Index

Symbols

== operator, 134
!= operator, 134
< operator, 134
<= operator, 134
> operator, 134
>= operator, 134
// sequence, 58
/* sequence, 58
*/ sequence, 58
+ (addition) operator, 104
+= (addition and assignment) operator, 114
& (And) operator, 148–149
-> (arrow) operator, 197, 700
\ back slash, 404
{} (braces), 56
&& (Conditional And) operator, 148–149
| | (Conditional Or) operator, 149–150
--(decrement) operator, 104
/ (division) operator, 104
/= (division and assignment) operator, 114
$ (dollar sign), 57
= (equal sign), 61
= = (equality) operator, 341, 359, 405
/ (forward slash), 387
> (greater than sign), 96
++ (increment) operator, 104
-(hyphen), 700
* (multiplication) operator, 104
*= (multiplication and assignment) operator, 114
! (Not) operator, 146–148
| (Or) operator, 149–150
|| (or-or) operator, 417
+ (plus sign), 87
% remainder (modulus), 104
%= (remainder and assignment) operator, 114
; (semicolon), 50
-(subtraction) operator, 104
-= (subtraction and assignment) operator, 114
^ (Xor) operator, 150–151

A

abortCountDown method, 681
abs(argument) method, 118
AbstractBorder class, 314
abstract classes, 311–314
AbstractCollection class, 314
AbstractMap class, 314
abstract method, 312, 690, 692, 695, 696, 698, 699, 702, 704, 706, 709–713
accessor patterns, 265
accessors, 264
action event, 767, 769–771
ActionListener interface, 323
actionPerformed method, 323
addAll() method, 531, 826
addFirst() method, 545
adding
  array list elements, 535–536
  elements to hash maps, 574–576
  fields to interfaces, 318–319
  items to linked lists, 549–551
  jar to ClassPath, 387
  Javadoc comments, 389–392
addLast() method, 545
add() method, 545
AddSubtract1 program, 785–787, 790
AddWithKey, 491
Alert class
  ClickCounterAlert Program, 812–813
  commonly used constructors and methods, 809, 810
  control using ButtonType parameter, 811
  dialog box, 810
  parameter, 810
algorithms, 583
  characteristics of, 584–585
    correct results, 584–585
    finite execution, 585
    well-defined, well-understood, and solvable problem, 584
    well-defined and predictable set of outputs, 584
    well-defined set of inputs, 584
  classification of, 585
    backtracking algorithm, 588–589
    brute-force algorithm, 585–586
    divide-and-conquer algorithm, 587–588
    dynamic algorithm, 590–591
  definition of, 584
  fun challenges, 592
    Fizz buzz, 592–593
    Roman numerals, 593–594
    schedule tournaments, 596–598
    spell out dollar amount, 595
    tic-tac-toe game, 595
ancestor, 496
And (&) operator, 148–149
animation, 759
anonymous classes, 373–376, 697–698, 784, 790, 791, 793, 794. *See also* anonymous inner classes
  creating, 373–375
  creating program with, 375–376
anonymous inner classes, 790–793
  with AddSubtract3 program, 791–793
  button events, 793
API. *See* application programming interface (API)
append method, 419
application launch, 766–767
application programming interface (API), 10–11, 611
  classes, importing, 65–66
  documentation, 25
applyAsLong, 710–712

COPYRIGHTED MATERIAL

archiving packages, 386–387
ArithmeticException, 220, 227, 459
arithmetic operators, 103–106
array initializers, 502
ArrayList class, 657, 660
about, 529–531
adding elements, 535–536
constructors, 531
creating objects, 533–534
deleting elements, 540–541
deleting items, 531
inserting elements, 530
vs LinkedList class, 544–545
methods, 531
printing array lists, 536
resizing array lists, 530
updating elements, 538–540
using iterator, 536–538
array lists, 480
ArrayList.stream() method, 721
arrays
about, 499–500
analyzing, 478–481
associative, 491
chessboard program, 516–523
comparing, 528
converting to strings, 528
copying, 525–526
creating, 500–501
defined, 478, 499
deleting items, 482
dynamic, 480
filling, 524–525
initializing, 502
inserting items, 481–482
jagged, 513–514
length of, 480
one-dimensional, 479
performance characteristics of, 480–481
searching, 527
solving homework problems with, 503–506
sorting, 527
three-dimensional, 479, 515
two-dimensional, 479, 509–514
using enhanced for loop, 506–507
using for loops with, 502–503
using with methods, 507–508
varargs, 508–509
Arrays class
comparing arrays, 528
converting arrays to strings, 528
copying arrays, 525–526
filling arrays, 524–525
methods, 524
searching arrays, 527
sorting arrays, 527
using, 523–528
arrows, drawing, 255–256
askForAnothersRound method, 209, 216
assignment operator, 61, 112–114
compound, 114–115
assignment statements, initializing variables with, 73–74
associative arrays, 491
AtomicInteger class, for thread safety, 666–670
constructors and methods of, 668
in CountDownClock class, 668–670
atomic operation, 667
@author tag, 390
AutoCloseable interface, 327
autonomous vehicle (AV), 184
error decoder
else–if version, 184–185
switch version, 185–186

B

BabyName object, 630–632, 642
Baby Name search program, 642–644
using binary search, 637–640
using linear search, 633–635
backspace, 404
backtracking algorithm, 588–589
BadTax program, 450–451
Ball class, 329
base, 476
BaseballBall class, 328
base classes, 251, 290
behavior, 248–249
Bennett, David, 625
BigDecimal abs() method, 455
BigDecimal add() method, 455
BigDecimal class
about, 451–452
arithmetic methods, 455
COBOL, comparison with, 456
comparing values, 461–462
comparison methods of, 460–461
conversion methods of, 463
creating BigDecimal objects, 452–454
RoundingMode enumeration, 459
rounding values, 457–460
scaling methods of, 457
BigDecimal divideAndRemainder() method, 455
BigDecimal divide() method, 455, 457
BigDecimal max() method, 455
BigDecimal min() method, 455
BigDecimal multiply() method, 455
BigDecimal negate() method, 455
BigDecimal pow() method, 455
BigDecimal remainder() method, 456
BigDecimal subtract() method, 456
binary operators, 105, 134
binary searching, 626, 635–644
binary search tree, 496
birthDate variable, 445
birthday calculator, 442–445
BirthdayFun program, 442–445
block lambda expression, 704–705
blocks, 56–57
body, 259
body handlers, 748–750
BodyHandlers class, 749
boolean contains(CharSequence) method, 406
boolean containsKey() method, 571
boolean containsValue() method, 571
boolean deepEquals() method, 524

boolean endsWith(String) method, 406
boolean equalsIgnoreCase (String) method, 406
boolean equals() method, 335, 461, 524
boolean equals(String) method, 406
boolean expressions, 133
  relational operators, 134
  simple, 134–136
boolean hasNextBoolean(), 101
boolean hasNextByte(), 101
boolean hasNextDouble(), 101
boolean hasNextFloat(), 101
boolean hasNextInt(), 101
boolean hasNextLong(), 101
boolean hasNextShort(), 101
boolean isAfter() method, 435
boolean isBefore() method, 435
boolean isBlank() method, 406
Boolean isEmpty() method, 406
boolean isEqual() method, 435
boolean nextBoolean(), 100
boolean operator. *See* logical operator
boolean remove() method, 572
boolean replace() method, 572
boolean startsWith() method, 407
boolean type, 82
Border, 822–824
  BorderPane class, 838, 840–841
  constructors and methods, 838, 839
  import statement, 840
  javafx.scene.layout package, 840
  regions, 838, 839
  traditional window arrangement, 838
border pane, 772, 778
BorderPane class, 838
boxing, wrapper classes, 91–92
break statement, 158, 159, 162, 187
  intentionally leaving out, 193–194
Broderick, Matthew, 586
brute-force algorithm, 585–586
btnAdd_Click method, 798
btnClose_Click event, 817
btnSubtract_Click method, 798
Bubble Sort algorithm, 611–615
buffer, 419
Builder pattern, 745–747
Burns, Robert, 13
Bush, George H. W., 583
business programs, 244
business rules layer, 252
buttonClicked method, 771
button creation, 768–769
ButtonType parameter, 811
bytecode, 8
byte nextByte(), 100

C

Calculate interface, 694
calculate() method, 690–692, 698, 699, 705, 710
calculateNewPos method, 523
Calculator interface, 690, 693, 694, 696, 698, 699, 701, 705, 710, 712, 713
calculator program
  using functional interfaces, 696–698
  using lambda expressions, 700–702
  using LongUnaryOperator interface, 711–712
callbacks, 320–323
calling, 199
capacity, hash map, 571
carriage return, 404
car wash application, 194
Car Wash program
  uses a string, 195–196
  uses a switch expression, 198
Cascading Style Sheets (CSS), 758–759
casting, 90, 91
catchBall, 319–320
catch block, 223, 225, 226, 229, 230
catching exceptions, 223–224
catch-or-throw compiler error, 234
catch-or-throw rule, 233, 308
cbrt(argument) method, 118
ceil(argument) method, 123
central processing unit (CPU), 470, 472–474
character cases, creation, 191–192
character constants, escape sequences for, 81–82
characters, extracting from strings, 409–410
charAt method, 409
char charAt(int) method, 406, 419, 422
CharSequence interface, 403, 422
char[] toCharArray() method, 408
charts, 759
char type, 80–81
check boxes, 865–867
checked exception, 221, 233–238
chessboard program, 516–523
child class, 251, 290
child nodes, 494
ChronoUnit enumeration, 439
Church, Alonzo, 686
ClassCastException, 302
Class class, 333, 357–359
classes. *See also specific types of class*
  about, 250, 258
  abstract, 311–314
  anonymous, 373–376
  base class, 251, 290
  body of, 259
  child class, 290
  collection, 529
  concrete, 311
  constructors, 268–272
  declaring, 258–261
  defined, 474
  derived class, 290
  diagramming with UML, 253–256
  drawing, 254–255
  final, 300–301
  getters, 264–265
  implementation of, 249
  importing, 380–381
  inheritance, 251
  initializers, 274–275
  inner, 259, 368–371
  interface of, 249, 251
  LinkedList, 543–553
  members, 261

classes *(continued)*
names, 258–259
nested, 259
overloading methods, 266–267
parent class, 290
public, 261
putting in JAR file, 384–389
records, 275–278
sealing, 327–332
setters, 264–265
singleton, 280, 284–285
static inner, 372–373
subclasses, 251, 290, 293–294
superclass, 251, 290, 299
class file, 26, 760
`Class getClass()` method, 335
classic factorial example calculation, 599
nonrecursive solution, 600
recursive solution, 600–602
Classic Locker Problem program, 504–505
`ClassName`, 258
`ClassNotFoundException`, 220
`ClassPath` variable, 382, 387
class variables, 69–70
`clear()` method, 532, 540
ClickCounterAlert Program, 812–813
ClickCounterExit Program, 818–820
click counter program, 774–779
`ClickHandler` class, 790
The Click Me program, 762–768, 770–774
clients and servers, 737
`Cloneable` interface, 320
`clone()` method
about, 336, 352
ArrayList class, 532
`CloneTest2` program, 352–355
CloneTest program, 348–349
creating deep copies, 351–357
creating shallow copy, 350–351
`HashMap` class, 571
implementing, 346–347
LinkedList class, 546
`CloneNotSupported Exception`, 320
`CloneTest2` program, 352–355
CloneTest program, 348–349
`CloseRequest` event, 816–818
COBOL, 244, 456
collection
classes, 529
vs maps, 570
`Collection values()` method, 572
`ColumnConstraints` Class, 845, 846
combining operators, 108–109
`com.lowewriter.payroll` module, 397
command line, 761–762
comments
crafting, 58
end-of-line, 58
JavaDoc, 59
traditional, 58–59
commit pane, 38
`compareTo` method, 461–462
comparing strings, 153–154
compiler, 8
catch-or-throw compiler error, 234
compiling modules, 399
compound assignment operator, 114–115
compound expressions, 146
compound operation, 666
compressed.zip file, 18
`computeIfAbsent()` method, 573
computer games, 244
concatenation operator, 405
concise constructors, 278
concise source files, 48
concrete classes, 311
Conditional And (&&) operator, 148–149
conditional operator, 105, 152–153
Conditional Or (||) operator, 149–150
`Console` class, 383–384
constants, 75
constant time, 476
constructors
ArrayList class, 531
basic, 268–269
calling other, 271–272
concise, 278
creating, 268–272
default, 269–270
defined, 249, 259, 268
`HashMap` class, 571
inheritance and, 398–299
LinkedList class, 545
vs methods, 268
superclass, 299
`Timer` class, 321
consumers, 734, 735
in `java.util.function`, 710
`containsAll()` method, 532
`ContainsKey`, 491
`contains()` method, 532
`continue` statement, 162–163
converting and casting numeric data, 90
automatic conversions, 90
casting types, 91
`convertSquareToPos` method, 522–523
`CopyDemo` program, 525–526
copying arrays, 525–526
`CountBothWays`, 174
`CountDownApp` class, 655–656, 680
countdown clock, 649
to abort, 677–682
coordinated countdown application, 658–661
countdown application with virtual threads, 683
`CountDownClock` class, 649
`AtomicInteger` class in, 669–670
extend `Thread` class, 651–652
`ReentrantLock` class in, 674–675
`synchronized` keyword in, 671–672
`volatile` keyword in, 664
`Runnable` interface implementation, 653–654
`CountDownApp` class, 655–657
`LaunchEvent` class, 654–655
`Thread` class, 649, 650
thread coordination, 657–661
thread creation, 649
create and start thread, 652
T minus 0, 649
T minus 6 seconds, 649
T minus 16 seconds, 649

thread safety in, 661–663
with virtual threads, 683
CountDownClock class, 649, 658, 660–663, 680, 681
AtomicInteger class in, 669–670
extend Thread class, 651–652
ReentrantLock class in, 674–675
run method, 651, 652
sleep method, 651, 652
synchronized keyword in, 671–672
volatile keyword in, 664
CountDownClock object, 660
CountDownClock thread, 657
counter variable, 168
for loop, 170–171
CountEven program, 171
count expression, for loop, 168, 170
counting even numbers, for loop, 171
CountTestApp program, 283–285
CountVowels program, 410
covariant return type, 296
C programming language, 244
C# programming language, 453
crafting comments, 58
end-of-line comments, 58
JavaDoc comments, 59
traditional comments, 58–59
creating
anonymous classes, 373–375
arrays, 500–501
constructors, 268–272
deep copy, 351–356
enum, 101–102
fields, 261–262
generic collection classes, 557–558
interfaces, 315–316
JAR files, 399–400
linked lists, 548–550
modular JAR file, 399–400
packages, 381–383
shallow copies, 350–351
subclasses, 293–294
subdirectories, 382
two-dimensional arrays, 510–511
CricketBall class, 328
custom exceptions, 307–310

D

The Daisy Hill Sloth Sanctuary Program, 725–730
dangling else problem, 143
data, 496
database layer, 252
data structures
about, 469
analyzing arrays, 478–481
binary trees, 496–498
characteristics of, 475
defined, 474
hash maps, 492–493
maps, 491–493
models of computation and, 470–474
operations, 475
quantifying performance, 476–478
queue, 490
stacks, 490
trees, 494–496
data types
enum creation, 101–102
primitive. *See* primitive data types
dates
calculating, 438–439
comparing, 437–438
extracting information about, 436–437
formatting, 440–441
Date-Time API
about, 425–427
calculating dates, 438–439
comparing dates, 437–438
creating objects using now() method, 428–429
creating objects using of() method, 431–434
creating objects using parse() method, 430–431
extracting date information, 436–437
formatting dates, 440–441
LocalDate class, 434–436
Month enumeration, 432–433
selecting classes for applications, 427–428
ZoneOffset class, 434
DateTimeFormatter class, 440–441
DateTimeParseException, 430–431
DayOfWeek getDayOfWeek() method, 435
declaring
class variables, 69–70
inner classes, 368–371
instance variables, 70–71
local variables, 71–72
method's return type, 205–206
two or more variables, 69
variables, 68
deconstruction, record, 364–365
decrement operators, 110–112
deep copies, creating, 351–356
default method, 252, 325, 692–694, 698, 702
default package, 381
degree, 496
Delegation Event Model, 370
Delegation pattern, 294
DeleteByKey, 491
deleteCharAt() method, 419
delete() method, 419
dependencies, 396
@deprecated tag, 390
dequeue, 490
derived class, 251, 290
descendent, 496
descendingIterator() method, 546
designing with objects, 252–253
deterministic function, 687
dialog box, Environment Variables, 20
diamond operator, 533
dictionary-entry, 491
directories, 602–606
Directory Listing application, 603–606
divide-and-conquer algorithm, 587–588, 590
divideTheseNumbers, 231
dividing integers, 106–108
division by repeated subtraction, 587, 588
division by zero, 129–131
DoBinarySearch method, 641, 642

documentation
API, 25
Java, 25–26
javadoc command, 394
page for String class, 25
Dodecaphobia program, 159, 162, 176
do loop, 163–165
double doubleValue() method, 463
double-elimination tournament, 596–598
DoubleIt class, 697, 698
double nextDouble(), 100
double trouble, 449
double variable, 447
doubly linked lists, 544
do–while loop, 163–164, 176, 209, 484
aware of concerning, 165
flowchart for, 164
validating input from user, 165–167
drawing
arrows, 255–256
classes, 254–255
dropBall, 319–320
dump, 219
dynamic algorithm, 590–591
dynamic arrays, 480

E

edge, 494
eight queens problem, 588–589
element() method, 546
elements
accessing, 535–536
adding to hash maps, 574–576
defined, 499
deleting, 540–541
updating, 538–540
else–if statement, 144–146, 183–185
email programs, 648
end-of-line comments, 58
enhanced for loop, 506–507
enqueue, 490
ensureCapacity() method, 420
entrySet method, 579
enum, 101–102, 185
Environment Variables dialog box, 20
equality (==) operator, 341, 359, 405
equalsIgnoreCase method, 154, 199, 405
equals method, 199
about, 336, 340–341
overriding, 342–345
testing for value equality with, 461
using, 341–342
Error class, 307
escape sequences, 404
Euclid, 587
event classes, 782
event handler, 770, 777, 781–784, 787, 788, 791, 794, 796
EventHandler Interface, 783–788, 790, 791, 793, 794
event handling techniques
with anonymous inner classes, 790–793
implementation, 785–788
with inner classes, 788–790
JavaFX event, 783–784
lambda expressions, 793–798
review, 782–783
event listener, 320, 370
event source, 320, 783, 784, 788, 791, 793
event target, 783
Exception class, 228, 238, 308
exception handling, 13, 219
catching all exceptions at once, 228–229
catching exceptions, 223–224
checked exceptions, 233–238
displaying exception message, 230
finally block, 230–233
finding the culprit, 222–223
preemptive, 227–228
swallowing exceptions, 237–238
throwing exceptions, 236, 238–239
try-with-resources statement, 239–240
understanding exceptions, 220–221
witnessing an exception, 221–222
exception object, 13, 219
exceptions
catching, 223–224
checked, 221
custom, 307–310
swallowing, 237–238
throwing, 236, 238–239
unchecked, 221
understanding, 220–221
witnessing, 221–222
Exclusive Or (Xor) (^) operator, 150–151
exp(argument) method, 118
exponential notation, 79
exponential time, 477–478
exponents, 477
expressions. *See specific types of expression*
extends clause, 334

F

Factorial class, 691, 693
factorial function, 687, 688
factorial method, 688
factorials, 599–601
factory pattern, 313
fields
adding to interfaces, 318–319
creating, 261–262
defined, 248, 259, 261–262
private, 262
public, 248, 262
static, 279–281, 286
visibility, 262
File class, 603, 605
FileInputStream, 234
FileNotFoundException, 234
catching, 234–235
throwing, 235–236
File object, 611, 613
file structure, 761
FileVisitor interface, 611
FileWriter object, 240
fill method, 524–525
filter method, 722
final classes, 300–301
finalize method, 336

final keyword, 300–301, 329
finally block, 223, 230–233
final method, 300
final variables (constants), 75
FindName method, 632, 637, 642
finite execution, of algorithm, 585
Fizz buzz, 592–593
flight-simulator programs, 244
floating-point numbers, 128
floating-point types, 78–80
floating-point weirdness, 128–129
floating-point zeros, 130
float nextFloat(), 100
floor(argument) method, 123
FlowPane, 822, 823
  button rearrangement, 838
  constructors and methods, 835–837
  horizontal flow layout, 834–835, 837
  import statement, 837
  javafx.scene.layout package, 837
  setPrefWrapLength method, 837
  vertical flow layout, 835
folder
  perusing the JDK, 19
  structure, 760, 761
forEach(), 724
for loop, 168, 600, 619, 620, 622
  breaking and continuing, 176–177
  count expression, 168, 170
  counting backward, 172–173
  counting even numbers, 171
  enhanced, 506–507
  flowchart for, 169
  formal format of, 168–170
  ganging up expressions, 174–175
  initialization expression, 168, 170, 174
  omitting expressions, 175–176
  scoping out the counter variable, 170–171
  test expression, 168, 170
  using with arrays, 502–503
  without bodies, 173–174
formal type parameters, 557–558
format(number) method, 126
formatting
  dates, 440–441
  numbers, 125–127
form feed, 404
for statement, 618
forward slash (/), 387
fullFormat formatter, 445
fun algorithms challenges, 592
  Fizz buzz, 592–593
  Roman numerals, 593–594
  schedule tournaments, 596–598
  spell out dollar amount, 595
  tic-tac-toe game, 595
function, 686, 687, 690. *See also* functional interfaces
  in java.util.function, 709
functional interface, 696–698, 793, 794
@FunctionalInterface annotation, 695–696
functional interfaces, 690
  add default methods, 692–693
  add static methods, 694–695
  creation of, 690
  @FunctionalInterface annotation, 695–696
  implementation with class, 690–691
functional programming, 686–689
  functional interfaces, 690–696
  lambda expressions, 686, 698–706
  languages, 687, 690

G

game play, 291–292
game programs, 648
garbage collector, 250
generic, 533
generic collection classes
  about, 555
  creating, 557–558
  generic queue class, 564–567
  generic stack class, 559–562
  wildcard-type parameters, 562–563
generic queue class, 564–567
generics, 530
generic stack class, 559–562
GenQueue class, 564–567
GenStack class, 559–562
get accessors, 264
GetByKey, 491
getChildren method, 826
getClass method, 336, 357–358
getCurrencyInstance() method, 126
getDayOfWeek method, 437
getDaysOfWeek, 508
getFirst() method, 546
getGuess method, 209, 216
GetInteger method, 227–228
getLast() method, 546
get() method, 532
getNumberInstance() method, 126
getOrDefault method, 576
getPath method, 605
getPercentInstance() method, 126
getProduct method, 310
getRandomNumber method, 205, 207, 209, 212–214, 216
getters, 264–265
getText method, 857
getTime() method, 658, 660–662, 670, 672, 675
Git, 28
  installation, 31–32
GitHub, 28
  dashboard, 29
  repository creation, 28–31, 34–35
  sending project to, 37–42
GoodTax program, 464–465
Gore, Al, 583
graphical user interface (GUI), 648, 757, 758, 762
Greeter class, 62–64
GridPane, 822
  add method, 844
  ColumnConstraints Class, 845, 846
  constructors and methods, 842–844
  example, 847–853
  important features, 841
  RowConstraints Class, 846, 847
  setColumnSpan method, 844
  setHalignment method, 844
  sketching out a panel, 841–842
guarded pattern, 363–364

GuessingGame class (GuessingGameMethod3), 222
Guessing–GameMethod3.java file, 223
guessing-game program, 209–212, 216–218
  playARound method, 203–205
GUI. *See* graphical user interface (GUI)

H

handle method, 784, 787, 788, 790, 791, 793, 794
hashCode method, 336
HashMap class
  about, 570–571
  adding elements to hash maps, 574–576
  capacity, 571
  constructors, 571
  creating hash maps, 573–574
  load factor, 571
  methods, 571–573
  removing entries from hash maps, 578–580
  replacing entries from hash maps, 577
  retrieving elements from hash maps, 576
  treating hash map as collection, 579–580
hash maps, 492–493
Haskell, factorial function in, 687–688
hasNextInt method, 228
hasNext() method, 537
HBox, 822, 823
  addAll method, 826
  adding space between nodes, 829–831
  adding space with margins, 828–829
  constructors and methods, 824–825
  creation of, 826
  import statement, 826
  Priority enumeration, 829, 830
  setHgrow method, 829, 830
  setMargin method, 828–830
  spacing and padding to pane, 826–828
head, 483
height of node, 495
Hello, World! program, 46–48, 202, 203
HelloApp class, 61–63
HelloApp program, running, 36–37
HelloApp project, 36, 42
HelloApp repository, 31, 42
HelloWorldMethod program, 202
hierarchies
  inheritance, 292–293
  throwable, 307
High-Performance Computing Act of 1991, 583
HTTP. *See* Hypertext Transfer Protocol (HTTP)
HttpClient
  import statements, 743
  interfaces, 743
  send and receive HTTP messages, 743, 744
HttpRequest, 743–749, 754
HttpResponse, 743, 747–750, 754
Hypertext Transfer Protocol (HTTP), 733
  basic understanding, 736–742
  elements, 737–741
    clients and servers, 737
    message, 739–740
    methods, 740–741
    Multipurpose Internet Mail Extensions (MIME), 738
    resources, 737
    session, 738
    Uniform Resource Identifier (URI), 738
    Uniform Resource Locator (URL), 738
  exchange, 742
  tester program, 751–754
hypot(arg1, arg2) method, 118

I

iCounter variable, 787, 788, 791, 794
IDE. *See* integrated development environment (IDE)
IDEA, IntelliJ, 28, 32
identifiers
  creation, 57
  enhanced for loop, 506–507
identity, 245–246
identity() method, 694–695, 698
if–else statement, 139–140
if statement, 133, 134, 136, 156, 207
  else if statement, 144–146
  inside switch statement, 190–191
  nested if statement, 140–144
  simple, 136–138
IllegalArgumentException, 220
The Imitation Game, 685
immutable objects, 277, 405
implementation, 249, 251, 252
implements keyword, 690
importing
  classes, 380–381
  packages, 380–381
import java.io statement, 234
incrementCounter method, 662
increment operators, 110–112
index, 475, 478
indexOf() method, 532
inferring the data type, 85–86
infinite loop, 158–162
inheritance, 12, 289
  business example of, 292
  constructors and, 298–299
  defined, 251
  delegation pattern, 294
  game play, 291–292
  hierarchies, 292–293
  vs interfaces, 315
  in terms of real-world objects, 291
initialization expression, for loop, 168, 170, 174
initializers
  about, 274–275
  array, 502
  initializing variables with, 74
  instance, 259
  static, 259, 275, 281, 287
  types of, 259
initializing
  arrays, 502
  two-dimensional arrays, 512–513

initializing variables, 72–73
with assignment statements, 73–74
with initializers, 74
inner classes, 788–790
about, 259, 368
with AddSubtract2 program, 788–790
anonymous, 373–376
declaring, 368–371
defined, 368
static, 372–373
TickTockInner class program, 369–371
InputMismatchException, 100, 220, 221, 226
inputs, of algorithm, 584
insert() method, 420
Insets class, 828
Insets Constructors, 827, 828
instance initializers, 259
instance methods, 262
instanceof operator, 303, 360–362
instances
counting, 283–285
defined, 245
preventing, 286
instance variables, 70–71, 254
Instant class, 428
int capacity() method, 420
int compareToIgnoreCase (String) method, 406
int compareTo(String) method, 406
integers
dividing, 106–108
overflow, 127–128
type, 77–78
integrated development environment (IDE), 17, 761
IntelliJ, 32, 37, 40
IntelliJ IDEA, 28, 32
clone repository, 34
customize, 33
learn, 33
new project, 34
open, 34
plugins, 33
projects, 33
interfaces
about, 249, 251
vs abstract classes, 314–315
adding fields to, 318–319
basic, 315–316
creating, 315–316
defined, 475
enabling Java language features, 326–327
extending, 319–320
implementing, 316–317
vs inheritance, 315
in java.util.function, 707–709
map, 570
marker, 320
method types, 324–326
public, 263
subinterface, 319
superinterface, 319
using as types, 317–318
using for callbacks, 320–323
internal members, 263
Internet, 583
InterruptedException, 676, 677
interrupt method, 676, 681
int getDayOfMonth() method, 435
int getDayOfYear() method, 435
int getMonthValue() method, 435
int getYear() method, 435
int hashCode() method, 335, 580
int indexOf() method, 406
int intValue() method, 463
int lastIndexOf() method, 407, 420
int length() method, 407, 420
int lengthOfMonth() method, 435
int lengthOfYear() method, 435
int nextInt(), 100
int scale() method, 457
int size() method, 572
int variable, 503
invoking, 199
IOException, 220, 229, 308
IP address, 734, 738
isBlank method, 417
isEmpty() method, 417
isPresent method, 811
items
about, 475
adding to linked lists, 549–551
removing from linked lists, 553
retrieving from linked lists, 551–552
updating in linked lists, 552–553
Iterable interface, 327
IterateByKey, 491
iterative method, binary search algorithm, 635
Iterator interface, 537–538
iterator() method, 532
iterator pattern, 537–538
iterators, 537

J

J2EE. *See* Java 2 Enterprise Edition (J2EE)
jagged arrays, 513–514
JAR files
about, 379, 384
adding jar to ClassPath, 387
archiving package, 386–387
creating, 399–400
jar command-line options, 385
modular, 396, 399–400
putting classes in, 384–389
running, 400
running program directly from archive, 388
Java
documentation, 25–26
exception handling, 13
features of, 11
platform independence, 8–9
type checking, 11–12
Java 2 Enterprise Edition (J2EE), 17
Java 2 Standard Edition (J2SE), 17
Java API, 10–11, 76
Java API classes, importing, 65–66
Java command-line tools, 17
Java Development Kit (JDK), 16, 18, 35
downloading, 18
installing, 18–19
JDK 9, 66

Java Development Kit (JDK) *(continued)*
perusing the JDK folders, 19
setting the JAVA_HOME and path variables, 19–22
Javadoc
about, 379, 389
adding comments, 389–392
comments, 59, 390–392
`javadoc` command, 392–393
tags, 390
viewing pages, 393–395
Java Enterprise Edition (Java EE), 17
JavaFX
action event, 770–771
`Alert` class, 809–813
application launch, 766–767
`buttonClicked` method, 771
button creation, 768–769
class files, 760
click counter program, 774–779
The Click Me program, 762–764
command line, 761–762
exit a program
ClickCounterExit Program, 818–820
`close` method, 815
`CloseRequest` event, 816–818
right way to, 814
extending application classes, 765–766
file structure, 761
folder structure, 760, 761
input data from user
text field, 856–863
using check boxes, 865–867
using radio buttons, 867–869
validating numeric data, 863–864
layout panes, 771–772. *See also* layout panes, JavaFX
life cycle, 765–766
package imports, 764–765
package names, 760
possibilities of, 758–759
primary stage, 800, 805
root folder, 760–761
`Scene` class, 803–805
scene creation, 773
`SceneSwitcher` Program, 805–808
`setScene` method, 805
`setText` method, 769
setup, 759–762
`Stage` class, 800–803
stage setting, 773–774
`start` method, 767–768
syntax, 770
`javafx.application`, 764, 765, 777
javafx.controls module, 399
`javafx.event`, 782, 783, 787
`javafx.scene`, 759, 764, 769
`javafx.scene.layout`, 764, 772
`javafx.scene.layout` package, 840
`javafx.stage`, 764
JAVA_HOME and path variables, setting, 19–22
JAVA_HOME variable, creation, 20
`java.lang` package, 380
Java math, recognizing weird things about, 127
division by zero, 129–131
floating-point weirdness, 128–129
integer overflow, 127–128
Java Module System, 14, 379, 395–396
Java Platform Module System, 66
Java program, creation of first, 23–24
Java project, creation, 33–36
Java Runtime Environment (JRE), 16
intercepts and folds errors, 13
`java.scene.control`, 765
`java.security.SecureRandom` class, 120
Java's HTTP client library
`HttpClient`, 743–744
`HttpRequest`, 744–747
`HttpResponse`, 747–748
`send` method, 748–749
`java.sql` module, 397
Java Standard Edition (SE), 17
Java Technologies page, 18
`java.time` package, 429
`java.util.concurrent.atomic`, 667
`java.util.function`, 706–710
consumers, 710
function, 709
interfaces in, 707–709
operators, 709–710
predefined functional interface extend, 710–712
predicate, 710
supplier, 710
`java.util.Random` class, 120
Java version, confirmation, 22
Java version 11, 758
Java Version Insanity, 13–16
Java Virtual Machine (JVM), 8, 9, 17, 648, 682
into memory, 47–48
JDK. *See* Java Development Kit (JDK)
`JOptionPane` class, 64, 65, 323
joy of methods, 200
JRE. *See* Java Runtime Environment (JRE)
JVM. *See* Java Virtual Machine (JVM)

K

keys, 569
`keySet` method, 579
key-value-pair, 491
keywords, dealing with, 50–53
`K getKey()` method, 580
`kickBall`, 319–320

J

J2EE. *See* Java 2 Enterprise Edition (J2EE)
jagged arrays, 513–514
JAR files
about, 379, 384
adding jar to `ClassPath`, 387
archiving package, 386–387
creating, 399–400
jar command-line options, 385
modular, 396, 399–400
putting classes in, 384–389

running, 400
running program directly from archive, 388
Java
documentation, 25–26
exception handling, 13
features of, 11
platform independence, 8–9
type checking, 11–12
Java 2 Enterprise Edition (J2EE), 15
Java 2 Standard Edition (J2SE), 15
Java API, 10–11, 76
Java API classes, importing, 65–66
Java command-line tools, 17
Java Development Kit (JDK), 14, 18, 35
downloading, 18
installing, 18–19
JDK 9, 66
perusing the JDK folders, 19
setting the JAVA_HOME and path variables, 19–22
Javadoc
about, 379, 389
adding comments, 389–392
comments, 59, 390–392
`javadoc` command, 392–393
tags, 390
viewing pages, 393–395
Java Enterprise Edition (Java EE), 15
JavaFX
action event, 770–771
`Alert` class, 809–813
application launch, 766–767
`buttonClicked` method, 771
button creation, 768–769
class files, 760
click counter program, 774–779
The Click Me program, 762–764
command line, 761–762
exit a program
ClickCounterExit Program, 818–820
`close` method, 815
`CloseRequest` event, 816–818
right way to, 814
extending application classes, 765–766
file structure, 761
folder structure, 760, 761
input data from user
text field, 856–863
using check boxes, 865–867
using radio buttons, 867–869
validating numeric data, 863–864
layout panes, 771–772. *See also* layout panes, JavaFX
life cycle, 765–766
package imports, 764–765
package names, 760
possibilities of, 758–759
primary stage, 800, 805
root folder, 760–761
`Scene` class, 803–805
scene creation, 773
`SceneSwitcher` Program, 805–808
`setScene` method, 805
`setText` method, 769
setup, 759–762
`Stage` class, 800–803
stage setting, 773–774
`start` method, 767–768
syntax, 770
`javafx.application`, 764, 765, 777
javafx.controls module, 399
`javafx.event`, 782, 783, 787
`javafx.scene`, 759, 764, 769
`javafx.scene.layout`, 764, 772
`javafx.scene.layout` package, 840
`javafx.stage`, 764
JAVA_HOME and path variables, setting, 19–22
JAVA_HOME variable, creation, 20
`java.lang` package, 380–381
Java math, recognizing weird things about, 127
division by zero, 129–131
floating-point weirdness, 128–129
integer overflow, 127–128
Java Module System, 379, 395–396
Java Platform Module System, 66
Java program, creation of first, 23–24
Java project, creation, 33–36
Java Runtime Environment (JRE), 15
intercepts and folds errors, 13
`java.scene.control`, 765
`java.security.SecureRandom` class, 120
Java's HTTP client library
`HttpClient`, 743–744
`HttpRequest`, 744–747
`HttpResponse`, 747–748
`send` method, 748–749
`java.sql` module, 397
Java Standard Edition (SE), 15
Java Technologies page, 18
`java.time` package, 429
`java.util.concurrent.atomic`, 667
`java.util.function`, 706–710
consumers, 710
function, 709
interfaces in, 707–709
operators, 709–710
predefined functional interface extend, 710–712
predicate, 710
supplier, 710
`java.util.Random` class, 120
Java version, confirmation, 22
Java version 11, 758
Java Version Insanity, 13–14
Java Virtual Machine (JVM), 8, 9, 15, 648, 682
into memory, 47–48
JDK. *See* Java Development Kit (JDK)
`JOptionPane` class, 64, 65, 323
joy of methods, 200
JRE. *See* Java Runtime Environment (JRE)
JVM. *See* Java Virtual Machine (JVM)

K

keys, 569
`keySet` method, 579
key-value-pair, 491

keywords, dealing with, 50–53
K getKey() method, 580
kickBall, 319–320

L

lambda calculus, 686
lambda expressions, 686, 698–700, 793–798
  about, 367
  AddSubtract4 program, 795–796
  AddSubtract5 program, 796–798
  calculator program, 700–702
  passing multiple parameters, 702–703
  pass to method, 705–706
  use block lambda expression, 704–705
lambda operator, 695, 700
lambdas, 698–700, 705, 706
Lamm, Robert, 425
Last-In, First-Out (LIFO) collections, 562
lastIndex() method, 532
lastIndexOf() method, 546
late binding, 306
LaunchController program, 175
LaunchEvent class, 653–655, 658, 677–678, 681
LaunchEvent objects, 656, 660
LaunchEvent threads, 657, 662, 663, 682
launch method, 787
layout manager, 764
layout panes, 768, 771–773
layout panes, JavaFX
  adding space by growing nodes, 829–831
  adding space with margins, 828–829
  aligning nodes, 833–834
  Border, 822–824
    BorderPane class, 838, 840–841
    constructors and methods, 838, 839
    import statement, 840
    javafx.scene.layout package, 840
    regions, 838, 839
    traditional window arrangement, 838
  combinations, 824
  creation of, 823–824
  FlowPane, 822, 823
    button rearrangement, 838
    constructors and methods, 835–837
    horizontal flow layout, 834–835, 837
    import statement, 837
    javafx.scene.layout package, 837
    setPrefWrapLength method, 837
    vertical flow layout, 835
  GridPane, 822
    add method, 844
    ColumnConstraints Class, 845, 846
    constructors and methods, 842–844
    example, 847–853
    features, 841
    RowConstraints Class, 846, 847
    setColumnSpan method, 844
    setHalignment method, 844
    sketching out a panel, 841–842
  HBox, 822–831
    addAll method, 826
    adding space between nodes, 829–831
    adding space with margins, 828–829
    constructors and methods, 824–825
    creation of, 826
    import statement, 826
    Priority enumeration, 829, 830
    setHgrow method, 829, 830
    setMargin method, 828–830
    spacing and padding to pane, 826–828
  Insets class, 828
  Insets Constructors, 827, 828
  Pos enumeration, 833–834
  setAlignment method, 833
  setMargin method, 828
  spacing and padding, 826–828
  VBox, 822–824, 831–833, 840
    constructors and methods, 831–832
    getChildren method, 832
    import statement, 832
    spacing and padding, 833
leaf node, 494
left child, 496–497
lengthOfMonth method, 437
length of path, 495
lengthOfYear method, 437
life cycle, 249–250, 765–766
  JavaFX, 765–766
linear searching, 626, 631–635
linear time, 477–478
LinkedList class
  adding items to linked lists, 549–551
  vs ArrayList class, 544–545
  constructors, 545
  creating linked lists, 548–550
  methods, 545–548
  removing linked list items, 553
  retrieving items from linked lists, 551–552
  updating linked list items, 553
linked lists
  about, 481–482
  adding items to, 549–551
  creating, 548–550
  defined, 482
  deleting items, 487
  doubly, 488–490
  head, 483
  inserting items at head of list, 485–486
  iterating backward, 487
  performance characteristics of, 485
  printing elements of, 483–484
  removing items, 553
  retrieving items from, 551–552
  storage for, 487
  tail, 482
  traversing, 488
  updating, 552–553
LISP programming language, 244
listDirectories method, 605, 606

listener object, 321
load factor, 571
LocalDate class, 428, 434–436
LocalDate method, 431
LocalDate minusDays() method, 436
LocalDate minusMonths() method, 436
LocalDate minusWeeks() method, 436
LocalDate minusYears() method, 436
LocalDate object, 430
LocalDate of() method, 435
LocalDate parse() method, 435
LocalDate plusDays() method, 436
LocalDate plusMonths() method, 436
LocalDate plusWeeks() method, 436
LocalDate plusYears() method, 436
LocalDateTime class, 428
LocalDateTime method, 431–432
LocalTime class, 428
LocalTime method, 431
local variables, 71–72
local variable type inference, 85
log10(argument) method, 118
log(argument) method, 118
logarithmic time, 476–477
logical operator, 146, 147
  combining, 151–152
  Conditional And (&&), 148–149
  Conditional Or (||), 149–150
  Not (!), 146–148
  And (&) operator, 148–149
  Or (|) operators, 149–150
  Xor (^), 150–151
logic layer, 252
long division, 590–591
long factorial, 600
long longValue() method, 463
long nextLong(), 100
long-term support (LTS) version, 13, 14
LongUnaryOperator, 710–712
long until() method, 436
loops, 155. *See also specific types of loop*
  breaking out of, 158
  infinite, 158–162
  while. *See* while loop
LTS. *See* long-term support (LTS)

M

main method, 46, 49, 60, 70, 200, 202, 236, 282
making methods, 200–205
manifest file, 384, 388
map(), 723
Map.Entry interface, 570
Map interface, 570
maps
  about, 491–492, 569–570
  vs collections, 570
  defined, 491, 569
  elements, 569–570
  HashMap class, 570–580
  interfaces, 570
marker interfaces, 320
Math class, 115–116
  constants of, 116–117
  mathematical functions, 117–120
mathematical functions, 117–120
Math.random method, 121, 207
max(arg1, arg2) method, 118
members
  about, 261
  defined, 261
  fields, 261–262
  internal, 263
  protecting, 296–297
memory, 470
message, HTTP, 739–740
  body, 740
  one or more headers, 740
  start line, 739–740
method, defined, 46
method declaration, 200
  method body, 201
  method-name, 201
  parameter-list, 201
  public, 200–201
  return-type, 201
  static, 201
method header, 46, 47
method reference operators, 712, 713
method references, 705, 712–713
methods. *See also specific types of method*
  abstract, 313
  ArrayList class, 531–532
  Arrays class, 524
  vs constructors, 268
  default, 325
  defined, 259
  ensureCapacity(), 535
  equalsIgnoreCase, 405
  fill, 524–525
  final, 300
  HTTP, 740–741
  interface, 324–326
  Map.Entry interface, 580
  overloading, 266–267
  overriding, 295–296
  private, 262, 325
  private static, 325
  return type, 267
  signature of, 267
  using arrays with, 507–508
  values, 579
  visibility, 263
MIME. *See* Multipurpose Internet Mail Extensions (MIME)
min(arg1, arg2) method, 118
minus method, 438
minus operators, unary plus and, 109–110
missing return statement, 208, 209
models of computation, 470
modern computers, 685
modern Java programming basics, 45–46
  crafting comments, 58–59
  dealing with keywords, 50–53
  Greeter class, 63–65
  identifiers creation, 57
  importing Java API classes, 65–66
  object-oriented programming, 59–64

modern Java programming basics *(continued)*
pulling back the curtain, 48–49
revisiting Hello, World!, 46–47
running the Hello, World! Program, 47–48
unpacking `System.out.println` statement, 49–50
working with blocks, 56–57
working with statements, 53–55
modular JAR file, 396, 399–400
`module-info.class` file, 396, 399–400
`module-info.java` file, 397–398
modules, 396
compiling, 399
defined, 396
setting up folders for, 398–399
modulus operator, 106
`MonthDay` class, 428
`MonthDay` method, 432
`Month` enumeration, 432–433
`Month getMonth()` method, 435
multidimensional arrays, 515
Multipurpose Internet Mail Extensions (MIME), 738, 740, 748

N

names
of classes, 258–259
of packages, 381–382
of parameters, 267
`NavigableMap` interface, 570
nested class, 259
`nested for` loop, 177–178
`nested if` statement, 140–144
`nested` loop, 177
guessing game, 178–181
simple `nested for` loop, 177–178
von Neumann, John, 685
newline character, 404, 413
`new Random().nextDouble()`, 120
`nextIn` method, 221, 223, 226
`nextInt` method, 199
`next()` method, 537
nodes
ancestor, 496
child, 494
descendent, 496
height of, 495
leaf, 494
parent, 494
nonrecursive solution, 600
`non-sealed` keyword, 229
nonstatic methods vs static method, 60
Not (!) operator, 146–148
`now()` method, 428–429
`NumberFormat` class, 449, 600
`NumberPhobia2`, 161
`NumberPhobia` program, 161
numbers
arithmetic operators, 103–106
assignment operator, 112–114
compound, 114–115
binary operators, 105
decrement operators, 110–112
dividing integers, 106–108
division by zero, 129–131
floating-point, 128
formatting, 125–127
increment operators, 110–112
Math class, 115–116
constants of, 116–117
pseudorandom, 120
random, creation, 120–123

O

`Object` class
about, 333
clone method, 320, 346–357
equals method, 340–346
`extends` clause, 334
methods of, 335–336
object as type, 334–335
primitive types, 337
`toString` method, 337–340
object-oriented programming, 59, 219
about, 243–244
classes and objects, 59–60
creating an object from a class, 60–61
static methods, 60
viewing a program that uses an object, 61–64
objects, 10
about, 245
ArrayList, 533–534
behavior, 248–249
comparing, 344–345
creation from class, 60–61
Date-Time, 428–432
designing programs with, 252–253
determining type of, 303–304
exception, 219
identity of, 245–246
immutable, 438
life cycle of, 249–250
listener, 321
observable, 370
observer, 370
orientation of Java, 10
state, 247–248
as type, 334–335
type of, 246–247
viewing a program that uses an, 61–64
observable object, 370
observer object, 370
Observer pattern, 370
`offerFirst()` method, 546
`offerLast()` method, 546
`offer()` method, 546
`OffsetDateTime` class, 428
`OffsetDateTime` method, 432
`OffsetTime` class, 428
`OffsetTime` method, 432
`of()` method, 431–432
Olivetti Programma 101, 9
omitting expressions, `for` loop, 175–176
`openFile` method, 235
operands, 103
number of, 105
operator, 146–152
arithmetic, 103–106
assignment, 112–115
binary, 105
boolean, 146
categorizing, 105

combining, 108–109
compound assignment, 114–115
conditional, 105, 152–153
decrement, 110–112
increment, 110–112
in `java.util.function`, 709–710
logical. *See* logical operator
prefix, 105
ternary, 105
unary, 105
unary plus and minus, 109–110
Oracle Java website, 18, 25
Or (|) operator, 149–150
or-or (||) operator, 417
outputs, of algorithm, 584
overloading methods, 266–267
overriding
equals method, 342–345
methods, 295–296
`toString` method, 339–340

P

package imports, 764–765
package names, 760
package-private visibility, 263
packages
about, 379
archiving, 386–387
creating, 381–383
default, 381
defined, 380
importing, 380–381
`java.lang`, 380
names, 381–382
parameter, 212–213
declaring, 213–214
names of, 267
pass-by-value, 215
scoping out, 214–215
@param tag, 390
parent, 496
parent class, 251, 290
parent node, 494
`parse()` method, 430–431
partitioning, 616
`partition` method, 617–620, 623
pass-by-value, 215
path analysis, 494–495
`Path` class, 603
path variable, editing, 21
pattern matching, 362–363
`peekFirst()` method, 546, 552
`peekLast()` method, 547, 552
`peek()` method, 546, 551, 560
Perl, 11
`permits` keyword, 328
pipelines, Streams API, 717–719
`ArrayList.stream()` method, 721
`forEach` method, 718
intermediate operations, 717
lazy evaluation, 719
`map` method, 718
source method, 717
`stream` method, 718
terminal operations, 717
pivot point, 615–616
platform independence of Java, 8–9
platform threads, 682
`playARound` method, 203–205, 209, 216
`plus` method, 438
`pollFirst()` method, 547, 552
`pollLast()` method, 547, 552
`poll()` method, 547
polymorphism, 304
polynomial time, 477–478
`pop()` method, 547, 552, 560
`Pos` enumeration, 833–834
`pow(arg1, arg2)` method, 118
predicate, in `java.util.function`, 710
prefix operator, 105
presentation layer, 252
primitive data types
boolean type, 82
char type, 80–81
floating-point types, 78–80
inferring the data type, 85–86
integer types, 77–78
vs reference types, 76, 83–84
working with, 76–86
wrapper classes, 82–83
primitive types, 337
`PrintArray` method, 613, 622, 623
`printBoard` method, 523
printing
array lists, 536
data with `System.out`, 95–97
standard error, 96
standard input, 95
standard output, 96
linked list elements, 483–484
`println` method, 199
`PrintList` method, 484
`println` method, 273
`print` method, 199
`printResult` method, 705–706, 712
`PrintSomeWords` method, 509
`Priority` enumeration, 829, 830
private fields, 263
`private` method, 262, 325
`private static` method, 325
private visibility, 263
`ProductDataException` class, 309
program, elements, 753–754
programming threads, 647–648. *See also* countdown clock
creation of, 649
examples of, 648–649
understanding, 648–649
programs and code listings. *See specific types*
Project Jigsaw, 395–396
property bindings, 759
`protected Object clone()` method, 335
protected visibility, 263, 294
`protected void finalize()` method, 335
provider, 734, 735
pseudorandom numbers, 120
`public` class, 261
public fields, 248, 262
public interfaces, 263
`public` keyword, 261
public visibility, 263
pure functional programming language, 687–690
pure functions, in Java, 689
`push()` method, 547, 560
`put` method, 575

Q

queue, 490, 564–567
Quicksort algorithm, 611, 615–617
  partition method, 618–620
  sort method, 617–618
Quicksort program, 620–623
quintuplet, 475
quotation mark, 404

R

radio buttons, 867–869
random access memory (RAM), 470–474
randomInt method, 121
random() method, 118, 121
random-number generator, 208
random numbers creation, 120–123
recite()method, 692–693, 697, 698, 702, 705, 712
records
  creating, 275–278
  deconstruction, 364–365
  defined, 275, 474
  using, 275–278
recursion, 588, 599
  classic factorial example calculation, 599
    nonrecursive solution, 600
    recursive solution, 600–602
  directories, 602–606
  Towers of Hanoi game, 606–608
    application, 608–609
recursive algorithm, 640–642
recursive solution, 600–602
ReentrantLock class, for thread safety, 672–675
reference data types vs primitive data types, 76
reference types, 83–84
registers, 473
relational operators, 359
  boolean expressions, 134
removeAll() method, 532, 541, 547
removeFirst() method, 547, 552
removeFirstOccurrence() method, 547
removeLast() method, 547, 552
removeLastOccurrence() method, 547
remove() method, 532, 537, 541, 547
repeated subtraction, 587, 588, 591
replaceAll method, 414–415
replaceFirst method, 414–415
replace() method, 420
repository, 28, 30
  creation, GitHub, 28–31, 34–35
  HelloApp, 31
REpresentational State Transfer (REST), 735
representations, 245
resizable arrays, 480
resources, 737
RESTful applications, 735, 736
  cachability, 736
  client–server architecture, 735
  code on demand, 736
  layered system, 736
  statelessness, 735
  uniform interface, 736
results, of algorithm, 584–585
retainAll() method, 532, 541, 548
retrieving
  elements from hash maps, 576
  items from linked lists, 551–552
return statement, 207–209, 230
  to return values, 206–207
@return tag, 390
return type, 267
return values, 205–212
  declaring the method's return type, 205–206
  return statement to, 206–207
reverse() method, 420
right child, 496–497
rint(argument) method, 123
Roman numerals, 593–594
root directory, 382
root folder, 760–761
round(argument) method, 123
rounding functions, 123–125
RoundingMode.CEILING, 459
RoundingMode.DOWN, 459
RoundingMode.FLOOR, 459
RoundingMode.HALF_DOWN, 459
RoundingMode.HALF_EVEN, 459
RoundingMode.HALF_UP, 459
RoundingMode.UNNECESSARY, 460
RoundingMode.UP, 460
RowConstraints Class, 846, 847
run method, 651–655
Runnable interface, 652–654, 689, 706
  CountDownApp class, 655–657
  create class, 654–655
Runnable object, 654
running modular JAR file, 400
RuntimeException, 308

S

sales tax calculator, 450–451, 457, 463–465
Scanner class, 221, 222, 227
  getting input values, 99–101
  importing, 99
  input with, 98
Scanner object, declaring and creating, 99
Scene class, 799
  commonly used constructors and methods, 804
  getHeight methods, 804
  getWidth methods, 804
  Parent class, 804
  root node, 804
  size of scene, 805
  Width and Height parameters, 804
scene creation, 773
SceneSwitcher Program, 805–808
schedule tournaments, 596–598
scientific notation, 79
scope of a variable, 92–94
scoping out parameter, 214–215
SDK. *See* software development kit (SDK)
sealed classes, 327–332
sealed keyword, 328
searching, 625–626
  Baby Name search program, 642–644

using binary search, 637–640
using linear search, 633–635
binary search, 635–644
create a list of baby names, 629–631
data structure creation, 626–631
linear searching, 631–635
session, 738
set accessors, 264
setAlignment method, 833
setCharAt() method, 420
setColumnSpan method, 844
Set entrySet() method, 572
setHalignment method, 844
setHgrow method, 829, 830
Set keySet() method, 572
setLength() method, 420
setMargin method, 828–830
setMaximumFractionDigits(int) method, 126
set() method, 532, 548
setMinimumFractionDigits(int) method, 126
setOnAction method, 784, 793, 794, 798
setOnCloseRequest method, 817
setPrefColumnCount method, 857
sets, 475
setScale() method, 457, 459
setScene method, 773, 778, 805
setSpacing method, 827
setters, 264–265
setText method, 769, 857
setting up folders for module, 398–399
setTitle method, 769, 774, 778
shadowing variables, 94–95
Shakespeare, William, 666, 799, 816
shallow copies, creating, 350–351
short nextShort(), 100
showAndWait method, 811
signature, of methods, 267
signum(argument) method, 118
Simula, 244
@since tag, 390
single case group, two or more values in, 192–193
single-elimination tournament, 596, 597
singleton class, 280, 284–285
size() method, 532, 548, 560, 565
snippets, 394
SoftballBall class, 328
software development kit (SDK), 15, 760
solvable problem, of algorithm, 583
SortArray method, 612–614
sorted(), 723–724
SortedMap interface, 570
sorting, 611
Bubble Sort algorithm, 611–615
program, 621–623
Quicksort algorithm, 615–617
partition method, 618–620
sort method, 617–618
Quicksort program, 620–623
sort method, 617–618, 622, 623
spacecraft, 648, 649, 655, 657
spell out dollar amount, 595
split command, 413
sqrt(argument) method, 118
SquareIt class, 696–698
stacks, 490
stack trace, 222
Stage class, 799
commonly used methods, 801–802
ins and outs, 802–803
primaryStage parameter, 800
stage setting, 773–774
start method, 767–768
Star Trek series (Kirk and Spock), 27
statements
types of, 54–55
white space, 55
working with, 53
static array copyOf() method, 524
static array copyOfRange() method, 524
static design, 279
static fields
about, 279–280
common uses, 280
counting instances, 283
preventing instances, 286
singleton pattern, 284–285
working with, 281
static initializers, 249, 259, 275, 281, 287
static inner classes, 372–373
static int binary() method, 524
static keyword, 281
static method, 325, 694–695
about, 279–280
common uses, 280
counting instances, 283
vs nonstatic method, 58
preventing instances, 286
Singleton pattern, 284–285
using, 282
static String toString() method, 524
static variables, 69
static void fill() method, 524
static void setAll() method, 524
static void sort() method, 524
Stream interface, 719–721
intermediation operations, 719, 720
methods, 720–721
source operations, 719, 720
terminal operations, 719, 721
stream() method, 721–722
streams API
The Daisy Hill Sloth Sanctuary Program, 725–730
operations, 716–724
Optional class, 730–731
StringBuffer class, 403, 418–421
StringBuilderApp program, 421
StringBuilder class, 403, 418–421
StringBuilder method, 419–421
String class, 154
documentation page for, 25
methods, 406–408
String getMessage(), 230
String indent(int) method, 406
StringIndexOutOfBoundsException, 409
String nextLine(), 100
String repeat() method, 407
String replaceAll() method, 407

String replaceFirst() method, 407
String replace() method, 407
strings, 86
  about, 403–405
  Car Wash program uses, 195–196
  combining, 87–88
  comparing, 153–154
  converting arrays to, 528
  converting BigDecimals to, 463
  converting primitives to, 88–89
  converting strings to primitives, 89
  declaring and initializing, 86–87
  determining whether empty, 416–417
  empty, 416–417
  escape sequences, 404
  extracting characters from, 409–410
  extracting substrings from, 411–412
  finding length of, 409
  modifying, 409
  replacing parts of, 414–415
  splitting, 413–414
  stripping, 415–416
  switching with, 195–196
  text blocks, 422–424
String split() method, 407
String stripLeading() method, 407
String strip() method, 407
String stripTrailing() method, 407
String substring() method, 407, 408, 420
String toLowerCase() method, 408
String toPlainString() method, 463
String toString() method, 230, 335, 408, 420, 463
String toUpperCase() method, 408
String trim() method, 408
String trimToSize() method, 421
String valueOf() method, 408
stripLeading method, 415–416
strip method, 409–416
stripTrailing method, 415–416
structure charts, 253
subclasses
  about, 251, 290
  creating, 293–294
  using super keyword in, 297–298
  using this keyword in, 297–298
subdirectories, creating, 382
subinterface, 319
subSequence method, 422
substrings, extracting from strings, 411–412
subtractive notation, 594
superclass, 251, 290, 299
superinterface, 319
super keyword, 297–298
supplier, in java.util.function, 710
swallowing exceptions, 237–238
swap method, 620, 623
switch expression, Car Wash program uses, 198
switch statement, 133, 183, 187, 188
  assigning a value with, 196–198
  flowchart for, 189
  if statements inside, 190–191
symbol table, 491
synchronized keyword, for thread safety, 670–672
syntax, 770
System.err, 97–98
System.out, 97–98
System.out.print, 125
System.out.println(), 49–50, 95–97, 125, 175, 441

T

tabs, 404, 413
tail, 483
TenPennies, 448
ternary operators, 105
test expression, for loop, 168, 170
text blocks, 422–424
text field
  constructors and methods, 856
  getText method, 857
  import statement, 856
  Role Player Program, 859–863
  setPrefColumnCount method, 857
  setPromptText method, 858
  setText method, 857
  TextField class, 856
this keyword, 272–273, 297–298
Thread class, 650, 653
  constructors and methods of, 650
  create and start thread, 652
  extend of, 651–652
thread creation, for countdown clock, 649
thread interruption, countdown clock, 676
  to abort countdown, 677–682
  checking, 676–677
Thread object, 652, 653
threads, 647, 648. *See also* countdown clock; programming threads
thread safe, countdown clock, 661–663
  AtomicInteger class for, 666–670
  ReentrantLock class for, 672–675
  synchronized keyword for, 670–672
  volatile keyword for, 663–666
Thread.sleep method, 654–655, 661
three-dimensional arrays, 479, 515
3-D objects, 759
Throwable class, 307
throwBall, 319–320
throwFor method, 222
throwing exceptions, 236, 238–239
throws keyword, 233
@throws tag, 390
throw statement, 233
TickTockAnonymous program, 375–376
TickTockInner class program, 369–371
TickTock program, 322–323
tic-tac-toe game, 585, 595
time, 426–427. *See also* Date-Time API

TimeMonitor interface, 660
Timer class, 321, 323
Timer class constructor, 321
toArray() method, 532, 548, 629
toLowerCase method, 409
toString method, 336–340, 422, 429, 528–529
touch interface, 759
toUpperCase method, 409
the Towers of Hanoi application, 608–609
the Towers of Hanoi game, 606–608
traditional comments, 58–59
trees
  defined, 494
  degree of, 496
  levels of, 495
  parts of, 494
  size of, 495
  terminology, 494–495
trim method, 409
try block, 223, 225, 230
try–catch–finally statement, 240
try–catch statement, 224
try statement, 219, 223, 227, 229
tryToChangeNumber method, 215
try-with-resources statement, 239–240, 327
tuple, 475
Turing, Alan, 685, 686
Turing machine, 685
two-dimensional arrays
  about, 479, 509–510
  accessing elements of, 511–512
  creating, 510–511
  defined, 509
  initializing, 512–513
  jagged arrays, 513–514
The Two Noble Kinsmen (Shakespeare and Fletcher), 666
type checking, 11–12

U

unary minus operators, 109–110
unary operators, 105
unary plus operators, 109–110
unchecked exception, 221
Unicode encoding scheme, 81
Unified Modeling Language (UML), diagramming classes with, 253–256
Uniform Resource Identifier (URI), 736, 738, 743, 745, 752, 754
Uniform Resource Locator (URL), 734, 736, 738, 740, 741, 751, 752, 754
until method, 438
upcasting, 301
UpdateByKey, 491
URI. *See* Uniform Resource Identifier (URI)
URL. *See* Uniform Resource Locator (URL)
UTF-16, 81

V

values maps, 569
values method, 579
varargs, 508–509
variable declaration, 68, 69
  class variables, 69–70
  instance variables, 70–71
  local variables, 71–72
  two or more variables, 69
variables, 67
  final variables (constants), 75
  initializing, 72–73
    with assignment statements, 73–74
    with initializers, 74
  instance, 254
  scope of, 92–94
  shadowing, 94–95
  visibility indicators for, 255
VBox, 822–824, 840
  constructors and methods, 831–832
  getChildren method, 832
  import statement, 832
  spacing and padding, 833
V computeIfPresent() method, 573
V compute() method, 573
vectors, 480
@version tag, 390
vertical bar, 413
V get() method, 572
V getOrDefault() method, 572
V getValue() method, 580
virtual machine, 8
VirtualMachineError, 307
virtual threads, 682–683
visibility, 263, 665
  indicators, 254–255
  keywords, 281
  levels of, 263
  of packages, 396
visual effects, 759
V merge() method, 573
void forEach() method, 573
void notifyAll() method, 335
void notify() method, 335
void printStackTrace(), 230
void putAll() method, 572
void replaceAll() method, 573
void wait() method, 335
volatile, description of, 665–666
volatile keyword, for thread safety, 663–666
V putIfAbsent() method, 572
V put() method, 572
V remove() method, 572
V replace() method, 572
V setValue() method, 580

W

web browsers, 648
web services
  characteristics of, 734–735
  consumers, 734, 735
  HTTP tester program, 751–754
  Java's HTTP client library, 742–749
  program integration, 749–751
  provider, 734, 735
  understanding HTTP, 736–742
website
  javadoc documentation, 396
  java.lang package, 380
  Oracle's Java, 18, 23

WebSocket, 743
while loop, 156, 163, 227, 228, 567, 619
  counting loop, 156–157
  flowchart for, 157
while statement, 156
white space, 55, 413, 414
Wikipedia, 734
wildcard-type parameters, 562–563
word processors, 648
word RAM model of computation, 470–474
World Wide Web, 733, 734, 737
wrap around effect, 128
wrapper classes
  boxing, 91–92
  for primitive types, 82–83
  using, 91

X

Xor (^) operator, 150–151

Y

Year class, 428
Year method, 432
YearMonth class, 428
YearMonth method, 432

Z

zero
  division by, 129–131
  floating-point, 130
ZonedDateTime class, 428, 433
ZoneId class, 433
ZoneOffset class, 434