
Java SE 9, 18–19
Java SE 10, 19
Java SE 11, 19
Java Virtual Machine (JVM), 10, 11, 14, 15, 20, 28, 29, 497, 520, 535
Java Web Start, 11, 19
java.awt package, 783, 786, 816, 894, 1036
classes, tables of some, 816–818
java.awt.Dimension class, reflection example using the, 1004–1005
java.awt.event package, 783, 785, 786, 797, 808, 1035, 1036
adapter classes, table of commonly used, 809
event classes, table of commonly used, 787
interfaces, table of commonly used, 797
java.awt.event.InputEvent class. *See* InputEvent class
java.awt.event.KeyEvent class. *See* KeyEvent class
java.awt.image package, 893, 899, 904, 915, 918
java.base module, 427, 477, 534, 537, 673, 727, 761, 783, 816, 920, 970, 995, 1004, 1011, 1016
java.beans package, 1112, 1114–1116
classes, table of, 1114–1115
interfaces, table of, 1114
java.desktop module, 427, 783, 894, 1030, 1114
java.instrument module, 535
java.io package, 315, 316–318, 324, 330, 679–680, 686, 727, 750
automatically imported by JShell, 1163
classes, list of, 680. *See also* Streams, byte; Streams, character
interfaces, list of, 680
java.io.Externalizable interface, 721, 1113
java.io.IOException. *See* IOException
java.io.Serializable interface, 1113. *See also* Serializable interface

java.lang package, 207, 240, 296, 305, 318, 324, 330, 366, 369, 449, 427, 477–536, 686
classes and interfaces, list of, 477
implicit importation of the, 207, 477
subpackages, 534–536
java.lang.annotation package, 295, 305, 312, 534, 535
java.lang.annotation.RententionPolicy enumeration, 295, 535
java.lang.image package, 901
java.lang.instrument package, 534, 535
java.lang.invoke package, 534, 535
java.lang.management package, 534, 535
java.lang.module, 529, 534, 535
java.lang.ref package, 535
java.lang.reflect package, 296, 301, 535–536, 995, 1004
classes, table of, 1004
java.management module, 535
java.net package, 759, 772, 777
classes and interfaces, list of, 760–761
java.net.http, 19, 759, 777–782
java.nio package, 316, 679, 727, 728
java.nio.channels package, 727, 730, 732
java.nio.channels.spi package, 727
java.nio.charset package, 727, 731
java.nio.charset.spi package, 727
java.nio.file package, 683, 727, 732
java.nio.file.attribute package, 727, 732, 736
java.nio.file.spi package, 727, 732
java.nio.file.WatchService, 758
java.prefs module, 676
java.rmi module, 1008
java.rmi package, 995, 1008
java.text package, 995, 1011
java.time package, 628, 995, 1015, 1016, 1021
java.time.Duration, 947, 1021
java.time.format package, 1015, 1017, 1018
java.time.temporal.ChronoUnits, 947

java.util package, 434, 537–538, 602, 619, 783, 786, 975, 990
automatically imported by JShell, 1163
classes, list of top-level, 537–538
interfaces defined by, list of, 538
subpackages, 673–677
java.util.concurrent package, 673, 920–921, 946, 952
java.util.concurrent.atomic package, 673, 920, 921, 950, 951
java.util.concurrent.locks package, 673, 920, 921, 948, 949, 950
java.util.function package, 18, 417, 533, 566, 584, 600, 619, 673, 976, 977, 982, 989
functional interfaces defined by, table of, 674–676
java.util.jar package, 673
java.util.List class. *See* List class (java.util)
java.util.logging package, 673, 676
java.util.prefs package, 673, 676
java.util.regex package, 673, 677, 995
java.util.spi package, 673, 677
java.util.stream package, 18, 673, 677, 970, 986, 994
java.util.zip package, 673, 677
java.xml module, 427
JAVA_HOME environmental variable, 444
javac (Java compiler), 27–28, 201, 308, 372, 1156, 1158, 1165, 1166
and modules, 419, 420, 424–425, 446
and multi-module compilation mode, 429
javadoc, 19, 1147–1153
tags, 1147–1153
JavaFX, 19, 315, 801
javap, 383
javax.imageio package, 894, 918
javax.servlet package, 1125, 1126–1130, 1132
interfaces and classes, list of core, 1126–1127
javax.servlet.http package, 1126, 1132–1137
interfaces and classes, list of some, 1132
javax.swing package, 1028, 1030, 1032, 1033, 1043, 1066
classes, list of, 1028–1029
javax.swing.event package, 1035, 1046, 1061, 1066

javax.swing.table package, 1069
javax.swing.tree package, 1066
JButton class, 1029, 1037, 1043, 1047–1049, 1074, 1077, 1095
JCheckBox class, 1043, 1047, 1050, 1052–1053, 1095
JCheckBoxMenuItem class, 1074, 1085, 1086–1087, 1095
JComboBox class, 1043, 1063–1065
JComponent class, 1028, 1029, 1038, 1039, 1043, 1047, 1075, 1085
JDialog class, 1029, 1105
JFormattedTextField class, 1105
JFrame class, 1029, 1030–1031, 1032, 1033–1034, 1042
    adding a menu bar to a, 1076, 1078, 1081
    content pane layout manager, 1033, 1036–1037
JIT (Just-In-Time) compiler, 10, 14
JLabel class, 1029, 1030, 1031, 1033, 1035, 1038, 1043–1045, 1081
JLayeredPane class, 1029
jlink tool, 11, 18, 444–445
JList class, 1043, 1060–1063
JMenu class, 1074, 1075, 1076–1077, 1078
    mnemonic, 1082
JMenuBar class, 1074, 1075–1076, 1078
JMenuItem class, 1074, 1075, 1076, 1077, 1078, 1085, 1086
    accelerator key, 1083, 1084
    action to create a, using an, 1095
    and action events, 1077, 1078, 1081, 1082
    mnemonic, 1082, 1083–1084
JMOD files, 18, 444, 445
join( ), 250, 258–260, 466–467, 522, 947, 953, 963, 964, 967
JOptionPane class, 1105
Joy, Bill, 6
JPanel class, 1029, 1030, 1040, 1042, 1056, 1058
JPEG image file format, 893
JPopupMenu class, 1074, 1087–1090, 1095
    and mouse events, 1088, 1089–1090
JRadioButton class, 1043, 1047, 1050, 1053–1055, 1095
JRadioButtonMenuItem class, 1074, 1086–1087, 1095
JRootPane class, 1029

JScrollBar class, 1029
JSeparator class, 1074, 1076
JScrollPane class, 1043, 1058–1060, 1061, 1066, 1067, 1069
jshell, 1155
JShell, 19, 1155–1164
    commands, 1158, 1163–1164
    and state information, 1155, 1157
JSpinner class, 1105
JTabbedPane class, 1043, 1056–1058
JTable class, 1043, 1069–1071
JTextComponent class, 1045
JTextField class, 1043, 1045–1047
JToggleButton class, 1043, 1047, 1049–1051, 1053
JToggleButton.ToggleButtonModel class, 1050
JToolbar class, 1074, 1091–1093
    adding an action to a, 1095
JTree class, 1043, 1065–1068
Jump statements, 87, 109–115
Just In Time (JIT) compiler, 10, 14
JVM (Java Virtual Machine), 11, 14, 15, 20, 28, 29, 497, 520, 535
JWindow class, 1029

K

Kernighan, Brian, 4
Key codes, virtual, 792, 806
KeyAdapter class, 809, 810
Keyboard events, handling, 804–808
KeyEvent class, 787, 789, 790, 792, 1082, 1083
KeyListener interface, 797, 798–799, 804–808, 809
keyPressed( ), 798–799, 804, 806
keyReleased( ), 798–799, 804
keys( ), 609, 610, 611
keySet( ), 571, 573, 612, 670, 772
KeyStroke class, 1083
keyTyped( ), 798–799, 804, 806
Keywords
    context-sensitive, 38, 420
    restricted, 420
    table of Java, 37

L

Label
    AWT standard control, 844, 845–846
    Swing, 1031, 1033, 1043–1045
    used with break statement, 111–113
    used with continue statement, 114
Label class, AWT, 817, 845
Lambda expression(s), 18, 389–406, 416–417, 566
    as arguments, passing, 399–401
    block, 390, 395–397
    body, 390, 395, 396
    and comparators, 586–587, 590
    definition of, 389
    and exceptions, 392, 402–403
    and generics, 397

to handle action events, 850–851, 1037–1038, 1054, 1075
parameters, 390–391, 392, 393–395, 403
and the stream API, 969
target type, 390, 391, 392, 397, 399, 401, 403
and var, 19
and variable capture, 403–404
Lambda arrow operator (–>), 18, 67, 85, 390
last( ), 546, 876
lastElement( ), 604, 605
lastIndexOf( ), 460, 461–462, 474, 475, 543, 544, 604, 605
lastKey( ), 575
Late binding, 194
Layered pane, 1029–1030
Layout managers, AWT, 819, 843, 846, 868–883
default, 869
LayoutManager interface, 869
Lazy behavior (stream API stream), 972
length instance variable of arrays, 155–157
length( ), 161, 452, 469, 532, 622
Lexer (lexical analyzer), 619
Libraries, class, 27, 38
Library, compact profiles of the API, 344
Lindholm, Tim, 6
LineNumberInputStream deprecated class, 680
LineNumberReader class, 318
lines( ), 715, 733, 973
LinkedBlockingDeque class, 947
LinkedBlockingQueue class, 947
LinkedHashMap class, 578, 581–582
LinkedHashSet class, 551, 557–558
LinkedList class, 551, 555–556
example program using the, 569–570
from a stream API stream, obtaining a, 989
LinkedTransferQueue class, 947
List
controls, 844, 858–861

event source, as an, 787, 791, 796, 859
pop-up, 855
List class (java.awt), 817, 858, 859
List interface (java.util), 540, 543–545, 551, 555, 556, 563, 564, 597, 603, 604
from a stream API stream, obtaining an instance of the, 986–988
methods, table of, 544
List, Swing, 1060–1063
list( ), 614, 733
and directories, 681, 683–685
list( ), ThreadGroup, 524
listFiles( ), 685–686
ListIterator interface, 540, 561, 563–565, 566
methods, table of, 563
listIterator( ), 544, 563
ListModel interface, 1061
ListResourceBundle class, 670–672
ListSelectionEvent class, 1061, 1062, 1069
ListSelectionListener interface, 1061, 1062
ListSelectionModel interface, 1061, 1069
Literals, 36, 45–48
class, 298
regular expression, 997
string, 48, 452
load( ), 434–435, 439–440, 499, 507, 614, 616–618
LocalDate class, 1016, 1017, 1020, 1021
LocalDateTime class, 1016, 1017, 1020, 1021
Locale class, 466, 635–636, 1012
Locale Data Markup Language (LDML), 636
Locale.Builder class, 636
Locale.IsoCountryCode, 636
LocalTime class, 1016, 1017, 1020, 1021
Lock interface, 921, 948, 949, 950
methods, table of, 948
lock( ), 921, 948, 949
Locks, 948–950
reentrant, 949, 950

Log, program, 509
log( )
  math method, 516
  servlet method, 1128, 1130
Logical operators
  bitwise, 73–75
  Boolean, 81–83
long, 39, 40, 41–42, 45
  and automatic type conversion, 52
  and automatic type promotion, 54
  literal, 46
Long class, 287, 288, 477, 483, 490, 491
  methods, table of, 488–490
LongAccumulator class, 951
LongAdder class, 951
LongBuffer class, 729
longs( ), 638
LongStream interface, 972, 973
longValue( ), 288, 478, 480, 482, 484, 485, 486, 489
Look and feels, 1026–1027
lookup( ), 1009
Loop(s), 87
  Boolean object to control, using a, 293
  break statement and, 109–111
  continue statement and, 113–114
  do-while, 96–99
  for. *See* for loop
  infinite, 102–103, 110
  nested, 108–109, 110, 111–112
  with polling, event, 248, 265
  while, 85–96
Low surrogate char, 495

M

main (default name of main thread), 252

main( ), 29–30, 117, 118, 120, 150, 153
  and the java application launcher, 29
  and command-line arguments, 29, 162–163
  and the source-file launch feature, 1165, 1166
  and Swing programs, 1034
major( ), 503
MalformedURLException, 767
Map interface, 571–574, 575, 578, 582, 609, 610, 612
  methods, table of, 572–574
map( ), 625, 626, 731, 742, 743, 746, 780
  and stream API streams, 971, 982–986
Map(s), 539, 570–582
  classes, 578–582
  collection-view of a, obtaining a, 539, 570, 571
  flat, 986
  interfaces, 571–577
  and stream API streams, 982–986
  submaps of, 574–575
Map.Entry interface, 571, 575, 579
  methods, table of non-static, 577
MapMode.PRIVATE, 742, 743
MapMode.READ_ONLY, 742, 743
MapMode.READ_WRITE, 742, 743, 746
MappedByteBuffer class, 729, 743
mapToDouble( ), 971, 985
mapToInt( ), 971, 985–986
mapToLong( ), 971, 985
mark( ), 689, 690, 695, 699, 702, 709, 715, 729
markSupported( ), 689, 690, 699, 702, 709, 715
Matcher class, 995, 996–997, 998, 999, 1002, 1003
matcher( ), 996
matches( ), 467, 996, 1000, 1003
Math class, 49, 139, 515–519
  exponential functions, methods for, 516
  miscellaneous methods, table of, 518–519
  rounding methods, table of, 517–518

and static import example, 339–341

trigonometric functions, methods, for, 515

max( ), 410–412, 480, 482, 486, 489, 517, 971, 976, 977

algorithm defined by Collections, 593, 596

MAX_EXPONENT, 479

MAX_PRIORITY, 260–261, 520

MAX_RADIX, 492

MAX_VALUE, 479, 490, 492

MediaTracker class, 817

Member, class, 23, 118

access and inheritance, 173–174

access, table of, 203

controlling access to a, 149–152, 202–206

static, 153–154

Member interface, 536, 1004

Memory

allocation using new, 56, 57, 121–122

deallocation, 133

leaks, 324, 329, 687

management, in Java, 14, 133

and the Runtime class, 497–498, 500

MemoryImageSource class, 899–901, 904

Menu bars and AWT menus, 843, 883–888

action command string of, 884

and events, 884

Menu, Choice control as a form of, 855

Menu class, AWT, 817, 883, 884, 888

Menu item as an event source, AWT, 787, 791, 796, 884

Menu(s), Swing, 1073–1105

accelerator keys, 1073, 1082, 1083–1084, 1097

action command string, 1074–1075, 1082

action to manage multiple components of a, using an, 1073, 1093–1098

and check boxes, 1085, 1086–1087

classes, interaction of core, 1074–1075

demonstration program, 1099–1105

events, 1074–1075, 1077, 1078, 1081–1082, 1085, 1086, 1088, 1089–1090

and images, 1084–1085
item as an event source, 786, 791, 796
main, creating a, 1078–1082
menu bar, 1073, 1075–1076, 1078, 1081, 1089, 1091
mnemonics, 1073, 1077, 1082, 1083–1084, 1097–1098
popup, 1073, 1074, 1087–1090
and radio buttons, 1085, 1086–1087
and submenus, 1074, 1076, 1077, 1078
and toolbars, 1073, 1074, 1091–1093
and tooltips, 1085, 1091, 1092, 1093
MenuBar class, AWT, 817, 883, 884
MenuDragMouseEvent, 1075
MenuEvent, 1075
MenuItem class, AWT, 817, 883–884
MenuKeyEvent, 1075
MenuListener, 1075
Metadata, 289, 294. *See also* Annotation(s)
Metal look and feel, 1027
Method class, 297, 300, 301, 536, 1004, 1006, 1116
Method reference(s), 389, 404–412
and the Collections Framework, 410–412
and generics, 409–412
to instance methods, 405–409
to static methods, 404–405
to a superclass version of a method, 409
method( ), 779
Method(s), 23, 118, 123–129
abstract. *See* Abstract method(s)
and annotations, 295, 314
and autoboxing/unboxing, 290–291
bridge, 382–384
calling, 125,126
collector, 986
default interface, 18, 19, 209, 219–223, 389, 391
dispatch, dynamic, 188–191
and the dot (.) operator, 119, 125

factory, 275–276, 761
final, 155, 194
general form, 123–124
generic, 346, 358, 364–367, 386
getter, 1110–1111
hidden, using super to access, 180–181, 186
inlining, 194
interface. *See* Interface methods
and Java Bean introspection, 1112
and JShell, 1159–1160
lookup, dynamic, 211
native, 336
overloading, 137–142, 166–168, 187, 384–385
overriding. *See* Overriding, method
and parameters, 124, 127–129
passing an object to, 145–146
recursive, 147–149
reference. *See* Method reference(s)
resolution, dynamic, 208, 211, 212, 216
returning an object from, 146–147
returning a value from, 124, 126–127
scope defined by, 50–52
setter, 1110–1111
static, 153–154, 223–224, 339–341, 404–405
subclasser responsibility, 192
synchronized, 250, 261–263, 921, 968
synthetic, 1156
type inference and, 366, 380–381
varargs. *See* Varargs
variable-arity, 163
MethodDescriptor class, 1112, 1115, 1116
MethodHandle class, 535
methodModifiers( ), 1007
MethodType class, 535
Microsoft, 8
MIME (Multipurpose Internet Mail Extensions) type, 1121, 1125

min( ), 480, 482, 486, 489, 517, 972, 975, 976, 977
  algorithm defined by Collections, 593, 596
minimumLayoutSize( ), 869
minor( ), 503
MIN_EXPONENT, 479
MIN_NORMAL, 479
MIN_PRIORITY, 260–261, 520
MIN_RADIX, 492
MIN_VALUE, 479, 490, 492
mismatch( ), 601
mkdir( ), 686
mkdirs( ), 686
Model-Delegate component architecture, 1028
Model-View-Controller (MVC) component architecture, 1027–1028
Modifier class, 1004, 1006, 1007
  “is” methods, table of, 1006
Module class, 528–529
module keyword, 37, 420
module statement, 420, 426, 443
Module(s), 18–19, 419–446
  and access control, 150, 419, 420, 426, 428–429, 443
  automatic, 446
  and compact profiles, 344
  compiling and running code based on, 424–425
  creating code based on, 420–424
  declaration, 419–420
  definition of the term, 419
  description, 420
  graph, 442–443
  Java API library packages and, 18, 419, 427
  layers, 446
  and legacy code, 427–428
  and multi-module compilation mode, 429, 433
  naming conventions, 421
  open, 443
  and packages, 201

path, 201, 424, 425, 428, 444
platform, 427, 445
and services and service providers, example application demonstrating, 435–442
unnamed, 427
module-info.java file, 420, 424, 425, 426
and javadoc, 1148
ModuleDescriptor class, 529, 535
ModuleFinder interface, 535
ModuleLayer class, 529
ModuleLayer.Controller class, 529
ModuleReader interface, 535
ModuleReference class, 535
Modulus operator (%), 67, 69
Monitor, 250, 261, 264, 265, 271
Mouse events, handling, 801–804
MouseAdapter class, 808, 809, 813, 1088, 1089–1090, 1117
mouseClicked( ), 799, 808, 1088
mouseDragged( ), 799, 808, 898
mouseEntered( ), 799, 1088
MouseEvent class, 787, 789, 790, 793–794, 1088
mouseExited( ), 799, 1088
MouseListener interface, 797, 799, 801–804, 809, 810, 1088 1089–1090
MouseMotionAdapter class, 808, 809
MouseMotionListener interface, 785, 797, 799, 801–804, 808, 809, 810
mouseMoved( ), 799, 808, 898
mousePressed( ), 799, 811, 1088, 1090, 1117
mouseReleased( ), 799, 1088, 1090
MouseWheelEvent class, 787, 794–795
MouseWheelListener interface, 797, 799, 801, 809, 810
mouseWheelMoved( ), 799
Multicore systems, 17–18, 248, 276, 389, 919, 920, 952, 956, 959, 979
Multitasking, 247
preemptive, 249
process-based versus thread-based, 247
Multithreaded programming, 8, 14, 247–276, 520

and context switching. *See* Context switching
effectively using, 276
and multicore versus single-core systems, 248
and spurious wakeup, 266
and StringBuilder class, 475
and synchronization. *See* Synchronization
and threads. *See* Thread(s)
versus the concurrency utilities, traditional, 919, 968
and parallel programming, 952
versus single-threaded system, 248
MutableComboBoxModel, 1064
MutableTreeNode interface, 1066
MVC (Model-View-Controller) component architecture, 1027–1028

N

Namespace collisions
between instance variables and local variables, 133
packages and, 199, 207, 341
and static import, 341
Naming class, 1008, 1009
NaN, 479, 482
nanoTime( ), 507, 508, 960
@Native built-in annotation, 305
native modifier, 336
Natural ordering, 532, 582
naturalOrder( ), 583
Naughton, Patrick, 6
NavigableMap interface, 571, 575, 580
methods, table of, 576–577
NavigableSet interface, 540, 547–548, 558, 559
methods, table of, 547
Negative numbers in Java, representation of, 72–73
NEGATIVE_INFINITY, 479
NegativeArraySizeException, 240, 598
Nest, 515

host, 515
.NET Framework, 8
NetBeans, 1122, 1123
Networking, 759–782
basics, 759–760
bidirectional communication and, 777, 782
classes and interfaces, list of, 760–761
new, 56, 57, 121–122, 129, 131, 133, 147, 192, 236, 237
constructor reference and, 412, 416
and enumerations, 278, 281
and type inference, 380–381
NEW, 274
New I/O. *See* NIO
newBuilder( ), 778, 779
newByteChannel( ), 731, 734, 739, 741, 742, 743, 744, 746
newCachedThreadPool( ), 941
newCondition( ), 948, 949
newDirectoryStream( ), 734, 752, 754–755
newFileSystem( ), 738
newFixedThreadPool( ), 941–942
newHttpClient( ), 778
newInputStream( ), 735, 748–749
Newline, inserting a, 649
newOutputStream( ), 735, 748, 749–750
newScheduledThreadPool( ), 941, 942
next( ), 562, 563, 566, 660, 876, 990, 991
nextBoolean( ), 636, 637, 660
nextBytes( ), 636, 637
nextDouble( ), 218, 636, 637, 660, 662, 665
nextElement( ), 603, 620, 703
nextFloat( ), 636, 637, 660
nextGaussian( ), 636–638
nextInt( ), 636, 637, 660, 665
nextLong( ), 636, 637, 660
nextToken( ), 620
nextX( ) Scanner methods, 658, 660–661, 662, 665

table of, 660
NIO, 679, 727–758
buffers, 728–730
channels, 728, 730–731
charsets, 732
and directories, 752–758
packages, list of, 727
for path and file system operations, using, 750–758
selectors, 732
and the stream API, 735
for stream-based I/O, using, 738, 748–750
NIO and channel-based I/O
copying a file using, 747–748
reading a file using, 739–743
writing to a file using, 743–747
NIO.2, 17, 727, 738, 750
noneMatch( ), 994
NORM_PRIORITY, 261, 520
NoSuchElementException, 546, 548, 550, 574, 603, 625, 661, 667
NoSuchMethodException, 241, 297
NOT operator
bitwise unary (~), 72, 73, 74–75
Boolean logical unary (!), 81, 82
NotDirectoryException, 752
notepad, 501, 505
notify( ), 197, 265–266, 268–269, 272–274, 510, 919, 949, 968
notifyAll( ), 197, 265–266, 510
NotSerializableException, 725
now( ), 1016–1017
null, 38, 131
alternative to using, 624
Null statement, 96
NullPointerException, 237, 240, 541, 543, 546, 548, 550, 561, 571, 574, 575, 598, 610, 668, 703
using Optional to prevent a, 624, 626
nullsFirst( ), 583–584

nullsLast( ), 583–584
Number class, 288, 355, 477, 478
NumberFormatException, 240, 288
Numbers
 formatting, 645–646, 649–655
 pseudorandom, 636

O

Oak, 6
Object class, 196–197, 265, 346, 348, 382, 509–510
 as a data type, problems with using the, 350–352
Object class methods
 and functional interfaces, 390
 table of, 195, 510
Object reference variables
 and abstract classes, 192–193, 194
 and argument passing, 144, 145–146
 assigning, 122–123
 declaring, 121
 and cloning, 510
 and dynamic method dispatch, 188–191
 to superclass reference variable, assigning subclass, 176, 180, 188
Object-oriented programming (OOP), 5, 6, 21–27, 117
 model in Java, 13
Object(s), 23, 117, 122
 creating/declaring, 119, 121–122
 exact copy (clone) of, 509–510
 initialization with a constructor, 129–132
 to a method, passing, 145–146
 monitor, implicit, 250, 264
 as parameters, 142–144
 returning, 146–147
 serialization of. *See* Serialization
 type at run time, determining, 333–335
Object.notify( ). *See* notify( )

Object.wait( ). *See* wait( )
ObjectInput interface, 723
  methods defined by, table of, 723
ObjectInputFilter, 724
ObjectInputFilter.Config, 724
ObjectInputFilter.FilterInfo, 724
ObjectInputFilter.Status, 724
ObjectInputStream class, 317, 723–724
  methods defined by, table of, 724
ObjectOutput interface, 721–722
  methods defined by, table of, 721
ObjectOutputStream class, 317, 722–723
  methods defined by, table of, 722
Objects class, 672
Observable class, 619
Observer interface, 619
Octals, 45
  as character values, 47
  and string literals, 48
of( ), 543, 544, 545–546, 561, 562, 571, 573, 574, 625, 626, 732, 736, 739, 741–742, 947, 994
offer( ), 548, 560
offerFirst( ), 549, 550, 555
offerLast( ), 549, 550, 555
ofFile( ), 780
offsetByCodePoints( ), 467, 474
ofInputStream( ), 780, 781
ofLines( ), 782
ofLocalizedDate( ), 1017, 1018
ofLocalizedDateTime( ), 1017
ofLocalizedTime( ), 1017
ofNullable( ), 625, 626
ofPattern( ), 1018–1020
  pattern letters, 1019
ofString( ), 780, 782
onAdvance( ), 937–938, 940

open
  keyword, 37, 420
  modifier, 443
open( ), 731
openConnection( ), 768, 769–770
OpenOption interface, 735
opens
  keyword, 37, 420
  statement, 443
Operator(s)
  arithmetic, 67–72
  assignment. *See* Assignment operator(s)
  bitwise, 72–80
  Boolean logical, 81–83
  conditional-and, 83
  conditional-or, 83
  diamond (<>), 380–381
  parentheses and, 45, 85
  precedence, table of, 85
  relational, 32, 44, 45, 80–81
  ternary if-then-else (?:), 82, 84
Optional class, 624–626, 975, 976, 977
  methods, table of, 624–625
optional( ), 503
OptionalDouble class, 624, 626
OptionalInt class, 624, 626
OptionalLong class, 624, 626
OR operator
  bitwise (|), 72, 73, 74–75
  bitwise exclusive (^), 72, 73, 74–75
  Boolean logical (|), 81, 82
  Boolean logical exclusive (^), 81, 82
OR operator, short-circuit (||) Boolean logical, 81, 83
or( ), 625, 626
Oracle, 17, 1122
Ordinal value, enumeration constant’s, 283, 284

ordinal( ), 283, 284–285, 531
orElse( ), 625, 626
orElseThrow( ), 625, 975
out output stream, 30, 38, 318, 319, 322, 323, 501, 505. *See also* System.out
out( ), 643, 645
OutputStream class, 316, 317, 322, 688, 689, 692, 698, 700, 704, 706, 718, 722, 726, 749, 1130
  methods, table of, 691
OutputStreamWriter class, 318
Overloading methods, 137–142, 166–168, 187, 384–385
@Override, built-in annotation, 305, 307
Overriding, method, 185–191
  and abstract classes, 191–194
  and bridge methods, 382–384
  and dynamic method dispatch, 188–191
  final to prevent, using, 194
  in a generic class, 379–380
  and run-time polymorphism, 188, 189, 191

P

Package(s), 150, 199–208, 225
  access to classes contained in, 202–206, 207
  built-in standard Java classes and, 206, 207
  the default, 200, 206
  defining, 200
  finding, 200–201
  importing, 206–208
  and JShell, 1162–1163
  and modules, 201
  Swing, 1030
Package class, 301, 527–528
  methods, table of, 527–528
package statement, 200, 206
Paint mode, setting the, 832–833
paint( ), 801, 804, 821, 822, 824, 898, 1038–1039, 1118

lightweight AWT components and overriding, 892
Paintable area, computing, 1039–1040
paintBorder( ), 1038–1039
paintChildren( ), 1038–1039
paintComponent( ), 1038–1039, 1042
Painting in Swing, 1038–1042
Panel class, 817, 818, 819, 875
Panes, Swing container, 1029–1030. *See also* Content pane
Parallel processing, 18, 389, 566, 568
of a stream API stream, 969, 972, 973, 979–981, 988, 991, 993
Parallel programming. *See* Programming, parallel
parallel( ), 970, 979–980
parallelPrefix( ), 600
parallelSetAll( ), 600
parallelSort( ), 599–600
parallelStream( ), 541, 543, 973, 979, 980
Parameter(s), 29, 124, 127–129
and constructors, 131–132
final, 155
and lambda expressions, 390–391, 392, 393–395, 403
names and instance variable names, 133
objects as, 142–144
and overloaded constructors, 140–141
and overloaded methods, 137, 187
and the scope of a method, 50
servlet, reading, 1130–1132
type. *See* Type parameter(s)
variable-length (varargs), 165, 545
Parameterized types, 346, 348
parameterModifiers( ), 1007
parse( ), 503, 1020
parseByte( ), 484, 490
parseDouble( ), 482
parseFloat( ), 480
parseInt( ), 487, 490–491
parseLong( ), 489, 490

parseShort( ), 485, 490
Parsing, definition of, 619
Pascal, 4
Passwords, reading, 720
patch( ), 502–503
Path interface, 681, 683, 732–733, 739, 750, 751, 752–753, 756
converting a File object into an instance of the, 683, 732, 750
instance for stream-based I/O, using a, 748–750
methods, table of a sampling of, 733
obtaining an instance of the, 735–736, 738–739, 741, 742, 743, 744, 746
Path.of( ), 736, 739, 741
Paths class, 735–736, 739
Paths.get( ), 735–736, 738–739
Pattern class, 995, 996, 999, 1002, 1003
Pattern matching, regular expressions, 997–1003
PatternSyntaxException, 997
Payne, Jonathan, 6
peek( ), 548, 607, 608
peekFirst( ), 549, 555
peekLast( ), 549, 555
Peers, native, 892, 1025–1026
Period class, 1021
Persistence (Java Beans), 1113
Phaser class, 920, 922, 934–941
compatibility with fork/join, 968
PI (Math constant), 515
PIPE, 505
Pipeline for actions on stream API streams, 18, 389, 972, 984–985
PipedInputStream class, 317
PipedOutputStream class, 317
PipedReader class, 318
PipedWriter class, 318
PixelGrabber class, 901–904
Pluggable look and feel (PLAF), 1026–1027, 1028
Plug-in and pluggable application architecture, 434, 436
PNG file format, 894

Point class, 793, 794, 817
poll( ), 548, 560
pollFirst( ), 547, 549, 555
Polling, 248, 265
pollLast( ), 547, 549, 555
Polygon class, 817, 826
Polymorphism, 5, 22, 25–27, 195
    and dynamic method dispatch, 188–191, 192
    and interfaces, 208, 211, 216
    and overloaded methods, 137, 139, 140
pop( ), 549, 550, 607, 608
Popup trigger, 1088, 1090
PopupMenu class, 817, 888
PopupMenuEvent, 1075
Port, 759, 760, 767
Portable Operating System Interface (POSIX), 737
Portability problem, 6–7, 8, 9–10, 12, 14, 20
    and data types, 40
    and thread context switching, 249
POSITIVE_INFINITY, 479
PosixFileAttributes class, 736, 737, 752
PosixFileAttributeView interface, 738
Possessive behavior (regular expression pattern matching), 1001
postVisitDirectory( ), 756
pow( ), 339–341, 516
pre( ), 503
Predicate<T> predefined functional interface, 417, 676, 976
preferredLayoutSize( ), 869
previous( ), 563, 876
preVisitDirectory( ), 756
PrimitiveIterator interface, 539, 990
PrimitiveIterator.OfDouble interface, 539, 990
PrimitiveIterator.OfInt interface, 990
PrimitiveIterator.OfLong interface, 990
print( ), 31, 38, 322, 323, 454, 704, 718, 1130
printf( )

function, C/C++, 642, 704
method, Java, 163, 657, 704–706, 718, 719
println( ), 30, 31, 38, 197, 322, 323, 454–455, 645, 704, 718, 1130
and Boolean output, 45
and String objects, 65
PrintStream class, 317, 319, 322, 657, 703–706
PrintWriter class, 318, 322–323, 657, 717–718, 1125
PriorityBlockingQueue class, 947
PriorityQueue class, 551, 559–560
private access modifier, 29, 150–152, 202–203
and inheritance, 173–174
Process class, 495–496, 501, 503
methods, table of, 498
Process, definition of, 247, 495
Process-based versus thread-based multitasking, 247
ProcessBuilder class, 495, 503–505
methods, table of, 503–504
ProcessBuilder.Redirect class, 505
ProcessBuilder.Redirect.Type enumeration, 505
ProcessHandle class, 495–496
ProcessHandle.Info, 495
Program log, 509
Programming
multithreaded. *See* Multithreaded programming
object-oriented. *See* Object-oriented programming
process-oriented, 21, 22, 26
structured, 4, 5
Programming, parallel, 17–18, 249, 566, 920, 921, 952, 979
and the level of parallelism, 955
and specifying the level of parallelism, 955, 959–962, 968
Project Coin, 17
Properties class, 538, 602, 613–618
methods, table of, 614
Properties, environment, 509
Property, Java Bean, 1110, 1118
boolean, 1111

bound and constrained, 1113, 1116
design patterns for, 1110–1111, 1113
PropertyChangeEvent, 1113
PropertyChangeListener interface, 1113, 1114
PropertyDescriptor class, 1112, 1115, 1116, 1118
PropertyPermission class, 672
PropertyResourceBundle class, 670
PropertyVetoException, 1113
protected access modifier, 150, 202–203
Protocols, overview of networking, 759–760
provides
keyword, 37, 420
statement, 435, 542
Pseudorandom numbers, 636
public access modifier, 29, 150–152, 202–203
Push buttons, AWT, 844, 847–851
action command string of, 847, 849, 850, 1047–1058
Push buttons, Swing, 1035–1038, 1047–1049
action command string of, 1047–1048
push( ), 550, 607, 608
Pushback, 700–702
PushbackInputStream, 317, 698, 700–702
PushbackReader class, 318, 716–717
put( ), 571, 573, 578, 580, 582, 609, 610, 611
and buffers, 729, 730, 745
putAll( ), 573, 582
PutField inner class, 722–723
putValue( ), 1094–1095, 1096

Q

Query string, 1138
Queue, double-ended, 549–550
Queue interface, 540, 548, 549, 555, 559, 560
methods, table of, 548
quietlyInvoke( ), 967

quietlyJoin( ), 967

R

Race condition, 263
Radio buttons, 853
  Swing, 1053–1055
  and Swing menus, 1086–1087
Radix, 483
radix( ), 667
Random class, 218, 636–638
  methods, table of core, 637
RandomAccess interface, 540, 570
RandomAccessFile class, 708, 730, 731
range( ), 561, 562
Raw types, 370–372
READ, 505
read( ), 99, 317, 318, 319–320, 324–325, 534, 690, 699, 701, 709, 717, 723, 724, 731, 740, 741, 748, 894–895
Readable interface, 534, 657, 663, 709
ReadableByteChannel interface, 657
readAttributes( ), 735, 737, 751–752
readBoolean( ), 707, 724
readDouble( ), 707, 724
Reader class, 317–318, 319, 688, 689, 708–709, 710, 712, 721
  methods defined by, table of, 709
readExternal( ), 721
readInt( ), 707, 724
readLine( ), 320–322, 490, 719, 1130
readObject( ), 723, 724, 725
readPassword( ), 719–720
readString( ), 735
ReadWriteLock interface, 950
Real numbers, 42
rebind( ), 1008–1009
receive( ), 774

Recursion, 147–149
    and the Fork/Join Framework divide-and-conquer strategy, 956
RecursiveAction class, 921, 952, 954, 956, 957, 958, 962
RecursiveTask class, 921, 952, 954, 956
    example program using, 962–965
Redirect class, 505
reduce( ), 972, 977–979, 980–981, 982–983
Reduction operations, 977–979
    mutable, 989
ReentrantLock, 949
ReentrantReadWriteLock, 950
Reflection, 296, 443, 535–536, 995, 1004–1007
    and annotations, 296–301
    and nests, 515
regionMatches( ), 457–458
register( ), 934
Regular expressions, 468, 619, 657, 665, 677, 995–1003
    definition of the term, 995
    syntax, 997
    wildcards and quantifiers, 995, 997, 999–1001
reinitialize( ), 966
Relational operators, 32, 44, 45, 80–81
Relative index, 655–656
Release cadence, 12, 502
release( ), 922–925
Reluctant behavior (regular expression pattern matching), 1000–1001
Remote interface, 1008
Remote method invocation (RMI), 15, 720, 995, 1008–1011
RemoteException, 1008
remove( ), 541, 542, 544, 548, 556, 561, 562, 563, 573, 609, 610, 611, 844, 1034, 1075, 1077
removeActionListener( ), 1037
removeAll( ), 541, 542, 844
removeAttribute( ), 1135, 1142
removeEldestEntry( ), 582
removeElement( ), 604, 605

removeElementAt( ), 604, 605
removeFirst( ), 550, 555
removeIf( ), 541, 542
removeKeyListener( ), 785
removeLast( ), 550, 555
removeTListener( ), 1112
removeTypeListener( ), 785
renameTo( ), 682
repaint( ), 801, 804, 809, 810, 811, 822–823, 1039, 1118
@Repeatable annotation, 305, 312, 313
REPL (read-evaluate-print loop) execution, 1155
replace( ), 445, 473, 573
replaceAll( ), 468, 543, 544, 573, 997, 1002
replaceRange( ), 867
ReplicateScaleFilter class, 904
requires keyword, 37, 420
requires statement, 420, 426, 427
    using static with a, 443, 444
    using transitive with a, 430–434
reset( ), 667, 689, 690, 695, 697, 699–700, 702, 709, 715, 729
resolve( ), 732, 733
Resource bundles, 667–672
ResourceBundle class, 667–670
    methods, table of, 668–670
    and modules, 670
ResourceBundle.Control class, 668
resume( ), 15, 272, 520, 522
retainAll( ), 541, 542
@Retention built-in annotation, 296, 305
RetentionPolicy enumeration, 296, 535
return statement, 114–115, 124
    in a lambda expression, 396
reverse( )
    algorithm defined by Collections, 593
    StringBuffer method, 472
reversed( ), 583, 586

reverseOrder( )
  collection algorithm, 593, 595–596
  Comparator method, 583
rewind( ), 729, 741, 744, 745
RGB (red-green-blue) color model, 829
  default, 899
RGBImageFilter class, 904, 906–907
  example program demonstrating the, 907–918
RGBtoHSB( ), 830
Richards, Martin, 4
Ritchie, Dennis, 4, 5
rmi protocol, 1009
RMI (Remote Method Invocation), 15, 720, 995, 1008–1011, 1122
rmic compiler, 1010
rmiregistry (RMI registry), 1010, 1011
Run-time
  system, Java,10. *See also* Java Virtual Machine (JVM)
  type information, 15, 333, 376–378, 514
run( ), 250, 253, 254, 390, 520, 522, 639, 966, 1034
  overriding, 255, 256, 639
  using a flag variable with, 272–274
RUNNABLE, 274
Runnable interface, 250, 252, 390, 520, 638, 639, 919, 965, 966, 1034
  implementing the, 253–254, 256
Runtime class, 495, 496–501, 962
  executing other programs and, 501–501
  memory management and, 497–498, 500
  methods, table of some, 499
RUNTIME retention policy, 295–296, 297, 300
Runtime.exec( ), 495, 499, 501
Runtime.Version class, 502–503
Runtime.version( ), 499, 502
RuntimeException class, 228, 229, 237, 240, 244
RuntimePermission class, 529

S

@SafeVarargs built-in annotation, 305, 307
SAM (Single Abstract Method) type, 390
Sandbox, 10
save( ), 613
Scanner, 619
Scanner class, 657–667
    closing an instance of the, 663
    constructors, 657, 658
    delimiters, 657, 665–666
    demonstration programs, 661–665
    hasNextX( ) methods, table of, 659
    how to use, 657–658, 660–661
    methods, miscellaneous, 666–667
    nextX( ) methods, table of, 660
schedule( ), 639, 640
ScheduledExecutorService interface, 941
ScheduledThreadPoolExecutor class, 921, 941
Scientific notation, 46, 645–646
Scopes in Java, 49–52
    nested, 50–51, 158
Scroll bars, 844, 861–864, 1058, 1059
    as an event source, 788, 796, 862
Scroll pane, 1058–1060
Scrollbar class, 817, 861
ScrollPane class, 817
search( ), 607, 608
Security manager, 324, 505, 529, 687, 704, 1011, 1122
Security problem, 8, 9, 10, 12, 20, 720
    and servlets, 1122
security( ), 503
SecurityException, 240, 324, 497, 505, 687, 704, 752, 756
SecurityManager class, 529
seek( ), 708
SeekableByteChannel interface, 731, 739, 742, 744

select( ), 856, 859, 864–865, 867
Selection statements, 87–95
Selectors, 731, 732
Semaphore, 919, 920, 922–927, 968
 and setting initial synchronization state, 927
Semaphore class, 920, 922
send( ), 774, 777, 778, 779, 781, 782
Separable Model architecture, 1028
Separators, 37
SequenceInputStream class, 317, 702–703
sequential( ), 970, 981
Serializable interface, 721, 725, 967, 1113
Serialization, 720–726
 example program, 725–726
 and Java Beans, 1113
 and security, 720
 and static variables, 721
 and transient variables, 721, 725
serialVersionUID constant, 726
Server process, 759
ServerSocket class, 730, 763, 773
service( ), 1122, 1125, 1127, 1131, 1137
ServiceLoader class, 434, 632
Services and service providers, 434–442
 definition of the terms, 434
 module-based example application demonstrating, 435–442
 and modules, 435
Servlet interface, 1126, 1127, 1130
 methods, table of, 1127
Servlet(s), 12, 1121–1143
 advantages of, 1122
 API, 1126
 development options, 1122–1124
 example program for a simple, 1124–1126
 life cycle of, 1122
 parameters, reading, 1130–1132

and portability, 12
and security, 1122
and session tracking, 1142–1143
using Tomcat to develop, 1122, 1123–1126
ServletConfig interface, 1126, 1128, 1130
ServletContext interface, 1126, 1128
methods, table of various, 1128
ServletException class, 1127, 1130
ServletInputStream class, 1127, 1130
ServletOutputStream class, 1127, 1130
ServletRequest interface, 1125, 1126, 1128, 1130
methods, table of various, 1129
ServletResponse interface, 1125, 1126, 1128
methods, table of various, 1129
Session tracking, HTTP, 1142, 1143
Set interface, 540, 545–546, 556, 561, 571, 575
from a stream API stream, obtaining an instance of the, 986–988
Set-view, obtaining, 579, 612
set( ), 543, 544, 556, 563, 622, 629, 951
setAccelerator( ), 1083
setActionCommand( ), 850, 884, 1047–1048, 1054
setAlignment( ), 845
setAll( ), 600
setAttribute( ), 1128, 1135, 1142
setBackground( ), 821, 829
setBlockIncrement( ), 862
setBorder( ), 1042
setBounds( ), 819, 869
setCharAt( ), 470
setColor( ), 830
setConstraints( ), 878
setContentType( ), 1125, 1129
setDefault( ), 634, 635
setDefaultCloseOperation( ), 1032–1033
setDisabledIcon( ), 1047, 1085
setDisplayedMnemonicIndex( ), 1082

setEchoChar( ), 865
setEditable( ), 865, 867
setEnabled( ), 883, 1077, 1095
setFont( ), 836
setForeground( ), 821, 829
setForkJoinTaskTag( ), 967
setHorizontalTextPosition( ), 1085
setIcon( ), 1044, 1084–1085
setJMenuBar( ), 1076, 1078
setLabel( ), 847, 851, 883–884
setLastModified( ), 683
setLayout( ), 846, 869, 1034
setLength( ), 470, 708
setLocation( ), 819
setMaxAge( ), 1136, 1142
setMnemonic( ), 1082
setN( ) setter method design pattern, 1110, 1111
setName( ), 251, 252, 522
setObjectInputFiller( ), 724
setPaintMode( ), 832
setPreferredSize( ), 819, 862
setPressedIcon( ), 1047
setPriority( ), 260–261, 522
setReadOnly( ), 683
setRolloverIcon( ), 1047
setSelectedCheckbox( ), 854
setSelectedIcon( ), 1047
setSelectionMode( ), 1061
setSize( ), 605, 819, 820, 824, 1032
setState( ), 851, 884
setText( ), 845, 864, 867, 1044, 1047
setTitle( ), 820
setToolTipText( ), 1085
setUnitIncrement( ), 862
setValue( ), 577, 861–862, 1136
setValues( ), 861

setVisible( ), 820, 823, 824, 1034
setXORMode( ), 832–833
Sheridan, Mike, 6
Shift operators, bitwise, 72, 75–79
Short class, 287, 288, 477, 483, 490, 491
    methods defined by, table of, 485
short data type, 39, 40, 41, 45
    and automatic type conversion, 52
    and automatic type promotion, 54, 75
ShortBuffer class, 729
shortValue( ), 288, 478, 480, 482, 484, 485, 487, 489
show( ), 876, 1088, 1089, 1090
shuffle( ) algorithm defined by Collections, 593, 594, 596
shutdown( ), 941, 943, 956, 967–968
shutdownNow( ), 967–968
Sign extension, 75, 77–78, 78–79
signal( ), 949
SimpleBeanInfo class, 1113, 1118, 1119
SimpleDateFormat class, 636, 1013–1015, 1019
    formatting string symbols, table of, 1014
SimpleFileVisitor class, 757, 758
SimpleTimeZone class, 634–635
sin( ), 42, 515
SingleSelectionModel, 1056
SIZE, 479, 490
size( ), 541, 542, 556, 574, 605, 609, 610, 611, 622, 735, 737, 743
skip( ), 667, 691–692, 699, 709, 723
SKIP_SIBLINGS, 756
SKIP_SUBTREE, 756, 757
sleep( ), 250, 251–252, 257, 522, 947
slice( ), 729
Slider box, 861
Snippet
    code, 1155, 1157, 1158, 1160, 1163
    ID, 1158
Socket class, 730, 731, 763–766, 773

Socket(s)
datagram, 773–774
overview, 759–760
and servlets, 1122
TCP/IP client, 763–766
TCP/IP server, 763, 773
SocketAddress class, 774
SocketChannel class, 731
SocketException, 774
sort( ), 543, 544, 599
algorithm defined by Collections, 594
sorted( ), 972, 976
SortedMap interface, 571, 574–575
methods, table of, 575
SortedSet interface, 540, 546, 547
methods, table of, 546
Source code file, naming a, 27
Source file(s), 27–28
annotations and, 294
compiling and running a program directly from a, 28, 1165–1166
naming a, 27
SOURCE retention policy, 295–296
split( ), 468, 1002–1003
Spliterator, 533, 539, 543, 566–568
and arrays, 600
characteristics, 568
Spliterator interface, 533, 539, 540, 561, 566–568
methods declared by, table of, 567
and streamAPI streams, 990, 991–993
spliterator( ), 533, 541, 543, 600, 970
Spliterator.OfDouble interface, 568
Spliterator.OfInt interface, 568
Spliterator.OfLong interface, 568
Spliterator.OfPrimitive interface, 568
sqrt( ), 42, 49, 339–341, 516
Stack

definition of, 25, 134
frame, 530
trace, 229–230, 236, 530
walking, 531
ways to implement a, 213
Stack class, 538, 551, 602, 607–609
methods, table of, 608
StackTraceElement class, 242, 530
methods, table of, 530
StackWalker class, 531
StackWalker.StackFrame interface, 531
StampedLock interface, 950
StandardCopyOption values, partial list of, 747
StandardOpenOption class, 735, 748, 749
enumeration, table of values for the, 736
StandardOpenOption.CREATE, 736, 744, 749
StandardOpenOption.READ, 736, 746, 748
StandardOpenOption.TRUNCATE_EXISTING, 736, 749
StandardOpenOption.WRITE, 736, 744, 749
Standard Template Library (STL), 539
start( ), 250, 253, 254, 255, 495, 505, 522, 996–997, 999
startsWith( ), 458, 733
State enumeration, 274
Statements, 30
null, 96
Statements, control, 31, 44
iteration, 87, 95–106
jump, 87, 106–114
selection, 87–95
static, 29, 153–154, 157, 339
member restrictions, 386
used in a requires statement, 444
Static import, 16, 339–341
statusCode( ), 779
stop( ), 15, 272, 520, 522
store( ), 613, 614, 616–618

str( ), 295
Stream API, 18, 969–994
and collections, 618, 969, 970, 973, 975, 988
interfaces, 970–973
and lambda expressions, 969
and NIO, 735
Stream interface, 543, 600, 715, 733, 971–972, 993, 994
methods, table of some, 971–972
and primitive types, 972, 973
Stream, intermediate operations on a stream API, 972
to create a pipeline of actions, 972, 984–985
lazy behavior of, 972
stateless versus stateful, 972
Stream, stream API
and accumulator operations, 977, 978, 980–981, 989
collection from a, obtaining a, 986–989
and combiner operations, 980–981, 983, 989
definition of the term, 969–970
iterators and a, 990–993
mapping a, 982–986
obtaining a, 973
operations on a, terminal versus intermediate, 972
ordered versus unordered, 981
parallel processing of a, 969, 972, 973, 979–981, 988, 991, 993
parallel, using a, 979–981, 993
reduction operations, 977–979
stream( ), 542, 543, 600, 622, 625, 973, 975, 980, 985
Stream(s), byte, 316–317, 319, 323, 689–708
buffered, 698–702
classes in java.io, table of, 317
filtered, 698
Stream(s), character, 316, 317–318, 319, 323, 689, 708–718
classes in java.io, table of, 318
Stream(s), I/O
benefits, 726
buffered, 698–702, 714–717

classes, top-level, 688–689
closing, 687–688
concatenating input to, 702–703
definition of the term, 316, 709
filtered, 698, 726
flushing, 686
and NIO, 738, 748–750
predefined, 318–319
reactive, 921
variables, predefined, 318–319
versus stream API streams, 970
strictfp, 335–336
StrictMath class, 519
String class, 29, 65, 160–162, 449, 468, 532, 657, 996
constructors, 450–452
methods, table of some, 467–468
String(s)
arrays of, 65, 162
changing case of characters in, 465–466, 492–494
comparison, 161, 456–460
concatenating, 31, 160–161, 453–454, 463, 466–467, 471
constants, 65, 160
converting data into a, 454–455, 465
creating, 160, 450–452
extracting characters from, 455–456
formatted, creating a, 643–645
formatting a, 645, 651
immutability of, 160, 449, 462, 468
joining, 466–467
length of a, obtaining the, 161, 452
literals, 48, 452
modifiable, creating and working with, 468–475
modifying, 462–464
numbers to and from, converting, 490–492
as objects, 48, 65, 160, 449
parsing a formatted input, 619–621

reading, 320–322

searching, 460–462

StringBuffer class, 160, 449, 451, 462, 468–475, 532

methods, table of some, 474

StringBufferInputStream deprecated class, 680

StringBuilder class, 160, 449, 451, 462, 475, 532, 642, 643

and synchronization, 475

StringJoiner class, 672

StringReader class, 318

StringTokenizer class, 619–621

methods, table of, 620

stringWidth( ), 840, 841

StringWriter class, 318

strip( ), 464

stripLeading( ), 465

stripTrailing( ), 465

Stroustrup, Bjarne, 6

Stubs (RMI), 1010

Subclass, 24, 171–174, 189

generic, 372–376

subList( ), 543, 544

subMap( ), 574, 575, 577

submit( ), 944

subSequence( ), 468, 474, 532

subSet( ), 546, 547, 559

substring( ), 462–463, 474

Sun Microsystems, 6, 15, 16, 17

super, 153, 177

and bounded wildcard arguments, 364

and interface default methods, 223

and method references, 409

and methods or instance variables, 180–181, 186

super( ), 344

and superclass constructors, 177–180, 184, 256

Superclass, 24, 171–174, 189, 199

abstract, 191–194

generic, 372–374
Supplemental character, definition of, 494
Supplier<T> predefined functional interface, 417, 676, 989
@SuppressWarnings built-in annotation, 305, 307
suspend( ), 15, 272, 520, 522
Swing, 15, 315, 800–801, 843, 1025–1071
applet, 1030
application, example of a simple, 1030–1035
and the AWT, 815, 843, 1025–1026
component classes, list of, 1028–1029
components. *See* Components, Swing
containers. *See* Container(s), Swing
event handling, 1034–1038
history of, 1025–1026
menus. *See* Menu(s), Swing
and MVC architecture, 1027–1028
packages, list of, 1030
and painting, 1038–1042
threading issues, 1034, 1038
Swing: A Beginner's Guide (Schildt), 1025
SwingConstants interface, 1044
SwingUtilities class, 1034
switch statement, 90–95, 109, 111
and auto-unboxing, 292
nested, 94
using enumeration constants to control a, 91, 278–280
using a String to control a, 16, 91, 93–94
versus the if statement, 94–95
Synchronization, 14, 249–250, 261–265
and atomic operations, 950–951
and collections, 550, 595, 602
and deadlock, 270–271
and interprocess communication, 265–270
objects, using, 922–941
race condition and, 263
and StringBuilder class, 475

via synchronized block, 263–265, 595
via synchronized method, 250, 261–262
versus concurrency utilities, traditional, 919, 968
synchronized modifier, 261, 919, 948, 968
used with a method, 261–263
used with an object, 263–265
synchronizedList( ), 594, 595
synchronizedSet( ), 594, 595
Synchronizers, 919, 920–921
SynchronousQueue class, 947
System class, 30, 38, 318, 505–509
methods, table of, 506–507
System.console( ), 506, 719
System.err standard error stream, 318, 319, 505
System.exit( ), 506, 801, 804, 821, 823, 1082
System.getLogger( ), 506, 509
System.getProperties( ), 506, 613
System.getProperty( ), 506, 509
System.in standard input stream, 318, 319, 501, 505, 657, 718
System.in.read( ), 99
System.Logger interface, 509
System.LoggerFinder class, 509
System.nanoTime( ), 507, 508, 960
System.out standard output stream, 30, 38, 318, 319, 322, 323, 501, 505, 657, 703, 705, 718
and static import, 341

T

Tabbed panes, 1056–1058
Table, Swing, 1069–1071
TableColumnModel, 1069
TableModel, 1069
TableModelEvent class, 1069
tailMap( ), 574, 575, 577
tailSet( ), 546, 547

@Target built-in annotation, 305, 306, 308
TCP/IP, 14, 760
- client sockets, 763–766
- disadvantages of, 773
- server sockets, 763, 773
- *See also* Transmission Control Protocol (TCP)

TemporalAccessor, 647
TERMINATE, 756, 757
TERMINATED, 274
Ternary if-then-else operator (?:), 82, 84
test( ), 417, 674, 675, 676, 976–977
Text
- area, 795, 799, 867–868
- components as an event source, 795, 796
- editing controls, 844, 864–868
- formatting using java.text classes, 995, 1011–1016
- output using font metrics, managing, 839–842

Text fields, 795
- AWT, 864–866
- Swing, 1045–1047

TextArea class, AWT, 818, 867–868
TextComponent class, 818, 864, 867
TextEvent class, 787, 795, 867
TextField class, AWT, 818, 864–865
TextListener interface, 797, 799
textValueChanged( ), 799
thenComparing( ), 584, 589–590
thenComparingDouble( ), 584
thenComparingInt( ), 584
thenComparingLong( ), 584
this, 132–133, 153
- and lambda expressions, 403
- and type annotations, 307, 308, 311

this( ), 341–344
Thompson, Ken, 4
Thread class, 15, 250, 251, 256, 272, 274, 520–522, 638, 919, 952

constructors, 253, 256, 520
extending, 255–256
methods, table of, 520–522
Thread pool, 919, 941–942, 953, 954–956, 959, 962, 967–968
common, 953, 954, 955, 956, 959, 962, 968
Thread(s)
communication among, 265–270
creating, 252–257
creating and starting a, 275–276
daemon, 639, 956, 965
and deadlock, 270–271, 522, 1034
definition of the term, 247
executors to manage, using, 921, 941–943
group, 252, 523–527
local variables, 527
main, 250–252, 254, 256, 257, 258, 1034
messaging, 250
priorities, 249, 260–261, 520
race condition of, 263
resuming, 272–274, 524–527
return a value, that, 943–946
and spurious wakeup, 266
states of, possible, 249, 274–275
stopping, 272
suspending, 250, 252, 272–274, 524–527
and Swing, event dispatching, 1034, 1038
synchronization. *See* Synchronization
Thread.UncaughtExceptionHandler interface, 534
ThreadGroup class, 520, 523–527, 534
methods, table of, 523–524
ThreadLocal class, 527
ThreadPoolExecutor class, 921, 941
throw, 227, 236–237, 246
Throwable class, 228, 229, 232, 236, 237, 241, 244, 332, 387, 529, 530
methods defined by, table of, 242
obtaining an object of the, 236–237

throws, 227, 237–238, 240
Thumb, 861
time Calendar class instance variable, 628
Time and date
  formatting, 647–648, 995, 1011–1015, 1017–1020
  java.util classes that deal with, 626–636
  strings, parsing, 1020–1021
Time and Date API, 628, 995, 1011, 1015–1021
  packages, list of, 1015
timedJoin( ), 947
timedWait( ), 947
TIMED_WAITING, 274
Timer class, 638–640
  methods, table of, 640
TimerTask class, 638–640
  methods, table of, 639
Timestamp, event, 787
TimeUnit enumeration, 921, 928, 944, 946–947
TimeZone class, 633–634
  methods defined by, table of some, 633–634
to
  keyword, 37, 420
  clause, 429, 443
to( ), 505
toAbsolutePath( ), 733, 750
toArray( ), 542, 553–554, 972
toBinaryString( ), 487, 489, 490, 491–492
toCharArray( ), 456
toChronoUnit( ), 947
toDays( ), 947
toDegrees( ), 519
ToDoubleFunction functional interface, 676, 985
toFile( ), 732
Toggle button, Swing, 1049–1051
toHexString( ), 481, 482, 487, 490, 491–492
toHours( ), 947

Tokens, 619, 657, 665
tokens( ), 667
toLanguageTag( ), 636
toList( ), 986, 988
toLocalDate( ), 1017
toLocalTime( ), 1017
toLowerCase( ), 465–466, 493, 495
Tomcat, 1122, 1123–1126
toMicros( ), 947
toMillis( ), 947
toMinutes( ), 947
toNanos( ), 947
toOctalString( ), 487, 490, 491–492
Toolbars, 1073, 1074, 1091–1093
    undocked, 1091
Tooltips, 1085, 1091, 1092, 1093
TooManyListenersException, 1112
toPath( ), 683, 732, 750
toRadians( ), 519
toSeconds( ), 947
toSet( ), 986, 988
toString( ), 197, 232, 241, 242, 288, 295, 301, 323, 454, 465, 481, 482, 484, 485, 487, 490, 491, 497, 510, 514, 522, 524, 528, 529, 530, 531, 532, 553, 600, 605, 611, 622, 623, 625, 627, 641, 643, 645, 704, 718, 732, 733, 763, 786, 834, 840, 967, 1004, 1066
    advantages to overriding, 197, 454–455
totalCPUDuration( ), 495
totalMemory( ), 498, 499, 500
toUpperCase( ), 465–466, 493
toZonedDateTime( ), 632
transient modifier, 333, 1113
transitive keyword, 37, 420, 430–434
translatePoint( ), 793
Transmission Control Protocol (TCP)
    definition of, 759–760
    and stream-based I/O, 760

See also TCP/IP
TreeExpansionEvent class, 1066
TreeExpansionListener interface, 1066
TreeMap class, 578, 580–581, 582, 618
example using a comparator, 587–590
TreeModel, 1066
TreeModelEvent class, 1066
TreeModelListener interface, 1066
TreeNode interface, 1066
TreePath class, 1066
Trees, Swing, 1065–1068
TreeSelectionEvent class, 1066
TreeSelectionListener interface, 1066, 1067
TreeSelectionModel, 1066
TreeSet class, 551, 557, 558–559, 582, 618
example using a comparator for sorting an instance of the, 585–587
trim( ), 464
trimToSize( ), 474, 553, 605
true, 38, 44, 45, 47, 81, 82, 83
TRUE, 495
True and false in Java, 47, 81
Truncation, 53–54
try block(s), 227, 228, 230–236, 237–239, 246
nested, 234–236
try-with-resources statement, 17, 228, 245, 324, 329–332, 534, 656, 663, 686, 687–688, 694, 730, 732, 739, 753, 764, 765–766, 774, 970
advantages to using, 329, 331, 688
and local variable type inference, 331
tryAdvance( ), 566–568, 991
tryLock( ), 921, 948–949
trySplit( ), 992–993
tryUnfork( ), 967
Two's complement, 72–73
TYPE, 479, 490, 492, 495
Type argument(s), 348–349, 350, 354
and bounded types, 355–357

and generic class hierarchies, 372
and generic interface implementation, 370
and type inference, 366, 380–381

Type conversion
automatic, 39, 52, 138–139
narrowing, 52
widening, 52

Type enumeration, 505

Type inference, local variable, 19, 62–65
advantages of using, 63
in a for loop, 107–108
and generics, 381
and inheritance, 195–196
and lambda expressions, 394
with reference types, 168–170
and streamlining code, 19, 63, 168–169, 381
and the try-with-resources statement, 331

Type interface, 536

Type parameter(s)
and bounded types, 354–357, 369–370
cannot create an instance of a, 385
and class hierarchies, 373–376
and erasure, 349, 382
and primitive types, 350
and static members, 386
and type safety, 350
used with a class, 348, 353–355 357
used with a method, 348, 364–367
and wildcard arguments, 357–366

Type safety
and collections, 595
and generic methods, 367
and generics, 345, 346, 349, 350–352
and raw types, 370–372
and wildcard arguments, 357–360, 361

type( ), 505

Type(s), 31
- annotations, 18, 307–312
- bounded, 355–357
- casting, 52–54
- checking, 13, 17, 39, 349, 350, 371, 387
- class as a data, 117, 119, 121, 122, 123, 134
- inference, 366, 380–381, 391, 394, 397, 403
- inference, local variable. *See* Type inference, local variable
- intersection, 357
- non-reifiable, 307
- parameterized, 346, 348
- promotion, 41, 54–55, 75–77
- raw, 370–372
- SAM (Single Abstract Method), 390
- simple, 39

Types, primitive (simple), 39–40, 122, 144, 286, 350, 478
- autoboxing/unboxing and, 289–292, 294, 554
- and collections, 554
- iterators for, 539
- to a string representation, converting, 453, 454, 465
- to or from a sequence of bytes, converting, 706–708
- wrappers for, 286–289, 294, 350, 478–495

Typesafe view of a collection, obtaining a dynamically, 595

U

UDP protocol, 760, 773

UI delegate, 1028

ulp( ), 516, 518

UnaryOperator<T> functional interface, 417, 676

UnavailableException, 1127, 1130

Unboxing, 289

UncaughtExceptionHandler interface, 534
- method uncaughtException( ), 534

Unchecked warnings and raw types, 372

UnicastRemoteObject, 1008

Unicode, 43, 44, 47, 316, 317, 451, 452, 456, 494, 708
    code points, table of some Character methods providing support for, 496
    support for 34-bit, 494–495
Unicode Technical Standard (UTS) 37, 636
Uniform Resource Identifier (URI), 772
Uniform Resource Locator (URL). *See* URL (Uniform Resource Locator)
UNIX, 5, 759
UnknownHostException, 762
unlock( ), 921, 948, 949
unmodifiable... collections methods, 594–595
unordered( ), 970, 981
Unreachable code, 115, 233
unread( ), 701, 716–717
UnsupportedOperationException, 222, 240, 540, 543, 561, 571, 595, 737
update( ), 502–503, 822, 823, 824
URI (Uniform Resource Identifier), 772
URI class, 772
URL (Uniform Resource Locator), 766–767, 772, 1121, 1122
    query string, 1138
    specification format, 766–767
URL class, 766–768, 769, 770
URLConnection class, 768–771, 772
    methods, list of some, 768–769
useDelimiter( ), 665–666
User Datagram Protocol (UDP), 760, 773
useRadix( ), 667
uses
    keyword, 37, 420
    statement, 435, 441, 442
UTS 37, 636
UUID class, 672

V

val( ), 295
value (annotation member name), 304, 305

valueChanged( ), 1061, 1062, 1066, 1067
valueOf( ), 280–281, 287, 288, 454, 465, 479, 481, 482, 483, 484, 485, 488, 490, 492, 495, 497, 531, 622
values( ), 280–281, 571, 574
van Hoff, Arthur, 6
var, context-sensitive reserved type name, 19, 38, 63–65, 107, 169, 195, 331, 381, 394
    restrictions on using, 64–65, 348
Varargs, 16, 163–168
    and ambiguity, 167–168
    methods, overloading, 166–167
    and Java's printf( ), 163
    parameter, 165, 545
Variable(s), 48–52
    capture, 403–404
    declaration, 31, 33, 48–49, 50–52, 62–65
    definition of the term, 30, 48
    dynamic initialization of, 49
    effectively final, 403–404
    enumeration, 278
    final, 155, 277
    instance. *See* Instance variables
    interface, 209, 216–218
    interface reference, 210–212, 216
    and JShell, 1156–1157
    loop control, 99, 100, 101, 102, 103
    member, 23
    object reference. *See* Object reference variables
    scope and lifetime of, 49–52
    temporary, 1162
    type inference, local. *See* Type inference, local variable
    volatile, 333
Vector class, 538, 551, 570, 602, 603–607
    legacy methods, table of, 604–605
Version information, obtaining, 502–503
version( ), 499, 503

VetoableChangeListener interface, 1113, 1114
Viewport, scroll pane, 1058, 1059
visitFile( ), 756, 758
void, 29, 123
Void class, 495
volatile modifier, 333

W

wait( ), 197, 265–266, 268–269, 272–274, 510, 919, 947, 949, 968
waitFor( ), 498, 501
WAITING, 274
WALL_TIME, 635
walk( ), 733
walkFileTree( ), 756–758
Warth, Chris, 6
Watchable interface, 732
WeakHashMap class, 578
Web. *See* World Wide Web
Web browser, 766, 1121, 1122, 1125, 1126
  executing applet in, 8, 11
Web server and servlets, 1122
Web Start. *See* Java Web Start
WebSocket
  class, 782
  protocol, 777
weightx constraint field, 879, 880
weighty constraint field, 879, 880
while loop, 95–96
  and break, 110
  and continue, 113
Whitespace, 36, 88
  characters, 619
  from a string, removing, 464–465
with keyword, 37, 420, 435
whois, 760, 764–766

Wildcard arguments, 357–364, 378
  bounded, 360–364
  used in creating an array, 387
Window, AWT-based
  class hierarchy, 818–819
  close event, 801, 804, 809, 821
  as an event source, 795, 796
  frame. *See* Frame window
  and graphics, 824
  paint mode, setting the, 832–833
  top-level, 819
Window class, 796, 818, 819, 821, 888
Window, Swing JFrame, 1032
windowActivated( ), 800
WindowAdapter class, 808, 809
windowClosed( ), 800
windowClosing( ), 800, 804, 808, 821, 823
WindowConstants interface, 1033
windowDeactivated( ), 800
windowDeiconified( ), 800
WindowEvent class, 787, 789, 795–796
WindowFocusListener interface, 797, 799, 809
windowGainedFocus( ), 799
windowIconified( ), 800
WindowListener interface, 797, 800, 804, 808, 809, 821, 823
windowLostFocus( ), 799
windowOpened( ), 800
WindowStateListener interface, 809
Work stealing, 955, 967
World Wide Web (WWW), 6, 7, 13, 20, 767
wrap( ), 729
Wrappers, primitive type, 286–289, 294, 350, 478–495
WRITE, 505
write( ), 317, 318, 322, 328–329, 691, 710, 721, 722, 731, 744, 745, 749
writeBoolean( ), 706, 722
writeDouble( ), 706, 722

writeExternal( ), 721
writeInt( ), 706, 722
writeObject( ), 721, 722, 725
Writer class, 317–318, 688, 689, 708, 709–710, 711, 721
    methods defined by, table of, 710
writeString( ), 735
writeTo( ), 698

X

XOR (exclusive OR) operator (^)
    bitwise, 72, 73, 74–75
    Boolean logical, 81,82
XOR paint mode, 832–833

Y

Yellin, Frank, 6

Z

Zero crossing, 73
ZIP file format, 677

ORACLE
PROFIT
CONFIDENT AND CONFIDENTIAL
Java
REACTIVE PROGRAMMING