


Edward Lavieri, Peter Verhas 著

精通Java 9

(影印版)

Mastering Java 9



 东南大学出版社
SOUTHEAST UNIVERSITY PRESS



Packt>

51

精通 Java 9(影印版)

Mastering Java 9

Edward Lavieri, Peter Verhas 著



南京 东南大学出版社

图书在版编目(CIP)数据

精通 Java 9: 英文/(美)爱德华·拉维耶里(Edward Lavieri),(美)皮特·维哈斯(Peter Verhas)著. —影印本. —南京: 东南大学出版社, 2018.8

书名原文: Mastering Java 9

ISBN 978-7-5641-7735-5

I. ①精… II. ①爱… ②皮… III. ①JAVA 语言—程序设计—英文 IV. ①TP312.8

中国版本图书馆 CIP 数据核字(2018)第 100079 号

图字: 10-2018-099 号

© 2017 by PACKT Publishing Ltd.

Reprint of the English Edition, jointly published by PACKT Publishing Ltd and Southeast University Press, 2018.
Authorized reprint of the original English edition, 2018 PACKT Publishing Ltd, the owner of all rights to publish and sell the same.

All rights reserved including the rights of reproduction in whole or in part in any form.

英文原版由 PACKT Publishing Ltd 出版 2017。

英文影印版由东南大学出版社出版 2018。此影印版的出版和销售得到出版权和销售权的所有者——PACKT Publishing Ltd 的许可。

版权所有, 未得书面许可, 本书的任何部分和全部不得以任何形式重制。

精通 Java 9(影印版)

出版发行: 东南大学出版社

地 址: 南京四牌楼 2 号 邮编: 210096

出 版 人: 江建中

网 址: <http://www.seupress.com>

电子邮件: press@seupress.com

印 刷: 常州市武进第三印刷有限公司

开 本: 787 毫米×980 毫米 16 开本

印 张: 28.5

字 数: 558 千字

版 次: 2018 年 8 月第 1 版

印 次: 2018 年 8 月第 1 次印刷

书 号: ISBN 978-7-5641-7735-5

定 价: 99.00 元

Credits

Authors

Dr. Edward Lavieri
Peter Verhas

Reviewer

Mandar Jog

Commissioning Editor

Kunal Parikh

Acquisition Editor

Denim Pinto

Content Development Editor

Lawrence Veigas

Technical Editor

Abhishek Sharma

Copy Editor

Safis Editing

Project Coordinator

Prajakta Naik

Proofreader

Safis Editing

Indexer

Francy Puthiry

Graphics

Jason Monteiro

Production Coordinator

Arvindkumar Gupta

About the Authors

Dr. Edward Lavieri is a veteran developer with a strong academic background. He earned a doctorate in computer science from Colorado Technical University, an MS in management information systems (Bowie State University), an MS in education (Capella University), and an MS in operations management (University of Arkansas).

He has been creating and teaching computer science courses since 2002. Edward retired from the U.S. Navy as a Command Master Chief after 25 years of service. As the founder and creative director of three19, a software design and development studio, Edward is constantly designing and developing software. He uses a variety of game engines and development tools. His passions include developing adaptive learning systems, educational games, and mobile apps.

Edward has authored *Adaptive Learning for Educational Game Design* (CreateSpace), *Getting Started with Unity 5* (Packt), *Learning AWS Lumberyard Game Development* (Packt), *LiveCode Mobile Development HOTSHOT* (Packt), *LiveCode Mobile Development Cookbook* (Packt), and *Software Consulting: A Revolutionary Approach* (CreateSpace). He was the technical editor for *Excel Formulas and Functions for Dummies* (Wiley Publishing). He has also developed numerous college courses involving computer science, information systems, and game development.

Peter Verhas is a senior developer and software system architect with more than 30 years of software development experience. He currently works for EPAM as a senior developer, where he is involved in many customer projects and actively participates in the education activities of the company. Peter writes a technical blog and is committed to open source software development. He has been using Java since 2005 and is also an author at Java Code Geeks.

About the Reviewer

Mandar Jog is an expert IT trainer with over 15 years of training experience. He is an expert in technologies such as Java, J2EE, and Android. He also holds SCJP and SCWCD certifications. He is an occasional blogger where he makes the readers feel "I can" for the complex concepts in Java and J2EE. He is a regular speaker at many engineering colleges for technical seminars and workshops.

He was also the technical reviewer of the book, *Modular Programming in Java 9* by Packt.

Thanks a lot Tejaswini, you have been my inspiration throughout this journey. I am equally grateful to my son, Ojas; his lovely smiles have always made me push myself further.

www.PacktPub.com

For support files and downloads related to your book, please visit www.PacktPub.com.

Did you know that Packt offers eBook versions of every book published, with PDF and ePub files available? You can upgrade to the eBook version at www.PacktPub.com and as a print book customer, you are entitled to a discount on the eBook copy. Get in touch with us at service@packtpub.com for more details.

At www.PacktPub.com, you can also read a collection of free technical articles, sign up for a range of free newsletters and receive exclusive discounts and offers on Packt books and eBooks.



www.packtpub.com/mapt

Get the most in-demand software skills with Mapt. Mapt gives you full access to all Packt books and video courses, as well as industry-leading tools to help you plan your personal development and advance your career.

Why subscribe?

- Fully searchable across every book published by Packt
- Copy and paste, print, and bookmark content
- On demand and accessible via a web browser

Customer Feedback

Thanks for purchasing this Packt book. At Packt, quality is at the heart of our editorial process. To help us improve, please leave us an honest review on this book's Amazon page at <https://www.amazon.com/dp/1786468735>.

If you'd like to join our team of regular reviewers, you can e-mail us at customerreviews@packtpub.com. We award our regular reviewers with free eBooks and videos in exchange for their valuable feedback. Help us be relentless in improving our products!

Table of Contents

Preface	1
Chapter 1: The Java 9 Landscape	7
Java 9 at 20,000 feet	7
Breaking the monolith	9
Playing around with the Java Shell	10
Taking control of external processes	11
Boosting performance with G1	11
Measuring performance with JMH	11
Getting started with HTTP 2.0	12
Encompassing reactive programming	12
Expanding the wish list	12
Summary	13
Chapter 2: Discovering Java 9	15
Improved Contended Locking [JEP 143]	16
Improvement goals	17
Segmented code cache [JEP 197]	17
Memory allocation	18
Smart Java compilation, phase two [JEP 199]	19
Resolving Lint and Doclint warnings [JEP 212]	19
Tiered attribution for javac [JEP 215]	20
Annotations pipeline 2.0 [JEP 217]	21
New version-string scheme [JEP 223]	23
Generating run-time compiler tests automatically [JEP 233]	23
Testing class-file attributes generated by Javac [JEP 235]	24
Storing interned strings in CDS archives [JEP 250]	26
The problem	26
The solution	27
Preparing JavaFX UI controls and CSS APIs for modularization [JEP 253]	27
JavaFX overview	27
Implications for Java 9	29
Compact strings [JEP 254]	30
Pre-Java 9 status	31
New with Java 9	31

Merging selected Xerces 2.11.0 updates into JAXP [JEP 255]	31
Updating JavaFX/Media to newer version of GStreamer [JEP 257]	32
HarfBuzz Font-Layout Engine [JEP 258]	33
HiDPI graphics on Windows and Linux [JEP 263]	34
Marlin graphics renderer [JEP 265]	35
Unicode 8.0.0 [JEP 267]	35
New in Unicode 8.0.0	35
Updated Classes in Java 9	36
Reserved stack areas for critical sections [JEP 270]	36
The pre-Java 9 situation	36
New in Java 9	37
Dynamic linking of language-defined object models [JEP 276]	38
Proof of concept	39
Additional tests for humongous objects in G1 [JEP 278]	39
Improving test-failure troubleshooting [JEP 279]	41
Environmental information	41
Java process information	42
Optimizing string concatenation [JEP 280]	42
HotSpot C++ unit-test framework [JEP 281]	43
Enabling GTK 3 on Linux [JEP 283]	43
New HotSpot build system [JEP 284]	45
Summary	45
Chapter 3: Java 9 Language Enhancements	47
Working with variable handlers [JEP 193]	48
Working with the AtoMiC Toolkit	49
Using the sun.misc.Unsafe class	51
Eliding depreciation warnings on import statements [JEP 211]	52
Milling Project Coin [JEP 213]	53
Using the @SafeVarargs annotation	53
The try-with-resource statement	54
Using the diamond operator	56
Discontinuing use of the underscore	57
Making use of private interface methods	58
Processing import statements correctly [JEP 216]	60
Summary	62
Chapter 4: Building Modular Applications with Java 9	63
A modular primer	64
Reviewing Java's platform module system [JEP-200]	66

Modularizing JDK source code [JEP-201]	70
Pre-Java 9 JDK source code organization	71
Development tools	72
Deployment	72
Internationalization	72
Monitoring	73
RMI	73
Security	73
Troubleshooting	73
Web services	74
JavaFX tools	74
Java runtime environment	74
Source code	74
Libraries	75
C header files	76
Database	77
JDK source code reorganized	77
Understanding modular run-time images [JEP-220]	77
Runtime format adoption	78
Runtime image restructure	78
Supporting common operations	80
De-privileging JDK classes	80
Preserving existing behaviors	80
Getting to know the module system [JEP-261]	80
Module paths	81
Access-control boundary violations	82
Runtime	82
Modular Java application packaging [JEP-275]	84
Advanced look at the Java Linker	85
Java Packager options	85
JLink - The Java Linker [JEP-282]	89
Encapsulating most internal APIs [JEP-260]	91
Summary	92
Chapter 5: Migrating Applications to Java 9	93
Quick review of Project Jigsaw	94
Classpath	94
The monolithic nature of the JDK	95
How modules fit into the Java landscape	96
Base module	97
Reliable configuration	98
Strong encapsulation	99

Migration planning	100
Testing a simple Java application	100
Potential migration issues	103
The JRE	104
Access to internal APIs	104
Accessing internal JARs	105
JAR URL depreciation	105
Extension mechanism	107
The JDK's modularization	108
Advice from Oracle	109
Preparatory steps	110
Getting the JDK 9 early access build	110
Running your program before recompiling	110
Updating third-party libraries and tools	110
Compiling your application	111
Pre-Java 9 -source and -target options	113
Java 9 -source and -target options	114
Running jdeps on your code	114
Breaking encapsulation	117
The --add-opens option	118
The --add-exports option	118
The --permit-illegal-access option	119
Runtime image changes	119
Java version schema	119
JDK and JRE layout	120
What has been removed	122
Updated garbage collection	123
Deployment	124
JRE version selection	124
Serialized applets	124
JNLP update	125
Nested resources	125
FX XML extension	125
JNLP file syntax	127
Numeric version comparison	127
Useful tools	128
Java environment - jEnv	129
Maven	130
Obtaining the M2Eclipse IDE	131
Summary	134
Chapter 6: Experimenting with the Java Shell	135
What is JShell?	136
Getting Started with JShell	136

Practical uses of JShell	142
Feedback modes	143
Creating a custom feedback mode	148
Listing your assets	150
Editing in the JShell	151
Modifying text	151
Basic navigation	152
Historical navigation	152
Advanced editing commands	153
Working with scripts	153
Start up scripts	153
Loading scripts	154
Saving scripts	154
Advanced scripting with JShell	155
Summary	156
Chapter 7: Leveraging the New Default G1 Garbage Collector	157
Overview of garbage collection	158
Object life cycle	158
Object creation	158
Object mid-life	159
Object destruction	159
Garbage collection algorithms	160
Mark and sweep	160
Concurrent mark sweep (CMS) garbage collection	160
Serial garbage collection	161
Parallel garbage collection	161
G1 garbage collection	161
Garbage collection options	162
Java methods relevant to garbage collection	167
The System.gc() method	168
The finalize() method	170
Pre-Java 9 garbage collection	171
Visualizing garbage collection	172
Garbage collection upgrades in Java 8	173
Case study - Games written with Java	174
Collecting garbage with the new Java platform	175
Default garbage collection	175
Deprecated garbage collection combinations	177
Unified garbage collection logging	178
Unified JVM logging (JEP-158)	179
Tags	179
Levels	180

Decorations	180
Output	181
Command-line options	181
Unified GC logging (JEP-271)	181
Garbage collection logging options	182
The gc tag	184
Macros	184
Additional considerations	185
Persistent issues	186
Making objects eligible for garbage collection	186
Summary	189
Chapter 8: Microbenchmarking Applications with JMH	191
Microbenchmarking overview	192
Approach to using JMH	193
Installing Java 9 and Eclipse with Java 9 support	193
Hands-on experiment	196
Microbenchmarking with Maven	198
Benchmarking options	205
Modes	206
Time units	206
Techniques for avoiding microbenchmarking pitfalls	207
Power management	207
OS schedulers	207
Time sharing	208
Eliminating dead-code and constant folding	208
Run-to-run variance	209
Cache capacity	210
Summary	210
Chapter 9: Making Use of the ProcessHandle API	211
What are processes?	212
The new ProcessHandle interface	213
Getting the PID of the current process	213
Getting information about a process	214
Listing processes	216
Listing children	216
Listing descendants	217
Listing all processes	218
Waiting for processes	219
Terminating processes	220
A small process controller application	222

Main class	223
Parameters class	224
The ParamsAndHandle class	225
The ControlDaemon class	226
Summary	229
Chapter 10: Fine-Grained Stack Tracing	231
Overview of the Java Stack	231
The importance of stack information	232
Example - Restricting callers	234
Example - Getting logger for caller	237
Working with StackWalker	238
Getting an instance of StackWalker	238
RETAIN_CLASS_REFERENCE	238
SHOW_REFLECT_FRAMES	239
SHOW_HIDDEN_FRAMES	239
Final thoughts on enum constants	242
Accessing classes	242
Walking methods	243
StackFrame	245
Performance	246
Summary	246
Chapter 11: New Tools and Tool Enhancements	247
The new HTTP client [JEP-110]	248
The HTTP client before Java 9	248
Java 9's new HTTP client	251
New API limitations	252
Simplified Doclet API [JEP-221]	254
The pre-Java 9 Doclet API	254
API enums	256
API classes	256
API interfaces	257
Problems with the pre-existing Doclet API	258
Java 9's Doclet API	258
Compiler tree API	258
Language model API	262
The AnnotatedConstruct interface	263
The SourceVersion enum	263
The UnknownEntityException exception	265
HTML5 Javadoc [JEP-224]	266
Javadoc search [JEP-225]	271
Introducing camel case search	272

Remove launch-time JRE version selection [JEP-231]	272
Parser API for Nashorn [JEP-236]	273
Nashorn	273
Using Nashorn as a command-line tool	274
Using Nashorn as an embedded interpreter	277
EMCAScript	278
Parser API	279
Multi-release JAR files [JEP-238]	281
Identifying multi-release JAR files	281
Related JDK changes	283
Java-level JVM compiler interface [JEP-243]	284
BeanInfo annotations [JEP-256]	285
JavaBean	285
BeanProperty	286
SwingContainer	287
BeanInfo classes	287
TIFF image input/output [JEP-262]	288
Platform logging API and service [JEP-264]	290
The java.util.logging package	291
Logging in Java 9	293
XML Catalogs [JEP-268]	294
The OASIS XML Catalog standard	294
JAXP processors	295
XML Catalogs prior to Java 9	295
Java 9 platform changes	295
Convenience factory methods for collections [JEP-269]	295
Using collections before Java 9	296
Using new collection literals	299
Platform-specific desktop features [JEP-272]	299
Enhanced method handles [JEP-274]	300
Reason for the enhancement	301
Lookup functions	301
Argument handling	302
Additional combinations	302
Enhanced deprecation [JEP-277]	303
What the @Deprecated annotation really means	304
Summary	304
Chapter 12: Concurrency and Reactive Programming	305
Reactive Programming	306

Reactive programming standardization	307
The New Flow API	309
The Flow.Publisher interface	310
The Flow.Subscriber interface	310
The Flow.Subscription interface	310
The Flow.Processor interface	311
Sample implementation	311
Additional Concurrency Updates	313
Java concurrency	313
Concurrency explained	313
System configurations	314
Java threads	315
Concurrency improvements	318
CompletableFuture API enhancements	319
Class details	320
Enhancements	324
Spin-Wait Hints	325
Summary	326
Chapter 13: Security Enhancements	327
Datagram Transport Layer Security	328
DTLS protocol version 1.0	328
DTLS protocol version 1.2	330
DTLS support in Java 9	333
Creating PKCS12 keystores	334
Keystore primer	334
Java Keystore (JKS)	334
Builder	335
The CallbackHandlerProtection class	336
The PasswordProtection class	336
The PrivateKeyEntry class	337
The SecretKeyEntry class	337
The TrustedCertificateEntry class	338
PKCS12 default in Java 9	339
Improving security application performance	339
Security policy enforcement	340
Permission evaluation	341
The java.Security.CodeSource package	341
Package checking algorithm	342
TLS application-layer protocol negotiation extension	343
TLS ALPN extension	344
The javax.net.ssl package	344