

Javier Fernández González

# Java 9 Concurrency Cookbook

**Second Edition**

Master the art of fast, effective Java development with the power of concurrent and parallel programming

This book is based on the Zero Bug Bounce milestone



**Packt**>

# Java 9 Concurrency Cookbook

*Second Edition*

Master the art of fast, effective Java development with the power of concurrent and parallel programming

**Javier Fernández González**

**Packt** >

---

BIRMINGHAM - MUMBAI

# Java 9 Concurrency Cookbook

## *Second Edition*

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 author, 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 2012

Second edition: April 2017

Production reference: 1170417

Published by Packt Publishing Ltd.

Livery Place  
35 Livery Street  
Birmingham  
B32PB, UK.

ISBN 978-1-78712-441-7

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

# Credits

**Author**

Javier Fernández González

**Copy Editor**

Gladson Monteiro

**Reviewer**

Piotr Bzdyl

**Project Coordinator**

Vaidehi Sawant

**Commissioning Editor**

Kunal Parikh

**Proofreader**

Safis Editing

**Acquisition Editor**

Denim Pinto

**Indexer**

Tejal Daruwale Soni

**Content Development Editor**

Nikhil Borkar

**Graphics**

Abhinash Sahu

**Technical Editor**

Subhalaxmi Nadar

**Production Coordinator**

Melwyn Dsa

# About the Author

**Javier Fernández González** is a software architect with almost 15 years of experience in Java technologies. He has worked as a teacher, researcher, programmer, analyst, and writer, and he now works as an architect in all types of projects related to Java, especially J2EE. As a teacher, has taken over 1,000 hours of training in basic Java, J2EE, and the Struts framework. As a researcher, he has worked in the field of information retrieval, developing applications for processing large amounts of data in Java, and has participated as a coauthor in several journal articles and conference presentations. Recently, he worked on developing J2EE web applications for various clients from different sectors (public administration, insurance, healthcare, transportation, and so on). Currently, he works as a software architect. He is the author of the book, *Java 7 Concurrency Cookbook* and *Mastering Concurrency Programming with Java 8* by Packt.

# About the Reviewer

Piotr Bzdyl is focused on Java concurrency topics, including other JVM languages and their libraries, aimed at helping in creating highly concurrent applications (async IO, non-blocking APIs, Scala, Akka, and Clojure). He has been helping teams with JVM tuning and troubleshooting.

He has also created a training course for Java concurrency topics, covering core JDK multithreading concepts as well as those from external libraries and languages (actors, STM, parallel collections, and functional languages).

You can connect with Piotr on LinkedIn at <https://www.linkedin.com/in/piotrbzdyl> and on GitHub at <https://github.com/pbzdyl>. You can follow him on Stack Overflow at <http://stackoverflow.com/cv/piotrebzdyl>.

# 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.



<https://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/178712441X>.

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!





*To Nuria, Paula, and Pelayo, for you infinite love and patience*



# Table of Contents

<b>Preface</b>	1
<b>Chapter 1: Thread Management</b>	7
<b>Introduction</b>	7
<b>Creating, running, and setting the characteristics of a thread</b>	8
Getting ready	9
How to do it...	10
How it works...	13
There's more...	15
See also	15
<b>Interrupting a thread</b>	16
Getting ready	16
How to do it...	16
How it works...	18
There's more...	19
<b>Controlling the interruption of a thread</b>	19
Getting ready	19
How to do it...	20
How it works...	22
There's more...	23
See also	23
<b>Sleeping and resuming a thread</b>	23
Getting ready	23
How to do it...	24
How it works...	25
There's more...	25
<b>Waiting for the finalization of a thread</b>	26
Getting ready	26
How to do it...	26
How it works...	28
There's more...	28
<b>Creating and running a daemon thread</b>	28
Getting ready	29
How to do it...	29
How it works...	31

There's more...	32
<b>Processing uncontrolled exceptions in a thread</b>	32
Getting ready	33
How to do it...	33
How it works...	35
There's more...	35
See also	36
<b>Using thread local variables</b>	36
Getting ready	37
How to do it...	37
How it works...	40
There's more...	40
<b>Grouping threads and processing uncontrolled exceptions in a group of threads</b>	41
Getting ready	41
How to do it...	42
How it works...	44
See also	45
<b>Creating threads through a factory</b>	45
Getting ready	46
How to do it...	46
How it works...	48
See also	49
<b>Chapter 2: Basic Thread Synchronization</b>	51
<hr/>	
<b>Introduction</b>	51
<b>Synchronizing a method</b>	52
Getting ready	53
How to do it...	53
How it works...	59
There's more...	60
See also	61
<b>Using conditions in synchronized code</b>	61
Getting ready	62
How to do it...	62
How it works...	66
There's more...	66
See also	66
<b>Synchronizing a block of code with a lock</b>	67
Getting ready	68

How to do it...	68
How it works...	71
There's more...	73
Avoiding deadlocks	73
See also	74
<b>Synchronizing data access with read/write locks</b>	74
Getting ready...	74
How to do it...	74
How it works...	78
See also	79
<b>Using multiple conditions in a lock</b>	79
Getting ready	80
How to do it...	80
How it works...	86
There's more...	87
See also	88
<b>Advanced locking with the StampedLock class</b>	88
Getting ready	89
How to do it...	89
How it works...	93
There's more...	94
See also	95
<b>Chapter 3: Thread Synchronization Utilities</b>	97
<hr/>	
<b>Introduction</b>	97
<b>Controlling concurrent access to one or more copies of a resource</b>	99
Getting ready	99
How to do it...	100
How it works...	103
There's more...	104
Fairness in semaphores	105
See also	105
<b>Waiting for multiple concurrent events</b>	105
Getting ready	106
How to do it...	106
How it works...	109
There's more...	110
<b>Synchronizing tasks in a common point</b>	110
Getting ready	111
How to do it...	111

How it works...	118
There's more...	119
Resetting a CyclicBarrier object	119
Broken CyclicBarrier objects	119
See also	119
<b>Running concurrent-phased tasks</b>	120
Getting ready	120
How to do it...	121
How it works...	126
There's more...	128
Registering participants in Phaser	129
Forcing the termination of Phaser	129
See also	130
<b>Controlling phase change in concurrent-phased tasks</b>	130
Getting ready	130
How to do it...	131
How it works...	135
See also	137
<b>Exchanging data between concurrent tasks</b>	137
Getting ready	137
How to do it...	138
How it works...	141
There's more...	141
<b>Completing and linking tasks asynchronously</b>	142
Getting ready	143
How to do it...	143
How it works...	148
There's more...	150
See also...	152
<b>Chapter 4: Thread Executors</b>	153
<b>Introduction</b>	153
<b>Creating a thread executor and controlling its rejected tasks</b>	154
Getting ready	155
How to do it...	155
How it works...	158
There's more...	160
See also	162
<b>Executing tasks in an executor that returns a result</b>	162
Getting ready	162

How to do it...	162
How it works...	166
There's more...	166
See also	167
<b>Running multiple tasks and processing the first result</b>	167
Getting ready	167
How to do it...	167
How it works...	171
There's more...	172
See also	172
<b>Running multiple tasks and processing all the results</b>	173
Getting ready	173
How to do it...	173
How it works...	177
There's more...	177
See also	177
<b>Running a task in an executor after a delay</b>	178
Getting ready	178
How to do it...	178
How it works...	180
There's more...	180
See also	181
<b>Running a task in an executor periodically</b>	181
Getting ready	181
How to do it...	182
How it works...	184
There's more...	185
See also	186
<b>Canceling a task in an executor</b>	186
Getting ready	186
How to do it...	186
How it works...	188
There's more...	188
See also	189
<b>Controlling a task finishing in an executor</b>	189
Getting ready	189
How to do it...	189
How it works...	192
See also	193



<b>Separating the launching of tasks and the processing of their results in an executor</b>	193
Getting ready	193
How to do it...	193
How it works...	198
There's more...	198
See also	199
<b>Chapter 5: Fork/Join Framework</b>	201
<b>Introduction</b>	201
<b>Creating a fork/join pool</b>	204
Getting ready	204
How to do it...	205
How it works...	209
There's more...	210
See also	211
<b>Joining the results of the tasks</b>	211
How to do it...	212
How it works...	219
There's more...	220
See also	221
<b>Running tasks asynchronously</b>	221
How to do it...	222
How it works...	226
There's more...	228
See also	229
<b>Throwing exceptions in the tasks</b>	229
Getting ready	229
How to do it...	230
How it works...	232
There's more...	234
See also	234
<b>Canceling a task</b>	235
Getting ready...	235
How to do it...	235
How it works...	241
See also	242
<b>Chapter 6: Parallel and Reactive Streams</b>	243
<b>Introduction</b>	243