## INDEX

<table>
<thead>
<tr>
<th>A</th>
<th>Stream, 283</th>
</tr>
</thead>
<tbody>
<tr>
<td>abstract classes, 73</td>
<td>Streaming, 491</td>
</tr>
<tr>
<td>versus interfaces, 78</td>
<td>applets, 83</td>
</tr>
<tr>
<td>abstract keyword, 73</td>
<td>RMI clients, 343</td>
</tr>
<tr>
<td>accept( ) method, 207</td>
<td>Application class (JavaFX), 236</td>
</tr>
<tr>
<td>access levels, 63</td>
<td>application servers versus containers, 350</td>
</tr>
<tr>
<td>acknowledgment modes (MOM), 442</td>
<td>application variable, 385</td>
</tr>
<tr>
<td>ActionEvent class, 102</td>
<td>applications</td>
</tr>
<tr>
<td>ActionListener interface</td>
<td>containers, 350</td>
</tr>
<tr>
<td>components, 101</td>
<td>distributed, 337</td>
</tr>
<tr>
<td>activation of objects, 188</td>
<td>Java Ee, 348</td>
</tr>
<tr>
<td>adapter classes, 108</td>
<td>arguments, 37</td>
</tr>
<tr>
<td>adapters, 109</td>
<td>command-line arguments, 58</td>
</tr>
<tr>
<td>add( ) method, 129</td>
<td>lambda expressions, 156</td>
</tr>
<tr>
<td>administered objects, 419</td>
<td>method extractors and, 290</td>
</tr>
<tr>
<td>Adobe Flex, 235</td>
<td>annotated endpoints, 403</td>
</tr>
<tr>
<td>@After annotation, 525</td>
<td>@Asynchronous annotation, 459</td>
</tr>
<tr>
<td>@AfterClass annotation, 526</td>
<td>asynchronous communications, 430</td>
</tr>
<tr>
<td>AnchorPane layout, 239</td>
<td>@Asynchronous method invocation, 452</td>
</tr>
<tr>
<td>AND operator, 57</td>
<td>asynchronous servlets, 377</td>
</tr>
<tr>
<td>annotated endpoints, 403</td>
<td>attributes, 21</td>
</tr>
<tr>
<td>annotation, 265, 266</td>
<td>mutable, 159</td>
</tr>
<tr>
<td>isAnnotationPresent( ) method, 331</td>
<td>autoboxing, 26</td>
</tr>
<tr>
<td>annotation processor, 324</td>
<td>AutoClosable interface, 121</td>
</tr>
<tr>
<td>annotations, 323</td>
<td>AWT (Abstract Windowing Toolkit), 83</td>
</tr>
<tr>
<td>@After, 525</td>
<td>Bean Validation framework, 477-479</td>
</tr>
<tr>
<td>@AfterClass, 526</td>
<td>@Before annotation, 525</td>
</tr>
<tr>
<td>@Asynchronous, 459</td>
<td>@BeforeClass annotation, 526</td>
</tr>
<tr>
<td>@Before, 525</td>
<td>Current, 473</td>
</tr>
<tr>
<td>@BeforeClass, 526</td>
<td>Java Logging API, 505</td>
</tr>
<tr>
<td>@Consumes, 496</td>
<td>JDBC, 299</td>
</tr>
<tr>
<td>@DELETE, 496</td>
<td>JMS, 429</td>
</tr>
<tr>
<td>@Deprecrated, 326</td>
<td>JPA, 469</td>
</tr>
<tr>
<td>@Documented, 327</td>
<td>JTA (Java Transaction API), 351</td>
</tr>
<tr>
<td>@Embeddable, 472</td>
<td>Object Model, 490, 492</td>
</tr>
<tr>
<td>@Entity annotation, 470</td>
<td>Persistence Criteria, 470</td>
</tr>
<tr>
<td>@FormParam, 496</td>
<td>Stream, 283</td>
</tr>
<tr>
<td>@FunctionalInterface, 326</td>
<td>Streaming, 491</td>
</tr>
<tr>
<td>@GET, 495</td>
<td>applets, 83</td>
</tr>
<tr>
<td>@Id, 471</td>
<td>RMI clients, 343</td>
</tr>
<tr>
<td>@Inherited, 326</td>
<td>Application class (JavaFX), 236</td>
</tr>
<tr>
<td>@LocalBean, 454</td>
<td>application servers versus containers, 350</td>
</tr>
<tr>
<td>@MessageDriven, 453</td>
<td>application variable, 385</td>
</tr>
<tr>
<td>classes, 73</td>
<td>applications</td>
</tr>
<tr>
<td>versus interfaces, 78</td>
<td>containers, 350</td>
</tr>
<tr>
<td>abstract keyword, 73</td>
<td>distributed, 337</td>
</tr>
<tr>
<td>accept( ) method, 207</td>
<td>Java Ee, 348</td>
</tr>
<tr>
<td>access levels, 63</td>
<td>arguments, 37</td>
</tr>
<tr>
<td>acknowledgment modes (MOM), 442</td>
<td>applications</td>
</tr>
<tr>
<td>ActionEvent class, 102</td>
<td>containers, 350</td>
</tr>
<tr>
<td>ActionListener interface</td>
<td>distributed, 337</td>
</tr>
<tr>
<td>components, 101</td>
<td>Java Ee, 348</td>
</tr>
<tr>
<td>activation of objects, 188</td>
<td>arguments, 37</td>
</tr>
<tr>
<td>adapter classes, 108</td>
<td>command-line arguments, 58</td>
</tr>
<tr>
<td>adapters, 109</td>
<td>lambda expressions, 156</td>
</tr>
<tr>
<td>add( ) method, 129</td>
<td>method extractors and, 290</td>
</tr>
<tr>
<td>administered objects, 419</td>
<td>annotated endpoints, 403</td>
</tr>
<tr>
<td>Adobe Flex, 235</td>
<td>@Asynchronous annotation, 459</td>
</tr>
<tr>
<td>@After annotation, 525</td>
<td>asynchronous communications, 430</td>
</tr>
<tr>
<td>@AfterClass annotation, 526</td>
<td>asynchronous method invocation, 452</td>
</tr>
<tr>
<td>AnchorPane layout, 239</td>
<td>asynchronous servlets, 377</td>
</tr>
<tr>
<td>AND operator, 57</td>
<td>attributes, 21</td>
</tr>
<tr>
<td>annotated endpoints, 403</td>
<td>mutable, 159</td>
</tr>
<tr>
<td>annotation, 265, 266</td>
<td>autoboxing, 26</td>
</tr>
<tr>
<td>isAnnotationPresent( ) method, 331</td>
<td>AutoClosable interface, 121</td>
</tr>
<tr>
<td>annotation processor, 324</td>
<td>AWT (Abstract Windowing Toolkit), 83</td>
</tr>
<tr>
<td>annotations, 323</td>
<td>Bean Validation framework, 477-479</td>
</tr>
<tr>
<td>@After, 525</td>
<td>@Before annotation, 525</td>
</tr>
<tr>
<td>@AfterClass, 526</td>
<td>@BeforeClass annotation, 526</td>
</tr>
<tr>
<td>@Asynchronous, 459</td>
<td>Current, 473</td>
</tr>
<tr>
<td>@Before, 525</td>
<td>Java Logging API, 505</td>
</tr>
<tr>
<td>@BeforeClass, 526</td>
<td>JDBC, 299</td>
</tr>
<tr>
<td>@Consumes, 496</td>
<td>JMS, 429</td>
</tr>
<tr>
<td>@DELETE, 496</td>
<td>JPA, 469</td>
</tr>
<tr>
<td>@Deprecrated, 326</td>
<td>JTA (Java Transaction API), 351</td>
</tr>
<tr>
<td>@Documented, 327</td>
<td>Object Model, 490, 492</td>
</tr>
<tr>
<td>@Embeddable, 472</td>
<td>Persistence Criteria, 470</td>
</tr>
<tr>
<td>@Entity annotation, 470</td>
<td>Stream, 283</td>
</tr>
<tr>
<td>@FormParam, 496</td>
<td>Streaming, 491</td>
</tr>
<tr>
<td>@FunctionalInterface, 326</td>
<td>annotations, 323</td>
</tr>
<tr>
<td>@GET, 495</td>
<td>@Before annotation, 525</td>
</tr>
<tr>
<td>@Id, 471</td>
<td>@BeforeClass annotation, 526</td>
</tr>
</tbody>
</table>
BiFunction interface, 169-171
Big O notation, 139
bind() method, 249, 341
bindBidirectional() method, 249
binding
binding in JavaFX, 248-251
BitSet class, 137-138
BitSet classes
block comments, 27
blocking calls, 430
body of a method, 6
boolean variables, 217
BorderLayout manager, 90
BorderPane layout, 239
bounded type parameters, 147-149
BoxLayout manager, 93
break keyword, 52
browser as WebSocket client, 402-403
buffered streams, 175
BufferedInputStream class, 175
BufferReader class, 176
buffers, 175
builds, 535
Ant "Hello World", 536-537
Gradle "Hello World", 541-544
Maven "Hello World", 537-540
business tier, 349
buttonClickHandler() method, 262
buttons
actionPerformed() method, 231
operation, 280
byte arrays, serialization
serialization, 194
byte streams, 174
calculator with Scene Builder, 253-262
CalculatorEngine class, 100, 101
call() method, 227
Callable interface, 212
CallableStatement class, 306
callback methods, 102
callbacks
capturing phase, events, 263
CardLayout manager, 95
casting, 68
downcasting, 69
catch keyword, 118
CDB (Clear Data Builder), 329
CDI (Contexts and Dependency Injection), 499-501
cell editors, 320-321
cell renderers, 320-321
certification, 563
char values, 176
character streams, 176
chat rooms, 432
CI (Continuous Integration), 535
class methods (see methods)
CLASS retention policy, 328
class versioning, 193
classes, 6
abstract, 73-76, 78
access level, 63
ActionEvent, 102
adapter classes, 108
anonymous inner classes, 156
attributes, 21
BitSet, 137-138
BufferedInputStream, 175
BufferReader, 176
CalculatorEngine, 100, 101
CallableStatement, 306
collection classes, 127
Collections, 133
Condition, 227
dataInputStream, 173
dataOutputStream, 180
declaring, 27, 29
descendant, 164
DirectMessageReceiver, 439
DirectMessageSender, 444
embeddable, 472
EntityManager, 470
Enumeration, 135
Executors, 227-230
fields, 21
File, 181
FileDownload, 201, 202
FileHandler, 512
FileInputStream, 173
FileReader, 173, 176
FileWriter, 176
final keyword, 64
Formatter, 513
generics and, 141-143
Hashmap, 132
Hashtable, 132
ImperativeVsFunctional, 154
InitialContext, 420
inner classes, 110
inner, anonymous, 111
InputStream, 173
instances, heap memory and, 23
Iterator, 135
JavaToJSONStreaming, 491
JTable, 313
LinkedBlockingDeque, 230
LinkedList, 135
methods, 21, 28
invoking, 337
ObjectInputStream, 190
ObjectOutputStream, 189-190
OutputStream, 173
packages, 61
parameterized, 141
custom, 146-147
PreparedStatement, 306
Properties, 134
Reader, 173
ReentrantLock, 226
ReflectionSample, 330
RemoteException, 124
ResultSetMetaData, 307
serializable, 188
ServerSocket, 205
SimpleFormatter, 513
Socket, 205
SocketServer, 207
StreamOfDates, 293
StringBuffer, 49
StringBuilder, 49
subclasses, 32
byte streams, 174
superclasses, 40
Swing, 84
Swingworker, 232
TestCallableThreads, 228
TestGenericBounded, 148
TestTaxLambda, 161
Thread, 212-213, 215, 217
TicTacToeController, 267-278
URL, 201
URLConnection, 200
UserPreferences, 187
variables, declaring, 22
WebSiteReader, 200
wrapper classes, 26
Writer, 173
clients
fat clients, 349
RESTful services, 499
RMI, 341
applets as, 343
thin, 349, 359
Closure, 153
closures, 223
lambda expressions, 169
code
embedding in HTML, 382-385
generating, 16-18
annotations and, 329
COLLECT() METHOD–DOGET() METHOD

collect() method, 284
Collection interface, 128
collections, 127
    ArrayList, 129
forEach() method, 162
interfaces, java.util, 128
java.util.concurrent package, 230
LinkedList, 129
mutable, 291
selection tips, 138
sorting, 288
    Comparable interface, 288
Comparator interface, 289
Collections class, 133
Command line arguments, 58
lesson, 59-60
comments, 26
    block comments, 27
    forward slashes, 27
commit() method, 310
Comparable interface, 288
Comparator interface, 289, 291
compareTo() method, 288
compiling
    errors, 142
    Hello World, 7
composition versus inheritance, 164
concurrency, 211
Condition class, 227
conditional statements, 57
    if, 30
    switch, 31
config variable, 385
configuration
    GlassFish, 361
    Gradle and, 549-555
ConsoleHandler object, 509
constructors, 39
@Consumes annotation, 496
container classes, JavaFX layouts, 239
containers
    EJB, 451
    asynchronous method invocation, 452
    embeddable, 452
    MDBs, 452
    JFrame, 85
    versus application servers, 350
contains() method, 155
containsKey() method, 133
containsValue() method, 133
contexts, naming services, 419
continue keyword, 52
Controller class
    event handling in JavaFX, 262, 266
    convention over configuration paradigm, 539
    conventions, 33
    cookies, 370
copy() method, 183
createDurableConsumer() method, 440
createNewFile() method, 181
Criteria API, 473
CriteriaBuilder interface, 474
CriteriaQuery interface, 474
CRUD (create, retrieve, update, delete), 470
CSS (Cascading Style Sheets)
    JavaFX, 236
    selectors, 243
    skinning, 243-246
CSS (cascading style sheets)
    tic-tac-toe game, 268-275
datasources, JNDI and, 423
DAS (Domain Administration Server), 422
data
    passing between objects, 104-106
    passing between view to controller, 265
    reading from Internet, 191-200
    stream data, 293
data streams, 173, 180
data types, 37
    erasure, 143
    parameterized, 141
    primitive, 24
    memory, 24
    wrapper classes, 26
    raw, 143, 143
type inference, 143
wildcards, 144
databases, 299
    @Column, 472
    JPQL, 473
    lesson, 322
    NoSQL, 299
    @NotNull, 472
tables
    mapping objects to, 470-472
    primary keys, 471
transactions
    committing, 310
    rolling back, 310
datagrams, 205
DataInputStream class, 173, 180
DataOutputStream class, 173, 180
DataSource interface, 310
dataSource, JNDI and, 423-425
DBMSes (Database Management Systems), 299
    metadata, 323
    stored procedures, 300
debugging, 54-56
    JavaScript, 413
declarations
    generics, 144
    JSP, 386
    variables, 23
declaring
    classes, 27, 29
    variables, 22
decode() method, 411
decoders, websockets, 410-413
defender methods, 67
@DELETE annotation, 496
DELETE method, 489
dependencies
    Gradle and, 546
    configurations and, 549-555
    repositories, 548
    JAR files, 551
dependency injection, 353
deployment, 58
    EJB, 462
    hot deployment, 367
    JSP (Java Server Pages), 382
    servlets, 360-361
@Deprecated annotation, 326
Deque interface, 230
derby DB, 300-302
descendant classes, 164
deserialization, 188
    ObjectInputStream class, 190
    properties, 194
design
    MVC (Model-View-Controller), 107
    Observer pattern, 249
destroy() method, 369, 377
developer interviews, 562
diamond operator, 142
directives
    JSP
        attribute, 386
        include, 386
        page directives, 386
    DirectMessageReceiver class, 439
    DirectMessageSender class, 444
directory services, 343, 419
distributed applications, 337
do-while loop, 53
document root, 360
@Documented annotation, 327
doGet() method, 360
doInBackground() method, 232
DOM (Document Object Model), 401
doPost() method, 360
doSomething() method, 162, 164
dot notation, 6
doubly linked lists, 135
downcasting, 69
downloading
   Eclipse IDE, 10
      from Internet, 201
driver types, JDBC, 300
DSL (Domain Specific Language), 540
DTOS (Data Transfer Objects), 389
durable subscribers, 440
dynamic arrays, ArrayList, 129-132
dynamic IPs, 197

Eclipse IDE, 32, 9, 9
   directory structure, 521
downloading, 10
GlassFish configuration, 361-364
Gradle in, 555, 556
   plug-ins, 555
Hello project, 11
HelloWorld class, 14
   installation, 10
RCP (Rich Client Platform), 9
   server adapters, 362
   servlet creation, 364-368
WAR files, 368
   workbench, 10
Eclipse IDE for Java EE Developers, 10
Eclipse Luna, lambda expressions and, 158
EE (Enterprise Edition), 31
EJB (Enterprise JavaBeans), 348, 451
   containers, 451
      asynchronous method invocation, 452
      embeddable, 452
      MDBs, 452
deployment, 462
ear (Enterprise Archive), 462
entity beans, 452
   HelloWorldBean, 453
      lesson, 467
local, 456-458
local no-interface beans, 454
   MDBs, 464
remote beans, 458
session beans
   singleton, 452, 461
      stateful, 452, 460
      stateless, 452, 453-460
timer service, 466
transactions, 465
@Embeddable annotation, 472
   embeddable classes, 472
   embedding code in HTML, 382-385
   encapsulation, 61, 62
   encoders, websockets, 410
   entities, 470
   @Entity annotation, 470
   entity beans (EJB), 452
   Entity Manager, 475-477
   EntityManager class, 470
   Enumeration class, 135
   Enumeration interface, 135, 135
   enumerations, 135
   equals() method, 50
   error pages (JSP), 388
   errors
      compilation and, 142
      I/O, 116
   event dispatch thread, 231
   event handling
      buttonClickHandler() method, 262
   JavaFX, 246-248
         Controller class, 262, 266
   event listeners, 99
   event-driven programming, 99
   events, 99
      bubbles, 263
      capturing phase, 263
      KeyEvent, 264
      source, 263
      sources, 102
   EventSource object, 401
   exception handling, 115
      final keyword, 64
   exception variable, 385
   exceptions, 116
      AutoCloseable interface, 121
      creating, 123-125
      finally clause, 120
      hierarchy, 117
      lesson, 125
      MalformedURLException, 198
      RemoteException, 124
      subclasses, 117
      throws clause, 119
   Executors framework, 227-230
   Executors class, 227-230
   expressions
JSP, 386
   lambda expressions, 153, 155
      method references, 157
   extended keyword, 147
   Externalizable interface, 191

fat clients, 349
File class, 181
FileDownload class, 201
   command-line parameters, 202
FileHandler class, 512
FileInputStream class, 173
FileReader class, 173, 176
   copying, 183
   downloading from Internet, 201
FileWriter class, 176
Filter interface, 376
filter() method, 284
   filters
      Java Logging API, 513
      servlets, 375
      streams, 285
final keyword, 23
   classes, 64
      exception handling and, 64
      methods, 64
      variables, 64
final variables, 23
   finally clause, 120
   finally keyword, 118
   firewalls, 197
      HTTP proxy servers and, 200
   FlowLayout layout manager, 86-87
   FlowLayout manager, 88
   FlowPane layout, 239
   FocusListener, 108
   for loop, 50
   for-each loop, 51
   forEach() method, 51, 162
   Formatter class, 513
   formatters, Java Logging API, 512
   @FormParam annotation, 496
   frameworks
      Executor, 227-230
      logging frameworks, 515-516
      ORM, 327
      RIA (Rich Internet Application), 235
      full-duplex communication, 399
      Function interface, 169-171
      functional interfaces, 100, 157-159
      functional programming languages, 153
functional style programming, 153
versus imperative, 154
@FunctionalInterface, 158
@FunctionalInterface annotation, 326
functions
hash functions, 133
higher order, 170
higher-order, 160
nested, closures, 223
passing to methods, 160
versus methods, 159-162
FXML, 235
annotation, 266
GUI and, 235
tic-tac-toe game, 268-275
GC (Garbage Collector), 25
generating code, 16-18
generics, 133, 141
classes and, 141-143
declaring, 144
lesson, 150
methods, 149
@GET annotation, 495
GET method, 489
Get method (HTTP), 369
get() method, 130, 133, 228
getConstructors() method, 331
getFields() method, 331
getInterfaces() method, 331
getProperty() method, 134
getSession() method, 372
getState() method, 221
GlassFish, 32
configuration, Eclipse IDE, 361-364
JMS objects, 445-447
JNDI objects, 422
Gosling, James, 1
Gradle, 540
conventions, 544-546
dependencies and, 546
configurations and, 549-555
repositories, 548
Eclipse IDE, 555, 556
plug-ins, 555
Hello World, 541-544
lesson, 557
projects, 540
tasks, 540
Gradle Wrapper, 553
GridLayout, 94
GridLayout manager, 88
GridPane layout, 239
sample application, 241-243
Groovy, 540
GUI (graphic user interface), 83
FXML and, 235
I/O streams and, 177-179
skinning with CSS, 243-246
half-duplex communication, 399
handlers, Java Logging API, 509-512
hash functions, 133
HashMap class, 132
Hashtable class, 132
Haskell, 153
hasMoreElements() method, 135
HBox layout, 239
sample application, 240
heap memory, 23
Hello WebSocket, 404-406
Hello World, 5-6
Ant, 536-537
Eclipse IDE, 11
Gradle, 541-544
Java Logging API, 507
class-level logger, 508
global logger, 507
Maven, 537-540
running, 7
HelloFX project, 237
HelloWorld class, Eclipse IDE, 14
HelloWorldBean, 453
hierarchy exceptions, 117
higher order functions, 170
higher-order functions, 160
hostnames in URLs, 197
hot deployment, 367
HTML (Hypertext Markup Language)
embedding code, 382-385
HTTP (Hypertext Transfer Protocol), 357
DELETE method, 489
drawbacks, 400-401
Get, 369
GET method, 489
Post, 369
POST method, 489
PUT method, 489
server-side data push, 401
session tracking, 370
cookies, 370
HttpSession, 372-375
HTTP proxy servers, 200
HTTPSession interface, 370
HttpSession object, 372-375
I/O errors, 116
I/O streams, 173
buffered streams, 175
BufferedInputStream class, 175
byte streams, 174
classer character streams, 176
data streams, 180
GUI and, 177-179
NIO, 184-185
Stream API, 294
@Id annotation, 471
IDE (Integrated Development Environment), 9
if statement, 30, 52, 57-58
if-else statements, 57
@Ignore annotation, 526
imperative programming versus functional, 154
ImperativeVsFunctional class, 154
implicit objects (JSP), 385
include directive, 386
infinite loops, 54
inheritance, 32, 61
casting, 68-70
eliminating, 167, 215
lambda expression comparison, 164-171
versus composition, 164
@Inherited annotation, 326
init() method, 236, 369
InitialContext class, 420
inner classes, 110
anonymous, 111, 156
InputStream class, 173
installation
Eclipse IDE, 10
JDK
MAC OS, 3
Windows, 4
instanceof operator, 130
instances, 22
heap memory, 23
locks, 219
integration testing, 519
IntelliJ IDEA, 9
interfaces, 65-82
ActionListener, 101
AutoClosable, 121
BiFunction, 169-171
Callable, 212
Collection, 128
collections, java.util, 128
Comparable, 288
Comparator, 289, 291
CriteriaBuilder, 474
CriteriaQuery, 474
DataSource, 310
Deque, 230
Enumeration, 135, 135
Externalizable, 191
Filter, 376
Function, 169-171
functional, 100
functional interfaces, 157
HTTPSession, 370
Iterable, 129
Iterator, 135
Lesson, 65-82
List, 129
Map, 129
marker interfaces, 66
MessageListener, 437
methods
default, 67
static, 68
no-interface beans (EJB), 454
Path, 182
programming to, 131
Property, 249
public, 63
Queue, 129
Recommendations, 499
remote, 338
implementation, 339
RMI, 337
Root, 474
Rowset, 309
Runnable, 212, 213-215
Serializable, 188
Set, 129
StockServer, 338
TableModel, 34-320
typedQuery, 474
versus abstract classes , 78
intermediate operations in streams
Java SE (Standard Edition), 3
applications
containers, 350
tiers, 348-350
lesson, 354
messages and, 443-445
multithreading, 352
profiles, 352
Project Explorer, 10
pruning, 352
reasons to use, 352
versions, 348
Java Logging API, 505
duplicate messages, 511
filters, 513
formatters, 512
handlers, 506, 509-512
Hello World, 507
class-level logger, 508
global logger, 507
lesson, 516
log levels, 509-512
logging frameworks, 515-516
logging.properties file, 511
message grouping, 506
Java SE (Standard Edition), 3
downloading, 3-5
java.io package, 173
java.nio package, 173
java.util collection interfaces, 128
java.util.Collections class, 133
java.util.concurrent package, 226-231
collections, 230
JavaBeans, 389
bean lifecycle, 390
JSP and, 390
RESTful services, 494
JavaDB (see Derby DB)
Javadoc
annotations, 323
@Deprecated, 326
@Documented, 327
@FunctionalInterface, 326
@Inherited, 326
@Override, 325-326
JavaFX, 83, 235
Application class, 236
binding, 248-251
CSS, 236
event handling, 246-248
Controller class, 262, 266
events
KeyEvent, 264
source, 263
HelloFX project, 237
layout container classes, 239
layouts, 238-241
BorderPane, 238
lesson, 252
mobile devices, 280
properties, 248-251
property classes, 249
root nodes, 236
Scene Builder, 253-262
scene graphs, 235
stages, 235
Swing and, 236
theater terminology, 235
tic-tac-toe game, 267-275
menus, 279
TicTacToeController class, 267-278
web, 280
JavaScript, 153
debugging, 413
JavaToJSONStreaming class, 491
JAX-RE, 487
JAX-WS, 487
JCP (Java Community Process), 348
JDBC
CallableStatement class, 306
collection pools, 310
DataSource interface, 310
drivers, 300
PreparedStatement class, 306
connection pooling, 304-305
result sets, 304-305
scrollable, 309
ResultSetMetaData class, 307
Rowset interface, 309
sample program, 302-304
JDBC API, 299
JDK (Java Development Kit), 32, 3
MAC OS, 3
Windows, 4
JFrame containers, 85
JMS (Java Messaging Service), 348, 429, 435
asynchronous communications, 430
message types, 435
MOM and, 429
objects in GlassFish, 445-447
OpenMQ MOM, 432-434
JNDI (Java Naming and Directory Interface), 419
administered objects, 419
datasource and, 423-425
InitialContext class, 420
lesson, 427
objects, GlassFish and, 422
resources, 421
join() method, 224
joining threads, 224
JPA (Java Persistence API), 469, 473
(see also JPQL)
bean validation, 477-479
Entity Manager, 475-477
lesson, 479
JPQL (Java Persistence Query Language), 470, 473
Criteria API, 473
JRE (Java Runtime Environment), 3
JSF (JavaServer Faces), servlets and, 357
JSON (JavaScript Object Notation), 353
Object Model API, 490
writing JSON, 492
RESTful services and, 489-493
Stream API and
reading JSON, 491
Streaming API, 490
writing JSON, 491
XML comparison, 489
JSP (Java Server Pages), 348, 381
actions, 387
declarations, 386
deployment, 382
directives
attribute, 386
include, 386
page directives, 386
error pages, 388
expressions, 386
forward element, 387
JavaBeans in, 390
lesson, 395
objects, implicit, 385
plugin element, 387
servlets and, 357
loading from, 391
tag libraries, 392-394
variables, 385
JSR (Java Specification Request), 348
JSTL (JSP Standard Tag Library), 394
JTA (Java Transaction API), 351, 465
JTable
annotations, 323
cell editors, 320-321
cell renderers, 320-321
UI class, 314
JTable class
MVC and, 313
JUnit framework, 519, 520
(see also unit testing)
annotations, 525
applying, 526-528
Eclipse directory structure, 521
imports, 527
installation, 521
lesson, 532
test case, 522-525
test runners, 531-532
test suites, 528-529
JVM (Java Virtual Machine), 1, 2

K

KeyEvent, 264
KeyListener, 108

keys
Hashmap class, 132
Hashtable class, 132
primary
databases, 471
keywords
abstract, 73
break, 52
catch, 118
continue, 52
extends, 147
final, 23, 64
finally, 118
null, 51
private, 63
public, 6
static, 6, 44
String, 6
super, 40, 149
synchronized, 219, 226
this, 40
throw, 118
throws, 118
transient, 190
try, 118
void, 6
'public, 63

L

lambda expressions, 153, 155
@FunctionalInterface, 158
arguments, 156
closures and, 169
curly braces, 156
inheritance comparison, 164-171
inner class alternative, 112
lesson, 172
Luna and, 158
method references, 157
polymorphism comparison, 164-171
syntax, 156
TableModelListener, 319
LANs (local area networks), 197
late binding, 77
launch() method, 236
layout managers (Swing)
absolute layout, 96
BorderLayout manager, 90
BoxLayout, 93
CardLayout, 95
combining, 90
FlowLayout, 86-87, 88
GridBagLayout, 94
GridLayout, 88
layouts
AnchorPane, 239
BorderPane, 239
FlowPane, 239
GridPane, 239, 241-243
HBox, 239, 240
JavaFX, 238-241
container classes, 239
StackPane, 239
TilePane, 239
VBox, 239
LDAP (Lightweight Directory Access Protocol), 425-426
lessons, 32
command-line arguments, 59
database, 322
EJB, 467
exceptions, 125
generics, 150
Gradle, 557
interfaces, 79
Java EE, 354
Java Logging API, 516
JavaFX, 252
passing data between objects, 104-106
between view and controller, 265
parameters, websockets, 405
passing values
by reference, 42
by value, 42
passivation, 188
@Path annotation, 496
Path interface, 182
@PathParam annotation, 405, 496
persist() method, 475
persistence context, 475
Persistence Criteria API, 470
platform independence, 1
plug-ins
E(fx)clipse, 236-238
Gradle in Eclipse IDE, 555
POJO (Plain Old Java Object), 350, 357, 451
polling, 401
polymorphism, 61, 76-78
lambda expression comparison, 164-171
populating arrays, 128
@POST annotation, 496
POST method, 489
Post method (HTTP), 369
Preferences panel, 187
PreparedStatement class, 306
presentation layer, 357
presentation tier, 349
primary keys, databases, 471
primitive data types, 24
memory, 24
wrapper classes, 26
println() method, 6, 505
private keyword, 63
private variables, 305
process() method, 233
processDigit() method, 264
processOperation() method, 264
@Produces annotation, 496
program life cycle, 2
programmatic endpoints, 403
programming
functional programming languages, 153
functional style programming, 153
imperative versus functional, 154
socket programming, 205-209
to interfaces, 131
Project Explorer (Java EE), 10
properties
Class, 133
deserialization, 194
JavaFX, 248-251
networking properties, 200
Properties class, 133, 134
Property interface, 249
protocols, 197
WebSocket, 206
proxies, 337
pruning, 352
public interface, 63
public keyword, 6, 63
publish() method, 233
publishing with WebSocket protocol, 413-415
push technology, 401
@PUT annotation, 496
PUT method, 489
QA testing, 520
Quartz Scheduler, 466
queries in JPQL, 473
@QueryParam annotation, 496
Queue interface, 129
queues, 230
race conditions, 131, 219-221
raw data types, 143, 143
RCP (Rich Client Platform), 9
RDBMSes (Relational Database Management Systems), 299
(see also databases)
Reader class, 173
readExternal() method, 191-200
readObject() method, 190
Recommendations interface, 499
ReentrantLock class, 226
reflection, 323, 330
introspection and, 332
ReflectionSample class, 330
regression, 519
remote interfaces, 338
implementation, 339
StockServer, 338
RemoteException class, 124
remove() method, 135
repositories, Gradle and, 548
request variable, 385
@Resource annotation, 421
Response variable, 385
REST (representation state of transfer), 488
RESTful services, 357, 487, 488
application creation, 494
CDI (Contexts and Dependency Injection), 499-501, 501
clients, 499
dependency class, 495-498
JavaBeans, 494
lesson, 502
websockets comparison, 490
result sets, scrollable, 309
ResultSetMetaData class, 307
reverse domain name, 61
reverse() method, 133
reverse-domain name conventions, 16
RIA (Rich Internet Application) framework, 235
RMI (Remote Method Invocation), 337
application development, 338-343
clients, 341
applets as, 343
distributed applications, 337
lesson, 343
object registration, 340
objects, finding, 343
remote interfaces, 338
implementation, 339
security, 342
stubs, 337
Root interface, 474
root nodes, JavaFX, 236
Rowset interface, 309
RPC (remote procedure calls), 429
Ruby, 153
run() method, 212, 213
run-time binding, 77
Runnable interface, 212, 213-215
runtime annotation processing, 332-334
runtime errors, stack traces, 115
RUNTIME retention policy, 328
Scala, 153
scalability, subscriptions and, 441
Scene Builder, 235
calculator, 253-262
@Schedule annotation, 466
scope, variables, 25, 43
screencasts, 32
security, RMI and, 342
sendBinary() method, 408
Scalability, subscriptions and, 441
Scene Builder, 235
calculator, 253-262
@Schedule annotation, 466
scope, variables, 25, 43
screencasts, 32
security, RMI and, 342
sendBinary() method, 408
sendObject( ) method, 408
sendStream( ) method, 408
sequential streams, 287
serial version unique ID, 194
Serializable interface, 188
serialization, 66, 187-188
byte arrays, 194
class versioning, 193
deserialization, 188
Externalizable interface, 191
lesson, 195
ObjectOutputStream class, 189-190
@ServerEndPoint annotation, 404
servers, websocket communication and, 403-406
ServerSocket class, 205
service( ) method, 359, 369
servlets, 348, 357
asynchronous, 377
creating, 359
Eclipse, 364-368
data flow, 368-369
deploying, 360-361
document root, 360
filters, 375
HTTP requests, 357
JSF and, 357
JSP and, 357, 381
loading from, 391
lesson, 378
online store example, 358
thin clients, 359
session beans (EJB)
singleton, 452, 461
stateful, 452, 460
stateless, 452, 453-460
asynchronous methods, 459
session tracking (HTTP), 370
cookies, 370
HttpSession, 372-375
session variable, 385
Set interface, 129
setConstraints( ) method, 242
setDoOutput( ) method, 200
setLevel( ) method, 509
setPriority( ) method, 219
short-circuit stream operations, 295
Silverlight, 235
SimpleFormatter class, 513
@Singleton annotation, 453
singleton session beans (EJB), 452, 461
test( ) method, 133
skinning with CSS, 243-246
SLA (Service Level Agreement), 520
sleep( ) method, 215
versus wait(), 222
SOAP (Simple Object Access Protocol), 357, 487
Socket class, 205
socket programming, 205-209
lesson, 209
sockets
non-blocking, 209
uses, 206
SocketServer class, 207
sort( ) method, 133
sorting
collections, 288
Comparable interface, 288
Comparator interface, 289
streams, 291
Comparator interface, 291
parallel, 292
source code, 33
SOURCE retention policy, 328
sources, events, 102
SPA (Single-Page Applications), 490
SSE (Server-Side Events), 401
stack trace, 115
StackPane layout, 239
start( ) method, 236
@Stateful annotation, 453
stateful session beans (EJB), 452, 460
stateless protocols, 370
stateless session beans (EJB), 452, 453-460
asynchronous methods, 459
statements
conditional, 30, 57
switch, 31
if, 30, 52, 57
if-else, 57
switch, 31
throw, 122
static final variable, 194
static keyword, 6, 44
static methods, interfaces, 68
stock market price program, 202-205
stock quote server, 206-208
StockServer remote interface, 338
stop( ) method, 217
stored procedures, 300
Stream API, 283
I/O streams, 294
Stream.generate( ) method, 293
Streaming API, 490
reading JSON, 491
writing JSON, 491
StreamOfDates class, 293
streams, 283-287
data generation, 293
filters, 285
finite size, 292
infinite-size, 293
intermediate operations, 284
lazy operations, 284
lesson, 296
maps, 285
parallel, 287
parallelStream( ) method, 287
reduce, 285
sequential, 287
short-circuit operations, 295
sorting, 291
Comparator interface, 291
parallel, 292
terminal operations, 284
streams of data (see data streams)
stress testing, 520
String keyword, 6
StringBuilder class, 49
StringBuffer class, 49
strings, 49-50
stubs (RMI), 337
subclasses, 32
byte streams, 174
exceptions, 117
Sublime Text, 3
super keyword, 40, 149
superclasses, 40
@SuppressWarning annotation, 325
swap( ) method, 133
Swing, 83
AbstractTableModel class, 314
adapters, 109
application creation, 234
classes, 84
DefaultTableModel class, 314
FocusListener, 108
GUI builders, 97
ItemListener, 108
JavaFx and, 236
JTable class, 313
JTable, MVC and, 313
KeyListener, 108
layout managers, 85
AbsoluteLayout, 96
BorderLayout, 90
BoxLayout, 93
CardLayout, 95
combining, 90
FlowLayout, 86-87, 88
GridBagLayout, 94
GridLayout, 88
library, 84
listeners, 108, 108
### MouseListener, 108
### MouseMotionListener, 108
### TableModel interface, 314-320
### threads, 231-233
### widgets, 96
### WindowListener, 108
### Swingworker class, 232
### switch statement, 31
### symbolic links, 182
### synchronization, 133
### synchronized keyword, 219, 226
### synchronous calls, 430
### syntax of lambda expressions, 156
### tables (databases)
- mapping objects to, 470-472
- primary keys, 471
### tabular data, 313
### tag libraries (JSP), 392-394
- JSTL, 394
### task scheduler, 466
### TCP/IP (Transmission Control Protocol/Internet Protocol), 205
### TDD (test-driven development), 519
### technical interviews, 559-561
### developers, 562
### offers, 561
### questions/answers, 563-571
### terminal operations in streams, 284
### @Test annotation, 521, 525
### TestCallableThreads class, 228
### TestGenericBounded class, 148
### testing
- SLA (Service Level Agreement), 520
- types, 519
### TestTaxLambda class, 161
### text editors, 3
### thin clients, 349
### servlets, 359
### this keyword, 40
### Thread class, 212-213
### sleep() method, 215
### stop() method, 217
### threads
- event dispatch thread, 231
- joining, 224

### tableChanged() method, 320
### TableModel interface, 314-320
### callbacks, 315
### TableModelListener, lambda expressions in, 319
### tables (databases)
- mapping objects to, 470-472
- primary keys, 471
### tabular data, 313
### tag libraries (JSP), 392-394
- JSTL, 394
### task scheduler, 466
### TCP/IP (Transmission Control Protocol/Internet Protocol), 205
### TDD (test-driven development), 519
### technical interviews, 559-561
### developers, 562
### offers, 561
### questions/answers, 563-571
### terminal operations in streams, 284
### @Test annotation, 521, 525
### TestCallableThreads class, 228
### TestGenericBounded class, 148
### testing
- SLA (Service Level Agreement), 520
- types, 519
### TestTaxLambda class, 161
### text editors, 3
### thin clients, 349
### servlets, 359
### this keyword, 40
### Thread class, 212-213
### sleep() method, 215
### stop() method, 217
### threads
- event dispatch thread, 231
- joining, 224

### validation, Bean Validation framework, 477-479
### values
- passing
  - by reference, 42
  - by value, 42
### variables
- application, 385
- boolean, 217
- class level declaration, 25
- config, 385
- declaring, 22, 23
- dot notation, 6
- exception, 385
- final, 23
- final keyword, 64
- JSP (Java Server Pages), 385
- member variables
  - access level, 63
  - out, 385
  - page, 385
  - pageContext, 385
  - private, 305
  - request, 385
  - response, 385
  - scope, 25, 43
  - session, 385
  - static final, 194
  - transient, 190, 191
- VBox layout, 239
- versioning, class versioning, 193
- vi, 3
- void keyword, 6

### wait() method, 221-224
- versus sleep(), 222
### WANs (wide area networks), 197
### WAR files, 368, 554
### warning messages, suppressing, 325
### Web Services, 488
- (see also RESTful services)
- JAX-RS, 487
- JAX-WS, 487
- SOAP, 487
- WSDL, 488
### WEB-INF directory, 361
### @WebFilter annotation, 376
### @WebServlet annotation, 359
### WebSiteReader class, 200
### WebSocket clients, browser as, 402-403
WebSocket protocol, 206, 399
decoders, 410-413
encoders, 410-413
Hello WebSocket, 404-406
Java-based clients, 409
publishing, 413-415
Sec-WebSocket-Key attribute, 403
standardization, 399
traffic monitoring, 406
websockets
annotated endpoints, 403
bidirectionality, 401
client-server communication, 401-409
data format, 403
endpoint instances, 409
lesson, 416
messages receiving, 408
sending, 407
parameter passing, 405
programmatic endpoint, 403
RESTful services comparison, 490
server communication, 403-406
subprotocols, 403
while loop, 52
WhileLoopDemo program output, 53
wildcards, 144
upper bounded, 148
willDecode( ) method, 411
WindowListener, 108
Windows, JDK installation, 4
workbench (Eclipse), 10
wrapper classes, 26
writeExternal( ) method, 191-193
writeObject( ) method, 189
Writer class, 173
WSDL (Web Services Description Language), 488
XML (eXtensible Markup Language), 235
JSON comparison, 489
zero-based arrays, 47
.zip files, 186