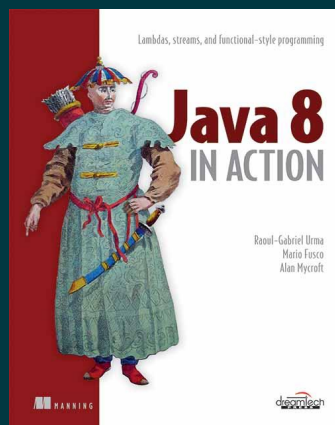




Java 8 in Action

By [Raoul-Gabriel Urma](#), [Mario Fusco](#), [Alan Mycroft](#)

Paperback

9789351197430

₹899.00

Pages: 424

Description

Every new version of Java is important, but Java 8 is a game changer. Java 8 in Action is a clearly written guide to the new features of Java 8. It begins with a practical introduction to lambdas, using real-world Java code. Next, it covers the new Streams API and shows how you can use it to make collection-based code radically easier to understand and maintain. It also explains other major Java 8 features including default methods, Optional, CompletableFuture and the new Date and Time API.

About the Author

Raoul-Gabriel Urma, Mario Fusco, Alan Mycroft

Raoul-Gabriel Urma is a software engineer

Table of Contents

Part 1 Fundamentals

Chapter 1 Java 8: why should you care?

- Why is Java still changing?
- Functions in Java
- Streams
- Default methods
- Other good ideas from functional programming

Chapter 2 Passing code with behavior parameterization

- Coping with changing requirements
- Behavior parameterization
- Tackling verbosity
- Real-world examples

Chapter 3 Lambda expressions

- Lambdas in a nutshell
- Where and how to use lambdas
- Putting lambdas into practice: the execute around pattern
- Using functional interfaces
- Type checking, type inference and restrictions

- Method references
- Putting lambdas and method references into practice!
- Useful methods to compose lambda expressions
- Similar ideas from mathematics

Part 2 Functional-style data processing

Chapter 4 Introducing streams

- What are streams?
- Getting started with streams
- Streams vs. collections
- Stream operations

Chapter 5 Working with streams

- Filtering and slicing
- Mapping
- Finding and matching
- Reducing
- Putting it all into practice
- Numeric streams
- Building streams

Chapter 6 Collecting data with streams

- Collectors in a nutshell
- Reducing and summarizing
- Grouping
- Partitioning
- The Collector interface
- Developing your own collector for better performance

Chapter 7 Parallel data processing and performance

- Parallel streams
- The fork/join framework
- Spliterator

Part 3 Effective Java 8 programming

Chapter 8 Refactoring, testing and debugging

- Refactoring for improved readability and flexibility
- Refactoring object-oriented design patterns with lambdas
- Testing lambdas
- Debugging

Chapter 9 Default methods

- Evolving APIs
- Default methods in a nutshell

- Usage patterns for default methods
- Resolution rules

Chapter 10 Using Optional as a better alternative to null

- How do you model the absence of a value?
- Introducing the Optional class
- Patterns for adopting Optional
- Practical examples of using Optional

Chapter 11 CompletableFuture: composable asynchronous programming

- Futures
- Implementing an asynchronous API
- Make your code non-blocking
- Pipelining asynchronous tasks
- Reacting to a CompletableFuture completion

Chapter 12 New Date and Time API

- LocalDate, LocalTime, Instant, Duration and Period
- Manipulating, parsing and formatting dates
- Working with different time zones and calendars

Part 4 Beyond Java

Chapter 13 Thinking functionally

- Implementing and maintaining systems
- What's functional programming?
- Recursion vs. iteration

Chapter 14 Functional programming techniques

- Functions everywhere
- Persistent data structures
- Lazy evaluation with streams
- Pattern matching
- Miscellany

Chapter 15 Blending OOP and FP: comparing Java 8 and Scala

- Introduction to Scala
- Functions
- Classes and traits

Chapter 16 Conclusions and where next for Java

- Review of Java 8 features
- What's ahead for Java?



