

## Index

Classes are listed by name, followed by the package in which they belong in parentheses, e.g. `HashMap (java.util)`. Methods are indicated by name followed by parentheses with no argument list, e.g. `println()`.

- `+`, *see* strings, *see* operators
- `@`, *see* annotation
- `\u`, escape sequence, 313
- `.` (dot operator), 67
- `.class` file, 15
- `.java` file, 14, 133
- `NaN`, infinity, *see* floating-point special values
- `<>`, 339
- `()`, 45
- abstract
  - class, 105–9
  - methods, 105–9
  - abstract modifier, 107–9
- access rules, 61, 140
  - default (package private), 140
  - see* packages
  - private, 140
  - protected, 140
  - public, 140
- Accessibility Framework, 163, 394
- `ActionListener`, *see* event listeners
- adapter classes, 240–2
  - see* event listeners
- `AffineTransformOp`
  - (`java.awt.image`), 381, 383
  - see* image filtering
  - see* Java 2D – transformations
- aliased, anti-aliasing, 185
- animations, 276–81
  - of images
    - with `BufferedImage`, 380
    - with `MemoryImageSource`, 376–8
  - see* threads
  - see* timers
- anonymous inner class, 239–40
- annotation, 25, 340
  - `@Override`, 98–9
- Ant tool, 521
- AOT (Ahead of Time) compilers, 658
- applets, 4, 15
  - client example, 459–64
  - directories, 148–9
  - implementing `Runnable`, 257, 277
  - `init()`, 50, 259
  - see* JAR
  - reading a file, 419–23
  - “Hello World” example, 19–20
  - hypertext tags, 19–20, 75
  - menu bar, 247
  - parameters, 75
  - security, 427
  - see* Swing classes – `JApplet`
  - `start()`, 259, 277
  - starter programs, 20–3
  - `stop()`, 259, 284
- appletviewer, 15, 16
  - vs. browsers, 22–3
- application programs, 4, 14, 31–2
  - see* frames,
  - “Hello World” example, 18
  - see* JAR
  - starter programs, 20–3
- arbitrary precision numbers, 356–9
  - `BigDecimal (java.math)`, 357, 359
  - `BigInteger (java.math)`, 357–8
- ARGB, *see* color
- array, 42, 79–80, 120–4, 328
  - see* Arrays
  - as object, 79, 122–3
  - copy with `System.arraycopy()`, 121
  - declaration, 79
  - default values, 79
  - formatted string, 25, 347–8
  - identifier, 31
  - length, 80
  - multi-dimensional, 121–2
  - of objects, 120–1
  - rectangular, 11
  - vectors and matrices, 123–4
- `ArrayList (java.util)`, 26, 118, 123, 336, 337, 338, 339
  - performance, 405–8, 409
- `Arrays (java.util)`, 328, 345–9
  - `binarySearch()`, 346–7
  - `equals()`, 345
  - `fill()`, 345
  - `sort()`, 346–7
  - `deepEquals()`, 348–9
  - `deepToString()`, 348–9
  - `toString()`, 347–8
- ASCII, 312
- atomic components, 161, 164
- audio, 402–4
- autoboxing and unboxing, 24, 76–9, 343
  - with generics, 339
  - and overloading, 78–9
  - see* wrappers for primitives
- AWT (Abstract Windowing Toolkit), 25, 161–2
- bands in image data buffer, 366
- `BasicStroke (java.awt)`, 184
  - see* Java 2D
- base class, 93
- big endian, 316, 648, 649, 667
- bit handling, 360–2
  - operations, 360–1, 689
  - `BitSet (java.util)`, 328, 361

- boolean, 30, 34
  - see* primitives
- Boolean (java.lang), *see* wrappers
  - for primitives
- Box (javax.swing), 226–8
  - see* layout managers
- break, 43, 44
- browsers, 15
  - see* applets
  - see* plug-in JVM
  - vs. appletviewer, 22–3
- BufferedImage (java.awt.image),
  - 190, 365–7, 386
  - animations, 380
  - create, 372–3
  - see* Image
  - see* images
  - loading, 370
  - pixel handling, 378–80
  - save, 373
- BufferedImageOp
  - (java.awt.image), 381
  - see* image filtering
- byte, 34
  - see* integers
  - see* primitives
- Byte (java.lang), *see* wrappers for
  - primitives
- bytecode, 3, 10, 15
  - verifier, 9
- C/C++
  - arrays, 11, 120, 123
  - 2D arrays, 124
  - and compute engines, 486
  - CORBA, 503
  - CYGWIN, 595
  - enumerated types, 343
  - include vs. import, 140
  - I/O, 289
  - Java GUI front-end, 10
  - JNI, 585–624
  - main(), 32
  - multiple inheritance, 93
  - operator precedence, 38
  - operator overloading, 11
  - performance, 9, 12
  - pointers vs. Java references, 66, 112
  - printf() function, 25, 52, 594
  - structures, 57
  - vs. Java, 3, 9, 10
- Calendar (java.util), 327, 353–6
  - GregorianCalendar
    - (java.util), 353
- callbacks, 192, 256, 490, 649
  - see* Interfaces
- cast
  - automatic conversions, 45, 116
  - container Object elements, 328
  - legal vs. illegal casts, 45
  - object references, 115–17
  - operator (parentheses), 45, 691
  - primitives, 44–7
- cast rules
  - integer to floating-point, 46
  - narrowing, 46
  - object references, 117
  - widening, 45
- CGI (Common Gateway Interface), 446
- ChangeListener, 221
  - see* event listeners
- char, 30, 33–4, 296
  - as unsigned short, 34
  - see* primitives
- Character (java.lang), *see*
  - wrappers for primitives
- character encoding, 34, 290, 312–13
  - ASCII, 312
  - see* Unicode
  - UTF-8, 312
- class, 57
  - " ." dot operator, 67
  - accessing methods and fields, 67
  - see* access rules
  - see* constructors
  - data type analogy, 57–8
  - definition, 58–65
  - definition files, 132–3
  - fields, 58, 59–60
  - see* inheritance
  - instantiation, 65–6
  - loader, 9, 191
  - see* methods
  - vs. structures, 57
  - see* variables
- class variables and methods, *see* static
- Class class, 119
- CLASSPATH, 134, 147–8, 511, 512, 513, 515, 524–5, 596
- client/server, 428–30, 471
  - applications in science, 471–3
  - callback, 490
  - conceptual model in UML, 475–7
  - see* CORBA
  - see* embedded Java
  - see* microservers
  - polling, 489
  - see* RMI
  - with sockets, 428–9, 431–45, 448–69
  - see* Web services
- cloning, 115, 118
- code block, 39
- coding conventions, 150–3
- collaboration diagrams, 475, 496
  - asynchronous operations (half arrowheads), 487
  - iteration notation, 480
  - messages, 477
  - nested sequences, 477
  - return messages (dashed arrows), 485
  - sequence numbers, 477
  - server example, 477–81, 483–7
  - synchronous operations (solid arrowheads), 478
  - see* UML
- Collections Framework, 6, 329, 331, 335–8
  - lists, maps, and sets, 336–7
  - object containers, 335
- color
  - ARGB (alpha-red-green-blue), 176, 366, 374–5, 378
  - background, foreground colors, 177
  - CIEXYZ, 176
  - color space, 176
  - ColorModel (java.awt.image), 366
  - see* graphics
  - see* gray scale
  - see* images
  - indexed color model, 367
  - pen color, 177, 183
  - RGB, 176, 366
  - sRGB, 176
- Color (java.awt), 176–8, 365
  - packed bits, 176, 360, 375–6
  - packing/unpacking ARGB values, 360–1, 374–5
  - constants, 177

ColorConvertOP  
     (java.awt.image), 381, 386  
     *see* image filtering  
 comments, 32–3  
     *see* Javadoc API  
 Comparable (java.lang), 346, 358  
 Comparator (java.util), 346  
 complex numbers, 11  
     class examples, 58, 85–6, 125–6  
 Component (java.awt), 161  
 compute engine, 482  
 concurrency, 26  
     concurrency utilities, 285, 341–3  
     Callable  
         (java.util.concurrent),  
         342–3  
     Executor  
         (java.util.concurrent),  
         341–2  
     *see* threads  
 console, 50  
     *see* I/O with console  
 constants, 140–1  
     *see* final  
     *and* interfaces, 114–15  
     *see* static import  
 constructors, 58, 63–5, 100–5  
     *see* class  
     construction sequence, 102–5  
     default, 64, 103  
     super(), 101–2  
     this(), 100–1  
 containers for graphics components,  
     164–6  
     Container (java.awt), 161  
     *see* Swing classes – JPanel  
     panes, 165–6, 179  
     getContentPane(), 166  
 containers of objects, 335  
     *see* Collections Framework  
 ConvolveOP (java.awt.image),  
     381, 383  
     *see* image filtering  
 CORBA (Common Object Request  
     Broker Architecture), 136, 137,  
     471, 479, 530–48  
     accessor (getter), 534  
     attributes, 534  
     C/C++, 503  
     client implementation, 544–5

constants, 534  
 CosNaming service, 479, 542, 543–4  
 exceptions, 534  
 IDL (Interface Definition Language),  
     530, 531, 535  
 idlj and IDL compilation, 535–7,  
     546  
 mutator (setter), 534  
 Object Management Group, 530  
 Object Request Broker (ORB), 540–2,  
     544  
 operations (methods), 533  
 orbd, 545  
 Portable Object Adapter (POA), 540,  
     542  
 run example on two machines, 547–8  
 servant, 538–40, 542–4  
 server implementation, 537–44  
 structures, 533  
 vs. RMI, 503, 530–2, 535  
 Web services, 573–4  
 cursor icons, 392–4  
  
 data fields, *see* class  
 data race, *see* synchronization  
 data types, 30, 33–4  
     *see* cast  
     *see* mixing  
     *see* primitives  
     *see* references  
 DataBuffer (java.awt.image), 366  
     *see* BufferedImage  
     *see* Raster  
 datagrams, 414–15  
 DatagramSocket (java.net), 426  
 Date (java.util), 327, 353–6  
 date and time formatting  
     clock examples, 282–3, 284–5  
     DateFormat (java.text), 353–6  
     SimpleDateFormat (java.text),  
     353–6  
 default package, *see* packages  
 deprecation, 258  
 design patterns, 471, 476  
     factory pattern, 341, 477, 496, 551–2  
         ServerFactory, 477–81  
         *see* RMI Client/Server example  
     *see* Model-View-Controller (MVC)  
     pattern  
     singleton, 479

desktop client, 24  
 device drivers, 656  
     *see* real-time  
 dialogs  
     Dialog (java.awt), 161  
     JDialog (javax.swing),  
     163  
 Dimension (java.awt), 175  
 direct memory access, 656  
     *see* real-time  
 directory separators, 135, 300–1,  
     516  
 distributed computing, 471–3, 501–3  
     client, 492–500  
     *see* design patterns  
     *see* RMI Client/Server example  
     scientific applications, 471–3  
     simple server example, 481–90  
     simulation compute engine, 483–7  
     *see* Web services  
 domain name servers (DNS), 415  
 double, 34  
     *see* floating-point  
     *see* primitives  
 Double (java.lang)  
     *see* wrappers for primitives  
 double buffering, 372  
 drag and drop, 163  
 dynamic binding, 9, 146  
  
 Ease of Development (EoD), 23, 24–6  
 embedded Java, 651–80  
     AOT (Ahead of Time) compilers,  
     658  
     EmbeddedJava API, 7, 652  
     embedded server example, 671–9  
     *see* Java Card API  
     *see* Java 2 Micro Edition (J2ME)  
     *see* microservers  
     PersonalJava API, 7, 652  
     science and engineering applications,  
     652  
 enhanced for loop, 24, 42  
 enumerated type, 26, 343–5  
 Enumeration (java.util), 327, 329,  
     337, 409  
 environment variables, 628–9  
     *see* CLASSPATH  
     path, 147  
 equals(), *see* Object class

- events, 201–10
  - button, 204–6
  - change, 221–3
  - item, 215
  - key, 399–402
  - mouse, 208–10
  - mouse buttons, 394–6
- event listeners, 202–3
  - adapter classes, 240–2
  - inner classes, 238–40
  - ActionListener
    - (java.awt.event), 202–3, 204–6, 216–18, 284
  - ChangeListener
    - (java.awt.event), 221
  - ItemListener
    - (java.awt.event), 215
  - KeyListener (java.awt.event), 399
  - ListSelectionListener
    - (javax.swing.event), 218–21
  - MouseListener
    - (java.awt.event), 202, 209–10, 399
  - MouseMotionListener
    - (java.awt.event), 202, 209
  - WindowListener
    - (java.awt.event), 202
- EventObject (java.util), 203
- exceptions, 9, 80–4
  - finally, 84
  - general (checked) exceptions, 82–3
  - run-time, 82, 83–4
  - throw statement, 82
  - throws statement, 82
  - try-catch syntax, 81
- expressions, 31, 35–6
  - associativity, 38–9
  - evaluation, 37–9
  - precedence, 38–9
- extends, 91
  - see* inheritance
- fields, *see* class
- File (java.io), 294, 298, 299–301
  - for a directory, 300
  - see* I/O with files
  - obtaining URL, 419
  - path names, 300–1
- FileFilter, *see* JFileChooser
- final modifier, 69, 140–1, 405
- finalize(), 119
  - see* Object
- float, 34
  - see* floating-point
  - see* primitives
- Float (java.lang)
  - see* wrappers for primitives
- floating-point, 12, 34, 47–50, 693–6
  - bit layout, 48
  - double, 30, 33–4
  - example, 52–3
  - extended exponents, 695
  - float, 30, 33–4, 57
  - see* mixing primitives
  - normalized and denormalized, 49, 693–4
  - specifications, 695
  - strictfp, 12, 50, 695
- floating-point special values, 694–5
  - NaN (Not-a-Number), 49, 53
  - infinity, 49, 53
- flush buffer, *see* streams
- focus, 400
- fonts for graphics
  - Font (java.awt), 181
  - FontMetrics (java.awt), 181–2
- for loop, 41
- formatted I/O, 25, 52, 153–7
  - arrays, 25, 347–8
  - DecimalFormat (java.text), 153–5
  - Format (java.text), 153–5
  - Formatter (java.util), 156, 293–5, 304–5
  - NumberFormat (java.text), 153–5
  - printf() in PrintStream, 25, 156–7, 289, 293–5
  - see* Scanner
- Fortran, 10, 11, 12, 85, 486, 487
- JNI, 587
- FPGA (Field Programmable Gate Array), 657
  - see* Java hardware processors
- frames,
  - centering, 629
  - closing, 243
  - menu bar, 244
  - Frame (java.awt), 161
  - JFrame (javax.swing), 163
- garbage collection, 9, 71, 357, 408, 502
  - and Javelin Stamp, 665
  - real-time systems, 655–6
  - System.gc(), 655
- generics, 26, 78, 338–40
  - autoboxing and unboxing, 339
  - compilation warnings, 340
- getContentPane(), 166
- getter methods, 153, 224
- glue, 227
- graphics
  - color, 176–8
  - coordinate system, 175–6
  - drawing, 174–6, 178–90
  - drawing example, 178–9
  - see* fonts for graphics
  - graphics context, 174–5
  - see* Java 2D API
  - rendering, 176
  - text, 171–4, 180–3
- Graphics (java.awt), 174–5, 178–83, 372
  - drawing an image, 191, 371
  - for printing, 390
- Graphics2D (java.awt), 174–5, 371–2
  - see* Java 2D API
- gray scale color model, 367, 379–80, 386
- GregorianCalendar (java.util), *see* Calendar
- GUI – Graphical User Interface, 10, 160, 201, 234, 242, 251
  - simple example, 206–8
- GUI builders, 15, 234
- HashMap (java.util), 327, 331, 336, 409
  - server pool example, 553
- HashSet (java.util), 336
- Hashtable (java.util), 327, 329–31, 335, 336, 409
- heavyweight, *see* lightweight components
- histograms
  - class examples, 86–9, 128–30, 194–9
  - graphics, 192–9, 247–51

- I/O, 320–5
  - destination, 321–3
  - filtering, 323–5
  - serialization, 320–1
- RMI example, 569–70
- Web client example, 459
- hostname, 415
- HotJava, 4
- Hotspot JVM, 6
- HTTP (Hypertext Transmission Protocol), 431, 432, 435–8
- indexed color model, 367
- IDE (Integrated Development Environment), 15
- identifiers, 30
  - see* literals
  - see* variables
- if statements, 40–1
  - see* statements
- Image (`java.awt.image`), 190, 365–7
  - see* `BufferedImage`
  - see* images
- image filtering, 380–8
  - color conversion, 381, 386
  - convolution, 381–3
    - kernel operator, 381
  - custom, 386–8
  - in-place, 381
  - lookup tables, 381, 384–5
  - rescaling, 381, 385
  - transforms, 381, 383–4
- `ImageIcon` (`javax.swing`), 367, 370
- `ImageObserver` (`java.awt.image`), 192, 367–8
- `ImageProducer` (`java.awt.image`), 377
- images, 190–2
  - see* animations
  - see* `BufferedImage`
  - creating, 372–3
  - display, 191, 371–2
  - double buffering, 372
  - encoding formats, 190, 370, 373
  - graphics context, 372–3
  - see* image filtering
  - `ImageIO` (`javax.imageio`), 287, 367, 373
    - `javax.imageio` packages, 373
  - in JAR, 191
  - loading, 190–2, 367–70, 417
  - pixel handling, 374–80
  - processing, 373–4
  - saving, 373
  - scaling, 371–2
  - tiling, 371
- immutable objects, 357
- implements, *see* interfaces
- import, 132, 137–40
  - see* static import
- `InetAddress` (`java.net`), 423–6
  - see* IP address
- inheritance, 91–3
  - base class, 93
  - extends, 91
  - see* interfaces
  - multiple inheritance, 93
  - see* overriding
  - subclass, 91, 93
  - superclass, 92, 93
- inner class, 238–40
  - anonymous, 239–40
- `instanceof`, 115, 203
- instances, *see* objects
- instantiation, *see* class
- int, 34
  - see* integers
  - see* primitives
- `Integer` (`java.lang`), *see* wrappers
  - for primitives
- Integers, 33–4
  - byte, 30
  - int, 30
  - long, 30
  - short, 30
  - see* mixing primitives
  - see* primitives
  - unsigned short, *see* `char`
  - see* wrappers for primitives
- interface, 109–15
  - for callbacks, 112–13
  - data fields, 114
  - implements, 109
  - interfacing classes, 111
  - marker, 115
  - multiple inheritance, 109–10, 113
  - to hold constants, 114–15
- internationalization, 363
- Internet basics, 413–14
- I/O in Java
  - binary, 309–12
  - bytes, 315–17
  - see* I/O with console
  - see* I/O with files
  - see* I/O with objects
  - see* I/O with text
  - `java.io`, 287, 289–90
  - `java.nio`, 287, 326
  - packages, 287–9
  - see* streams
- I/O with console, 18, 50–2, 291–9
  - see* formatted output/I/O
  - see* stream classes – `PrintStream`
    - `print()`, `println()`, 32, 50–1, 62–3, 289
  - `System.err`, 291
  - `System.in`, 289, 291, 295–6
  - `System.out`, 62, 289, 291, 295
- I/O with files, 301–12
  - binary, 309–12
  - see* `File`
  - text input, 305–9
    - see* `Scanner`
  - text output, 302–5
- I/O with objects, 313–15, 417
  - histogram example, 320–1
  - `ObjectOutputStream`, `ObjectInputStream`, *see* stream classes
  - `Serializable` (`java.io`), 314, 320
  - transient, 314
- I/O with text
  - see* formatted I/O
  - see* I/O with console
  - input, 295–9
    - see* `Scanner`
  - output, 291–5
    - to file, 302–5
  - see* stream classes – `PrintStream`
  - see* stream classes – `Reader`
  - see* stream classes – `Writer`
- `IOException` (`java.io`), 296
- IP (Internet Protocol), 413, 414–15, 423–6
- IP address, 414
  - applet access, 427
  - `InetAddress`, 423–6
  - loopback, 414, 438

- ItemListener, *see* event listener
- Iterator (`java.util`), 42, 329, 337–8, 409
- JAR (Java ARchive), 25, 132, 143–6, 524
  - applet, 145, 149
  - application, 145–6
  - `java.util.jar`, 363
  - manifest, 144
- jar tool, 143–5
  - see* pack200 tool
- Java
  - code compatibility, 16–17, 340
  - core language, 16
    - optional packages and APIs, 16, 631
    - standard extensions (`javax` packages), 162–3
  - definition of, 3–4
  - documentation, 16
  - editions, 5
  - features and benefits, 8–12
  - history, 4–5
  - Java Community Process (JCP), 8
    - see* Java programming
  - Java 2 Standard Edition (J2SE), 5–6
    - keywords, 29, 30, 683
    - legacy code and Java, 10
    - naming conventions, 7–8
    - networking, 9, 10, 416–17
    - see* operators
    - see* performance
    - portability, 9, 10, 160, 417
    - see* real-time
    - real world applications, 12–14
    - shortcomings, 10–12
    - symbols, 30, 684
    - technical programming, 10–12
    - versions, 5–8
  - Java Card API, 5, 7, 651, 652, 653
  - Java Foundation Classes (JFC), 6, 160, 163
    - see* Java 2D
    - see* Swing
  - Java hardware, 657–80
    - aJile board, 663–4
    - aJile processors, 658–9, 660
    - ARM Jazelle, 660
    - Aurora, 660
    - benefits, 658
    - boards, 661–5
    - chips, 658–9
    - co-processor, 657
    - cores, 657, 659–60
    - DCT cores, 660
    - FPGA (Field Programmable Gate Array), 657
    - Imsys Cjip processor, 659
      - see* SNAP from Imsys Technologies
    - instruction sets, 657–8
    - Nazomi JA108, 659
      - see* Javelin Stamp from Parallax
    - performance, 663, 680
    - picoJava, 657
    - real-time, 659
    - standalone, 657
    - Synopsys JvXtreme, 660
      - see* TINI
    - Virtual Peripheral (VP), 659, 665, 667
    - Vulcan Machines Moon 2, 660
    - Xilinx lavaCORE, 660
- Java IDL, 530
  - see* CORBA
- Java networking, 416–17
  - `java.net`, 417, 431
    - see* Client/Server
    - see* CORBA
    - see* embedded Java
    - see* RMI
    - see* sockets
    - see* Web services
- Java programming
  - basics, 14–23
  - first applet, 19–20
  - first application, 18
  - getting started, 17–23, 50–3
  - JDK (Java Development Kit), 5, 7
  - SDK (Software Development Kit), 3, 6, 7, 15, 17–18
  - starter programs, 20–3
  - tools, 15–16
- Java Runtime Environment (JRE), 3, 147
- java tool, 15, 147
  - D option, 628–9
- Java Virtual Machine, *see* JVM
- Java 2 Enterprise Edition (J2EE), 7, 9, 162, 431
- Java 2 Micro Edition (J2ME), 7, 163, 652–4, 659
- configurations, 652, 653–4
  - CDC, 653–4
  - CLDC, 653–4, 659, 663, 671, 672
  - see* Java hardware
  - KVM, 653–4
  - prelinked, 653
  - preverification, 672
  - profiles, 653, 654
    - MIDP, 654, 659
    - MIDlets, 654
    - see* SNAP from Imsys Technologies
- Java 2 Standard Edition (J2SE), 5–6, 182
  - see* Java
- Java 2D API, 163, 183–90, 390
  - clipping, 185
  - compositing, 185
  - drawing example, 187–90
    - see* Graphics2D
    - see* images
  - Line2D (`java.awt`), 186
  - paint, 183–4
  - Rectangle2D (`java.awt`), 186, 188
  - rendering hints, 184–5, 186
  - Shape (`java.awt`), 186
  - stroke, 184
  - transformations, 185, 190
- Java 5.0 (J2SE5.0), 6, 8, 23–6, 327
- javac tool, 3, 15, 147, 523, 671
  - bootclasspath option, 672
  - target option, 672
- Javadoc API, 33, 149–50
  - Doclets API, 150
  - Taglet API, 150
- javax packages, 162–3
  - see* Java
- `javax.comm`, *see* port communications
- `javax.swing`, *see* Swing
- `java.io`, 289–90
  - see* I/O
  - see* streams
- `java.lang`, 137
- Javelin Stamp from Parallax, 659, 664–5
  - programming, 665–70
  - example program, 640, 643, 667–70
  - see* Java hardware
- JFileChooser (`javax.swing`), 318–20
  - FileFilter (`javax.swing.filechooser`), 318

- JNI (Java Native Interface), 585–624
  - arrays
    - get and set region functions, 605–6
    - of primitives, 604–7
    - of object, 606–7
  - C tools, 595–6
  - calling conventions, 600
  - calling Java methods from native code, 615–18
  - client/server applications, 562
  - exceptions, 618–22
  - field ID, 608–11
  - Fortran, 587
  - getting Java field values from native code, 613–15
  - Hello World example, 587–97
  - interface pointer, 598–600
  - introduction, 585–7
  - "j" data types, 593
  - javah and header file, 588, 591–3
  - jclass, 593, 597, 611–12
  - jobject, 593, 597, 608, 612
  - jobjectArray, 606–7
  - local and global references, 622
  - method ID, 615–18
  - method signature, 616
  - name mangling, 593
  - native implementation, 594–6
  - native keyword, 589, 590
  - native method, 586
  - objects on C side, 607–15
  - object methods, 612
  - passing parameters, 597–8
  - shared library, 596
  - string objects, 600–4
  - System.loadLibrary(), 590–1, 597
  - threads, 623–4
  - type signatures, 610
- JSP (Java Server Pages), 446–7
- JTabbedPane (javax.swing), *see* Swing components
- JTextField (javax.swing), *see* Swing components
- JVM (Java Virtual Machine), 3, 8, 11, 15, 625, 658
  - Hotspot JVM, 6
  - instruction set, 657–8
  - J2ME, 653–4
- Just-In-Time (JIT) compilers, 11
- KVM, *see* Java 2 Micro Edition
- real-time, 654–7
- threads, 253, 259–62
  - see* threads
- KeyAdapter (java.awt.event), 400
  - adapter classes, 240–2
- KeyEvent (java.awt.event), 244, 400
  - see* events
- KeyListener, *see* event listener
- keystroke handling, 399–402
- keywords, *see* Java
- KVM, *see* Java 2 Micro Edition
- laying out components, 167–71
  - set layout manager, 224
  - see* containers
  - see* layout managers
- layout managers, 167, 223–37
  - BorderLayout (java.awt), 173, 228–9
  - BoxLayout (javax.swing), 225–8
    - in Box, 226–8
  - CardLayout (java.awt), 229–30
    - vs. JTabbedPane, 230–1
  - FlowLayout (java.awt), 168, 173, 224–5
  - GridBagLayout (java.awt), 234–7
  - GridBagConstraints (java.awt), 234–7
  - GridLayout (java.awt), 168–71, 224
  - set layout manager, 224
  - SpringLayout (javax.swing), 231–4
- lightweight components, 162, 174
  - see* AWT
  - heavyweight, 162
  - peer component, 162
  - see* Swing
- LinkedList (java.util), 336
- Linux/Unix and Java, 6
  - continuation characters, 524
  - directory listing, 630
  - directory separators, 516
  - JNI, 586, 588, 591
  - ports, 416, 432
  - storing preferences, 335
  - SDK, 17
    - opening comm ports, 635
  - List (java.util), 336
    - and* generics, 339–40
  - ListSelectionListener, *see* event listeners
  - literals, 31
    - floating-point types, 696
    - integers, 97
    - string, 35
  - little-endian, 316
  - localhost, 510, 527
    - see* IP Address – loopback
  - logging (java.util), 363
  - long, 34
    - see* integers
    - see* primitives
  - Long (java.lang), *see* wrappers for primitives
  - Look and Feel, 163
    - Swing Synth, 24
  - LookupOP (java.awt.image), 381, 384
    - see* image filtering
  - LookupTable (java.awt.image), 384–5
    - threshold filter, 385
  - loopback address, *see* IP Address
  - Mac computers, 6, 50
  - main(), 32, 50, 253
    - JNI, 589
  - Map (java.util), 336
  - math
    - Math (java.lang), 53–5, 69
    - Math functions, 54–5
    - StrictMath (java.lang), 55
    - vectors and matrices, 123–4
  - MediaTracker (java.awt), 367, 368–9, 370
    - see* images
  - MemoryImageSource (java.awt.image), 376–8, 380
    - see* animations
  - member variables, 59
  - menus, 244



- metadata, *see* annotation
- methods, 58, 60–2
  - see* class
  - name, 61
  - overloading, 62–3, 95, 96–8
  - see* overriding
  - parameters, 61, 62, 71–3
  - polymorphism, 96
  - return type, 61
- Microsoft Windows, 6
  - directory separators, 523
  - JNI, 586, 588, 591
  - JVM 1.1, 6, 8
  - screen resolution, 629
  - storing preferences, 335
  - SDK, 17
- microservers, 431, 432–5, 448–69
  - see* client/servers
  - see* embedded Java
- middleware, 431
  - see* Java 2 Enterprise Edition (J2EE)
- MIDlets, 654
  - see* Java 2 Micro Edition
- mixing primitives, 46–7
  - see* cast
  - see* cast rules
  - numeric promotion, 46
- Model-View-Controller Pattern (MVC),
  - 482–3, 493–6
  - compute engine, 482
  - controller, 482
  - model, 482
  - Observable (`java.util`),
    - 493–5
  - Observer (`java.util`), 493–5
  - view, 482
- mouse buttons, 394–6
  - see* events
- MouseAdapter (`java.awt.event`),
  - 399
  - adapter classes, 240–2
- MouseListener, *see* event listener
- MouseMotionListener, *see* event listener
- multiprocessing, 259–62
- namespaces, 136–7
- naming service, 479
  - see* CORBA
  - see* RMI
- network architectures, 413–14
  - application layer, 413, 415
  - network layer, 413
  - transport layer, 413, 415
- networking, *see* Java networking
- new operator, 35, 66
- notify(), notifyAll(), *see* threads
- null modem, 640
- null reference, 40, 331
- NumberFormat (`java.text`), *see*
  - formatting
- Oak project, 4
- Object (`java.lang`), 73, 115, 118–19,
  - 328, 337–8
  - clone(), 115, 118
  - equals(), 73, 118, 330
  - finalize(), 119
  - hashCode(), 330
  - methods, 118–19
  - notify(), notifyAll(), and
    - wait(), *see* synchronization
  - toString(), 119–20
- Object-Oriented Analysis and Design (OOAD), 473, 474, 558
- Object-Oriented Programming (OOP), 9,
  - 10, 58, 84–5, 89–90, 130
- objects (or instances), 58
  - immutable, 357
  - instance data, 67
  - see* references
  - see* I/O with objects
- operators, 30, 36–9
  - arithmetic, 53–4, 686
  - assignment, 36, 37, 685
  - associativity, 38–9, 692
  - bitwise, 360–1, 689
  - Boolean, 688
  - cast, 691
  - class and object, 690
  - comparison, 688
  - conditional, 691
  - decrement and increment, 37,
    - 687
  - precedence, 38–9, 51, 691
  - returned value, 36–7
- Observable and Observer, *see*
  - Model-View-Controller Pattern
- Outputable interface, 171–2, 270,
  - 649
- overloading,
  - see* methods
- overloading operators, 11, 35
- overriding, 93, 94–100
  - @Override annotation, 98–9
  - data fields, 99
  - shadowing variables, 100
  - vs. overloading, 96–8
- pack200 tool, 25, 146, 149
- packages, 132, 133–6
  - see* access rules
  - unnamed (default) package, 136, 140
- packet communications, 413–14
- paintComponent() in JComponent,
  - 179, 183, 193, 390
  - see* graphics
  - see* Swing classes – JComponent
- Panel (`java.awt`), 161
- panes, *see* containers for graphics
  - components
- parallel port I/O, *see* port
  - communications
- path, 147
  - see* CLASSPATH
  - see* directory separators
  - see* File
- peer component, 162
  - see* lightweight components
- performance, 9, 23, 404–9
  - inlining, 405
  - see* Java hardware – performance
- permissions, *see* SecurityManager
- PersonalJava API, *see* embedded java
- PixelGrabber (`java.awt.image`),
  - 375–6
  - see* color
- plug-in JVM for browsers, 22, 149, 164
- policy file, *see* SecurityManager
- policytool, 442, 514
  - see* SecurityManager
- polymorphism, 96
- popup menu, 396–9
- port communications, 631–49
  - javax.comm, 631–40
    - CommPort, 632, 639
    - CommPortIdentifier, 632–5
    - ParallelPort, 632
    - SerialPort, 632, 637–40
  - communication procedure, 637



- finding ports, 632–4
- opening ports, 635–7
- port status and ownership, 634–5, 637
- serial port I/O, 637–40
  - RS232, 637–40
  - DB9 connector pins, 638
  - null modem, 640
  - example program, 640–9
- USB, 631
- ports, 415–16
  - see* sockets
- Preferences API, 327, 331–5
  - examples, 333–4
  - exporting and importing, 334
  - listeners, 335
  - platform independence, 332
  - system, 334–5
- prefs (java.util), 332
- primitives, 29, 30, 33–4
  - cast, 44–7
  - converting, 46
  - see* data types
  - see* floating-point
  - method parameters, 69–70
  - string conversions, 74–6
  - to/from byte array, 315–17, 649
  - see* variables
  - see* wrappers
- PrintStream, *see* stream classes
- print(), println(), 32, 50–1, 62–3, 289
  - see* stream classes – PrintStream
  - see* I/O with console
- printf(), 25, 156–7, 289, 293–5
  - see* formatted I/O
  - see* stream classes – PrintStream
- printing graphics, 390–2
  - JobAttributes (java.awt), 392
  - PageAttributes (java.awt), 392
  - PrintGraphics (java.awt), 390
  - PrintJob (java.awt), 392
- priority inversion, *see* threads
- private, *see* access rules
- Process (java.lang), 629–31
- Properties (java.util), 327, 330–1, 335, 336, 392
  - see* Hashtable
  - see* Preferences API
  - system properties, 625–9
- protected, *see* access rules
- public, *see* access rules
- Queue (java.util), 336
- random numbers, 126–8
  - Gaussian, 88, 127
- Random (java.util), 127
- RandomAccessFile (java.io), 302
- Raster (java.awt.image), 366
  - see* BufferedImage
- Reader and its subclasses, *see* streams
- real-time, 654–7
  - asynchronous event handlers, 656
  - garbage collection, 655–6
  - happenings, 656
  - hard vs. soft, 655
  - immortal memory, 656
  - Java processors, 659
  - memory direct access, 656
  - Real-Time Specification for Java (RTSJ), 656–7
  - scoped memory, 656
  - timing, 656
  - threads, 656
- refactoring, 285
- reference, 31, 66–7
  - accessing methods and fields, 67
  - see* cast
  - conversions, 116
  - in method parameters, 71–3
  - modifying, 70–1
  - super, 99–100, 183
  - this, 99–100, 256, 612
  - variables, 69–70
  - vs. C/C++ pointers, 66, 112
- regular expressions, 352
  - java.util.regex, 363
- rendering hints, 184–5, 186
  - see* Java 2D
- RescaleOp (java.awt.image), 381, 385
  - see* image filtering
- RGB, *see* color
- rigid area, 227
- RMI (Remote Method Invocation) API, 26, 417, 429–30, 471, 479, 501–29
  - binding, 507–9
  - client, 509–11
  - client JAR file, 524
  - client/server example, 515–26
  - codebase, 512–13, 524
  - see* distributed computing
  - exceptions, 503–4, 506, 509, 510, 518
  - exporting, 506
  - marshalling, 504
  - name lookup, 508–11, 543
  - policy file, 513–15, 525, 526
  - registry, 507–9, 524–5, 552, 554–5
    - rmiregistry, 479
  - Remote (java.rmi), 506
  - remote implementation class, 506–7
  - remote interface, 505–6
  - remote objects, 503–5
  - rmic, 504, 523
  - run example on two machines, 527
  - security manager, 511–12
  - security, 511–15
  - skeletons, 504
  - stub, 504
  - vs. CORBA, 503, 530–2, 535
  - vs. socket programming, 469, 502
  - Web services, 573–4
  - and XML, 574–6
- RMI Client/Server example, 549–72
  - client implementation, 568–9
    - with histogram, 569–70
  - factory interface, 551–2, 553–6
  - JNI and RMI, 562
  - security, 551
  - server implementation, 556–67
  - server interface, 552–3
  - server pool, 553
  - spring simulation, 549–50, 560–7
- RPC (Remote Procedure Calls), 502
- run external programs, 629–31
  - Process, 629–31
  - Runtime, 629–31
- Runnable (java.lang), 109–10, 254, 256–8
  - concurrency utilities, 341, 342
  - see* applets
  - see* threads
- Runtime (java.lang), 629–31

- SampleModel (java.awt.image),  
   366  
   *see* BufferedImage  
   *see* Raster  
   sample, 366  
 sandbox, 625  
 Scanner (java.util), 289, 295,  
   298–9, 307–9  
   *see* I/O with text  
 scope, 238  
 screen properties, 629  
 scrollbars, 171, 221  
 SDK, *see* Java programming  
 security, 9, 417, 439–43, 551  
   with RMI, 511–15  
 SecurityManager (java.lang), 9,  
   424, 439–40, 511–12  
   and applets, 419, 626  
   policy file, 440–1, 513–15  
   permissions, 440, 442–3, 513–15  
   policytool, 442, 514  
   with RMI, 511–12  
 serial port I/O, *see* port communications  
 serialization, *see* I/O with objects  
 servers, *see* client/server  
 server pool, 553  
 ServerSocket, 428–9, 432–3, 434,  
   448, 449, 450  
   *see* sockets  
 servlets, 446–7  
   *see* Web services  
 Set (java.util), 336  
 setter methods, 153  
 shadowing variables, *see* overriding  
 short, 34  
   *see* integers  
   *see* primitives  
 Short (java.lang), *see* wrappers for  
   primitives  
 SIMM (Single Inline Memory Module),  
   661  
 simulations with Java, 10  
   *see* animations  
   *see* RMI Client/Server example  
 SNAP from Imsys Technologies, 663  
   *see* Java hardware  
   programming SNAP, 671–9  
   Web server example, 671–9  
 Socket (java.net), 426, 432–3,  
   443  
   sockets, 426–8, 434–5, 448  
     benefits and shortcomings, 469  
     Client/Server, 428–9  
     *see* DatagramSocket  
     ports, 415–16  
     *see* ServerSocket  
     Web server, 431  
 SortedMap (java.util), 336  
 SortedSet (java.util), 336  
 sRGB, *see* colors  
 Stack (java.util), 336, 362  
 standard extensions, *see* Java  
 starter programs, 20–3  
 statements, 31, 39–44  
   compound, 39  
   conditional, 40–1  
   declarations, 40  
   enhanced for loop,  
   flow control, 41–4  
   loops, 41–3  
   switch, 43–4  
 static keyword, 67  
 static (or class) members, 67–9  
 static import, 25, 115, 141–3  
   *see* constants  
   *see* import  
 streams, 287, 289–90  
   buffered, 290–1, 296–8, 317–18  
   choosing, 315  
   destination, 317–18  
   histogram example, 321–3  
   file, 301–12  
   filters, 317–18  
   histogram example, 323–5  
   flushing buffers, 290, 293, 456  
   *see* I/O  
   input examples, 315  
   networking, 416–17  
   output examples, 315  
   pipelined, 326  
   wrappers, 287, 290–1  
   8-bit, 291–9  
   16-bit, 291  
 stream classes from java.io  
   BufferedReader, 290–1, 419  
     for socket stream, 426, 434–5, 453  
   ByteArrayInputStream, 316,  
     649  
   ByteArrayOutputStream, 316,  
     321  
   DataInputStream, 310, 316, 649  
   DataOutputStream, 309, 316,  
     453  
   FileInputStream, 301–12, 316  
     web server example, 436  
   FileOutputStream, 301–12,  
     316  
   FileReader, FileWriter, 301–9  
   FilterInputStream,  
     FilterOutputStream, 313,  
     317–18, 323–5  
   InputStream, 289, 295  
     from URL, 419  
   InputStreamReader, 290, 419  
     for socket stream, 426, 434–5  
   ObjectOutputStream,  
     ObjectInputStream, 313  
   OutputStream, 289, 294, 295, 321,  
     323  
   OutputStreamWriter, 291, 313  
   PrintStream, 289, 293, 295, 674  
     deprecated constructors, 291  
     *see* formatted I/O – printf()  
     *see* I/O with console  
     print(), println(), 32, 50–1,  
     62–3, 289  
     printf(), 25, 156–7, 289, 293–5  
   PrintWriter, 291, 293  
     for socket stream, 426, 434–5, 436,  
     453  
   Reader, 289, 291  
   Writer, 289, 291–5  
 strictfp, 12, 50, 695  
   *see* floating-point  
 StrictMath (java.lang), 55  
   *see* math  
 String (java.lang), 12, 35, 50,  
   76  
   `+' append operator, 35, 76  
   array example, 120  
   from objects, 119–20  
   methods, 349–50  
   split(), 352–3, 435, 674  
   valueOf(), 76  
 StringBuffer (java.lang), 351  
 StringBuilder (java.lang), 26,  
   295, 351  
 StringTokenizer (java.util),  
   327, 352  
 strut, 227

- subclass, *see* inheritance
- super, *see* reference
- super(), 101–2
  - see* constructors
- superclass, *see* inheritance
- Swing, 25, 160, 162–4
  - class hierarchy, 163–4
  - double buffering, 372
  - see* graphics
  - javax.swing, 163
    - see* Swing components
  - J2SE5.0 improvements, 24
  - see* laying out components
  - see* layout managers
  - text display, 171–4
- Swing components
  - Box, 226–8
  - JApplet, 164–6, 188, 247
  - JButton, 166–7
  - JCheckBox, 212, 214–16
  - JComponent, 163
  - JDialog, 163, 164–6
  - JFileChooser, 318–20
  - JFrame, 163, 164–6
  - JLabel, 167
  - JList, 212, 218–21
  - JMenu, 244
  - JMenuBar, 244
  - JMenuItem, 244
  - JPanel, 166–7, 174, 183, 224
  - JPopupMenu, 396–9
  - JRadioButton, 212, 216–18
  - JScrollPane, 171, 221
  - JSlider, 212, 221–3
  - JTabbedPane, 230–1
  - JTextArea, 170, 171–3
  - JTextField, 167, 168, 171, 251, 449
    - JPasswordField, 449
  - JWindow, 166
- switch statement, 43–4
  - break, 44
  - case, 44
  - default, 44
  - type of switch variable, 44
- symbols, 684
  - see* Java
- synchronization, 119, 267, 272–6
  - data race, 253, 267, 271
  - lock, 267, 269–70, 272
  - notify(), notifyAll(), wait(), 119, 269–70, 272–3
    - see* Object
  - synchronized, 268, 269
  - System.arraycopy(), 121
  - System.err, *see* I/O with console
  - System.getProperties(), 330
  - System.in, *see* I/O with console
  - System.out, *see* I/O with console
  - system properties, 625–9
    - see* properties
- Taglets API, *see* Javadoc
- TCP (Transmission Control Protocol), 413, 415
- text display, *see* Swing
- text drawing, *see* Graphics
- this, *see* reference
- this(), 100–1
  - see* constructors
- threads, 9, 10, 253–8
  - see* animations
  - communicating, 262, 272–6
  - data race, *see* synchronization
  - see* concurrency features
  - exclusive, 262, 267–72
  - non-interacting, 262, 263–4
  - priority, 261–2, 264
  - priority inversion, 656
  - producer/consumer, 272–6
  - real-time, 656
  - resources per thread, 260
  - run(), 253, 254–5, 256–8
  - servers, 433–4, 449
  - stopping, 258–9
  - see* synchronization
  - task splitting, 262, 265–6
  - Thread subclass vs. Runnable, 257–8
    - see* thread context switching
  - using Runnable, 256–8
    - see* Runnable
  - using Thread subclass, 254–5
- Thread (java.lang), 254
  - sleep(), 254, 260, 262, 277, 281
  - subclass, 254–5, 257–8
  - yield(), 260, 262
  - start(), 254, 256
- thread context switching, 260–1
  - non-preemptive (cooperative), 260
  - preemptive (time-slicing), 260
- time
  - see* date and time formatting
  - System.currentTimeMillis(), 356, 405
- timers, 281–5
  - clock examples, 282–3, 284–5
  - Timer (java.swing), 282, 284–5
  - Timer (java.util), 282–3
  - TimerTask (java.util), 282–3
- timing, 260, 356, 404–9
  - and profilers, 404
- TINI (Tiny InterNet Interface), 661–3
  - SLUSH command shell, 661
  - SNAP card from Imsys Technologies, 663
    - see* Java hardware
- transient, 314
  - see* I/O with objects
- TreeMap (java.util), 336
- TreeSet (java.util), 336
- Toolkit (java.awt), 191, 392, 394, 629
- UDP (User Datagram Protocol), 415
  - see* datagrams
- UML (Unified Modeling Language), 130, 471, 494–6
  - association, 494
  - class diagrams, 474, 487–90, 494, 498–500
  - client/server, 475–7, 481–90
    - see* collaboration diagrams
  - collaborator, 478
  - conceptual model, 474, 476
  - see* design patterns
  - generalization (inheritance), 488, 494
  - interaction diagrams, 474–5
  - interface, 488
  - introduction, 473–5
  - iteration notation, 480
  - operations and attributes, 474, 487
  - polling, 489
  - realization, 489, 494
  - sequence diagrams, 475
  - sequence numbers, 477

- Unicode, 34, 79, 290, 312–13
  - see* char
  - see* character encoding
- unsigned short,
  - see* char
  - see* integers
  - see* primitives
- URL
  - from File, 419
  - loading image, 190, 417
  - MalformedURLException
    - (java.net), 419
  - reading file, 419–23
  - URL (java.net), 417–23, 512–13
  - URL components, 417
- utilities,
  - see* bit handling
  - see* Collections Framework
  - concurrency utilities, 285, 341–3
  - see* generics
  - java.util, 327–8
  - see* Preferences API
- varargs (variable argument list), 25
- variables
  - default values, 60
  - local, 61, 62
  - member, 62
  - primitives, 69–70
  - passed by value, 69–70
  - see* references
- Vector (java.util) 123, 327, 328–9, 335, 337
  - in example program, 452
  - performance, 405–8, 409
- Virtual Peripheral (VP), *see* Java hardware
- web client application, 443–5
- web server, 431–45, 448–69
  - applications of, 445–6
  - client application example, 443–5
  - see* client/server design, 432–5
  - embedded example, 671–9
  - see* microservers
  - security, 439–43
  - servlets and JSP, 432–3, 579–80
  - threads, 433–4
  - socket based, 431
- web services, 12
  - and distributed computing, 573–4
  - introduction, 573–81
  - Java, 578–80
  - JWS DP (Java Web Services Developer Pack), 13, 579, 581
  - servlets, 579–80
  - SOAP, 580
  - technical applications, 580
  - UDDI, 580
  - XML, 574–6
- while, do-while loops, 42–3
- whois service, 426–8
- Window (java.awt), 161
- WindowListener, *see* event listener
- wrappers for primitives, 73–9, 329
  - see* autoboxing and unboxing
  - Double, 74
    - parseDouble(), 75
  - Float, 76
  - Integer, 74, 77, 78
    - parseInt(), 75, 80–3
  - string conversions, 74–6
  - see* primitives
  - table, 74
- wrappers for streams, 287, 290–1
  - see* streams
- WritableRaster
  - (java.awt.image), 379, 380
  - see* BufferedImage
  - see* Raster
- Writer and its subclasses, *see* streams
- XML, 12, 574–6
  - DOM, 578
  - DTD, 577, 578
  - Namespaces, 577
  - parsers, 578
  - preferences export/import, 334
  - SAX, 578
  - Schema, 577, 578
  - Swing Synth, 24
  - tag definitions, 577
- zip compression
  - see* JAR
  - java.util.zip, 287, 363