

Dr. Edward Lavieri, Peter Verhas

# Mastering Java 9

Write reactive, modular, concurrent, and secure code



**Packt>**



# Mastering Java 9

Java 9 and its new features add to the richness of the Java language, one of the languages most used by developers to build robust software applications. Java 9 comes with a special emphasis on modularity with its integration with Jigsaw. This book is your one-stop guide to mastering the language.

You'll be provided with an overview and explanation of the new features introduced in Java 9, and the importance of the new APIs and enhancements. Some of the new features of Java 9 are groundbreaking, and if you are an experienced programmer, you will be able to make your enterprise application leaner. You'll be provided with practical guidance in applying your newly acquired knowledge of Java 9 and further information on the future developments of the Java platform. This book will improve your productivity, making your application faster. By learning best practices in Java, you'll become the "go-to" person in your organization.

By the end of the book, you'll not only know the important concepts of Java 9, but you'll also have a nuanced understanding of the important aspects of programming with this great language.

## Things you will learn:

- Write modular Java applications using the newly introduced module system
- Migrate existing Java applications to modular ones
- Understand how to use the G1 garbage collector to improve the performance of your applications
- Discover the possibilities provided by the newly introduced Java shell
- Test your application's effectiveness with the JVM harness
- See how Java 9 provides support for the HTTP 2.0 standard
- Use the new process API
- Discover additional enhancements and features provided by Java 9

**Packt**  
www.packtpub.com

\$ 44.99 US  
£ 37.99 UK

Prices do not include local sales  
Tax or VAT where applicable





# Mastering Java 9

Dr. Edward Lavieri, Peter Verhas



# Mastering Java 9

Write reactive, modular, concurrent, and secure code

**Dr. Edward Lavieri**  
**Peter Verhas**



**BIRMINGHAM - MUMBAI**

# Mastering Java 9

Copyright © 2017 Packt Publishing

All rights reserved. No part of this book may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, without the prior written permission of the publisher, except in the case of brief quotations embedded in critical articles or reviews.

Every effort has been made in the preparation of this book to ensure the accuracy of the information presented. However, the information contained in this book is sold without warranty, either express or implied. Neither the authors, nor Packt Publishing, and its dealers and distributors will be held liable for any damages caused or alleged to be caused directly or indirectly by this book.

Packt Publishing has endeavored to provide trademark information about all of the companies and products mentioned in this book by the appropriate use of capitals. However, Packt Publishing cannot guarantee the accuracy of this information.

First published: October 2017

Production reference: 1031017

Published by Packt Publishing Ltd.

Livery Place

35 Livery Street

Birmingham

B3 2PB, UK.

ISBN 978-1-78646-873-4

[www.packtpub.com](http://www.packtpub.com)

# 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](http://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](http://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](mailto:service@packtpub.com) for more details.

At [www.PacktPub.com](http://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](http://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](mailto: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

<b>Preface</b>	1
<b>Chapter 1: The Java 9 Landscape</b>	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
<b>Chapter 2: Discovering Java 9</b>	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



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

<b>Modularizing JDK source code [JEP-201]</b>	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
<b>Understanding modular run-time images [JEP-220]</b>	77
Runtime format adoption	78
Runtime image restructure	78
Supporting common operations	80
De-privileging JDK classes	80
Preserving existing behaviors	80
<b>Getting to know the module system [JEP-261]</b>	80
Module paths	81
Access-control boundary violations	82
Runtime	82
<b>Modular Java application packaging [JEP-275]</b>	84
Advanced look at the Java Linker	85
Java Packager options	85
<b>JLink - The Java Linker [JEP-282]</b>	89
<b>Encapsulating most internal APIs [JEP-260]</b>	91
<b>Summary</b>	92
<b>Chapter 5: Migrating Applications to Java 9</b>	93
<b>Quick review of Project Jigsaw</b>	94
Classpath	94
The monolithic nature of the JDK	95
<b>How modules fit into the Java landscape</b>	96
Base module	97
Reliable configuration	98
Strong encapsulation	99

<b>Migration planning</b>	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
<b>Advice from Oracle</b>	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
<b>Useful tools</b>	128
Java environment - jEnv	129
Maven	130
Obtaining the M2Eclipse IDE	131
<b>Summary</b>	134
<b>Chapter 6: Experimenting with the Java Shell</b>	135
<b>What is JShell?</b>	136
<b>Getting Started with JShell</b>	136

<b>Practical uses of JShell</b>	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
<b>Working with scripts</b>	153
Start up scripts	153
Loading scripts	154
Saving scripts	154
Advanced scripting with JShell	155
<b>Summary</b>	156
<b>Chapter 7: Leveraging the New Default G1 Garbage Collector</b>	157
<b>Overview of garbage collection</b>	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
<b>Pre-Java 9 garbage collection</b>	171
Visualizing garbage collection	172
Garbage collection upgrades in Java 8	173
Case study - Games written with Java	174
<b>Collecting garbage with the new Java platform</b>	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
<b>Persistent issues</b>	186
Making objects eligible for garbage collection	186
<b>Summary</b>	189
<b>Chapter 8: Microbenchmarking Applications with JMH</b>	191
<b>Microbenchmarking overview</b>	192
Approach to using JMH	193
Installing Java 9 and Eclipse with Java 9 support	193
Hands-on experiment	196
<b>Microbenchmarking with Maven</b>	198
<b>Benchmarking options</b>	205
Modes	206
Time units	206
<b>Techniques for avoiding microbenchmarking pitfalls</b>	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
<b>Summary</b>	210
<b>Chapter 9: Making Use of the ProcessHandle API</b>	211
<b>What are processes?</b>	212
<b>The new ProcessHandle interface</b>	213
<b>Getting the PID of the current process</b>	213
<b>Getting information about a process</b>	214
<b>Listing processes</b>	216
Listing children	216
Listing descendants	217
Listing all processes	218
<b>Waiting for processes</b>	219
<b>Terminating processes</b>	220
<b>A small process controller application</b>	222