# Index

**Note to the Reader:** Throughout this index **boldfaced** page numbers indicate primary discussions of a topic. *Italicized* page numbers indicate illustrations.

---

**A**

absolute() method, 530–532, 530–531
absolute paths, 408–409
  checking for, 465–466
  vs. relative, 457
relativize(), 467–468
resolve(), 469
toRealPath(), 470
abstract classes, 5–6
accept() method, 211, 213
access modifiers, 2–4
accessibility, file, 480
accumulator parameter for reduce(),
  373–375
add() method
collections, 125–126
lists, 130
queues, 135
sets, 133
afterLast() method, 529, 529, 532
allMatch() method, 192
ambiguous lambda expressions, 342–343
angle brackets (->) for generics, 109
annotating overridden methods, 11–13
anonymous inner class, 29–31
anti-patterns, 86–87
anyMatch() method, 192
Apache Commons Lang library
equals(), 17
toString(), 14–15
applications. See JDBC (Java Database Connectivity)
apply() methods, 212–213
arguments in NIO.2, 461–462
ArithmeticException class, 286
ArrayDeques
  applications, 142
description, 135
  methods, 135–138, 137
ArrayIndexOutOfBoundsException class, 286
arrays and ArrayLists
  applications, 142
Atomic class, 353
bounds, 117–122
diamond operator, 107–108
implementations, 128–129
overview, 104–105
searching and sorting, 105–106
wrapper classes and autoboxing, 106–107
ArrayStoreException class, 287
arrow operator (->) for lambda expressions,
  56–57, 57
assert statements, 308–309
AssertionError class, 308–309
assertions
  assert statements, 308–309
  enabling, 309–310
  exam essentials, 315
  outcomes altered by, 312–313
  overview, 308
  review questions, 316–323
  summary, 314
  working with, 310–314
Atomic classes, 352–354, 355
ATOMIC_MOVE value, 461–462
atomic operations, 461
AtomicMoveNotSupportedExceptions, 462
attributes
  files, 478–486
  modifying, 486
  reading, 485
autoboxing, 106–107
AutoCloseable interface, 300–302
automatic resource management, 297
averagingDouble() method, 217
averagingInt() method, 217
averagingLong() method, 217
await() method, 379–380
awaitTermination() method, 344–345
base cases in recursion – access modifiers

base cases in recursion, 381–382
base classes in streams, 415–416
BasicFileAttributes class, 484–485
BasicFileAttributeView class, 484, 486
beforeFirst() method, 529, 529, 532
BiConsumer interface, 175–177
BiFunction interface, 178–179
big O notation, 128–129
BigDecimal type, 268
binarySearch() method, 106, 15
BinaryOperator interface, 180
BiPredicate interface, 177–178
blocking private variable access, 68–69
blocking queues, 361–363
blocks
    streams, 411–412, 412
    synchronized, 354–356
boolean type
    assert statements, 308
    Atomic class, 353
    functional interfaces, 211
    ResultSet, 527
    SQL statements, 523
    wrapper class, 106
Boolean wrapper class, 106
BooleanSupplier type, 211
bounds for generics, 117–122
braces {} for lambda expressions, 58–59
breadth-first searches, 488
buffer size, 423
Buffered class, 415
BufferedInputStream class
    high-level streams, 415–416
    working with, 422–423
BufferedOutputStream class, 422–423
BufferedReader class, 437
    description, 417
    vs. FileReader, 414
    working with, 424–426
BufferedWriter class
    description, 417
    high-level streams, 414–416
    working with, 424–426
builder pattern, 86–89
built-in functional interfaces
    checking, 181–182
    Consumer and BiConsumer, 175–177
    default methods, 178
    Function and BiFunction, 178–179
    overview, 173–174
    Predicate and BiPredicate, 177–178
    Supplier, 174–175
    UnaryOperator and BinaryOperator, 180–181
byte streams vs. character streams, 413–414
Byte wrapper class, 106
bytes in streams, 412

call() method, 342
Callable interface
    checked exceptions, 344
    overview, 342–344
    vs. Supplier, 342–343
cancel() method, 340
casting references, 64–65
catch clauses
    multi-catch, 291–296, 293
    overview, 288–289, 288
ceiling() method, 134
chaining
    high-level streams, 414
    Optionals, 214–217
char type for wrapper class, 107
collection, 215–216
circular paths in directory walking, 490
class design
    abstract classes, 5–6
    access modifiers, 2–4

ChronoUnit class, 248–249
checkError() method, 433
Character wrapper class, 107
characteristics() method, 376
checked exceptions, 285–286
    Callable and Runnable, 344
    functional interfaces, 215–216
    OCP, 286–287
character streams
    benefits, 413
    vs. byte streams, 413–414
characteristics for directories, 484–486
Characteristics() method, 376
checkError() method, 433
ChronoUnit class, 248–249
circular paths in directory walking, 490
class design
    abstract classes, 5–6
    access modifiers, 2–4
annotating overridden methods, 11–13

design patterns and principles. See design patterns and principles

eenums, 20–24
equals(), 15 18
exam essentials, 34–35
hashCode(), 18–20
imports, 6–7
instanceof operator, 7–9
nested classes. See nested classes
overloading and overriding, 4–5
review questions, 36–46
summary, 33–34
toString(), 13–15
virtual method invocation, 9–11
class invariants with assertions, 310
ClassCastException class, 65, 109, 286
classes
 concurrent, 360–361
genrics, 109–111
 resource bundles, 262–263
 system, 309
ClassNotFoundException class, 431
cleaning up paths, 469–470
clear() method
collections, 127
 maps, 139
close() method
 FileReader and FileWriter, 424
 streams, 414
Closeable interface, 302
closing
database resources, 533–535
 resources, 297
 streams, 418
collect() method
 combining results, 375–377
 streams, 194–196
collections, 104
 arrays and ArrayLists, 104 108
 common methods, 125–127
 comparing, 140–143
 concurrent. See concurrent collections
 exam essentials, 161
 generics, 115
 interfaces, 124–125, 125
 List interface, 127–131, 128
looping through, 155
Map interface, 138–140, 138
Queue interface, 134–138, 135, 137
review questions, 162–170
Set interface, 132–134, 132–133
summary, 159–161
tasks, 339–340
collectors
 basic, 218–219
 examples, 218–220
 grouping, partitioning, and mapping, 220–223
 maps, 219–220
Collectors class, 376
colons (:)  
 JDBC URLs, 514–515, 514
 for parameters, 153
 property files, 260
 in time formats, 272
columns in ResultSet data, 526–529
 combiner parameter for reduce(), 373–374
commas (,)
 date formats, 272
 numbers, 269–270
comments in property files, 260–261
common methods for collections, 125–127
Comparable interface
 vs. Comparator, 148
 as functional interface, 148
 overview, 143–146
 for searching and sorting, 150–152
Comparator interface
 vs. Comparable, 148
 overview, 146–150
 for searching and sorting, 150–152
compareTo() method, 143–146
comparing multiple fields, 149–150
compiler warnings, 115–116
composing objects, 74 75
compute() method, 382–383, 387
computeIfAbsent() method, 157–159
computeIfPresent() method, 157–159
concatenation of strings, 253–254
CONCUR_READ_ONLY mode, 520
CONCUR_UPDATABLE mode, 520
concurrency and Concurrency API
 Callable interface, 342–344
collections. See concurrent collections
exam essentials, 393
ExecutorService. See ExecutorService
interface
overview, 326
parallel streams, 366–377
pools, 348–350
processes, 377–387
race conditions, 391–392, 391
review questions, 394–404
summary, 392–393
synchronizing data access, 350–358,
351–352
threads. See threads
concurrency mode in ResultSet, 520
concurrent collections
blocking queues, 361–363
classes, 360–361
CopyOnWrite collections, 364
memory consistency errors, 359–360
overview, 358–359
parallel streams, 372
SkipList classes, 363
synchronized, 365–366
concurrent processes, 377
CyclicBarrier, 377–381
fork/join framework, 381–387
RecursiveTask, 385–386
ConcurrentHashMap class, 359–360
ConcurrentLinkedDeque class, 360
ConcurrentLinkedQueue class, 360
ConcurrentModificationException class, 359
ConcurrentSkipListMap class, 360, 363
ConcurrentSkipListSet class, 360, 363
conditional removing, 154
connecting to relational databases, 513–518,
514
Connection interface
description, 512, 512
working with, 515–518
consistency errors, memory, 359–360
consistent methods, 16
Console class, 433, 438–442
core() method, 438
constant time in big O notation, 129
constants, enums for, 20–22
constructor references, 154
constructors
dates and times, 239–240
with enums, 22–24
factory pattern, 92
generics, 113
immutable objects, 85
Consumer interface, 175–177
contains() method, 127
containsKey() method, 139–140
containsValues() method, 139
tables for processes, 329
control flow invariants with assertions, 310
Coordinated Universal Time (UTC), 236
COPY_ATTRIBUTES value, 461
copy() method
FileInputStream and FileOutputStream, 422
Files, 473–475
copying file contents, 473–475
CopyOnWrite collections, 364
CopyOnWriteArrayList class, 360
CopyOnWriteArraySet class, 360
cost, synchronization, 357–358
COUNT() function, 511
count() method, 190
counting() method, 217
country codes, 256–257
covariant return types, 5
createDirectories() method, 473
createDirectory() method, 473
createStatement() method, 518
creational patterns, 76
creationTime() method
Files, 485
views, 484
currency
formatting, 267–268
parsing, 268–270
cursors in ResultSet, 524, 524
cycles from symbolic links, 490
CyclicBarrier class, 377–381

D
daemon threads, 328
data models, 66
databases. See relational databases
dataInputStream class, 437
DataOutputStream class, 437
DataSource class, 515, 517
dates and times
creating, 235–241
daylight savings time, 251–253, 251
durations, 247–250
exam essentials, 247–250
file modifications, 481–482
formatting, 270–273
instants, 250–251
JDBC, 528
manipulating, 241–244
overview, 234
periods, 244–247, 246
ResultSet, 527
review questions, 276–282
summary, 273–274
java.time package
DateTimeException class, 240, 287
DateTimeFormatter class, 270–273
daylight savings time, 251–253, 251
days
in date formats, 272
with of(), 246–247, 246
java.util.DAYS values, 342
deadlocks, 388–390
debugging generics, 222
decimal points for numbers, 269
DECIMAL type in SQL, 510
declare rules for exceptions, 285
decrementAndGet() method, 353
default methods
functional interfaces, 178
interfaces, 51
defaultCharset() method, 476
deferred execution, 56
defining functional interfaces, 53–54
delete() method
File, 409
Files, 475–476
DELETE statements in SQL, 510–511
deleteIfExists() method, 475–476
depth-first searches, 488
deserialization
object creation, 431–432
objects, 430
streams, 426
Design Patterns, 92
design patterns and principles, 48
builder, 86–89
capability, 66–68
case class, 91
encapsulation, 66–68
decorator pattern, 52–60
factory, 89–92, 91
functional programming, 52–60
has-a relationships, 73–74, 73
immutable objects, 82–86
interfaces, 48–52
is-a relationships, 71–73, 71–72
JavaBeans, 69–71
object composition, 74–75
polymorphism, 61–65, 64
review questions, 95–102
singleton, 76–82
summary, 93
working with, 75–76
diamond operator, 107–108
directories
attributes, 478–480
File class, 407–411, 407
class paths, 406–407, 407
class, 406–407, 407
class, 406–407, 407
listing contents, 491–492
making, 473
searching, 488, 490–491
walking, 487, 490, 487
java.nio.file.Files
DirectoryNotEmptyException class, 475
disabling assertions, 310
distinct() method, 196–197
dollar signs ($) for currency, 270
DosFileAttributes class, 484
dos file metadata, 484
dots (.)
numbers, 269
paths, 467
double-checked locking, singleton patterns
with, 81–82
double type
functional interfaces, 211–213
for money, 268
ResultSet, 527
wrapper class, 107
Double wrapper class, 107
DoubleBinaryOperator interface, 212
DoubleConsumer interface, 211
DoubleFunction interface, 212
DoublePredicate interface, 211
DoubleStream class, 205–208
DoubleSupplier interface, 211
DoubleToIntFunction interface, 213
DoubleUnaryOperator interface, 212
downstream collectors, 221
Driver interface, 512, 512
DriverManager class, 515–517
durations and Duration class, 247–250

E
Eclipse tool, 80
driver interface, 512
edge cases in daylight savings time, 252
effectively final concept, 28, 172
empty Optionals, 183
enabling assertions, 309–310
encapsulation
JavaBeans, 69–71
overview, 66–68
enums
constructors, fields, and methods, 22–24
in switch statements, 21–22
working with, 20–21
EOFException class, 431
ePOCH date, 245
equals() method, 15
and compareTo(), 146
contract, 16–17
writing, 17–18
Error types, 286
event handlers, inner classes as, 31
exceptions
categories, 285–286, 285
checked, 215–216
creating, 289–291
exam essentials, 315
JDBC, 535
multi-catch, 291
on OCP, 286–288
overview, 284
rethrowing, 305–307
review questions, 316–323
summary, 314
suppressed, 302–304
terminology, 284–285
throw vs. throws, 289
try statements, 288–289, 288
try-with-resources, 296–304, 298
execute() method
SQL statements, 521, 523
vs. submit(), 338
tasks, 338
threads, 335
executeQuery() method, 521, 523–524
executeUpdate() method, 521, 523
executing SQL statements, 520–523
Executors class, 348–350
ExecutorService interface, 335
pools, 348–350
shutting down thread executors, 337–338, 337
single-thread executors, 335–336
submitting tasks, 338–340
task results, 340–345
existence, file, 469–470
eXists() method
File, 409
Files, 471
explicit casts, 64
extending interfaces, 51

F
factorials, 382
factory classes, instances with, 455–456, 456
factory pattern, 89–92
fields with enums, 22–24
FIFO (first-in, first-out) queues, 135
File class, 407
vs. Files, 471
object creation, 408–409
working with, 409
file systems, 406–407
FileInputOutputStream class
description, 417
and FileOutputStream, 414
vs. FileReader, 413
low-level streams, 415
working with, 420–422
fileKey() method, 485
FileNotFoundException class, 287, 421
FileOutputStream class
description, 417
and FileInputStream, 414
working with, 420–422
FileReader class
vs. BufferedReader, 417
description, 417
vs. FileInputStream, 413
overview, 424
files
accessibility, 480
attributes, 478–486
copying contents, 473–475
directory creation, 473
directory walking, 487–490, 487
event, 469–470
File class, 407–411
file systems, 406–407, 407
interacting with, 471
length, 481
moving, 475
ownership, 482–483
path testing, 471–472
printing, 492–493
reading, 476–478
removing, 475–476
time and date modifications, 481–482
uniqueness, 472–473
views, 483–486
visibility, 480
writing, 476–477
Files class
vs. File, 471
file attributes, 478–486
file interaction, 471–478
helper classes, 456, 456
FileSystem class, 459
FileTime class, 481
FileVisitor interface, 489
FileWriter class, 414
description, 417
working with, 424
filter() method, 196
FilterInputStream class, 437
FilterOutputStream class, 437
final modifiers
with interfaces, 51
overview, 6
finally clauses, 288–289, 288, 298
find() method, 490–491
findAny() method
order-based tasks, 372–373
streams, 191
findFirst() method
order-based tasks, 373
streams, 191
finite streams, 186, 188–189
“fire-and-forget” methods, 338
first-in, first-out (FIFO) queues, 135
first() method, 529, 529, 532
flatMap() method, 198
flatMapToDouble() method, 208
flatMapToInt() method, 208
first() method, 529, 529, 532
flatMapToDouble() method, 208
flatMapToInt() method, 208
Float wrapper class, 107
floor() method, 134
flush() method
Console, 440
FileReader and FileWriter, 424
overview, 418
flushing streams, 418
FOLLOW_LINKS value, 461, 490
forEach() method
independent operations, 370
streams, 192–193
forEachOrdered() method, 368, 372
Fork/Join class, 386–387
core/join framework, 381–387
fork() method, 386–387
ForkJoinTask interface, 382
formal type parameters for generics, 109–111
format() method
Console, 433, 439
dates and times, 270–271
numbers, 267
PrintStream, 433–434
formatting
currency, 267–268
dates and times, 270–273
dates and times, numbers, 267–268
forward-only ResultSets, 524
fromMillis() method, 481
Function interface, 178–179
function package, 174
functional interfaces and programming, 172
getPercentInstance() method

G

Gamma, Erich, 92
Gang of Four, 92
generate() method, 206
generics
bounds, 117–122
classes, 109–111
debugging, 222
exam essentials, 161
example problems, 122–124
interfaces, 112–113
legacy code, 114–117
limitations, 113
methods, 114
naming conventions, 110
overview, 108–109
review questions, 162–170
summary, 159–161
supertypes, 122
type erasure, 111–112
get() method
Atomic, 353
get() methods in ResultSet, 527
getAbsolutePath() method, 409
getAndDecrement() method, 353
getAndIncrement() method, 353
getAndSet() method, 353
getAsBoolean() method, 211
getAsDouble() method, 209, 211
getAsInt() method
IntSupplier, 211
getAsLong() method, 209, 211
getAvailableZoneIds() method, 239
getBundle() method, 261, 263–264
getConnection() method, 515–516
getCurrencyInstance() method, 267–268
default() method
FileSystems, 482
Locale, 256–258
getDelay() method, 346
getFileAttributeView() method, 484, 486
getFileName() method, 464–465
getFileSystem() method, 482
getInstance() method, 267
getIntegerInstance() method, 267
getLastModifiedTime() method, 481–482, 485
getMessages() method, 536
getName() method
File, 409
Path, 463
getNameCount() method, 463
getNumberInstance() method, 267
getObject() method, 528–529
getOwner() method, 482–483
gerParent() method
File, 409
Path, 464–465
gerPath() method
FileSystem, 459
Path, 459
Paths, 457
gerPercentInstance() method, 267
lists, 130
maps, 139
Optional, 184
Paths, 457–459
tasks, 340
getProperty() method
resource bundles, 262
System, 457
getRoot() method, 464–465
getSQLState() method, 536
getTime() method, 528
getTimestamp() method, 528
GMT (Greenwich Mean Time), 236
Greenwich Mean Time (GMT), 236
grouping collectors, 220–223
groupingBy() method, 217, 220–222
groupingByConcurrent() method, 376–377

H
handles, exception, 285
has-a relationships, 73–74, 73
hashCode() method
sets, 133
working with, 18–19
writing, 19–20
HashMaps
applications, 142
description, 138–139
HashSets
applications, 142
description, 132–133, 133
Hashtable class, 138
Helm, Richard, 92
helper classes, instances with, 455–456, 456
high-level streams vs. low-level, 414–415
higher() method, 134
hours
in methods, 238
in time formats, 272–273
HOURS values, 342

I
I/O fundamentals
exam essentials, 443–444
files and directories. See directories; files
NIO.2. See NIO.2 API
overview, 406
review questions, 445–451
streams. See streams and Stream interface
summary, 442
user interaction, 437–442
Idempotent methods, 301
identity parameter for reduce(), 373–374
ifPresent() method, 184
IllegalArgumentException class, 286
IllegalStateException class, 288
immutable objects
constructors, 85
creating, 82
modifying, 85–86
strategies, 82–84
importing static nested classes, 32
imports, 6–7
incrementAndGet() method, 353
independent operations, 370–371
indexes, string, 254–255
indexOf() method, 130–131
indirection, 76
infinite streams, 186
inner classes
anonymous, 29–31
local, 27–29
member, 25–27
InputStream class, 415, 417, 437
InputStreamReader class
description, 417
overview, 436–437
INSERT statements in SQL, 510–511
instanceof operator
generics, 113
working with, 7–9
instances
casting, 65
with factory and helper classes, 455–456, 456
legacy files, 460
instantiation, lazy, 79–80
instants and Instant class, 250–251
int type
Atomic class, 353
vs. enum, 21–22
functional interfaces, 211–213
for money, 268
ResultSet, 527
SQL statements, 523
wrapper class, 106
IntBinaryOperator interface, 212
IntConsumer interface, 211
INTEGER type in SQL, 510
Integer wrapper class, 106
interfaces
designing, 48–51
functional. See functional interfaces and programming
generic, 112–113
purpose, 51
intermediate stream operations, 186–187, 186
distinct(), 196–197
filter(), 196
flatMap(), 198
limit() and skip(), 197
map(), 197
peek(), 199–200
sorted(), 198–199
internal invariants with assertions, 310
internationalization and localization
data formats, 274–275
formatting numbers and currency, 267–268
locale selection, 256–258, 256
overview, 255–256
resource bundles, 258–266
review questions, 276–282
summary, 273–274
IntFunction interface, 212
IntPredicate interface, 211
ints() method, 207
IntStream class, 205–208
IntSupplier interface, 211
IntToDoubleFunction interface, 213
IntToLongFunction interface, 213
IntUnaryOperator interface, 212
invalid lambda expressions, 58–59
invariants
with assertions, 310
encapsulation, 66–67
invokeAll() method
fork/join framework, 387
tasks, 339–340
invokeAny() method, 339–340
IOException class, 286–287
is-a relationships, 71–73, 71–72
isAbsolute() method, 465–466
isCancelled() method, 340
isDirectory() method
File, 409
Files, 478, 480, 485
isDone() method, 340
isEmpty() method
collections, 126–127
maps, 139
isExecutable() method, 480
isFile() method, 409
isHidden() method, 480
isOther() method, 485
isParallel() method, 367
isPresent() method, 184
isReadable() method, 480
isRegularFile() method, 478, 480
IsSameFile() method, 472–473
isShutdown() method, 337
isSymbolicLink() method, 478–480
isTerminated() method
tasks, 345
threads, 337
iterate() method, 207

J

JAR files, 512
Java 7 and earlier
data formats, 272–273
dates and times in, 240–241, 243–244
Java Collections Framework. See collections
Java Persistence API (JPA), 507
JavaBeans, 69–71
JDBC (Java Database Connectivity), 506
closing database resources, 533–535
exam essentials, 537–538
exceptions, 535–536
interfaces, 511–513, 512
reading data from, 524
relational databases, 507–511, 508
ResultSet data, 524–532, 529–531
review questions, 539–545
SQL statements, 518–523
summary, 536–537
URLs, 513–515, 514
Johnson, Ralph, 92
join() method, 386–387
joining() method, 217
joining paths, 468–469
JPA (Java Persistence API), 507

K
keys in relational databases, 508, 508
keyset() method, 139

L
lambda expressions
  ambiguous, 342–343
  invalid, 58–59
  overview, 55–56
  stateful, 371
  syntax, 56–58, 57
  variables in, 172–173
languages. See internationalization and localization
last-in, first-out (LIFO) queues, 135
last() method, 529, 529, 532
lastAccessTime() method, 485
lastIndexOf() method, 130
lastModified() method, 409
lastModifiedTime() method, 484
lazy instantiation applied to singleton patterns, 79–80
lazy traversals for directories, 488
leaks in database resources, 533
leap years, 242
legacy code and generics, 114–117
legacy file instances, 460
legacy methods vs. NIO.2 API, 494–495
length, file, 409, 481
length() method, 409
levels of indirection, 76
LIFO (last-in, first-out) queues, 135
limit() method
  order-based tasks, 373
  streams, 197
linear time in big O notation, 129
lines() method, 492–493
LinkedBlockingDeque class, 360, 363
LinkedBlockingQueue class, 360, 362
LinkedHashMaps class, 138
LinkedLists
  applications, 142
  description, 129
linking streams to underlying data, 214–215
list() method, 491–492
listFiles() method, 409
listing directory contents, 491–492
ListResourceBundle class, 263
lists and List interface
  description, 124–125, 125
  implementations, 128–129
  looping through, 131
  methods, 130–131
  overview, 127–128, 128
livelock, thread, 390
liveness, thread, 387–388
local inner classes, 27–29
LocalDate class
  description, 235
  long conversions, 245
  methods, 243
  periods and durations, 250
  working with, 238–239
LocalDateTime class
  description, 235
  long conversions, 245
  methods, 243
  periods and durations, 250
  working with, 236–243
Locale class, 256–258
Locale.Builder class, 257–258
localization. See internationalization and localization
LocalTime class
  description, 235
  methods, 243
  periods and durations, 250
  working with, 238–239
locking, double-checked, 81–82
locks, 354
logarithmic time in big O notation, 129
long type
  Atomic class, 353
dates and times, 245
functional interfaces, 211–213
ResultSet, 527
wrapper class, 107
Long wrapper class, 107
LongBinaryOperator interface, 212
LongConsumer interface, 211
LongFunction interface, 212
LongPredicate interface, 211
LongStream class, 205–208
LongSupplier interface, 211
LongToDoubleFunction interface, 213
LongToIntFunction interface, 214
looping
collections, 155
with forEach(), 192–193
lists, 131
loose coupling, 89
low-level streams vs. high-level, 414–415
lower-bounded wildcards, 121–122
lower() method, 134

M
map() method
independent operations, 370–371
streams, 197, 207
mapping() method, 217, 222–223
maps and Map interface
collectors, 219 223
computeIfPresent() and
computeIfAbsent(), 157–159
description, 125, 125, 138, 138
implementations, 138
merge(), 156–157
methods, 139–140
new APIs, 155–156
mapToDouble() method, 207
mapToInt() method, 207
mapToLong() method, 207
mapToObj() method, 207
mark() method, 418–419
marking streams, 418 419
markSupported() method, 419
max() method, 190–191
MAX_PRIORITY constant, 329
MAX_VALUE value, 489
maxBy() method, 217
MILLISECONDS values, 341
MICROSECONDS values, 341
MIN_PRIORITY constant, 329
MINUTES values, 342
MINUTES values, 342
MissingResourceException class, 288
mkdir() method, 409, 473
mkdirs() method, 409, 473
mock objects, 51–52
modifications, file, 481–482
modified elements in concurrent collections, 364
modifying
attributes, 486
immutable objects, 85–86
monitors, 354
months
in date formats, 272–273
with of(), 246–247, 246
move() method, 475
moving files, 475
multi-catch clauses, 291–296, 293
multi-threaded processes, 327
multiple fields, comparing, 149–150
multiple tasks, 336–337
mutable reductions, 194
mutable strings, 254–255

N
n squared time in big O notation, 129
name() method, 20
naming conventions
generics, 110
JavaBeans, 69–70
nanoseconds, 238
NANOSECONDS values, 341
natural ordering, 146
NavigableSets, 132, 134
nested classes, 24–25
anonymous, 29–31
local, 27–29
member, 25–27
static, 31–32
summary, 33
New I/O. See NIO.2 API
newBufferedReader() method, 476–477
newBufferedWriter() method, 476–477
newCachedThreadPoolExecutor() method, 348–349
newFixedThreadPool() method, 348–349
newSingleThreadExecutor() method, 335–336, 348
newSingleThreadScheduledExecutor() method, 348
next() method, 529, 532
NIO.2 API
eaxm essentials, 495–496
file attributes, 478–483
file interaction, 471–478
vs. legacy methods, 494–495
optional arguments, 461–462
overview, 454–455
Path class, 456–460
path interaction, 460–471
Path interface, 455–456, 456
review questions, 497–503
streams, 487–493, 487
summary, 495
views, 478–483
NOFOLLOW_LINKS value, 461
nomenclature in streams, 412–413
Non-blocking I/O. See NIO.2 API
noneMatch() method, 192
NORM_PRIORITY constant, 329
normalize() method, 468–469
NoSQL databases, 507
NoSuchFileException class, 476
NotSerializableException class, 287, 427, 429
now() method, 235
null values
collections, 141–142
with instanceof, 8
merge and compute methods, 159
vs. Optional, 185
NullPointerException class, 285–286
NumberFormat class, 267–268
NumberFormatException class, 287, 427
numbers
formatting, 267–268
parsing, 268–270

O
ObjDoubleConsumer interface, 213
Object class, 8
anonymous inner classes, 30
ResultSet, 527
ObjectInputStream class
high-level streams, 415–417
working with, 426–432
ObjectOutputStream class
description, 417
working with, 426–432
null
file existence, 469–470
interacting with, 460–461
joining, 468–469
symbols, 467
testing, 471–472
Paths class
 FileSystem class, 459
legacy file instances, 460
vs. Path, 458
path creation, 456–459
peek() method
 queues, 136
 streams, 199–200, 203–204
 performance improvements in parallel streams, 368–370
 performTask() method, 380
 periods and Period class, 244–250, 246
 pipelines
 putting together, 200–202, 201
 working with, 185–188, 186–188
 plus signs (+) for string concatenation, 253
 plusDays() method, 241
 plusxxx() methods for dates and times, 243
 poll() method
 BlockingQueue, 362
 queues, 136
 pollFirst() method, 363
 pollLast() method, 363
 polls, thread, 334–335
 polymorphism
 casting objects, 64–65
 objects and references, 63, 64
 overview, 61–63
 pools
 concurrency, 348–350
 cyclic barriers, 380–381
 strings, 253
 pop() method, 136
 PosixFileAttributes class, 484
 PosixFileAttributeView class, 484
 post conditions with assertions, 310
 preconditions with assertions, 310
 Predicate interface, 177–178
 predicate interfaces, 60
 PreparedStatement class, 522
 previous() method, 529, 532
 primary keys in relational databases, 508, 508
 primitives
 functional interfaces, 210–213
 Optional with, 208–210
 overview, 205
 statistics, 210
 stream types, 205–208
 print() method, 433–434
 printf() method
 Console, 433, 439
 PrintStream, 433–434
 printing
 files, 492–493
 streams, 204
 println() method, 434
 PrintStream class, 415, 437
 description, 417
 working with, 432–435
 PrintWriter class
 access to, 438–439
 description, 417
 working with, 432–435
 priority, thread, 329
 private access modifiers, 2–4
 private variables in blocking access, 68–69
 processes, concurrent, 377–387
 processRecord() method, 370
 Properties class, 261–262
 property files, 258
 creating, 259–262
 format, 260–261
 protected access modifiers, 2–4
 public access modifiers, 2–4
 push() method, 136
 put() method, 139
 putIfAbsent() method, 156

Q

question marks (?) for wildcards, 117–118
queues and Queue interface
 blocking, 361–363
 description, 124–125, 125
 implementations, 135
 methods, 135–138, 137
 overview, 134–135, 135
race conditions, 351, 391–392, 391
Random class, 207
range() method, 207
rangeClosed() method, 207
raw collections, 115
read() method, 414
BufferedInputStream, 422
FileInputStream, 420–421
FileReader, 424
readAllLines() method
Files, 477–478
vs. lines(), 493
readAttributes() method
BasicFileAttributeView, 486
Files, 484–485
Reader class, 415, 417, 437–438
reader() method, 438–439
Readers/Readers vs. streams, 413
reading
attributes, 478–480, 485
files, 476–478
ResultSet data, 524–526, 524
readLine() method
BufferedReader, 424–425
Console, 440–441
high-level streams, 414
readObject() method, 430–431
readPassword() method, 441–442
recursion
overview, 381 382
RecursiveTask, 385–386
RecursiveAction class, 382–383
RecursiveTask class, 382–386
reduce() method
order-based tasks, 373–375
streams, 193–194
reductions
mutable, 194
parallel, 372–377
terminal operations, 189–190
refactoring, 87
references
Atomic class, 353
casting, 64–65
constructor, 154
methods, 152–154
vs. objects, 63, 64
reflection technique, 15
reflexive methods, 16
regular files, attributes for, 478 480
reifiable types, 113
relational databases
accessing, 507
closing resources, 533–535
connecting to, 513–518, 514
picking, 507–508
setting up, 509–510
SQL. See Structured Query Language (SQL)
structure, 508, 508
relationships
has-a, 73 74, 73
is-a, 71–73, 71–72
relative() method, 531–532
relative paths, 408–409
vs. absolute, 457
checking for, 465–466
relativize(), 467–468
resolve(), 469
toRealPath(), 470
relativize() method, 467–468
remove() method
collections, 126
lists, 130
maps, 139
queues, 136
removeIf() method, 154
removing
conditional, 154
files, 475–476
renameTo() method, 409
REPLACE_EXISTING value, 461
replaceAll() method, 155
reset() method, 418–419
resolve() method, 468 469
resource bundles
Java classes, 262–263
overview, 258–259
property files, 259–262
selecting, 263–266
variables, 266
ResourceBundle class, 261
resources, closing, 297, 533–535
results, task, 340–345
ResultSet interface
  closing resources, 533–535
  column data, 526, 529
  concurrency mode, 520
  description, 512, 512
  reading data from, 524–526, 524
  scrolling, 529–532, 529–531
types, 519
rethrowing exceptions, 305–307
return statements for lambda expressions, 58–59
return types, overriding, 5
root directory, 406
round-robin schedules, 328
run() method, 332–333
Runnable interface
  checked exceptions, 344
  description, 174
  overview, 330–331
runtime exceptions, 285–288

S
scaling parallel streams, 370
schedule() method, 346
scheduleAtFixedDelay() method, 346–347
scheduleAtFixedRate() method, 346–347
ScheduledExecutorService interface, 345–347
schedulers, thread, 328–329
scheduling tasks, 345–347
ScriptException class, 287
scrolling ResultSet data, 529–532, 529–531
searching
  arrays, 105–106
    Comparator and Comparable for, 150–152
  directories, 488, 490–491
seconds in methods, 238
SECONDS values, 342
SELECT statements in SQL, 510–511
semicolons (;) for variables, 29
serial streams, 366
Serializable interface, 427–429
serialization
  object creation, 431–432
  objects, 429–431
  streams, 426
serialVersionUID variable, 428–429
server environments, singletons in, 80
set() method
  Atomic, 353
  lists, 130
setDefault() method, 258
setLastModifiedTime() method, 481–482
setOwner() method, 482–483
sets and Set interface
  description, 124–125, 125
  implementations, 132, 133
  methods, 133
  overview, 132, 132
setTimes() method, 486
shared environments, 327
SHORT format for dates and times, 272
Short wrapper class, 106
shutdown() method
  tasks, 344
  threads, 337
shutdownNow() method, 337
shutting down thread executors, 337–338, 337
signatures, method, 4–5
define Parallel streams, 335–336
single-threaded processes, 327
singleton patterns
  applying, 76–79
  with double-checked locking, 81–82
  lazy instantiation applied to, 79–80
  unique, 80–81
size
  buffers, 423
  files, 481
  thread pools, 349–350
size() method
  collections, 126–127
  Files, 481
  maps, 139
skip() method
  description, 420
  order-based tasks, 373
  streams, 197
SkipList classes, 363
slashes (/, \)
paths, 408–409
root directory, 406
sleep() method, 334–335
sort() method, 106, 151
sorted() method, 198–199
sorting
arrays, 105–106
Comperator and Comparable for, 150–152
sources for streams, 186, 186, 188–189
SQL (Structured Query Language), 507
executing statements, 520–523
obtaining statements, 518–520
writing statements, 510–511
SQLException class, 287, 515, 535–536
stack traces, 291
stacks
description, 129
example problems, 137–138, 137
start() method, 331–332
starvation, thread, 390
stateful operations, 371–372
Statement interface
description, 512, 512
executing statements, 520–523
obtaining statements, 518–520
static modifiers, 6
static nested classes, 31–32
statistics with primitives, 210
streams and Stream interface
base classes, 415–416
BufferedInputStream and
BufferedOutputStream classes, 422–423
BufferedReader and BufferedWriter
classes, 424–426
byte vs. character, 413–414
chaining Optionals, 214–217
class names, 416–418
class review, 435, 436
closing, 418
directory content listing, 491–492
directory walking, 487–490, 487
InputStream and FileOutputStream
classes, 420–422
FileReader and FileWriter classes, 424
flushing, 418
fundamentals, 411–412, 412
input and output, 414
intermediate operations, 196–200
linking to underlying data, 214–215
low-level vs. high-level, 414–415
marking, 418–419
miscellaneous classes, 436–437
nomenclature, 412–413
ObjectInputStream and
ObjectOutputStream classes, 426–432
overview, 185–188, 186–188
parallel. See parallel streams
pipelines, 200–202, 201
primitives. See primitives
printing, 204
printing files, 492–493
PrintStream and PrintWriter classes,
432–435
results, 217–223
searching directories, 490–491
skipping over data, 420
sources, 188–189
terminal operations, 189–196
unordered, 373
StringBuffer class, 254–255
StringBuilder class, 254–255
strings and String class
from enums, 21
exam essentials, 274–275
overview, 253–255
pools, 253
ResultSet, 527
review questions, 276–282
summary, 273–274
Structured Query Language (SQL), 507
executing statements, 520–523
obtaining statements, 518–520
writing statements, 510–511
submit() method
Callable, 342
tasks, 338–339
submitting tasks, 338–340
subpath() method, 466–467
summarizingDouble() method, 218
summarizingInt() method, 218
summarizingLong() method, 218
summingDouble() method, 218
summingInt() method, 218
summingLong() method, 218
supertypes for generics, 122
Supplier interface
  vs. Callable, 342–343
  implementing, 174–175
suppressed exceptions, 302–304
switch statements, enums in, 21–22
symbolic links
  attributes for, 478–480
  circular paths from, 490
  optional arguments, 461
symbols, path, 467
symmetric methods, 16
synchronized collections, 365–366
synchronizing data access
  Atomic classes, 352–354, 352
  blocks, 354–356
  cost, 357–358
  methods, 356–357
  overview, 350–352, 351
system classes, 309
System.err object, 433
System.in object, 437–438
System.out object, 433, 437–438
system threads, 328
systemDefault() method, 239

T
tables in relational databases, 507–508, 508
tasks
  description, 327
  multiple, 336–337
  order-based, 372–373
  parallel processing, 367–368
  results, 340–345
  scheduling, 345–347
  submitting, 338–340
  waiting for, 344–345
  telescoping constructor anti-patterns, 86–87
TemporalUnit interface, 248
terminal stream operations, 186–187, 186

allMatch(), anyMatch() and noneMatch(), 192
collect(), 194–196
count(), 190
findAny() and findFirst(), 191
forEach(), 192–193
min() and max(), 190–191
overview, 189–190
reduce(), 193–194
test() method, 211
testing paths, 471–472
Thread class, 331–333
thread safety with singletons, 81
thread schedulers, 328–329
threads
  concurrency, 328–329
  creating, 331–333
  deadlocks, 388–390
  ExecutorService. See ExecutorService
    interface
  livelock, 390
  liveness, 387–388
  overview, 327, 327
  polls, 334–335
  pools, 348–350
  priority, 329
  problems, 387–392
  Runnable interface, 330–331
  with singletons, 80–81
  starvation, 390
  synchronizing, 350–358, 351–352
  types, 328
  throw vs. throws, 289
  tightly coupled code, 88–89
time package, 234
Time type
  JDBC, 528
  ResultSet, 527
time zones
  in computations, 236–237
  retrieving, 239
  ZonedDateTime, 238–239
times. See dates and times
TimeSta personne type
  JDBC, 528
  ResultSet, 527
TIMESTAMP type in SQL, 510
TimeUnit class, 341–342
- toAbsolutePath() method, 465–466
- toCollection() method, 218
- toConcurrentMap() method, 376
- ToDoubleBiFunction interface, 212
- ToDoubleFunction interface, 212
- toEpochDay() method, 245
- toEpochSecond() method, 245
- ToIntBiFunction interface, 212
- ToIntFunction interface, 212
- toList() method, 218
- toLocalDateTime() method, 528
- toEpochDay() method, 245
- toEpochSecond() method, 245
- ToIntBiFunction interface, 212
- ToIntFunction interface, 212
- toList() method, 218
- toLocalDateTime() method, 528
- ToLongBiFunction interface, 212
- ToLongFunction interface, 212
- toMillis() method, 481
- toPath() method, 460
- toRealPath() method
  - File, 460
  - Path, 468–471
- toSet() method, 218, 376
- toString() method
  - Path, 463
  - for valueOf(), 434
  - working with, 13–14
  - writing, 14–15
- toUri() method, 459
- transitive methods, 16
- traversing directories, 487–490, 487
- TreeMap class, 138, 140
- TreeSet class
  - applications, 142
  - description, 132, 133
  - NavigableSets, 134
- trim() method, 254
- try statements, 288–289, 288
- try-with-resources
  - AutoCloseable interface, 300–302
  - basics, 298, 299, 298
  - database resources, 533–534
  - overview, 296–297
  - summary, 305
- type erasure for generics, 111–112
- TYPE_FORWARD_ONLY mode, 519
- type parameters for generics, 109–111
- TYPE_SCROLL_INSENSITIVE mode, 519
- TYPE_SCROLL_SENSITIVE mode, 519

U
- UnaryOperator interface, 155, 180–181
- unbounded wildcards, 118–119
- uniform resource identifiers (URIs), 458–459
- unique singletons, 80–81
- uniqueness, file, 472–473
- unmodifiableList() method, 83
- unordered streams, 373
- UnsupportedOperationException class, 288, 484
- UPDATE statements in SQL, 510–511
- upper-bounded wildcards, 119–121
- URIs (uniform resource identifiers), 458–459
- URLs in JDBC, 513–515, 514
- user-defined threads, 328
- user interaction, 437
  - new method, 438–442
  - old method, 437–438
- UserPrincipalLookupService class, 482

V
- validating parameters, 313–314
- valueOf() method, 433–434
- VARCHAR type in SQL, 510
- variables
  - access modifiers, 2–4
  - blocking access, 68–69
  - inner classes, 26–28
  - in lambdas, 172–173
  - resource bundles, 266
- Vectors, 129
- views, file, 483–486
- virtual method invocation, 9–11
- visibility, file, 480
- Vlissides, John, 92

W
- waiting for tasks, 344–345
- walk() method, 488–490
- walking directories, 487–490, 487
warnings, compiler, 115–116
waves in streams, 411–412, 412
WHERE clause in SQL, 511
wildcards, 117–118
   lower-bounded, 121–122
   unbounded, 118–119
   upper-bounded, 119–121
wrapper classes, 106–107
wrapping high-level streams, 414
write() method
   BufferedOutputStream, 422
   BufferedWriter, 424–425
   FileOutputStream, 421
   FileWriter, 424
   PrintStream, 433
   Writer, 424
writeObject() method, 429
Writer class, 415, 417, 424
writer() method, 438–439
writing files, 476–477

Y

years
   in date formats, 272
   with of(), 246–247, 246

Z

ZonedDateTime class, 238–239
   description, 235
   methods, 243
   periods and durations, 250