

PopupControl class, 1173, 1185
 PopupMenu class, 799, 876
 PopupMenuEvent, 1071
 Port, 727–728, 735
 Portability problem, 6–7, 8, 9, 10, 12, 16
 and data types, 36
 and native methods, 328
 and thread context switching, 235
 Pos enumeration, 1119
 POSITIVE_INFINITY, 443
 PosixFileAttributes class, 699, 714
 PosixFileAttributeView interface, 699
 postVisitDirectory(), 718
 pow(), 331–333, 478
 Predicate<T> predefined functional interface, 409, 503, 638, 972
 preferredLayoutSize(), 856
 previous(), 523, 863
 preVisitDirectory(), 718
 PrimitiveIterator interface, 499, 986
 PrimitiveIterator.OfDouble interface, 499, 986
 PrimitiveIterator.OfInt interface, 499, 986
 PrimitiveIterator.OfLong interface, 499, 986
 print(), 27, 34, 308, 309, 418, 666, 680, 1220
 printf()
 function, C/C++, 605, 666
 method, Java, 155, 620, 666–667, 679, 680, 681
 println(), 26, 27, 34, 186, 308, 309, 418–419, 608, 665, 666, 679, 680, 1220
 and applets, 748, 767
 and Boolean output, 41
 and String objects, 59
 printStackTrace(), 228
 PrintStream class, 303, 305, 308, 620, 665–667
 PrintWriter class, 304, 308–309, 620, 679–680, 1215
 PriorityBlockingQueue class, 943
 PriorityQueue class, 511, 519–520
 private access modifier, 25, 142–144, 190–191
 and inheritance, 163–164
 Process class, 460–461, 464, 465
 methods, table of, 461
 Process, definition of, 233, 460
 Process-based versus thread-based multitasking, 233
 ProcessBuilder class, 460, 465–467
 methods, table of, 465–466
 ProcessBuilder.Redirect class, 465
 ProcessBuilder.Redirect.Type enumeration, 465
 Program, creating a windowed, 809–810
 Programming
 multithreaded. *See* Multithreaded programming
 object-oriented. *See* Object-oriented programming
 process-oriented, 17, 18, 22
 structured, 4, 5
 Programming, parallel, 15, 235, 916, 917, 947–948, 975
 and specifying the level of parallelism, 950, 955–958, 963
 Project Coin, 14
 Properties class, 498–499, 561, 572–577
 methods, table of, 573–574
 Properties, environment, 470
 Property, Java Bean, 1208
 bound and constrained, 1203, 1206
 design patterns for, 1200–1201, 1203

PropertyChangeEvent, 1203
 PropertyChangeListener interface, 1203, 1204
 PropertyDescriptor class, 1202, 1205, 1206, 1208
 PropertyPermission class, 635
 PropertyResourceBundle class, 633
 PropertyVetoException, 1203
 protected access modifier, 126, 142, 190–191
 Protocols, overview of networking, 727–728
 Pseudorandom numbers, 596
 public access modifier, 25, 142–144, 190–191
 Push buttons, AWT, 751, 836–840
 action command string of, 836, 838, 839
 Push buttons, JavaFX, 1115–1119
 adding an image to, 1130–1133
 Push buttons, Swing, 1030–1033, 1045–1047
 action command string of, 1045
 push(), 510, 567
 Pushback, 662
 PushbackInputStream, 303, 659, 662–663
 PushbackReader class, 304, 678–679
 put(), 531, 533, 537, 539, 541, 568, 569, 570
 and buffers, 691, 692, 706–707, 724
 putAll(), 533, 541
 PutField inner class, 684
 putValue(), 1090–1091, 1092

Q

Query string, 1228
 Queue interface, 501, 508, 515, 519, 520
 methods, table of, 508
 quietlyInvoke(), 962
 quietlyJoin(), 962

R

Race condition, 248–249
 Radio buttons, 842
 JavaFX, 1135–1138, 1139–1142
 Swing, 1051–1053
 and Swing menus, 1082–1083
 RadioButton class, 1135, 1136, 1142
 RadioMenuItem class, 1172, 1183
 Radix, 447
 radix(), 630
 Random class, 205, 596–598
 methods, table of, 596
 random(), 480
 RandomAccess interface, 501, 530
 RandomAccessFile class, 669–670, 692, 724
 range(), 521, 522
 Raw types, 362–364
 READ, 465
 read(), 93, 303, 304, 305–306, 310–311, 495, 651, 660, 662, 671, 678, 685, 686, 693, 701, 702, 703, 710, 721
 Readable interface, 495, 620, 626, 670
 ReadableByteChannel interface, 620
 readAttributes(), 697, 699, 712–714
 readBoolean(), 668, 686
 readDouble(), 668, 686
 Reader class, 303–304, 305, 650, 670, 672, 674, 688
 methods defined by, table of, 671

- `readExternal()`, 683
- `readInt()`, 668, 686
- `readLine()`, 306–308, 454, 680, 681, 751, 1220
- `readObject()`, 685, 686
- `readPassword()`, 680, 681
- ReadWriteLock interface, 946
- Real numbers, 38
- `rebind()`, 1006
- `receive()`, 743
- Recursion, 139–141
 - and the Fork/Join Framework divide-and-conquer strategy, 951–952
- RecursiveAction class, 917, 948, 949–950, 951, 952, 954, 958
- RecursiveTask class, 917, 948, 950, 951
 - example program using, 958–960
- Redirect class, 465
- `reduce()`, 968, 973–977, 978–979
- Reduction operations, 973–975
 - mutable, 985
- ReentrantLock, 944
- ReentrantReadWriteLock, 946
- Reflection, 281, 496, 991, 1001–1005
 - and annotations, 281–286
- ReflectiveOperationException, 227
- Region class, 1107, 1115
- `regionMatches()`, 421–422
- `register()`, 930
- Regular expressions, 432, 621, 628, 991, 993–1001
 - syntax, 995
 - wildcards and quantifiers, 993, 995, 997–999
- `reinitialize()`, 961
- Relational operators, 28, 40, 41, 74–75
- Relative index, 618–619
- `release()`, 918–921
- `remainderUnsigned()`, 451, 453
- Remote interface, 1006
- Remote method invocation (RMI), 12, 682, 991, 1005–1009
- RemoteException, 1006
- `remove()`, 502, 503, 505, 508, 516, 521, 522, 523, 533, 568, 569, 570, 834, 1029, 1071, 1072, 1114, 1166, 1173, 1174
- `removeActionListener()`, 1032
- `removeAll()`, 502, 503, 834
- `removeAttribute()`, 1225, 1232
- `removeEldestEntry()`, 541
- `removeElement()`, 563, 564
- `removeElementAt()`, 563, 564
- `removeFirst()`, 510, 515
- `removeIf()`, 502, 503
- `removeKeyListener()`, 771
- `removeLast()`, 510, 515
- `removeTListener()`, 1202
- `removeTypeListener()`, 771
- `renameTo()`, 644
- `repaint()`, 756–759, 803, 1037, 1119, 1208
- @Repeatable annotation, 290, 297, 298
- `replace()`, 427, 437, 533
- `replaceAll()`, 431, 504, 505, 533, 995, 999–1000
- `replaceFirst()`, 431
- `replaceRange()`, 854
- ReplicateScaleFilter class, 899
- `reset()`, 630, 651, 652, 657, 660–661, 663, 671, 676, 691
- `resolve()`, 694, 695
- Resource bundles, 630–634
- ResourceBundle class, 630–633
 - methods, table of, 631–632
- ResourceBundle.Control class, 631
- `resume()`, 13, 257, 482, 488
- `retainAll()`, 503
- @Retention built-in annotation, 281, 290
- RetentionPolicy enumeration, 281, 496
- return statement, 108, 116
 - in a lambda expression, 388–389
- `reverse()`, 436, 451, 453, 553
- `reverseBytes()`, 449, 451, 453
- `reversed()`, 542, 545–546
- `reverseOrder()`
 - collection algorithm, 553, 555–556
 - Comparator method, 543
- `rewind()`, 691, 703, 706, 707, 721, 724
- RGB (red-green-blue) color model, 816–817
 - default, 895
- RGBImageFilter class, 899, 902
 - example program demonstrating the, 902–912
- RGBtoHSB(), 816
- Richards, Martin, 4
- `rint()`, 479
- Ritchie, Dennis, 4
- rmi protocol, 1007
- RMI (Remote Method Invocation), 12, 682, 991, 1005–1009
- rmic compiler, 1008
- rmiregistry (RMI registry), 1008, 1009
- Rotate class, 1166–1167
 - program demonstrating, 1167–1170
- `rotateLeft()`, 451, 453
- `rotateRight()`, 451, 453
- `round()`, 479
- Run-time
 - system, Java, 9. *See also* Java Virtual Machine (JVM)
 - type information, 13, 322, 368–370, 376
- `run()`, 236, 239, 240, 382, 481, 483, 602, 603, 962, 1030
 - overriding, 241, 242, 602
 - using a flag variable with, 257–259, 759
- RUNNABLE, 260
- Runnable interface, 236, 238, 382, 481, 602, 758–759, 915, 961, 962, 1030
 - implementing the, 239–240, 242
- Runtime class, 460, 461–464, 958
 - executing other programs and, 464
 - memory management and, 462–464
 - methods, table of some, 461–462
- RUNTIME retention policy, 281, 282, 285
- RuntimeException class, 214–215, 223, 226, 230
- RuntimePermission class, 490

S

- @SafeVarargs built-in annotation, 290, 292
- SAM (Single Abstract Method) type, 382
- `save()`, 573
- `scalb()`, 478

- Scale class, 1166, 1167
 - program demonstrating, 1167–1170
- Scanner, 579
- Scanner class, 620–630
 - closing an instance of the, 626
 - constructors, 620, 621
 - delimiters, 621, 628–629
 - demonstration programs, 624–628
 - hasNextX() methods, table of, 622
 - how to use, 620–621, 623–624
 - methods, miscellaneous, 629–630
 - nextX() methods, table of, 623
- Scene class, 1106, 1107, 1110, 1111
- schedule(), 602, 603
- ScheduledExecutorService interface, 937
- ScheduledThreadPoolExecutor class, 917, 939
- Scientific notation, 42, 607, 609–610
- Scopes in Java, 45–48
- Scroll bars, 773, 782, 849–851, 1056, 1149–1150, 1157–1159, 1160
- Scroll pane, 1056–1057, 1157–1159
- Scrollbar class, 799, 849
- ScrollPane class, 1157–1159
- search(), 567
- Security manager, 310, 467, 490, 660, 1212
- Security problem, 8, 9–10, 16
 - and native methods, 328
 - and servlets, 1212
- SecurityException, 226, 310, 461, 467, 649, 666, 714, 717
- SecurityManager class, 490
- seek(), 670
- SeekableByteChannel interface, 693, 701, 704, 705
- select(), 844, 847, 852, 854
- selectedItemProperty(), 1147, 1149, 1160
- selectedToggleProperty(), 1139
- Selection statements, 81–89
- SelectionMode, 1150
- SelectionModel class, 1147
- Selectors, 693
- Semaphore, 915, 916, 918–923
 - and setting initial synchronization state, 923
- Semaphore class, 916, 917, 918
- send(), 743
- Separable Model architecture, 1023
- Separator class, 1142
- SeparatorMenuItem class, 1172, 1174
- Separators, 33
- SequenceInputStream class, 303, 663–665
- sequential(), 966, 977
- Serializable interface, 682–683, 687, 962, 1203
- Serialization, 682–688
 - example program, 686–688
 - and Java Beans, 1203
 - and static variables, 683
 - and transient variables, 683, 687
- Server, 727
- ServerSocket class, 692, 731, 741–742
- service(), 1212, 1215, 1217, 1221, 1227
- ServiceLoader class, 635
- Servlet interface, 1216, 1217, 1220
 - methods, table of, 1217
- Servlet(s), 10, 16, 1211–1233
 - advantages of, 1212
 - API, 1216
 - development options, 1212–1214
 - example program for a simple, 1214–1216
 - life cycle of, 1212
 - parameters, reading, 1220–1222
 - and portability, 10
 - and security, 1212
 - and session tracking, 1232–1233
 - using Tomcat to develop, 1212, 1213–1216
- ServletConfig interface, 1216, 1218, 1220
- ServletContext interface, 1216, 1218
 - methods, table of various, 1218
- ServletException class, 1217, 1220
- ServletInputStream class, 1217, 1220
- ServletOutputStream class, 1217, 1220
- ServletRequest interface, 1215, 1216, 1218, 1220
 - methods, table of various, 1219
- ServletResponse interface, 1215, 1216, 1218
 - methods, table of various, 1219
- Session tracking, HTTP, 1232–1233
- Set interface, 501, 504, 506, 516, 521, 531, 536
 - from a stream API stream, obtaining a, 982–984
- Set-view, obtaining, 538–539, 571–572
- set(), 504, 505, 516, 523, 582, 589, 946
- setAccelerator(), 1079, 1181
- setActionCommand(), 839, 872, 1045, 1052
- setAlignment(), 835, 1119
- setAll(), 560
- setAllowIndeterminate(), 1145
- setAngle(), 1166–1167
- setAttribute(), 1218, 1225, 1232
- setBackground(), 754–755, 816
- setBlockIncrement(), 849
- setBorder(), 1040
- setBounds(), 801, 856
- setChanged(), 598–599
- setCharAt(), 434
- setColor(), 817
- setConstraints(), 866
- setContent(), 1157
- setContentDisplay(), 1129–1130, 1132–1133, 1190
- setContentType(), 1215, 1219
- setContextMenu(), 1186
- setDefault(), 593, 595
- setDefaultCloseOperation(), 1028
- setDisable(), 1170, 1175
- setDisabledIcon(), 1045, 1081
- setEchoChar(), 852
- setEditable(), 852, 854, 1153–1154
- setEffect(), 1165
- setEnabled(), 871, 1073, 1091
- setFill(), 1120–1121
- setFont(), 822
- setForeground(), 754–755, 816
- setForkJoinTaskTag(), 962
- setGraphic(), 1130, 1174, 1183
- setHorizontalTextPosition(), 1081
- setHvalue(), 1157–1158
- setIcon(), 1042, 1080–1081
- SetIntField(), 327

- setJMenuBar(), 1072, 1074
- setLabel(), 836, 840, 871
- setLastModified(), 645
- setLayout(), 856, 1029
- setLength(), 433–434, 670, 744
- setLevel(), 1165
- setLocation(), 801
- setMaxAge(), 1226, 1232
- setMnemonic(), 1078
- setMnemonicParsing(), 1181
- setMultipleMode(), 882
- setN() setter method design pattern, 1200, 1201
- setName(), 237, 238, 484
- setOnAction(), 1116, 1118, 1119, 1175
- setOnContextMenuRequested(), 1187–1188
- setOrientation(), 1189
- setPaintMode(), 818
- setPannable(), 1157
- setPivotX(), 1166–1167
- setPivotY(), 1166–1167
- setPrefColumnCount(), 1154
- setPreferredSize(), 801, 850
- setPrefHeight(), 1147
- setPrefSize(), 1147
- setPrefViewportHeight(), 1157
- setPrefViewportWidth(), 1157
- setPrefWidth(), 1142, 1147
- setPressedIcon(), 1045
- setPriority(), 246, 484
- setPromptText(), 1154
- setReadOnly(), 645
- setRolloverIcon(), 1045
- setRotate(), 1166
- setScaleX(), 1166
- setScaleY(), 1166
- setScene(), 1111
- setSelected(), 1136, 1139, 1183
- setSelectedCheckbox(), 842
- setSelectedIcon(), 1045
- setSelectionMode(), 1058, 1150
- setSize(), 564, 801, 802, 803, 1028
- setStackTrace(), 228
- setState(), 840, 871
- setStroke(), 1120–1121
- setText(), 835, 852, 854, 1042, 1045, 1118, 1154, 1174
- setTitle(), 802, 1110
- setToggleGroup(), 1136
- setTooltip(), 1170, 1189
- setToolTipText(), 1081
- setTranslateX(), 1166
- setTranslateY(), 1166
- setUnitIncrement(), 849
- setValue(), 537, 849, 1090, 1091, 1226, 1152
- setValues(), 849
- setVvalue(), 1157–1158
- setVisible(), 802, 803, 1029
- setX(), 1167
- setXORMode(), 818–819
- setY, 1167
- Shear class, 1166
- Sheridan, Mike, 6
- Shift operators, bitwise, 66, 69–73
- Short class, 272, 273, 447, 454
 - methods defined by, table of, 449
- short data type, 35, 36, 37, 41
 - and automatic type conversion, 48
 - and automatic type promotion, 50, 69
- ShortBuffer class, 691
- shortValue(), 273, 442, 444, 446, 448, 449, 451, 453
- show(), 863, 1084, 1085, 1086, 1111, 1188
- showDocument(), 764, 765–766
- showStatus(), 750, 759, 766, 792, 793–794
- shuffle(), 553, 555–556
- shutdown(), 937, 939, 951, 963
- shutdownNow(), 963
- Sign extension, 72–73
- signal(), 944
- signum(), 451, 453, 480
- SimpleBeanInfo class, 1203, 1208, 1209
- SimpleDateFormat class, 596, 1011–1013, 1017
 - formatting string symbols, table of, 1012
- SimpleFileVisitor class, 718, 719
- SimpleTimeZone class, 594
- sin(), 38, 477
- SingleSelectionModel, 1053
- sinh(), 477
- SIZE, 443, 447
- size(), 503, 516, 533, 564, 568, 569, 570, 582, 697, 698, 705, 1173, 1174
- skip(), 630, 651, 652–654, 660, 671, 685
- SKIP_SIBLINGS, 718
- SKIP_SUBTREE, 718
- sleep(), 236, 237–238, 243, 484, 943
- slice(), 691
- Slider box, 849
- Socket class, 692, 731–734, 741, 742
- Socket(s)
 - datagram, 742–743, 744–745
 - overview, 727
 - TCP/IP client, 731–734
 - TCP/IP server, 731, 741–742
- SocketAddress class, 742–743
- SocketChannel class, 692, 693
- SocketException, 743
- sort(), 504, 505, 554, 558–559
- sorted(), 968, 972
- SortedMap interface, 531, 534
 - methods, table of, 534
- SortedSet interface, 501, 506
 - methods, table of, 506
- Source code file, naming a, 23
- SOURCE retention policy, 281
- split(), 431, 432, 1000–1001
- Spliterator, 499, 504, 526–529
 - and arrays, 559
 - characteristics, 528
- Spliterator interface, 495, 499, 501, 523, 526–529
 - methods declared by, table of, 527
 - and streamAPI streams, 986, 987–989
- spliterator(), 494–495, 503, 504, 559, 966
- Spliterator.OfDouble interface, 529
- Spliterator.OfInt interface, 529
- Spliterator.OfLong interface, 529
- Spliterator.OfPrimitive interface, 529
- sqrt(), 38, 45, 331–333, 478
- Stack
 - definition of, 21, 127
 - ways to implement a, 201

- Stack class, 498–499, 511, 561, 566–568
 - methods, table of, 567
- Stack frame, 491
- Stack trace, 215–216, 222, 491
- StackTraceElement class, 228, 491
 - methods, table of, 491
- Stage class, 1106, 1107, 1110, 1111
- StampedLock interface, 946
- StandardCopyOption values, partial list of, 709
- StandardOpenOption class, 695, 710, 711
 - enumeration, table of values for the, 697
- StandardOpenOption.CREATE, 697, 705, 711
- StandardOpenOption.READ, 697, 708, 710
- StandardOpenOption.TRUNCATE_EXISTING, 697, 711
- StandardOpenOption.WRITE, 697, 705, 711
- Standard Template Library (STL), 499
- start(), 236, 239, 240, 241, 460, 466, 467, 484, 750, 751, 753, 755, 759, 803, 994, 997, 1033, 1035, 1107, 1108, 1110, 1111
- startsWith(), 422, 695
- State enumeration, 259
- Statements, 26
 - null, 90
- Statements, control, 28, 40
 - iteration, 81, 89–102
 - jump, 81, 102–108
 - selection, 81–89
- static, 25, 145–146, 149, 325, 331, 332–333
 - member restrictions, 377
- Static import, 14, 331–334
- stop(), 13, 257, 482, 750, 751, 753, 756, 759, 767, 803, 1033, 1107, 1108, 1110, 1111, 1180
- store(), 573, 576–577
- Stream API, 16, 965–990
 - and collections, 577, 965
 - interfaces, 966–968
 - and lambda expressions, 965
 - and NIO, 695
- Stream interface, 504, 559, 676, 695, 967–968, 989, 990
 - methods, table of some, 967–968
- Stream, intermediate operations on a stream API, 968
 - to create a pipeline of actions, 968, 980
 - lazy behavior of, 968
 - stateless versus stateful, 968
- Stream, stream API
 - collection from a, obtaining a, 982–985
 - definition of a, 965–966
 - iterators and a, 986–989
 - mapping a, 978–982
 - obtaining a, 969
 - operations on a, terminal versus intermediate, 968
 - ordered versus unordered, 977
 - parallel processing of a, 965, 968, 969, 975–977, 984, 986, 987, 989
 - parallel, using a, 975–977, 989
 - reduction operations, 973–975
- stream(), 503, 504, 559, 582, 969, 971, 976, 981
- Stream(s), byte, 302–303, 305, 309, 650, 651–670
 - classes in java.io, table of, 303
- Stream(s), character, 302, 303–304, 305, 309, 650, 670–680
 - classes in java.io, table of, 304
- Stream(s), I/O
 - benefits, 688
 - buffered, 659–663, 676–678
 - classes, top-level, 650
 - closing, 649–650
 - concatenating input to, 663–665
 - definition of, 302, 641
 - filtered, 659, 688
 - flushing, 648
 - and NIO, 709–711
 - predefined, 304–305
- strictfp, 324
- StrictMath class, 481
- String class, 25, 58–59, 152–154, 413, 493, 620, 761, 994
 - constructors, 414–416
 - methods, table of some, 431–432
- String(s)
 - arrays of, 58, 154
 - changing case of characters in, 429–430, 456, 457
 - comparison, 153, 420–424
 - concatenating, 152–153, 417–418, 427, 435
 - constants, 58, 152
 - converting data into a, 418–419, 428–429
 - creating, 152, 414–416
 - extracting characters from, 419–420
 - formatted, creating a, 607–608
 - formatting a, 609, 614
 - immutability of, 152, 413, 426, 432
 - joining, 430–431
 - length, obtaining, 153, 416
 - literals, 43–44, 416
 - modifiable, creating and working with, 432–439
 - modifying, 426–428
 - numbers to and from, converting, 454–455
 - as objects, 44, 58–59, 152, 413
 - parsing a formatted input, 579
 - reading, 306–308
 - searching, 424–426
- StringBuffer class, 152, 413, 415, 426, 432–439, 493
 - methods, table of some, 439
- StringBufferInputStream deprecated class, 642
- StringBuilder class, 152, 413, 415–416, 426, 439, 493, 606
 - and synchronization, 439
- StringIndexOutOfBoundsException exception, 226
- StringJoiner class, 635
- StringReader class, 304
- StringTokenizer class, 579–580, 832
 - methods, table of, 580
- stringWidth(), 826
- StringWriter class, 304
- strokeLine(), 1120
- strokeOval(), 1120
- strokeRect(), 1120
- strokeText(), 1120
- Stroustrup, Bjarne, 6
- Stubs (RMI), 1007–1008
- Subclass, 20, 161–164, 179
- subList(), 504, 505
- subMap(), 534, 536
- submit(), 940
- subSequence(), 432, 438, 493
- subSet(), 506, 507, 519

- substring(), 426–427, 437
 - subtractExact(), 480
 - sum(), 444, 446, 451, 453
 - Sun Microsystems, 6, 14
 - super, 145, 167
 - and bounded wildcard arguments, 356
 - and interface default methods, 211
 - and method references, 401
 - and methods or instance variables, 170–171, 176
 - super(), 336
 - and superclass constructors, 167–170, 174
 - Superclass, 20, 161–164, 179, 187
 - abstract, 181–184
 - Supplemental character, definition of, 458
 - Supplier<T> predefined functional interface, 409, 638, 985
 - @SuppressWarnings built-in annotation, 290, 292
 - suspend(), 13, 257, 482, 488
 - Swing, 13, 16, 301, 319, 747, 797, 833, 1021–1068, 1105
 - applet, example of a simple, 1033–1035
 - application, example of a simple, 1026–1030
 - and the AWT, 797, 1021–1022
 - component classes, list of, 1024–1025
 - components. *See* Components, Swing
 - event handling, 1030–1033
 - history of, 1021–1022
 - and JavaFX, 1106, 1119
 - menus. *See* Menu(s), Swing
 - and MVC architecture, 1023
 - packages, list of, 1026
 - and painting, 1033, 1036–1040
 - threading issues, 1029–1030, 1033
 - Swing: A Beginner's Guide* (Schildt), 1021
 - SwingConstants interface, 1042
 - SwingUtilities class, 1030
 - switch statement, 84–89
 - and auto-unboxing, 277
 - nested, 88
 - using enumeration constants to control a, 84, 264–265
 - using a String to control a, 15, 84–85, 87–88
 - versus the if statement, 88–89
 - Synchronization, 12, 235–236, 247–250
 - and atomic operations, 946–947
 - and collections, 510, 550
 - and deadlock, 255–257
 - and interprocess communication, 251–257
 - objects, using, 917–936
 - race condition and, 248–249
 - and StringBuilder class, 439
 - via synchronized block, 249–250, 550
 - via synchronized method, 236, 247–249
 - versus concurrency utilities, traditional, 915, 964
 - synchronized modifier, 247, 915, 943, 964
 - used with a method, 247–249
 - used with an object, 249–250
 - synchronizedList(), 550, 554
 - synchronizedSet(), 550, 554
 - Synchronizers, 915, 916–917
 - SynchronousQueue class, 943
 - System class, 26, 34, 304, 467–470
 - methods, table of, 467–468
 - System.console(), 467, 680
 - System.err standard error stream, 304, 305, 467
 - System.exit(), 1078, 1180
 - System.getProperties(), 468, 572
 - System.getProperty(), 468, 470
 - System.in standard input stream, 304, 305, 464, 467, 620, 680
 - System.in.read(), 93
 - System.nanoTime(), 468, 469, 955
 - System.out standard output stream, 26, 34, 304, 305, 308, 309, 464, 467, 620, 665, 666, 680
 - and static import, 333, 334
 - System.out.println() and applets, 748, 767
- ## T
-
- Tabbed panes, 1053–1055
 - Table, Swing, 1066–1068
 - TableColumnModel, 1067
 - TableModel, 1067
 - TableModelEvent class, 1067
 - tailMap(), 534, 536
 - tailSet(), 506, 507
 - tan(), 477
 - tanh(), 477
 - @Target built-in annotation, 290, 291
 - TCP/IP, 12, 728
 - client sockets, 731–734
 - disadvantages of, 742
 - server sockets, 731, 741–742
 - See also* Transmission Control Protocol (TCP)
 - TERMINATE, 718
 - TERMINATED, 260
 - Ternary if-then-else operator (?:), 75, 77–78
 - test(), 638, 972
 - Text area, 780, 854–855
 - Text class, 1170
 - Text components as an event source, 782
 - Text fields, 780
 - AWT, 852–854
 - Swing, 1043–1044
 - Text formatting using java.text classes, 991, 1009–1013
 - Text output using font metrics, managing, 825–832
 - TextArea class
 - AWT, 800, 854–855
 - JavaFX, 1156
 - TextComponent class, 800, 852, 854
 - TextEvent class, 772, 780, 854
 - TextField class
 - AWT, 800, 852
 - JavaFX, 1154–1156
 - TextInputControl, 1154
 - TextListener interface, 782, 784
 - textValueChanged(), 784
 - thenComparing(), 543–544, 548–550
 - thenComparingDouble(), 544
 - thenComparingInt(), 544
 - thenComparingLong(), 544
 - this, 124–125, 145
 - and lambda expressions, 395
 - and type annotations, 292, 293, 296
 - this(), 334–336
 - Thompson, Ken, 4

- Thread class, 13, 236, 237, 481, 482–484, 602, 759, 915
 - constructors, 239, 242, 482
 - extending, 241–242
 - methods, table of, 482–484
- Thread(s)
 - creating, 238–243
 - daemon, 602, 951, 961
 - and deadlock, 255–257, 482, 1030
 - definition of, 233
 - executors to manage, using, 917, 937–942
 - group, 238, 484–488
 - JavaFX, 1112
 - local variables, 488
 - main, 236–238, 240, 242, 243, 244
 - messaging, 236, 251–255
 - pool, 937–939, 949, 950, 951, 954–955, 958, 963
 - priorities, 235, 246, 482
 - resuming, 257–261, 486–488
 - and spurious wakeup, 251
 - states of, possible, 235, 259–261
 - stopping, 257
 - suspending, 236, 237–238, 257–261, 486–488
 - and Swing, event dispatching, 1029–1030, 1033, 1034, 1035
 - synchronization. *See* Synchronization
- Thread.UncaughtExceptionHandler interface, 495
- ThreadGroup class, 481, 484–488, 495
 - methods, table of, 484–485
- ThreadLocal class, 488
- ThreadPoolExecutor class, 917, 937
- throw, 213, 222–223, 232
- Throwable class, 214–215, 218, 222, 223, 227, 230, 318, 379, 490, 491
 - methods defined by, table of, 228
 - obtaining an object of the, 222–223
- throws, 213, 223–224, 226
- Thumb, 849
- time, 588
- Time and date
 - formatting, 610–612, 1009–1013, 1015–1017
 - java.util classes that deal with, 586–596
 - strings, parsing, 1017–1018
- Time and Date API, 1009, 1013–1018
 - packages, list of, 1013
- timedJoin(), 943
- timedWait(), 943
- TIMED_WAITING, 260
- Timer class, 602–604
 - methods, table of, 603
- TimerTask class, 602–604
 - methods, table of, 602
- Timestamp, event, 773
- TimeUnit enumeration, 917, 923–924, 940, 942–943
- TimeZone class, 593, 594
 - methods defined by, table of some, 593
- to(), 465
- toAbsolutePath(), 695, 712
- toArray(), 504–504, 513–514, 968
- toBinaryString(), 451, 453, 455
- toCharArray(), 420
- toDays(), 943
- toDegrees(), 480, 481
- ToDoubleFunction functional interface, 636, 981
- toFile(), 695
- Toggle button, JavaFX, 1133–1135
 - adding an image to a, 1133
- Toggle button, Swing, 1047–1049
- Toggle interface, 1133, 1135, 1142
- Toggle group, 1136, 1138
 - handling change events in a, 1138–1139
 - and RadioMenuItems, 1183
- ToggleButton class, 1133–1135
- ToggleGroup class, 1136, 1138, 1142
- toHexString(), 444, 446, 451, 453, 455
- toHours(), 943
- toIntExact(), 480
- Tokens, 579, 620–621, 628
- toLanguageTag(), 595
- toList(), 982, 984
- toLocalDate(), 1015
- toLocalTime(), 1015
- toLowerCase(), 429–430, 457, 458
- Tomcat, 1212, 1213–1216
- toMicros(), 942
- toMillis(), 942
- toMinutes(), 943
- toNanos(), 943
- toOctalString(), 451, 453, 455
- ToolBar class, 1173, 1189
- Toolbars, 1069, 1070, 1087–1089, 1171, 1173, 1189–1190
- Tooltip class, 1170
- Tooltips, 1081, 1087, 1088–1089, 1170, 1189–1190
- TooManyListenersException, 1202
- toPath(), 645, 695, 712
- toRadians(), 480, 481
- toSeconds(), 943
- toSet(), 982
- toString(), 185, 186, 218, 227, 228, 230, 273, 280, 286, 309, 418–419, 429, 444, 446, 448, 449, 451, 453, 454, 460, 471, 475, 484, 485, 490, 491, 492, 493, 513, 560, 564, 570, 582, 583, 585, 587, 604, 606, 608, 666, 680, 694, 695, 731, 772, 820, 826, 963, 1064
- totalMemory(), 462–463
- toUnsignedInt(), 448, 449
- toUnsignedLong(), 448, 449, 451
- toUnsignedString(), 451, 453
- toUpperCase(), 429–430, 457
- toZonedDateTime(), 592
- transient modifier, 322, 1203
- Transform class, 1166
- Transforms, 1166–1167
 - program demonstrating, 1167–1170
- Translate class, 1166
- translatePoint(), 779
- Transmission Control Protocol (TCP)
 - definition of, 727
 - and stream-based I/O, 728
 - See also* TCP/IP
- TreeExpansionEvent class, 1063
- TreeExpansionListener interface, 1063
- TreeItem class, 1160, 1161, 1164
- TreeMap class, 537 539–540, 542, 577
 - example using a comparator, 549–550

- TreeModel, 1063
 - TreeModelEvent class, 1063
 - TreeModelListener interface, 1063
 - TreeNode interface, 1064
 - TreePath class, 1064
 - Trees
 - JavaFX, 1160–1164
 - Swing, 1063–1066
 - TreeSelectionEvent class, 1063
 - TreeSelectionListener interface, 1063, 1064
 - TreeSelectionModel, 1063
 - TreeSet class, 511, 517, 518–519, 542, 577
 - TreeView class, 1160–1163
 - trim(), 428
 - trimToSize(), 438, 513, 564
 - true, 34, 40, 41, 43, 75, 76
 - TRUE, 458
 - True and false in Java, 43, 75
 - Truncation, 49
 - try block(s), 213, 214, 216–222, 224–225, 232
 - nested, 220–222
 - try-with-resources statement, 15, 214, 231, 310, 315–318, 495, 619, 626, 648, 649–650, 656, 692, 694, 701, 714, 720, 732, 734, 743, 966
 - advantages to using, 650
 - tryAdvance(), 526, 527–528, 987
 - tryLock(), 917, 944
 - trySplit(), 988–989
 - tryUnfork(), 962
 - Two's complement, 66–67
 - TYPE, 443, 447, 455, 458, 460
 - Type argument(s), 340, 342, 346
 - and bounded types, 347–349
 - and generic class hierarchies, 364
 - and type inference, 358, 372–373
 - Type conversion
 - automatic, 35, 48, 130–131
 - narrowing, 48
 - widening, 48
 - Type enumeration, 465
 - Type interface, 496
 - Type parameter(s)
 - and bounded types, 346–349, 361–362
 - cannot create an instance of a, 377
 - and class hierarchies, 365–368
 - and erasure, 341, 373
 - and primitive types, 342
 - and static members, 377
 - and type safety, 342
 - used with a class, 340, 345–346, 347
 - used with a method, 340, 356–359
 - Type safety
 - and collections, 500, 550
 - and generic methods, 359
 - and generics, 337, 338, 341, 342–344, 500
 - and raw types, 362–364
 - and wildcard arguments, 349–352, 353
 - type(), 465
 - Type(s), 27
 - annotations, 16, 292–297
 - bounded, 347–349
 - casting, 48–49, 50
 - checking, 10, 11, 35, 341, 342–344, 363, 379
 - class as a data, 109, 110, 113, 114, 116, 126
 - inference, 358, 372–373, 383, 386, 395
 - non-reifiable, 292
 - parameterized, 338, 340
 - promotion, 37, 50–51, 69–70
 - raw, 362–364
 - simple, 35
 - TypeNotPresentException, 226
 - Types, primitive (simple), 35–36, 114, 136, 272, 342
 - autoboxing/unboxing and, 274–277, 279, 500, 514
 - and collections, 500, 514
 - iterators for, 499
 - to a string representation, converting, 417, 418, 428–429
 - to or from a sequence of bytes, converting, 667–669
 - wrappers for, 272–274, 279, 342, 442–460
 - Typesafe view of a collection, obtaining a dynamically, 550
-
- ## U
- UDP protocol, 727, 728, 742
 - UI delegate, 1023, 1024
 - ulp(), 478, 479
 - UnaryOperator functional interface, 504, 639
 - UnavailableException, 1217, 1220
 - Unboxing, 274
 - uncaughtException(), 495
 - UncaughtExceptionHandler interface, 495
 - Unchecked warnings and raw types, 364
 - UnicastRemoteObject, 1006
 - Unicode, 39, 40, 43, 302, 303, 415, 416, 420, 458, 670
 - code points, table of some Character methods providing support for, 459
 - support for 32-bit, 458
 - Unicode Technical Standard (UTS) 35, 595
 - Uniform Resource Identifier (URI), 741
 - Uniform Resource Locator (URL). *See* URL (Uniform Resource Locator)
 - UNIX, 4, 727
 - UnknownHostException, 729, 730
 - unlock(), 917, 944
 - unmodifiable... collections methods, 554–555
 - unordered(), 966, 977
 - Unreachable code, 108, 219
 - unread(), 662, 678
 - UnsupportedOperationException, 226, 501, 502, 504, 521, 531, 550, 699
 - update(), 598, 599, 754, 756, 757, 811, 832
 - overriding, 754
 - URI (Uniform Resource Identifier), 741
 - URI class, 741
 - URL (Uniform Resource Locator), 735, 741, 760, 1211
 - specification format, 735
 - URL class, 735–736, 738, 739, 766
 - methods, list of some, 737
 - URLConnection class, 736–739, 741
 - useDelimiter(), 628–629
 - User Datagram Protocol (UDP), 727, 728, 742

useRadix(), 630
 UTS 35, 595
 UUID class, 635

V

value (annotation member name), 289, 290
 valueChanged(), 1058, 1059, 1063, 1064
 valueOf(), 266–267, 418, 428–429, 444, 446, 448, 449, 451, 453, 460, 492, 582
 values(), 266–267, 531, 533
 van Hoff, Arthur, 6
 Varargs, 14, 155–160
 and ambiguity, 159–160
 methods, overloading, 158–159
 and Java's printf(), 155
 parameter, 157, 521
 Variable(s), 44–48
 capture, 395–396
 declaration, 27, 29, 44–45, 46–48
 definition of, 26, 44
 dynamic initialization of, 45
 effectively final, 395–396
 enumeration, 264
 final, 147, 263
 instance. *See* Instance variables
 interface, 197, 204–206
 interface reference, 198–199, 204
 object reference. *See* Object reference variables
 scope and lifetime of, 45–48
 Vector class, 498–499, 511, 530, 561, 562–566
 legacy methods, table of, 563–564
 VetoableChangeListener interface, 1203, 1204
 Viewport, scroll pane, 1056, 1157
 visitFile(), 718, 719
 void, 25, 116
 Void class, 460
 volatile modifier, 322
 VSPACE, 760, 761

W

wait(), 185, 186, 251, 253–255, 258–259, 471, 915, 943, 944, 964
 waitFor(), 461, 464
 WAITING, 260
 WALL_TIME, 594
 walk(), 695
 walkFileTree(), 717–718
 Warth, Chris, 6
 Watchable interface, 694
 WeakHashMap class, 537
 Web browser, 767
 executing applet in, 320, 321, 747–748, 751, 753, 759, 760, 761, 766, 801
 using status window of, 759
 Web server and servlets, 1211, 1212
 WebView class, 1196
 weightx constraint field, 867, 868
 weighty constraint field, 867, 868
 while loop, 89–90

Whitespace, 32, 82
 from a string, removing, 428
 whois, 728, 732–734, 735
 WIDTH, 760, 761
 Wildcard arguments, 349–356, 370
 bounded, 352–356
 used in creating an array, 379
 Window, AWT-based
 displaying information within an, 811
 as an event source, 780–781, 782
 frame. *See* Frame window
 fundamentals, 800–801
 and graphics, 811
 status, using, 759–760
 Window class, 781, 800, 801, 876
 Window, Swing JFrame, 1028
 windowActivated(), 785
 WindowAdapter class, 792
 windowClosed(), 785
 windowClosing(), 785, 803
 WindowConstants interface, 1028
 windowDeactivated(), 785
 windowDeiconified(), 785
 WindowEvent class, 772, 774, 780–781
 WindowFocusListener interface, 782, 785, 792
 windowGainedFocus(), 785
 windowIconified(), 785
 WindowListener interface, 782, 785, 792, 803
 windowLostFocus(), 785
 windowOpened(), 785
 WindowStateListener interface, 792
 Work stealing, 951, 962
 World Wide Web (WWW), 6, 7, 11, 735
 wrap(), 691
 Wrappers, primitive type, 272–274, 279, 342, 442–460
 WRITE, 465
 write(), 303, 304, 308, 314–315, 652, 672, 683, 684, 693, 706, 707, 711, 723, 724
 writeBoolean(), 668, 684
 writeDouble(), 668, 684
 Writer class, 303–304, 650, 670, 673, 688
 methods defined by, table of, 671–672
 writeExternal(), 683
 writeInt(), 668, 684
 writeObject(), 683, 686
 writeTo(), 659

X

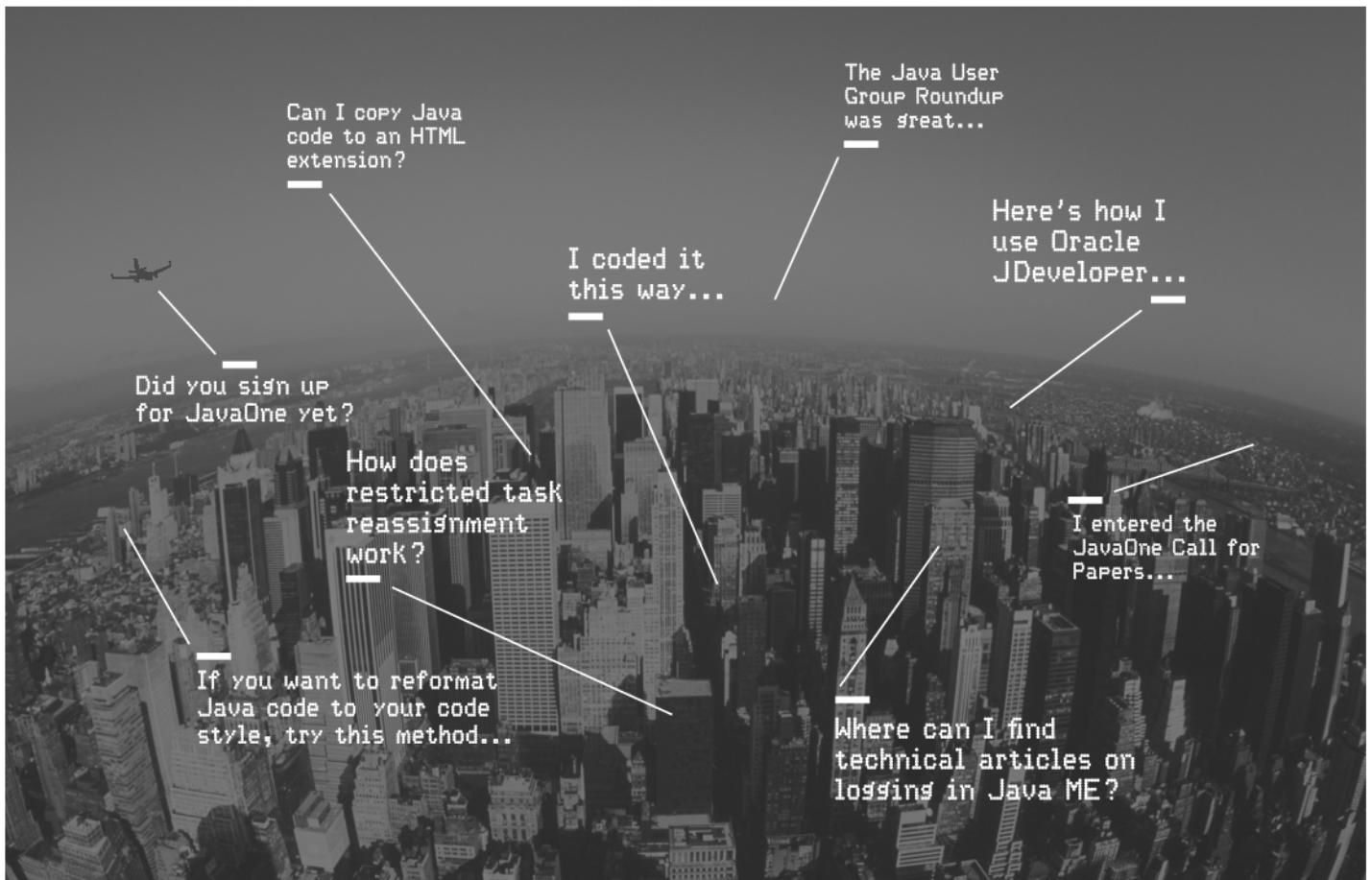
XOR (exclusive OR) operator (^)
 bitwise, 66, 67, 68–69
 Boolean logical, 75–76

Y

Yellin, Frank, 6

Z

Zero crossing, 67
 ZIP file format, 639



Oracle Technology Network. It's code for sharing expertise.

Come to the best place to collaborate with other IT professionals on everything Java.

Oracle Technology Network is the world's largest community of developers, administrators, and architects using Java and other industry-standard technologies with Oracle products.

Sign up for a free membership and you'll have access to:

- Discussion forums and hands-on labs
- Free downloadable software and sample code
- Product documentation
- Member-contributed content

ORACLE®
TECHNOLOGY NETWORK

Take advantage of our global network of knowledge.

JOIN TODAY ► Go to: oracle.com/technetwork/java

ORACLE®

Our Technology. Your Future.

Fast-track your career with an Oracle Certification.

Over 1.5 million
certifications testify to
the importance of these
top industry-recognized
credentials as one of the
best ways to get ahead.
AND STAY THERE.

START TODAY

ORACLE®
CERTIFICATION PROGRAM



certification.oracle.com

**Hardware and Software
Engineered to Work Together**



Stay Connected

oracle.com/technetwork/oracleace

 [oracleaces](https://www.facebook.com/oracleaces)

 [@oracleace](https://twitter.com/oracleace)

 blogs.oracle.com/oracleace

Need help? Need consultation?
Need an informed opinion?

You Need an Oracle ACE

Oracle partners, developers, and customers look to Oracle ACEs and Oracle ACE Directors for focused product expertise, systems and solutions discussion, and informed opinions on a wide range of data center implementations.

Their credentials are strong as Oracle product and technology experts, community enthusiasts, and solutions advocates.

And now is a great time to learn more about this elite group—or nominate a worthy colleague.

For more information about the
Oracle ACE program, go to:
oracle.com/technetwork/oracleace

ORACLE®



Reach More than 700,000 Oracle Customers with Oracle Publishing Group

Connect with the Audience
that Matters Most to Your Business



Oracle Magazine

The Largest IT Publication in the World

Circulation: 550,000

Audience: IT Managers, DBAs, Programmers, and Developers



Profit

Business Insight for Enterprise-Class Business Leaders to
Help Them Build a Better Business Using Oracle Technology

Circulation: 100,000

Audience: Top Executives and Line of Business Managers



Java Magazine

The Essential Source on Java Technology, the Java
Programming Language, and Java-Based Applications

Circulation: 125,000 and Growing Steady

Audience: Corporate and Independent Java Developers,
Programmers, and Architects



For more information
or to sign up for a FREE
subscription:
Scan the QR code to visit
Oracle Publishing online.

ORACLE®