
Index

A

About dialog box, 463–464
Absolute positioning, 445
Absolute resource name, 498
Abstract classes, 186–191, 691–692
 and interfaces, 248–249
 PersonTest.java, 189–191
Abstract methods, 187–188
AbstractAction class, 343, 346, 407, 409
AbstractButton class, 408–409, 410, 414
AbstractCollection superclass, LinkedList
 class, 664
ACCELERATOR_KEY, 343
Accelerators, 413–414
Access control, 259
Access order, 687
Action interface, methods, 342
actionCommand property, ButtonModel
 interface, 367
ACTION_COMMAND_KEY, 343
ActionEvent objects, 324
ActionListener interface, 325–327, 332,
 339, 342, 359
ActionMap class, 345
actionPerformed method, 256, 325, 335,
 343, 359, 806
Actions, 324
 defined, 342
 predefined action table names, 343
ActionTest.java, 346–348
activeCaption, 308
activeCaptionBorder, 308
activeCaptionText, 308

ActiveX controls, 13
Adapter classes, 338–341
add method, 765
 Set interface, 691
addActionListener method, JButton class,
 600
addAll method, TreeSet class, 674
addBall method, 716
addChangeListener method, JSlider class,
 600
addItem method, 397
addWindowListener method, 341
AdjustmentEvent, 359
AdjustmentListener interface, 359
adjustmentValueChanged method, 359
Advanced logging, 576–577
Aggregation, UML connector, 110
AlgorithmAnimation.java, 789–792
Algorithms, 700–707
 binary search, 704–705
 collections, 700–707
 merge sort algorithm, 702
 ShuffleTest.java, 703
 simple, 705–706
 sorting and shuffling, 701–704
 writing, 706–707
Algorithms + Data Structures = Programs
 (Wirth), 106
Algorithms in C++ (Sedgewick), 702
Andreessen, Mark, 9–10
Animation, pausing/resuming
 (example), 788–793
Anonymous arrays, 92

- Anonymous inner classes, 258–259, 269–271, 332
 - AnonymousInnerClassTest.java, 270–271
 - defined, 269
- Applet class, 520–521, 529–530, 538–539
 - play method, 530
- AppletContext class, 531, 539
- AppletFrame.java, 535–538
- Applets, 7–8, 516–539
 - applet HTML tag, 521–524
 - AppletContext interface, 531, 534–535
 - AppletFrame.java, 535–538
 - AppletStub interface, 534–535
 - building and running, 31–34
 - Chart.java, 527–529
 - context, 531–539
 - converting applications to, 520
 - defined, 516
 - image and audio files, accessing, 530
 - inheritance diagram, 517
 - inter-applet communication, 523, 531
 - Java Plug-in, 32, 516
 - “Not Hello World” applet, 516–518
 - NotHelloWorldApplet.java, 518
 - object tag, 524
 - serialized applet object, 522
 - showStatus method, 531–532
 - using parameters to pass information to, 524–529
 - viewing in a browser, 519–520
 - viewing in applet viewer, 519
 - WelcomeApplet.html, 32–33
 - WelcomeApplet.java, 33–34
- Application preferences, 539
 - Preferences class, 544–546
 - PreferencesTest.java, 546–549
 - PropertiesTest.java, 541–543
 - property maps, 539–544
- Application Programming Interface (API), 58
 - reading on-line documentation, 59–62
- ArgoUML (software), 110
- args parameter, 93–94
- Arguments, 39
- Arithmetic operators, 46–47
- Array class, 231
- Array list elements:
 - accessing, 206–209
 - ArrayListTest.java, 208–209
- Array lists, 91, 668
 - raw, compatibility between typed array lists and, 210–211
- Array variables, defining, 90
- ArrayBlockingQueue, 766
- arraycopy method, 231
- ArrayDeque class, as concrete collection type, 658
- ArrayGrowTest.java, 232–234
- ArrayIndexOutOfBoundsException, 554, 794
- ArrayList class, 204, 614, 659, 691
 - as concrete collection type, 658
- Arrays, 90–104
 - anonymous, 92
 - array entries, numbering of, 91
 - array lists, 204–211
 - clone method, 253
 - command-line parameters, 95
 - copying, 93–94
 - declaring, 90
 - defined, 90
 - “for each” loop, 91–92
 - initializers, 92–93
 - multidimensional, 99–102
 - ragged, 102–104
 - sorting, 95–97
 - LotteryDrawing.java, 96–97
- Arrays class, 698
 - asList method, 694
- Ascender, 312–313
- Ascent, 312–313
- ASCII, 43
- asList method, Arrays class, 694
- Assertions, 571–575
 - defined, 571
 - enabling/disabling, 572–573
 - using for documenting assumptions, 574
 - using for parameter checking, 573–574
- Association, UML connector, 110
- AtomicBoolean, 757
- Autoboxing, defined, 211
- await method, 761

awaitUninterruptibly method, 761
AWT event hierarchy, 357–360
AWT events:
 EventTracer class, 599
 EventTracer.java, 600–601
 EventTracerTest.java, 602
 tracing, 598–602
AWTEvent class, 357

B

Background color, specifying, 308
Backslash, escape sequence, 43
Backspace, escape sequence, 42
BadCastException, 641
BallComponent.java, 721–722
Ball.java, 720–721
Bank.java, 738–739, 748–750, 752–753
Barriers, 787–788
Baseline, 312–313
Basic logging, 575
Basic multilingual planes, 43
BasicButtonUI, 368
BasicService interface, 509, 514
Beans, 494
Behavior, objects, 108
BIG-5, 43
BigDecimal class, 88, 90
BigInteger class, 88, 90
BigIntegerTest.java, 89
Binary search, 704–705
binarySearch method:
 Arrays class, 277
 Collections class, 704–705
Bit sets, 710–713
BitSet class, 707, 710–711
 “Sieve of Eratosthenes” benchmark,
 711–713
 Sieve.cpp, 712–713
 Sieve.java, 711–712
BitSet interface, 650
Bitwise operators, 49
Black box behavior, objects, 107
Blank final variable, 268
Block scope, 71–72
Block statements, 72
Blocked threads, 731–732

Blocking queues, 764–771
 defined, 764
 methods, 765
BlockingQueue, 770
BlockingQueueTest.java, 766–769
Blocks, 38
 defined, 73
boolean operators, 48–49
boolean type, 44
 conversion character, 66
Border layout manager, 371–373
BorderFactory method, 392
BorderLayout, 373
Borders, 392–396
 BorderTest.java, 393–394
Bounce.java, 718–720
BounceThread.java, 725–727
Bounded collections, 653
Braces styles, 38
break statements, 86–87
Bridge methods, 623–624
brighter() method, Color class, 308
BrokenBarrierException, 787
Bucket, 669–670
BuggyButtonTest.java, 607–608
Bulk operations, 699
Button click:
 ButtonTest.java, 329–331
 handling, 326–331
ButtonModel interface, properties of, 367
ButtonTest program, 607
ButtonUIListener class, 368
byte type, 40

C

C/C++/C#, 12
C++ bitset template, 710
Calculator.java, 374–377
CalendarTest.java, 118–120
Callable interface, 774–778
Callbacks, and interfaces, 255–258
Camel case (CamelCase), 37
Carriage return, escape sequence, 42
Casting, 184–186
Casts, 51–52
Catching exceptions, 219–221

- char type, 42–44
- Character class:
 - isJavaIdentifierPart method, 44
 - isJavaIdentifierStart method, 44
- Character, conversion character, 66
- Chart.java, 527–529
- Checkbox menu items, 409–410
- Checkboxes, 385–387, 388
- CheckBoxTest.java, 386–387
- Checked exceptions, 218–219, 554
 - declaring, 555–557
- Checked views, 696–697
- Child classes, 171
- Choice components, 385–406
 - borders, 392–396
 - checkboxes, 385–387
 - combo boxes, 396–399
 - radio buttons, 388–392
 - sliders, 400–406
- CircleLayoutTest.java, 447–450
- Clark, Arthur C., 440
- Clark, Jim, 9
- Class class, 204, 217–218, 220, 225, 230, 500, 640, 642, 646
 - forName method, 219
 - parameterization of, 218
- Class constants, 46
- Class diagrams, 109–110
- Class files, 160
- Class loaders, 276
 - defined, 572–574
- Class path, 160–162
 - setting, 161–162
- Class relationships, UML notation for, 110
- ClassCastException, 697
- Classes, 36–38, 105–169, 171–192
 - abstract, 186–191
 - adapter, 338–341
 - class-based access privileges, 131
 - class path, 160–162
 - setting, 161–162
 - constructors, 126–127
 - defined, 107
 - defining your own, 122–132
 - design hints, 167–169
 - Employee class, 122–123
 - dissecting, 125–126
 - EmployeeTest.java, 124–125
 - encapsulation, 107
 - benefits of, 129–131
 - final instance fields, 132
 - identifying, 108–109
 - immutable, 132
 - implicit and explicit parameters, 127–128
 - multiple source files, use of, 125
 - names, 36–37
 - nested, 259
 - predefined, 111–121
 - private methods, 132
 - protected access, 191–192
 - relationships between, 109–110
- ClassLoader, 574–575
- CLASSPATH environment variable, 162
- Client-side locking, 754
- clone method, 238, 251–252
- Cloneable interface, 251–253
- CloneNotSupportedException, 253
- CloneTest.java, 254–255
- Code errors, 553
- Code planes, 43
- Code point, 43
- Code units, 43–44
- Collection interface, 653–657, 664, 689–690, 693
 - “for each” loop, 654
 - generic utility methods, 656–657
 - methods, 653
- Collections, 649–713, *See also* Collections framework
 - algorithms, 700–707
 - bounded, 653
 - concrete, 658–689
 - framework, 681, 689–700
 - fundamental interfaces, 689
 - inserting elements into, 689
 - interfaces, 650–658
 - separating implementation and, 650–653
 - legacy, 707–713
 - ordered collection, 690
 - thread-safe, 771–774
- Collections class, 697–698, 703–706

- binarySearch method, 704–705
- synchronizedMap method, 696
- Collections framework, 681, 689–700
 - bulk operations, 699
 - classes in, 692
 - converting between collections and arrays, 700
 - optional operations, 697–699
 - views and wrappers, 693–697
- Color, 307–310
 - background, 308
 - custom color, specifying, 307
 - system colors, 308
- Color choosers, 485–491
 - ColorChooserTest.java, 487–490
 - HSB pane of, 486
 - RGB pane of, 486
 - “swatches” pane of, 485
- Color class, 307, 309
 - brighter() method, 308
 - darker() method, 308
- Color dialogs, 452
- ColorAction class, 343–344
- ColorChooserTest.java, 487–490
- Combo boxes, 396–399
 - ComboBoxTest.java, 398–399
- ComboBoxTest.java, 398–399
- Comma-separated list of expressions, 52
- Command-line tools, using, 21–25
- Comments, 39–40
 - marking, 39
- Comparable interface, 246, 619, 673–678, 702
- Comparator interface, 674
- compare method, Comparable interface, 673
- compareTo method, Comparable interface, 243–244, 673–674
- Compilation errors, locating, 27–28
- Compiler, 45, 264, 756
- Component class, 288–289, 293, 299, 310, 357, 379, 446
 - inheritance hierarchy for, 370
- Components, 298, 369
 - displaying information in, 294–299
 - NotHelloWorld.java, 297–298
- ComponentTreeUI method, 335
- Compound statement, 73
- CompoundInterest.java, 101–102
- Concrete classes, 692
- Concrete collections, 658–689
- Concurrent hash map, 771
- Concurrent modification detection, 664
- ConcurrentHashMap class, 771–772
- ConcurrentLinkedQueue class, 771–772
- ConcurrentModificationException, 663, 771, 773
- ConcurrentSkipListMap class, 771–772
- ConcurrentSkipListSet class, 771–772
- Condition objects, 745–750
 - Bank.java, 748–750
 - defined, 745
- Condition variables, 745
- Conditional statements, 72–75
- Console class, 65
- ConsoleHandler class, 579, 583, 590
- ConsoleWindow class, 597–598
- const keyword, 46
- Constants, 46
- Construction, overloading, 144–145
- Constructor class, 641
- Constructors, 126–127, 135
 - calling another constructor, 148
 - ConstructorTest.java, 150–152
 - default constructors, 145–146
 - default field initialization, 145
 - explicit field initialization, 146–147
 - initialization blocks, 148–150
 - parameter names, 147
- ConstructorTest.java, 150–152
- Container class, 331, 370
- Containers, 297, 369
- Content pane, 295
- continue statement, 87–88
- control, 309
- Control flow, 71–88
 - block scope, 71–72
 - conditional statements, 72–75
 - determinate loops, 80–83
 - loops, 76–83
- controlDkShadow, 309
- controlHighlight, 309
- controlLtHighlight, 309
- controlShadow, 309
- controlText, 309
- Conversion character, 66

- Copying arrays, 93–94
- CopyOnWriteArrayList, 773
- CopyOnWriteArraySet, 773
- Core Java program examples, installing, 20–21
- Countdown latches, 786–787
- CountDownLatch class, 785, 786–787
- createCustomCursor method, Toolkit class, 352
- CROSSHAIR_CURSOR, 351
- Cursor shapes, 351
- Custom color, specifying, 307
- Custom layout management, 446–450
 - CircleLayoutTest.java, 447–450
- CyclicBarrier class, 785, 787–788

- D**
- Daemon threads, 734
- darker() method, Color class, 308
- Data exchange, 467–474
 - DataExchangeTest.java, 469–473
- Data types, 40–44
 - boolean type, 44
 - char type, 42–44
 - floating-point types, 41–42
 - integer types, 40–41
- Date and time, conversion character, 66–68
- DateInterval class, 623–624
- Deadlocks, 757–760
- Debuggers, 7, 591–611
 - AWT events:
 - tracing, 598–602
 - using the AWT robot, 603–606
 - BuggyButtonTest.java, 607–608
 - console window, using, 597–598
 - ConsoleWindow.java, 597–598
 - using, 607–611
- Decimal integer, conversion character, 66
- Declaration of variable, 45
- Decrement operators, 47–48
- Deep copy, 251
- DEFAULT, 343
- DefaultButtonModel, 368
- DEFAULT_CURSOR, 351
- Definition of variable, 45
- DelayQueue, 766, 770

- Delegates, 234
- Dependency, UML connector, 110
- Derived classes, 171
- deriveFont method, 312
- Descender, 312–313
- Descent, 312–313
- Design patterns, 362–363
- Design Patterns—Elements of Reusable Object-Oriented Software* (Gamma et al.), 363
- desktop, 308
- Determinate loops, 80–83
- Development environment, choosing, 21–22
- Device errors, 553
- Dialogs, 452–491
 - color choosers, 485–491
 - creating, 462–467
 - data exchange, 467–474
 - default button, 468–469
 - DialogTest.java, 464–466
 - file dialogs, 474–485
 - option dialogs, 453–462
 - password dialog box, 467
- Digitally signed JAR file, 505–506
- Dijkstra, Edsger, 786
- Directed association, UML connector, 110
- do/while loop, 77–78
- DocCheck, 167
- Doclets, 166
- Documentation comments, 162–166
- Double-precision numbers, 41
- Double quote, escape sequence, 43
- double type, 41, 51
- Doubly linked lists, 660
- drawImage method, Graphics class, 318
- DrawTest.java, 304–306
- Dynamic binding, 181–182

- E**
- Echo character, 381
- Eclipse, 25–28, 591
 - configuring a project, 26
 - editing a source file, 26
 - error messages in, 28
 - error report, 27
 - running a program in, 27

- ECMA-262, 13
- element method, 765
- Ellipse2D class, 302
- Ellipse2D.Double, 307
- Ellipses, constructing, 302–304
- Employee class, 122–123, 171–172
 - dissecting, 125–126
- EmployeeSortTest.java, 245–246
- EmployeeTest.java, 124–125
- enabled property, ButtonModel interface, 367
- Encapsulation, 107
 - benefits of, 129–131
- Enterprise Edition (EE), 17
- Enum class, 216, 216–217
- Enumerated types, 53
- Enumeration classes, 215–217
 - EnumTest.java, 216–217
- Enumeration interface, 650, 708–709
- Enumeration objects, 531
- Enumeration sets and maps, 687–688
- Enumerations, 708–709
- EnumMap class, 687
 - as concrete collection type, 659
- EnumSet class, 687–688
 - as concrete collection type, 658
- EnumTest.java, 216–217
- Equality testing, and inheritance, 194–196
- equals method:
 - Object class, 193–194
 - Set interface, 691
- EqualsTest.java, 200–203
- E_RESIZE_CURSOR, 351
- Error hierarchy, 553
- Errors:
 - code, 553
 - dealing with, 552–553
 - device, 553
 - input, 552
 - physical limitations, 553
- Escape sequences, special characters, 42–43
- Event delegation model, 324
- Event dispatch thread, 286–287
- Event handling, 323–360
 - basics of, 324–341
 - defined, 324
 - summary, 359–360
- Event listeners, 324
- Event notification, 325–326
- Event object, 324
- Event procedure, 324
- Event source, 324–325
 - relationship between listeners and, 325
- EventObject class, 357
- Exception, 568
- Exception handler, 218, 220
- Exception handling, 552
- Exception hierarchy, 553–554
- Exceptions:
 - ArrayIndexOutOfBoundsException, 554
 - BadCastException, 641
 - catching, 559–568
 - checked, 554
 - declaring, 555–557
 - classCastException, 697
 - classification of, 553–555
 - ConcurrentModificationException, 663, 771
 - exception classes, creating, 558–559
 - finally clause, 563–565
 - InterruptedException, 760–761
 - IOException, 560, 565
 - multiple, catching, 561
 - NoSuchElementException, 653
 - rethrowing and chaining, 562
 - RuntimeException, 553–555
 - stack trace:
 - defined, 566
 - StackTraceElement class, 566
 - StackTraceTest.java, 567
 - throwing, 557–558
 - TimeoutException, 774
 - tips for using, 568–571
 - uncaught exceptions, handlers for, 735–736
 - unchecked, 554
 - UnsupportedOperationException, 694, 697
- Exchanger class, 785, 787
- Exchangers, 787
- Executable JAR files, 496–497
- Execution path, setting, 18–20
- ExecutorCompletionService, 784–785

- Executors, 778–785
 - groups of tasks, controlling, 784–785
 - scheduled execution, 783–784
 - thread pools, 779–784
 - purpose of, 778
 - Executors class, 778–785
 - newCachedThreadPool method, 779
 - newFixedThreadPool method, 779
 - newScheduledThreadPool method, 779, 783
 - newSingleThreadExecutor method, 779
 - newSingleThreadScheduledExecutor method, 779, 783
 - ExecutorService interface, 779, 780, 783
 - Expected behavior, 239
 - Explicit parameters, 127–128
 - Exponential floating-point, conversion
 - character, 66
 - ExtendedService class, 509
- F**
- Factory methods, 135
 - Field width, 65
 - File access warning, 509
 - File dialogs, 452, 474–485
 - FileChooserTest.java, 479–483
 - File filter, 475
 - File handler configuration parameters, 581
 - FileContents class, 515
 - FileFilter interface, 476, 484
 - FileHandler class, 583, 590
 - FileNameExtensionFilter, 484
 - FileOpenService class, 515
 - openFileDialog method, 509
 - FileSaveService class, 515
 - FileView class, 477, 484–485
 - Filter class, 591
 - final keyword, 46
 - finally clause, 563–565
 - First Person, Inc., 9
 - FirstSample.java, 39
 - Fixed-point floating-point, conversion
 - character, 66
 - float type, 41, 41–42, 51
 - Floating-point overflow, 47
 - Floating-point types, 41–42
 - Flow layout manager, 369
 - FlowLayoutManager, 371, 372–373
 - FocusableComponent method, JComponent
 - class, 452
 - FocusEvent, 359
 - focusGained method, 360
 - FocusListener interface, 360
 - focusLost method, 360
 - Font, 316–317
 - createFont method, 312
 - Font face name, 310–311
 - Font family name, 310
 - FontMetrics, 318
 - FontRenderContext class, 312
 - FontTest.java, 314–316
 - for loop, 80–83
 - LotteryOdds.java, 83
 - Format specifiers, 66
 - Formattable interface, 66
 - Formatter superclass, 591
 - formatTo method, 66
 - and Formattable interface, 66
 - Frame class, 285, 293–294
 - Frames:
 - creating, 285–288
 - defined, 285
 - frame size, determining, 290–292
 - positioning, 288–294
 - properties of, 290
 - SimpleFrameTest.java, 286
 - SizedFrameTest.java, 292
 - Framework:
 - collections, *See* Collections framework
 - defined, 689
 - Function object, 674
 - FutureTask wrapper, 774
 - FutureTest.java, 775–777
- G**
- GB18030, 43
 - GBC.java, 433–435
 - General floating-point, conversion
 - character, 66
 - General Public License, 12
 - Generic classes, 616
 - Generic code, and the virtual machine, 621–626

- Generic collection interfaces, advantage of, 700
- Generic expressions, translating, 622–623
- Generic methods, 618–619
 - translating, 623
- Generic programmers, 615
- Generic programming, 613–648
 - ArrayList class, 614
 - defined, 614
 - generic types, inheritance rules for, 630–632
 - and inheritance, 614
 - reflection, 640–647
 - restrictions and limitations, 626–630
 - simple generic class, defined, 616–617
 - skill levels, 615
 - wildcard types, 632–640
- GenericArrayType interface, 642, 647
- GenericReflectionTest.java, 643–645
- Generics:
 - defined, 614
 - generic type information in the virtual machine, 641–643
 - GenericReflectionTest.java, 643–645
 - and reflection, 640–647
 - and type parameters, 614
 - using `Class<T>` parameters for type matching, 641
- GentleWare (software), 110
- getActionCommand method, 357
- getAddListenerMethod method,
 - EventSetDescriptor class, 599
- getAppletContext method, 531
- getAvailableGraphicsEnvironment method, 310
- getBalances method, 763
- getClickCount method, 349
- getCodeBase method, 530
- getConstructor method, 641
- getConstructors method, Class class, 221
- getDeclaredConstructor method, 641
- getDeclaredConstructors method, Class class, 221
- getDeclaredFields method, Class class, 221
- getDeclaredMethods method, Class class, 221, 235
- getDelay method, 766
- getDocumentBase method, 530
- getEnumConstants method, 641
- getField method, Class class, 235
- getFields method, Class class, 221
- getFontRenderContext method, Graphics2D class, 312
- getListenerType method,
 - EventSetDescriptor class, 599
- getLocalGraphicsEnvironment method, 310
- getMethod method, Class class, 235
- getMethods method, Class class, 221
- getModifiers method, 221
- getModifiersEx method, 350
- getPredefinedCursor method, Cursor class, 351
- getSelectedItem method, 397
- getSource method, 357
- getStackTrace method, 566
- getStringBounds method, 312–313
- getTime and setTime methods, Calendar class, 183
- getTotalBalance method, 744
- getValue method, 342
- Goetz, Brian, 716, 756
- Gosling, James, 9–10
- goto statements, 86
- Graphical application, running, 28–31
- Graphics class, 309, 321
 - drawImage method, 318
- Graphics class, methods, 299
- Graphics debugger, Swing, 594
- Graphics programming, 281–322
- Graphics2D class, 310, 317
- GraphicsEnvironment class, 310, 606
- Green project, 9
- GregorianCalendar class, 114–115
- Grid bag layout, 426–436
 - constraints, 430
 - defined, 426
 - external padding, 428
 - fill and anchor parameters, 428
 - GBC.java, 433–435
 - GridBagLayoutTest.java, 431–432
 - gridx, gridy, gridwidth, and gridheight parameters, 427–428
 - alternative method to, 428–429
 - internal padding, 428

- Grid bag layout (*continued*):
 - padding, 428
 - weight fields, 428
 - Grid layout, 373–377
 - Calculator.java, 374–377
 - GridBagConstraints, 431–432
 - GridBagLayoutTest.java, 431–432
 - GridLayout class, 370, 377
 - Group layout, 436–445
 - GroupLayoutTest.java, 441–444
 - GroupLayout class, 436, 444–445
 - Groups of tasks, controlling, 784–785
- H**
- HAND_CURSOR, 351
 - Handler class, 589–590
 - Hansen, Per Brinch, 755–756
 - Hash code, 196–198
 - conversion character, 66
 - defined, 668–669
 - Hash collision, 669–670
 - Hash map, 681
 - Hash set iterator, 670
 - Hash sets, 668–669
 - adding elements into, 672
 - Hash tables, 668–670
 - defined, 668
 - hashCode method, 196–198, 669, 689
 - Set interface, 691
 - HashMap class, 681, 708
 - as concrete collection type, 659
 - HashSet class, 670
 - Hashtable class, 626, 707–708
 - elements method, 708
 - Hashtable interface, 650, 709
 - as concrete collection type, 658
 - hasMoreElements method, Enumeration
 - interface, 708–709
 - Height, 313
 - Hexadecimal floating-point, conversion
 - character, 66
 - Hexadecimal integer, conversion
 - character, 66
 - Hierarchies, inheritance, 179–180
 - Hoare, Tony, 755
 - Hold count, 744
 - HotJava browser, 10
 - HTML, 11
- I**
- Identity hash maps, 688
 - Identity, objects, 108
 - IdentityHashMap, 668
 - as concrete collection type, 659
 - if/else statement, 73–75
 - if statement, 72
 - IllegalStateException, 655–656
 - Images, displaying, 318–322
 - ImageTest.java, 319–320
 - ImageViewer.java, 29–31
 - Immutable classes, 132
 - implements keyword, 619
 - Implicit parameters, 127–128
 - inactiveCaption, 308
 - inactiveCaptionBorder, 308
 - inactiveCaptionText, 308
 - Increment operators, 47–48
 - Index, 69, 90
 - info, 309
 - Information hiding, 107
 - infoText, 309
 - Inheritance, 108, 109, 171–240
 - defined, 171
 - design hints, 238–240
 - dynamic binding, 181–182
 - and equality testing, 194–196
 - final classes and methods, 183–184
 - hash code, 196–198
 - hierarchies, 179–180
 - multiple, 179, 249
 - polymorphism, 179–180
 - preventing, 183–184
 - UML connector, 110
 - Inheritance chain, 179
 - Inheritance hierarchy, 369
 - for the Component class, 370
 - Inlining, 184
 - Inner classes, 242, 258–275, *See also*
 - Anonymous inner classes
 - to access object state, 259
 - anonymous, 258–259, 269–271, 332
 - becoming comfortable with, 331–333

- defined, 258
- InnerClassTest.java, 262–263
- local, 259, 266–267
- need for, 264–266
- special syntax rules for, 263–264
- static, 259, 271–275
- syntax, 264
- usefulness of, 264–266

Input:

- InputTest.java, 64
- reading, 63–65

Input errors, 552

Input maps:

- JComponent class, 344–345
 - conditions, 345

InputEvent, 356

Insecure certificate, 507

Instance, 107

Instance fields, 107

int type, 40

Integer class, 213, 277

Integer types, 40–41

Integrated development environment, using, 25–28

Inter-applet communication, 523, 531

Interface inheritance, UML connector, 110

Interfaces, 242–249

- and abstract classes, 248–249
- and callbacks, 255–258
- collections, 650–658
- defined, 242
- marker, 252
- properties of, 247–248
- tagging, 252

Internal padding, 428

Internationalization API, 498–499

InterruptedException, 760–761

Introduction to Algorithms (Cormen/Leiserson/Rivest/Stein), 672

Invocation handler, 275

InvocationHandler interface, 275

invokeAny method, 784

IOException, 560, 565

Is-a relationship, 109, 171, 238

isJavaIdentifierPart method, Character class, 44

isJavaIdentifierStart method, Character class, 44

ISO 8859-1, 43

isTraversable method, 477

ItemEvent, 359

ItemListener interface, 360

itemStateChanged method, 360

Iterator interface, 653–657, 661

- “for each” loop, 654
- generic utility methods, 656–657
- methods, 653–655
- remove method, 655–656

IzPack, 497

J

JApplet class, 520

JAR files, 160, 494–501

- executable, 496–497
- manifest, 495–496
- and resources, 497–498

jar tool, 494

- program options, 495

Java, 16–18

- applets, 7–8, 13
 - building and running, 31–34
- architecture-neutral object file format, 5–6
- casting, 184–186
- classes, 36–38
- command-line tools, using, 21–25
- comments, 39–40
- common misconceptions about, 11–14
- constants, 46
- control flow, 71–88
- Core Java* program examples, installing, 20–21
- data types, 40–44
- development environment, choosing, 21–22
- directories, navigating, 21
- directory tree, navigating, 21
- documentation, installing, 20
- as a dynamic language, 7
- evolution of (table), 11
- fundamental programming structures in, 35–104

- Java (*continued*):
 - integer types, 40–41
 - interfaces, 234
 - and the Internet, 7–8
 - interpreter, 6
 - jargon (table), 17
 - libraries, 11, 21
 - library source files, installing, 20
 - methods, 37
 - multithreading, 7
 - network-savvy nature of, 4
 - object-oriented design, 3–4
 - operators, 46–53
 - performance, 6–7
 - portability of, 6, 47
 - as a programming platform, 2
 - robustness of, 4
 - security of, 5
 - security risk, 13
 - short history of, 9–11
 - simple Java program, 36–39
 - simplest form for a class definition in, 122
 - simplicity of, 3
 - strings, 53–63
 - syntax, 3
 - variables, initializing, 45–46
 - versions/editions, 10
 - White Paper buzzwords, 2–3
- Java 2 (J2), 17
- Java 2D library, 299
- Java collections library, 689
- Java Community Process, 12
- Java Concurrency in Practice* (Goetz), 716
- Java Development Kit (JDK), 5, 16–18, 160
 - documentation, 346
 - downloading, 16–17
 - execution path, setting, 18–19
 - installation directory, 18
 - installing, 16
 - setup procedure, 18
- Java FAQ, 14
- Java Language Specification, 37
- Java Memory Model and Thread Specification, 756
- Java Micro Edition, 3
- Java Plug-in, 32, 516
- Java Runtime Environment (JRE), 17
- Java virtual machine (JVM), 9, 497
- Java Virtual Machine Specification, 37
- Java Web Start, 501–515
 - application characteristics, 501
 - calculator application, 501–504
 - defined, 501
 - desktop integration warning, 504
 - launching, 502
 - sandbox, 504–505
 - security advisory, 508
 - security manager, 504–505
 - signed code, 505–507
- java.applet.Applet, 520–521, 529, 530, 538–539
- java.applet.AppletContext, 539
- java.awt.BorderLayout, 373
- java.awt.Color, 309
- java.awt.Component, 293, 299, 310, 357, 379, 446
- java.awt.Container, 331, 370
- java.awt.event.ActionEvent, 333
- java.awt.event.InputEvent, 356
- java.awt.event.MouseEvent, 356, 411
- java.awt.event.WindowEvent, 341
- java.awt.event.WindowListener, 341
- java.awt.event.WindowStateListener, 341
- java.awt.FlowLayout, 371
- java.awt.Font, 316–317
- java.awt.font.LineMetrics, 317
- java.awt.FontMetrics, 318
- java.awt.Frame, 293–294
- java.awt.geom.Ellipse2D.Double, 307
- java.awt.geom.Line2D.Double, 307
- java.awt.geom.Point2D.Double, 307
- java.awt.geom.Rectangle2D.Double, 306
- java.awt.geom.Rectangle2D.Float, 306
- java.awt.geom.RectangularShape, 306
- java.awt.Graphics, 309, 321
- java.awt.Graphics2D, 310, 317
- java.awt.GraphicsEnvironment, 606
- java.awt.GridBagConstraints, 431–432
- java.awt.GridLayout, 377
- java.awt.LayoutManager, 450
- java.awt.Toolkit, 294, 357
- java.awt.Window, 293, 377
- java.beans.EventHandler, 333

- javac compiler, 161
- avadoc utility, 162–166
 - class comments, 163
 - comment extraction, 166
 - comment insertion, 162–163
 - field comments, 164
 - general comments, 164–165
 - method comments, 164
 - on-line documentation, 166
 - package and overview comments, 165–166
- java.io.Console, 65
- java.io.File, 71
- java.io.PrintWriter, 71
- java.lang.Class, 204, 220, 225, 230, 500, 641, 646
- java.lang.ClassLoader, 574–575
- java.lang.Comparable, 246, 677
- java.lang.Enum, 216–217
- java.lang.Exception, 568
- java.lang.Integer, 213
- java.lang.Object, 108, 198, 203, 672, 753–754
- java.lang.reflect.AccessibleObject, 230
- java.lang.reflect.Constructor, 220, 225, 641
- java.lang.reflect.Field, 225, 230
- java.lang.reflect.GenericArrayType, 647
- java.lang.reflect.InvocationHandler, 280
- java.lang.reflect.Method, 225, 238, 646
- java.lang.reflect.Modifier, 225–226
- java.lang.reflect.ParameterizedType, 646–647
- java.lang.reflect.Proxy, 280
- java.lang.reflect.TypeVariable, 646
- java.lang.reflect.WildcardType, 646
- java.lang.Runnable, 727
- java.lang.StackTraceElement, 568
- java.lang.System, 65, 99, 543–544, 689
- java.lang.Thread, 727, 730, 733, 734
- java.lang.ThreadGroup, 736
- java.lang.Thread.UncaughtExceptionHandler, 736
- java.lang.Throwable, 221, 559, 567–568
- java.math.BigDecimal, 90
- java.math.BigInteger, 90
- JavaScript, 13, 523–524
 - accessing an applet from, 523
 - JavaScript: The Definitive Guide* (Flanagan), 523
- JavaScript-to-Java communication, 523
- java.text.DateFormatSymbols, 121
- java.text.NumberFormat, 214
- java.util.ArrayDeque, 679
- java.util.ArrayList, 206, 210
- java.util.Arrays, 97–98, 198, 246, 698
- java.util.BitSet, 710–711
- java.util.Collection, 657–658
- java.util.Collections, 697–698, 703, 774
- java.util.Comparator, 677
- java.util.concurrent framework, 742
- java.util.concurrent.ArrayBlockingQueue, 769
- java.util.concurrent.BlockingDeque, 770–771
- java.util.concurrent.BlockingQueue, 770
- java.util.concurrent.Callable, 778
- java.util.concurrent.ConcurrentHashMap, 772
- java.util.concurrent.ConcurrentLinkedQueue, 772
- java.util.concurrent.ConcurrentSkipListMap, 772
- java.util.concurrent.ConcurrentSkipListSet, 772
- java.util.concurrent.Delayed, 770
- java.util.concurrent.DelayQueue, 770
- java.util.concurrent.Executors, 783
- java.util.concurrent.ExecutorService, 783
- java.util.concurrent.FutureTask, 778
- java.util.concurrent.Future, 778
- java.util.concurrent.LinkedBlockingDeque, 769
- java.util.concurrent.LinkedBlockingQueue, 769
- java.util.concurrent.locks.Condition, 750, 761
- java.util.concurrent.locks.Lock, 745, 750, 761
- java.util.concurrent.locks.ReentrantLock, 745
- java.util.concurrent.locks.ReentrantReadWriteLock, 761–762
- java.util.concurrent.PriorityBlockingQueue, 770

- java.util.concurrent.ThreadPoolExecutor, 783
- java.util.Deque, 679
- java.util.Enumeration, 708
- java.util.EnumMap, 689
- java.util.EnumSet, 689
- java.util.EventObject, 324, 333
- java.util.GregorianCalendar, 120–121
- java.util.HashMap, 684
- java.util.HashSet, 671
- java.util.Hashtable, 709
- java.util.IdentityHashMap, 689
- java.util.Iterator, 658
- java.util.LinkedHashMap, 688
- java.util.LinkedList, 668
- java.util.List, 667, 698
- java.util.ListIterator, 667
- java.util.logging.ConsoleHandler, 590
- java.util.logging.FileHandler, 590
- java.util.logging.Filter, 591
- java.util.logging.Formatter, 591
- java.util.logging.Handler, 589–590
- java.util.logging.Logger, 588
- java.util.logging.LogRecord, 590–591
- java.util.Map, 684
- java.util.Map.Entry, 684
- java.util.NavigableMap, 699
- java.util.NavigableSet, 677–678, 698
- java.util.prefs.Preferences, 549–550
- java.util.PriorityQueue, 680
- java.util.Properties, 543
- java.util.Queue, 678–679
- java.util.Random, 152
- java.util.Scanner, 64–65, 71
- java.util.SortedMap, 685, 698
- java.util.SortedSet, 677, 698
- java.util.TreeMap, 685
- java.util.TreeSet, 672, 678
- java.util.Vector, 709
- java.util.WeakHashMap, 688
- javax.awt.Toolkit, 258
- javax.imageio.ImageIO, 321
- javax.jnlp.BasicService, 514
- javax.jnlp.FileContents, 515
- javax.jnlp.FileSaveService, 515
- javax.jnlp.ServiceManager, 514
- javax.swing.AbstractAction, 409
- javax.swing.AbstractButton, 392, 408–409, 410, 414
- javax.swing.Action, 348
- javax.swing.BorderFactory, 394–395
- javax.swing.border.LineBorder, 396
- javax.swing.border.SoftBevelBorder, 396
- javax.swing.ButtonGroup, 391
- javax.swing.ButtonModel, 392
- javax.swing.event package, 359
- javax.swing.event.MenuListener, 415
- javax.swing.filechooser package, 477
- javax.swing.filechooser.FileFilter, 476, 484
- javax.swing.filechooser.FileNameExtensionFilter, 484
- javax.swing.filechooser.FileView, 484
- javax.swing.GroupLayout, 444–445
- javax.swing.GroupLayout.Group, 445
- javax.swing.GroupLayout.ParallelGroup, 445
- javax.swing.GroupLayout.SequentialGroup, 445
- javax.swing.ImageIcon, 331
- javax.swing.JButton, 331, 474
- javax.swing.JCheckBox, 387
- javax.swing.JCheckBoxMenuItem, 410
- javax.swing.JColorChooser, 490–491
- javax.swing.JComboBox, 399
- javax.swing.JComponent, 299, 318, 348, 379, 396, 412, 424, 474
- javax.swing.JDialog, 467
- javax.swing.JFileChooser, 483–484
- javax.swing.JFrame, 298, 408
- javax.swing.JLabel, 380
- javax.swing.JMenu, 407
- javax.swing.JMenuItem, 408, 414, 415
- javax.swing.JOptionPane, 258, 460
- javax.swing.JPasswordField, 381
- javax.swing.JPopupMenu, 411
- javax.swing.JRadioButton, 391
- javax.swing.JRadioButtonMenuItem, 410
- javax.swing.JRootPane, 474
- javax.swing.JScrollPane, 384
- javax.swing.JSlider, 405–406
- javax.swing.JTextArea, 384
- javax.swing.JTextField, 379
- javax.swing.JToolBar, 424

- javax.swing.KeyStroke, 348
- javax.swing.SwingUtilities, 473
- javax.swing.SwingWorker, 806
- javax.swing.text.JTextComponent, 377
- javax.swing.Timer, 258
- javax.swing.UIManager, 338
- javax.swing.UIManager.LookAndFeelInfo, 338
- JButton class, 331, 368–369, 474
 - addActionListener method, 600
- JCheckBox class, 387
- JCheckBoxMenuItem class, 410
- JColorChooser class, 490–491
- JColorChooser dialog, 452, 485–491
- JComboBox class, 396, 399
- JComponent class, 295–297, 299, 318, 348, 368–369, 379, 392, 396, 412, 424, 474
 - input maps, 344–345
 - conditions, 345
 - Metrics method, 314
- jconsole, 595–596, 758–759
- JDialog class, 462, 467–468, 474
- JDK, *See* Java Development Kit (JDK)
- JFileChooser class, 479–484, 485
 - setFileFilter method, 476
- JFileChooser dialogs, 474–475
- JFrame class, 285–288, 294–295, 298, 369, 408
 - defined, 285
 - internal structure of, 295
 - methods, 288
- JLabel class, 380
- jmap utility, 596
- JMenu class, 407, 408
- JMenuBar class, 406
- JMenuItem class, 408–409, 414, 415
- Jmol applet, 8
- JNLP API, 507–514
- JOptionPane class, 258, 452, 453, 460–462
- JPanel class, 369
- JPasswordField class, 377, 381
- JPopupMenu class, 411
- JRadioButton class, 388, 391
- JRadioButtonMenuItem class, 410
- JRootPane class, 474
- JScrollPane class, 384
- JSlider class, 405–406

- addChangeListener method, 600
- JSmooth, 497
- JTextArea class, 377, 381, 384
- JTextComponent class, 377–379
- JTextField class, 377–379
- JToolBar class, 424
- JUnit (unit testing framework), 592
- JVM, *See* Java virtual machine (JVM)

K

- Keyboard mnemonics, 412–413
- KeyEvent, 359
- KeyListener interface, 360
- keyPressed method, 360
- keyReleased method, 360
- KeyStroke class, 348
- keyTyped method, 360
- Knuth, Donald, 86
- KOI-8, 43

L

- labeled break statement, 86–87
- Labels and labeling components, 379–380
- Langer, Angelika, 647
- Layout management, 368–377
 - border layout manager, 371–373
 - custom, 446–450
 - defined, 369
 - flow layout manager, 369
 - grid bag layout, 426–436
 - grid layout, 373–377
 - group layout, 436–445
 - omission of use of, 445–446
 - sophisticated, 424–452
 - traversal order, 450–452
- LayoutFocusTraversalPolicy, 452
- LayoutManager class, 446, 450
- Leading, 313
- Legacy code, calling, 625–626
- Legacy collections, 705–713
 - bit sets, 710–713
 - enumerations, 708–709
 - Hashtable class, 707–708
 - property maps, 709
 - stacks, 709–710
- Legacy container classes, 693
- Lexicographic order, 673

- Library source files, installing, 20
 - Lightweight collection wrappers, 694
 - Line2D.Double, 307
 - LineBorder class, 392
 - Linefeed, escape sequence, 42
 - LineMetrics class, 313, 317
 - Linked hash map, 685–687
 - access order, 687
 - Linked hash sets, 685–687
 - Linked lists, 659–668
 - defined, 660–661
 - doubly linked, 660
 - LinkedBlockingDeque, 769
 - LinkedBlockingQueue, 765–766
 - LinkedHashMap class, 686–687
 - as concrete collection type, 659
 - LinkedHashSet class, 686–687
 - as concrete collection type, 658
 - LinkedList class, 664
 - AbstractCollection superclass, 664
 - as concrete collection type, 658
 - listIterator method, 661
 - LinkedListTest.java, 665–666
 - List interface, 690–691, 698
 - List wrapper, 694
 - Listener interface, 324
 - Listener object, 324–325
 - listener.actionPerformed(event), 325
 - Listeners, containing a single method call, 334
 - ListIterator interface, 661, 691
 - listIterator method, LinkedList class, 661
 - LiveScript, *See* JavaScript
 - Load factor, 670
 - Local inner classes, 259, 266–267
 - accessing final variables from outer methods, 267–269
 - Locale specific formatting rules, 69
 - Localization, 579
 - lock method, 760
 - Lock objects, 742–745
 - Lock testing and timeouts, 760–761
 - lockInterruptibly method, 760
 - Log file pattern variables, 581
 - Logger, 588–589
 - Logging, 575–582
 - advanced, 576–577
 - basic, 575
 - filters, 583
 - formatters, 583
 - handlers, 579–582
 - localization, 579
 - log manager configuration, changing, 578
 - LoggingImageViewer.java, 584–588
 - recipe, 583–584
 - resource bundles, 579
 - Logging proxy, 592
 - LoggingImageViewer.java, 584–588
 - Logical font names, mapping to physical font names, 311
 - LogRecord class, 590–591
 - long type, 40
 - LONG_DESCRIPTION, 343
 - Loops, 76–83
 - break statements, 86–87
 - continue statement, 87–88
 - labeled break statement, 86–87
 - LotteryOdds.java, 83
 - Retirement2.java, 79–80
 - Retirement.java, 78–79
 - switch statement, 84–86
 - LotteryArray.java, 104
 - LotteryOdds.java, 83
 - Low-level events, 357–358
- M**
- main method, 122, 132–133, 135–138
 - body, 38
 - StaticTest.java, 136–138
 - Manager class, 173–174, 179, 253
 - getSalary method, 174–176
 - ManagerTest.java, 177–178
 - Map classes, specialized, 685
 - Map interface, 566, 689–690, 693
 - Maps, 680–685, 681–685
 - data structure, 680
 - implementations for, 681
 - MapTest.java, 682–683
 - property, 709
 - Marker interfaces, 252
 - Math class, 50
 - Mathematical functions and constants, 49–50

- Menu, 309
 - MouseListener, 415
 - Menus, 406–419
 - accelerators, 413–414
 - building, 406–407
 - enabling/disabling menu items, 414–415
 - keyboard mnemonics, 412–413
 - menu bar, 406–407
 - menu items, 406
 - icons in, 408–409
 - MenuTest.java, 416–419
 - pop-up, 410–412
 - submenus, 406
 - MenuTest.java, 416–419
 - menuText, 309
 - Merge sort algorithm, 702
 - Method parameters, 138–144
 - ParamTest.java, 142–144
 - Method pointers, 234–235
 - MethodPointerTest.java, 236–237
 - Method table, 182
 - Methods, 38–39, 107
 - body, 38
 - signature, 145
 - syntax, 38
 - with a variable number of parameters, 214–215
 - Metrics method, JComponent class, 314
 - Micro Edition (ME), 17
 - mnemonic property, ButtonModel interface, 367
 - MNEMONIC_KEY, 343
 - Modal dialog box, 452, 463
 - Model-view-controller design pattern, 362–368
 - analysis of Swing buttons, 367–368
 - ButtonModel interface, properties of, 367
 - classes, 364
 - defined, 364
 - separate views of same model, 365
 - Modeless dialog box, 452, 463
 - Modulus, 46
 - Monitor concept, 755–756
 - Mosaic, 9–10
 - Mouse events, 349–357
 - mouse test program, 349–350
 - mouseClicked method, 349–350, 352, 360
 - mouseDragged method, 352, 360
 - mouseEntered method, 360
 - MouseEvent class, 356, 359, 411
 - mouseExited method, 352, 360
 - MouseHandler class, 352
 - MouseListener interface, 351, 360
 - MouseMotionHandler class, 352
 - MouseMotionListener interface, 351–352, 360
 - mouseMoved method, 352, 360
 - mousePressed method, 349–350, 360
 - mouseReleased method, 349, 360
 - MouseTest.java, 353–356
 - MouseWheelEvent method, 359
 - MouseWheelListener interface, 360
 - mouseWheelMoved method, 360
 - MOVE_CURSOR, 351
 - Multidimensional arrays, 99–102
 - CompoundInterest.java, 101–102
 - Multiple inheritance, 179, 249
 - Multitasking, 716
 - Multithreading, 715–807
 - BallComponent.java, 721–722
 - Ball.java, 720–721
 - Bounce.java, 718–720
 - BounceThread.java, 725–727
 - defined, 716
- N**
- NAME, 343
 - Name control, 259
 - Naughton, Patrick, 9–10
 - NavigableMap interface, 691, 699
 - NavigableSet interface, 691, 695, 698
 - NE_RESIZE_CURSOR, 351
 - Nested classes, 259
 - NetBeans, 17, 591
 - NetScape, 9–10
 - New threads, 731
 - newCachedThreadPool method, Executors class, 779
 - newCondition method, 746
 - newFixedThreadPool method, Executors class, 779
 - newInstance method, 219
 - newProxyInstance method of the, Proxy class, 276

- newScheduledThreadPool method, Executors class, 779, 783
- newSingleThreadExecutor method, Executors class, 779
- newSingleThreadScheduledExecutor method, Executors class, 779, 783
- nextElement method, Enumeration interface, 708–709
- NoSuchElementException, 653
- NotHelloWorld.java, 297–298
- notify method, Object class, 751
- notifyAll method, Object class, 751
- N_RESIZE_CURSOR, 351
- NumberFormat, 135
- Numeric types, conversions between, 51
- NW_RESIZE_CURSOR, 351
- O**
- Oak language, 9
- Object class, 108, 192, 198, 203, 672, 753–754
 - equals method in, 193–194
- Object cloning, 249–255
- Object comparison, 673–678
- Object-oriented programming (OOP), 3–4, 106–107, 114, 133, 152, 172–173, 242, 256, 299
- Object wrappers, 211–213
- ObjectAnalyzerTest.java, 228–230
- Objects, 105–169
 - characteristics of, 108
 - construction, 144–153
 - overloading, 144–145
 - destruction, 152–153
 - finalize method, 152–153
- Octal integer, conversion character, 66
- offer method, 765
- Opaque panel, 297
- Operator hierarchy, 52–53
- Operators:
 - arithmetic, 46–47
 - bitwise, 49
 - casts, 51–52
 - conversions between numeric types, 51
 - increment/decrement, 47–48
 - mathematical functions and constants, 49–50
 - parentheses and operator hierarchy, 52–53
 - relational and boolean, 48–49
- Option dialogs, 453–462
 - OptionDialogTest.java, 455–460
- Optional operations, 697–699
- Ordered collection, 660, 690
- Output, formatting, 65–69
- Overloading, defined, 144
- Overloading resolution, 144, 181
- P**
- Packages, 153–159
 - addition of a class into, 155–156
 - class importation, 153–154
 - default, 155–156
 - Employee.java, 157–158
 - PackageTest.java, 157
 - scope, 158–159
 - static imports, 155
- paintComponent method, 295–296, 357, 763, 806
- Pair class, 616–622
- PairTest1.java, 617
- PairTest2.java, 620–621
- PairTest3.java, 637–640
- ParameterizedType interface, 642, 646–647
- Parameters, 39
- ParamTest.java, 142–144
- Password fields, 381
- Payne, Jonathan, 10
- peek method, 765
- Percent symbol, conversion character, 66
- PersistenceService class, 509, 515
- PersonTest.java, 189–191
- Physical limitations, 553
- PlafFrame class, 335
- PlafTest.java, 336–337
- Platform-dependent line separator, conversion character, 66
- Point2D class, 302
- Point2D.Double, 307
- poll method, 765
- Polymorphism, 179–180, 239

Pop-up menus, 410–412
Pop-up trigger, 411
Positioning frames, 288–294
PostScript Type 1 format, 312
Preconditions, 574
Predefined action table names, 343
Predefined classes, 111–121
 CalendarTest.java, 118–120
 GregorianCalendar class, 114–115
 mutator and accessor methods,
 115–118
 objects and object variables, 111–114
Preferences class, 544–546, 549–550
pressed property, ButtonModel interface,
 367
Primitive types, 40, 192
printf:
 conversions for (table), 66
 flags for, 67
 parameters, 214–215
printStackTrace method, Throwable class,
 220, 566, 593
PrintWriter, 71
Priorities, threads, 733–734
Priority queues, 679–680
 PriorityQueueTest.java, 680
PriorityBlockingQueue, 766, 770
PriorityQueue, as concrete collection type,
 659
Procedural vs. OOP programming,
 106–107
Producer threads, 764
Profiler, 596
Properties class, 707
Properties, frames, 290
PropertiesTest.java, 541–543
Protected fields, 238
Proxies, 275–280
 defined, 275
 proxy class methods, 275
 proxy mechanism, 275
Proxy class:
 newProxyInstance method of, 276
 ProxyTest.java, 278–279
Proxy classes, properties of, 279–280
public keyword, 242–243

Pure virtual function, 188
put method, 765
putIfAbsent method, 771
putValue method, 342

Q

Queue interface, defined, 650
Queues and dequeues, 650, 678–679
 priority queues, 679–680

R

Radio buttons, 388–392
 event notification mechanism for, 388
 JRadioButton, 388
 menu items, 409–410
 RadioButtonTest.java, 389–391
Ragged arrays, 102–104
 LotteryArray.java, 104
RandomAccess interface, 691
Rational Rose (software), 110
Raw array lists, compatibility between
 typed array lists and, 210–211
Read/write locks, 761–762
Rectangle2D class, 300–301
 Double, 306
 Float, 306
 methods, 301
RectangularShape class, 302, 306
Red-black tree, 672
Reentrant lock, defined, 744
ReentrantLock class, 742, 745
ReentrantReadWriteLock class, 761–762
Reflection, 171, 217–237
 Class class, 217–218
 and generics, 640–647
 ObjectAnalyzerTest.java, 228–230
 overuse of, 239
 ReflectionTest.java, 222–224, 264–265
 using to analyze objects at runtime,
 226–230
 using to analyze the capabilities of
 classes, 221–226
 using to write generic array code,
 231–234
Reflection library, 217
Rehashing a table, 670

- Relational operators, 48–49
- Relative resource name, 498
- remove method, 765
 - Iterator interface, 655–656
- removeItem method, 397
- removeItemAt method, 397
- repaint method, 296, 717, 756
- Resource bundles, 579
- ResourceBundle class, 579
- Resources:
 - absolute resource name, 498
 - defined, 497
 - internationalization API, 498–499
 - and JAR files, 497–498
 - relative resource name, 498
 - ResourceTest.java, 499–500
- Retirement2.java, 79–80
- Retirement.java, 78–79
- Retroweaver program, 621
- Robot class, 603, 606–607
- RobotTest.java, 604–606
- rollover property, ButtonModel interface, 367
- Rounding mode, 90
- run method, TransferRunnable class, 759
- Runnable threads, 731
- RuntimeException, 553–555

- S**
- Scanner class, 64–65, 71
- Scheduled execution, 783–784
- ScheduledExecutorService interface, 783–784
- Scroll panes, 382–384
- scrollbar, 309
- Sealing, 500–501
- Secure certificate, 506
- selected property, ButtonModel interface, 367
- Self-signed certificate, 507
- Semantic events, 357–358
- Semaphore class, 786
- Semaphores, 786
- SE_RESIZE_CURSOR, 351
- Serialized applet object, 522
- ServiceManager interface, 514
- Set classes, specialized, 685–689
- Set interface, 691
- setAccelerator method, 413
- setBackground method, 592
 - Component class, 308
- setBorder method, JComponent class, 392
- setBounds method:
 - Component class, 289–290
 - JFrame, 288–289
- setDebugGraphicsOptions method, JComponent class, 594–595
- setDefaultUncaughtExceptionHandler method, Thread class, 735
- setEditable method, JComboBox class, 396
- setFileFilter method, JFileChooser class, 476
- setFileView method, 478
- setIconImage method, JFrame, 288
- setJMenuBar method, 406
- setLocation method:
 - Component class, 290
 - JFrame, 288
- setMultiSelectionEnabled method, 475
- setPaint method, Graphics2D class, 307
- setResizable method, JFrame, 288
- Sets, 680
- setSelectedFile method, 475
- SetTest.java, 671
- setTitle method, JFrame, 288
- Shallow copy, 250–251
- Shape classes, relationships between, 302–303
- Shell window, running programs in, 24–25
- short type, 40
- SHORT_DESCRIPTION, 343
- Show Java Console box, 597
- showConfirmDialog method, 454
- showDialog method, 468, 475
- showDocument method, 532
- showInputDialog method, 454
- showMessageDialog method, 454
- showOpenDialog method, 474–475
- showOptionDialog method, 454
- showSaveDialog method, 474–475
- showStatus method, 531–532
- ShuffleTest.java, 703
- shutdownNow method, 784

- “Sieve of Eratosthenes” benchmark, 711–713
- Sieve.cpp, 712–713
- Sieve.java, 711–712
- signal method, 748, 759–761
- signalAll method, 747–748, 759–761
- Signed code, 505–507
- Signed JAR file, 505–506
- Simple algorithms, 705–706
- Simple Java program, 36–39
- SimpleFrameTest class, 286–287
- SimpleFrameTest.java, 286
- Single quote, escape sequence, 43
- Single-thread rule for Swing
 - programming, 795–796
- SizedFrameTest.java, 292
- Sliders, 400–406
 - constructors, 400
 - SliderTest.java, 402–405
- SMALL_ICON, 343
- SoftBevelBorder class, 392
- Software Development Kit (SDK), 17
- SortedMap interface, 691, 699
- SortedSet interface, 691, 695, 698
- Sorting, arrays, 95–97
- Special characters, escape sequences, 42–43
- Special fonts for text, 310–318
- Special syntax rules for inner classes, 263–264
- sqrt method, 49–50
- src directory, 21
- src.zip file, 20
- S_RESIZE_CURSOR, 351
- Stack interface, 650
- Stack trace, 758
 - defined, 566
 - StackTraceElement class, 566
 - StackTraceTest.java, 567
- Stacks, 709–710
- StackTraceElement, 568
- Standard Edition (SE), 17
- Standard Template Library (STL), 650, 654
- State, objects, 107, 108
- Static constants, 133–134
- Static fields, 132–133, 135
- Static inner classes, 259, 271–275
 - StaticInnerClassTest.java, 273–275
- static keyword, 135
- Static methods, 134–135
- static void, 38
- StaticTest.java, 136–138
- stop method, 762–764
- StreamHandler class, 582
- StrictMath class, 50
- String class, 57–59
- String type, conversion character, 66
- StringBuilder class, 62–63
- Strings, 53–63
 - building, 62–63
 - code points/code units, 56–57
 - concatenation, 54
 - defined, 53
 - file input and output, 69–71
 - formatting output, 65–69
 - immutability of, 54–55
 - InputTest.java, 64
 - String class, 57–59
 - substrings, 54
 - testing for equality, 56
- Strongly typed language, defined, 40
- Subclasses, 171, 173–174
- sublist method, 694
- Submenus, 406
- Subranges, views, 694–695
- Substrings, 54
- Suffix, numbers of float type, 41
- super keyword, 175
- Superclasses, 171, 174–175
- Supplementary characters, 43
- Surrogates area, 43
- suspend method, 762–764
- suspendRequested variable, 763–764
- Swing, 282–285
 - graphics debugger, 594
 - and the model-view-controller design pattern, 362–368
 - and threads, 794–807
 - running time-consuming tasks, 795–796
 - single-thread rule, 806–807
 - SwingThreadTest.java, 796–799
 - SwingWorkerTest.java, 802–805
 - user interface components with, 361–492

- SwingConstants interface, 380
 - SwingThreadTest.java, 796–799
 - SwingUtilities class, 473
 - SwingUtilities.updateComponentTreeUI, 335
 - switch statement, 84–86
 - SW_RESIZE_CURSOR, 351
 - SynchBankTest program, 759
 - synchronized keyword, 750–754
 - Bank.java, 752–753
 - Synchronization, 736–764
 - Bank.java, 738–739
 - condition objects, 745–750
 - deadlocks, 757–760
 - lock objects, 742–745
 - lock testing and timeouts, 760–761
 - monitor concept, 755–756
 - race condition:
 - defined, 736
 - example of, 736–740
 - explained, 740–742
 - read/write locks, 761–762
 - stop method, 762–764
 - suspend method, 762–764
 - synchronized keyword, 750–754
 - synchronized blocks, 754–755
 - TransferRunnable.java, 740
 - UnsynchBankTest.java, 738
 - volatile fields, 756–757
 - Synchronization primitive, 786
 - Synchronization wrapper, 773
 - Synchronized blocks, 754–755
 - client-side locking, 754
 - Synchronized views, 696
 - synchronizedMap method, Collections class, 696
 - Synchronizers, 785–793
 - barriers, 787–788
 - countdown latches, 786–787
 - exchangers, 787
 - semaphores, 786
 - synchronous queues, 787–788
 - SynchronousQueue class, 786, 787
 - System class, 65, 71, 99, 543–544, 689
 - System colors, 308
 - System of Patterns, A* (Buschmann et al.), 363
 - SystemColor class, 308
 - System.out object, 38
- T**
- Tab, escape sequence, 42
 - Tagging interfaces, 252
 - take method, 765
 - TalkingClock class, 260
 - TalkingClock\$1TimePrinter class, 267–268
 - Template code bloat, 622
 - text, 309
 - Text areas, 381–382
 - Text input, 377–384
 - labels and labeling components, 379–380
 - password fields, 381
 - scroll panes, 382–384
 - text areas, 381–382
 - text fields, 378–379
 - TextComponentTest.java, 382–384
 - Text, special fonts for, 310–318
 - TextComponentTest.java, 382–384
 - TEXT_CURSOR, 351
 - textHighlight, 309
 - textHighlightText, 309
 - textInactiveText, 309
 - textText, 309
 - Thread class, 727, 730, 733–736
 - setDefaultUncaughtExceptionHandler method, 735
 - Thread group, defined, 735
 - Thread pools, 779–784
 - purpose of, 778
 - ThreadPoolExecutor class, 780
 - ThreadPoolTest.java, 780–782
 - Thread priorities, 733–734
 - system dependence, 733
 - Thread-safe collections, 771–774
 - copy on write arrays, 773
 - efficient maps, sets, and queues, 771–773
 - synchronization wrapper, 773
 - Thread states, 730–733
 - blocked threads, 731–732
 - new threads, 731
 - runnable threads, 731

terminated threads, 733
 timed waiting threads, 731–732
 waiting threads, 731–732
 ThreadDeath error, 733
 ThreadDeath objects, 736, 763
 ThreadGroup class, 735–736, 736
 ThreadPoolExecutor class, 779, 780, 783
 Threads, 7, 716–722
 daemon threads, 734
 defined, 716
 handlers for uncaught exceptions,
 735–736
 interrupting, 728–730
 priorities, 733–734
 properties, 733–736
 and Swing, 794–807
 running time-consuming tasks,
 795–796
 single-thread rule, 806–807
 SwingThreadTest.java, 796–799
 SwingWorker class, 799–806
 SwingWorkerTest.java, 802–805
 termination of, 728
 thread states, 730–733
 using to give other tasks a chance,
 722–724
 Thread.UncaughtExceptionHandler, 736
 Throwable class, 221, 553, 559, 567–568
 printStackTrace method, 220, 566, 593
 Throwing an exception, 557–558
 use of term, 219
 Timed waiting threads, 731–732
Timeless Way of Building, The (Alexander),
 362
 TimeoutException, 774
 TimePrinter class, 260–261, 265, 267
 Timer class, 255–257
 constructor, 256
 TimerTest.java, 257–258
 TimeUnit, 760
 Together (software), 110
 Tomcat, 501
 Toolbars, 419–421
 defined, 419
 detached, 420
 dragging, 420

 programming, 421
 ToolBarTest.java, 422–424
 Toolkit-modal dialog, 463
 Tooltips, 421–422
 toString method, 200–203
 and Formattable interface, 66
 Total ordering, 675
 TransferRunnable class, run method, 759
 TransferRunnable.java, 740
 Traversal order, 450–452
 Tree sets, adding elements into, 672
 TreeMap class, 681, 691
 as concrete collection type, 659
 TreeSet class, 691
 addAll method, 674
 as concrete collection type, 658
 TreeSetTest.java, 675–677
 TrueType, 312
 tryLock method, 760
 Two-dimensional array, 102
 2D shapes, 299–307
 complexity of, 300
 DrawTest.java, 304–306
 Type variables, bounds for, 619–621
 Typesetting terms, 313
 TypeVariable interface, 642, 646

U

UIManager.setLookAndFeel method, 335
 UML (Unified Modeling Language)
 notation, 109–110
 Uncaught exceptions, handlers for,
 735–736
 UncaughtExceptionHandler, 736
 Unchecked exceptions, 219, 554
 Unicode, 43–44
 Unmodifiable views, 695–696
 UnsupportedOperationException, 694, 697
 UnsynchronBankTest.java, 738
 Update (u), 17

V

val\$beep instance variable,
 TalkingClock\$1TimePrinter class,
 268
 Variables, initializing, 45–46

Vector class, 691, 708–709, 755
Vector interface, 650
Views, 693–699

- checked, 696–697
- defined, 693–694
- subranges, 694–695
- synchronized, 696
- unmodifiable, 695–696

Violet (software), 110
Virtual machines, 9
Visual Basic, 3–4
Volatile fields, 756–757
volatile keyword, 756–757

W

wait method, Object class, 751
WAIT_CURSOR, 351
Waiting threads, 731–732
Weak references, defined, 685
WeakHashMap class, 685

- as concrete collection type, 659

WeakReference object, 685
WebStartCalculator application, 510
WebStartCalculator.java, 511–514
WelcomeApplet.html, 32–33
WelcomeApplet.java, 33–34
Welcome.java, 23–24
WHEN_ANCESTOR_OF_FOCUSED_COMPONENT,
345–346
WHEN_FOCUSED, 345
WHEN_IN_FOCUSED_WINDOW, 345
while loop, 76–77
White Paper buzzwords, Java, 2–3

Wildcard types, 632–640
Wildcards:

- capture, 636–637
- PairTest3.java, 637–640
- subtype relationships with, 633
- supertype bounds for, 634–636
- unbounded, 636

WildcardType interface, 642
window, 308
windowActivated method, 360
WindowAdapter class, 339–341
windowBorder method, 308
windowClosed method, 360
windowClosing method, 338–339, 360
WindowEvent, 341, 359

- objects, 324

WindowFocusListener interface, 360
windowGainedFocus method, 360
windowIconified method, 360
WindowListener, 341
WindowListener interface, 338, 360
windowLostFocus method, 360
windowOpened method, 360
windowStateChanged method, 360
WindowStateListener interface, 341, 360
windowText, 309
Wirth, Niklaus, 5, 9, 106
Wrappers, 694

- defined, 211

W_RESIZE_CURSOR, 351

X

XML, 11