

Index

■ A

- Aggregator modules, 17
- Applet API, 497–498
- Arrays, 495–497
- Automatic modules
 - Automatic-Module-Name, 88–89
 - definition, 86–87
 - deriving names and versions, 88
 - explicit modules, 90
 - exported packages, 87
 - _.jar file, 88, 89
 - jdojo-logging1.5.0, 88
 - jdojo.person-2.2.jar, 89
 - jdojo-trans-api_1.5_spec-1.0.0, 89
 - jdojo.util-2.2.jar file, 90
 - MANIFEST.MF file, 87
 - manifest.txt file, 89–90
 - rules, 90

■ B

- Boot loader, 137
- Breaking module encapsulation
 - com.jdojo.intro
 - module, 201–204
 - com.jdojo.intruder, 201–202, 204, 206
- command-line options
 - add-exports option, 198
 - add-opens option, 199
 - add-reads option, 200
 - illegal-access option, 200–201
 - manifest attributes, 198
- definition, 197
- executable JAR, 211–212
- TestIllegalAccess class, 206–208
- TestNonOpen class, 204–206
- using manifest attributes, 209

■ C

- CallerClassTest class
 - getCallerClass(), 404
 - getClassContext(), 404
 - StackWalker, 404–407
- ClassLoader, 478
- Client endpoint
 - build endpoint, 370–371
 - create listener
 - default methods, 368
 - onBinary() and onText() methods, 369
 - onClose() method, 369
 - onError() method, 369
 - onOpen() method, 369
 - onPing() method, 369
 - onPong() method, 369
 - text messages, 369–370
 - defined, 365
 - run program, 374–375
 - send messages to peer, 371
- Client module, 118–119
- Clients, 105
- Collection
 - JDK 9, 335–337
 - List, 338–340
 - Map, 344–347
 - Set, 341–342, 344
- Collectors
 - Employee class, 433–436
 - filtering(), 433
 - filtering and flat mapping, 436–438
 - flatMapMapping(), 433
- CompletableFuture<T>, 481
- Cookie manager, 352
- Current class, 391
- Current frame, 391
- Current method, 391
- current() static method, 316

■ **D**

Deep reflection

- com.jdojo.reflect module
 - declaration, 75
 - error messages, 77, 80
 - final version, module graph, 83
 - Item class, 76
 - modified version, 81, 82
 - open module, modified version, 81–82
- com.jdojo.reflect.test module
 - access public APIs, 82
 - adding, com.jdojo.reflect module, 77–80
 - ClassNotFoundException, 77–78
 - declaration, 76
 - declarations, modified versions, 83
 - final version, module graph, 83
 - module graph, 78–79
 - open module, modified version, 82
 - recompilation, 81
 - ReflectTest class, 76–77
 - run command, 80

- description, 73
- opening packages, 74–75
- open modules, 74
- rules, 73
- Spring and Hibernate, 73
- strong encapsulation, 73

Default methods, 470

@Deprecation

- annotation, 377
- FileCopier, 378–379
- Javadoc tag, 377
- JDK 9
 - since and forRemoval, 379–380

Deprecation

- Box, 382
- BoxTest, 382–386
- @deprecated Javadoc tag, 377
- @deprecated annotation, 377
- dynamic analysis, 389
- import statements, 389, 390
- static analysis, 386–388
- warnings, 381

Desktop features

- execution, 502–503
- system and application
 - event notifications, 501
- userSessionActivated(), 504
- userSessionDeactivated(), 504
- UserSessionListener, 503

destroyForcibly() method, 331

Diamond operator (<>)

- Callable<V>, 468
- intersection type, 469
- Magic, 469

DISCARD, 474–475

■ **E**

Eclipse, 4

Edit-Compile-Execute loop (ECEL), 254

■ **F**

Faster prime service provider, 115–116

Fastest prime service provider, 117–118

■ **G**

Generic prime service provider, 113–115

get() method, 122

getLocalDate() method, 130

■ **H**

HTTP/2 Client API

- abstract class, 351
- authenticator, 352
- benefits, 350
- com.jdojo.http.client module, 352
- cookie manager, 352
- executor, 352
- HttpClient object, 353–354
- HTTP request
 - body, 357
 - builder object, 354–355
 - create, 358
 - headers, 356
 - response body, 360–363
 - response status code and headers, 359–360
 - response trailers, 364
 - steps, 351, 354
 - string, 351
 - 3XX status code, 365
- incubator module, 349
- proxy selector, 352
- redirect policy, 352
- request priority, 352
- SSL
 - context, 352
 - parameters, 352
- supports, 350
- version, 352

I

- Incubator module, 349
- Integrated development environment (IDE), 4
- Interactive language shell, 253
- I/O API
 - Buffer, 513
 - transferTo(), 512
- isPrime() method, 114–115, 118

J, K

- JAR, 128
- JAR-hell, 8
- Java command, 3
- Java 8, 128
- Java Debugger API, 255
- Java Development Kit (JDK) 9, 128
 - classes, 129
 - IDE, 4
 - installation, 3
 - issues, 1
 - Java command, 3, 4
 - Jigsaw, 1
 - JMOD, 138
 - JShell, 1
 - module system, 1
- javadoc, 498–500
- Java SE 6, 105
- Java shell (JShell), 1, 253–254
 - API
 - abstract classes and interface, 305
 - components, 306
 - creating, 306–307
 - example, 309–313
 - handling snippet events, 309
 - working with snippets, 307–309
 - architecture, 255
 - auto-completion, 282–285
 - configuring
 - feedback mode, 294–300
 - snippet editor, 292–294
 - startup snippets, 300–303
 - declarations
 - method, 275–276
 - type, 276–279
 - declaring variables, 268–271
 - documentation, 303–305
 - Edit Pad, 267
 - expressions, evaluating, 260–262
 - import statements, 271–275
 - no checked exceptions, 281–282
 - sessions, reusing, 287–289
 - setting execution environment, 279–281
 - stack trace, 286–287

- state
 - reloading, 290–292
 - resetting, 289
- tool
 - exiting, 258
 - keys to modify text, 266
 - navigation keys, 266
 - snippets and commands, 259–260
 - starting, 256–258

- JDK 8, 129, 134
- JDK 9, *see* Java Development Kit (JDK) 9
- Jigsaw code, 1
- jlink tool, 141
- jmod tool, 138–140, 142–143
- JMOD file, 25
 - creating files, 140
 - describing, 142
 - extracting contents, 141
 - jlink tool, 141
 - jmod tool
 - command, 139
 - options, 139–140
 - listing contents, 141–142
 - record hashes, 142–145
- JVM logging
 - decorations, 454
 - destinations, 455
 - levels, 454
 - tags, 453
 - Xlog, 452, 455–459

L

- Legacy mode, 123–125
- List
 - emptyList(), 339
 - of() method, 338–340
- LocalDate class, 130

M

- Main class, 130, 135
- Map
 - entry(), 345–346, 348
 - of(), 344–346, 348
 - ofEntries(), 345–346, 348
- Matcher, 492–494
- META-INF directory, 128
- META-INF/services directory, 123–124
- META-INF/versions/9 directory, 128–129, 132, 135–136
- Modular JARs
 - class path, 123
 - META-INF/services directory, 123–124

Module dependency

- accessing, reflection (*see* Deep reflection)
 - com.jdojo.person.test module
 - compiler message, 68
 - Main class, 68
 - modified declaration, 68, 70
 - module declaration, 67
 - module graph, 69–71
 - module path setting, 68–69
 - requires statement, 70
 - declaration
 - Add Project dialog box, 62–63
 - Address class, 58–59
 - build\classes directory, 65
 - com.jdojo.address module, 58
 - com.jdojo.common, 61
 - com.jdojo.person module, 57–58, 60, 62
 - exports statement, 59
 - java.base module, 61
 - Main class, com.jdojo.person module, 64–65
 - NetBeans and select properties, 62
 - NetBeans project added, module path, 63–64
 - package, 57
 - Person class, 60–61
 - requires statement, 61
 - disassembling module
 - definitions, 98–102
 - migration path, JDK 9, 97–98
 - optional dependency, 72–73
 - qualified exports, 71–72
 - restrictions and declarations, 85
 - splitting packages, 84–85
 - troubleshooting
 - empty package error, 65
 - module not found error, 66
 - module resolution exception, 66
 - package does not exist error, 66
 - type accessibility, 84
- Module system, 1, 121
- add-modules option, 16–17
 - aggregator modules, 17
 - before Java 9
 - compact profile, 8
 - JAR-hell, 8
 - Java application steps, 7
 - JDK/JRE categories, 9
 - problem in Java code, 8
 - claim module, 14
 - declare modules
 - control access, 19–20
 - declares dependence, 21
 - names, 19

- open module, 18–19
 - service interface, 21
 - definition, 10–11
 - dependencies, 11–12
 - descriptors
 - compile module declarations, 22
 - source files structure, 23
 - version, 22
 - exporting, 11
 - JAR file, 8
 - java.se module, 15–16
 - java.sql module, 14, 15
 - observable modules, 13, 27–29
 - packaging modules
 - in directory, 24
 - JMOD file, 25
 - modular JAR, 24
 - path, 25
 - cases, 25
 - defined, 25
 - GNU-style command-line options, 26
 - GNU-style option, 27
 - Java tool, 26
 - UNIX-like platform, 26
 - Windows, 26
 - platform modules, 10–11
 - qualified export, 11
 - reliable configuration, 9
 - root, 13
 - strong encapsulation, 10
 - terms, 11
- Module types
- automatic (*see* Automatic modules)
 - definitions, 86
 - degree of string encapsulation and reliable configuration, 98
 - JDK 9, 85
 - normal, 87
 - open, 87
 - pictorial view, 86
 - unnamed (*see* Unnamed modules)
- Multi-release JARs (MRJAR)
- and boot loader, 137
 - class file search, 129
 - create, 129–135
 - defined, 128
 - and encapsulation, 136
 - and JAR URL, 137
 - JDK 9, 129
 - modular, 136
 - multi-release manifest attribute, 138
 - rules, 136
 - same versioned files, 137

N

- NetBeans, 123
 - classes, 5
 - creates one module per project, 4
 - directory, 5
 - JDK 8, 4
 - JDK 9, 4
- newInstance() method, 111
- Normal modules
 - definition, 87
 - to unnamed modules, 92–93

O

- Object deserialization filters
 - array length, 508
 - Item, 509
 - ObjectFilterTest, 509–512
 - ObjectInputFilter, 505
 - ObjectInputFilter.Config
 - createFilter(), 505
 - getSerialFilter() and setSerialFilter(), 506
 - ObjectInputFilter.FilterInfo, 507
 - ObjectInputFilter.Status, 508
- Objects, 494–495
- Observable modules, 13, 27–29
- Open modules, 18–19, 87
- Optional<T>, 478–481
- Ordinary deprecation, 380

P

- Peer, 365
- Platform logging
 - configuration, 441
 - Log4j 2.0
 - configuration file, 444–445
 - library, 442
 - Logger Finder, 447
 - module, definition, 443–444
 - NetBeans, 442–443
 - System.Logger, 445–446
 - testing, 448–450
- Platform modules, 10–11
- Prime checker, 110
- Prime number service, 105
- Prime service provider
 - faster, 115–116
 - fastest, 117–118
 - generic, 113–115
- Private method, 471
- Probable prime service provider, 117

Process API

- classes and interfaces, 315
- com.CurrentProcessInfo class, 319–320
- com.jdojo.process.api module, 318–319
- creating
 - addJvmPath() method, 326
 - command and arguments, 321
 - CurrentProcessInfo class, 326
 - inheritIO() method, 321
 - Job class, 322–326
 - onExit(), 327
 - printlnInfo(), 326
 - start() method, 321
 - startProcess() method, 326
 - StartProcessTest class, 327–328
- current process, 316
- destroyForcibly() method, 331
- java.lang package, 315
- ManageProcessPermission class, 331–334
- methods
 - ProcessHandle.Info interface, 318
 - ProcessHandle interface, 317–318
- native processes, 316
- permissions, security manager, 331–334
- ProcessBuilder class, 316
- process comparison, 320
- ProcessHandle.Info interface, 316
- ProcessHandle interface, 316
 - allProcesses() method, 329
 - children() method, 329
 - computing process statistics, 329–331
 - current() static method, 329
 - descendants() method, 329
 - methods, 329
 - of() static method, 329
 - parent() method, 329
- termination, 331
- Provider constructor, 109
- Provider method, 109
- provides statement, 109, 113
- Proxy selector, 352

Q

- Qualified export, 11

R

- Read-Eval-Print loop (REPL), 1, 253
- Reactive streams
 - classes, 414
 - processors, 413, 424–427

Reactive streams (*cont.*)

- publisher, 412, 415
 - publisher-subscriber interactions, 414–415
 - publishing elements, 416
 - SubmissionPublisher, 417–419
 - subscribers, 412, 419–424
 - subscription, 412
- Redirect policy, 352
- Request priority, 352
- requires statement, 113, 115, 117
- Root modules, 13

■ S

- @SafeVarargs
- constructors and methods, 473
 - private method, 474
 - reifiable and non reifiable type, 472
 - unsafe(), 473
- Scanner, 491
- Server endpoint, 365–367
- Service
- classes/interfaces, 106
 - consumers, 105
 - definition, 105
 - description, 106
 - discovering, 107–108
 - implementations, 108–110
 - interface, 105, 108, 110–112
 - Java, 105
 - module, 106
 - prime testing, 118–121
 - provider, 105, 108
 - faster, 115–116
 - fastest, 117–118
 - generic prime, 113, 115
 - interface, 105
 - type, 105
- ServiceLoader class, 106–107, 109–110, 112–113, 116, 118, 124
- Set, of() method, 341–344
- Snippets
- and commands, 259–260
 - history, 285–286
 - defined, 259
 - editing, 266–268
 - listing, 262–266
 - rerunning previous, 268
- Source code analysis, 305
- Spin-Wait, 482–483
- SSLcontext, 352

- SSL parameters, 352
- Stack, 391–392
- Stack traversal, *see* Stack walking
- Stack walking
- drawbacks, 395
 - JDK 8
 - LegacyStackWalk, 393–395
 - StackTraceElement, 392–393
 - Throwable and Thread, 393
 - JDK 9
 - caller's class, 404–407
 - current thread, 399–403
 - options, 396
 - permission check, 407–408
 - stack frames, 396–398
 - static factory
 - methods, 398–399
- stream() method, 122
- Stream, 411
- Collector (*see* Collectors)
 - dropWhile(), 430
 - execution, 431–432
 - flatMap(), 431
 - ofNullable(), 431
 - ofNullable(T t), 430
 - takeWhile(), 430
- StrictMath, 475–477

■ T

- Terminal deprecation warnings/removal
- warnings, 380
- Test class, 136
- Time API
- Clock class, 483
 - dividedBy(), 484
 - Duration class, 484
 - duration parts, 485
 - epoch seconds, 487
 - Julian day formatter, 490
 - LocalDate, 488–489
 - ofInstant(), 486–487
 - tickMillis(), 483
 - time zone names, 490–491
 - truncating duration, 486
- TimeUtil class, 130, 135
- try-with-resources block
- AutoCloseable, 463, 466
 - final, 465
 - ResourceTest, 466–467
- type() method, 122

■ **U, V**

UnderscoreTest, 462

UNIX shell, 253

Unnamed modules

- automatic and explicit modules, 91

- com.jdojo.unnamed.test module
 - adding, com.jdojo.reflect project, 92–93
 - declaration, 92

 - Main class, 92

- declaration, 91

- definition, 91

- rules, 91

- to normal modules, 94–97

URL, 137

uses statement, 107

■ **W, X, Y, Z**

WebSocket protocol

- client endpoint

 - build endpoint, 370–371

 - create listener, 368–370

 - defined, 365

 - run program, 374–375

 - send messages to peer, 371

- closing handshake, 365

- error message and solution, 375–376

- opening handshake, 365

- peer, 365

- server endpoint, 365–367

- WebSocketClient class, 371