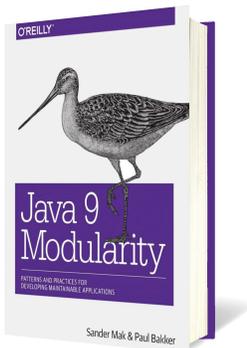


Book Reviews



Java 9 Modularity

By Sander Mak and Paul Bakker
300 pages

Java 9 Modularity

Reviewed by Andrew Binstock

The introduction of modules in Java 9 significantly changed how Java applications are built and delivered. These changes are particularly important for developers of Java libraries, who need to work out their strategy for delivering modular JAR files while continuing to provide the traditional bits that run on JVMs prior to this new release. Although you certainly can run apps on the Java 9 runtime without using modules, it is expected that most sites will switch over to module-based binaries during the next few years. Some sites, especially those wrestling with so-called “classpath hell,” will likely find incentive to move to modules as quickly as possible. Those sites will discover a trove of useful information in Mak and Bakker’s new work.

The book opens with a detailed explanation of what modules are and how they work. The first four chapters cover the anatomy and use of modules in detail, with plenty of examples. It’s a very readable guide. The remaining 180 pages are where the value is really apparent. These pages start out covering modularity patterns, which are ways of architecting modules so that they work together ideally. The goal is to find the balance between devising modules that naturally (that is, conceptually) fit together while creating the minimum number of dependencies on external modules. This tension is familiar to Java architects designing JARs. However, the impetus to get the design right in traditional JARs historically has been more of a desirable goal than an imperative. With modules, it becomes a much more serious proposition. The whole point of modularity is to manage dependencies intelligently to get rid of classpath conflicts and enable delivery of modestly sized binaries.

The authors tackle many aspects of the problem: splitting modules, aggregating modules, and creating new modules as facades. They then move into technical problems such as encapsulating resources, after which they explore building modules specifically for use in containers.

The final sections have pointers for library designers and a handy section on configuring tools, such as Maven and Gradle, for modules.

Wedge into all this goodness is a lengthy discussion of how to run Java 9 without migrating to modules. This will be particularly useful to sites that are planning to adopt modules at some future point and want to understand the full scenario, starting with limited migration to the new runtime and then slowly implementing the considerations presented in the rest of the book.

Taken together, all these topics represent a comprehensive overview of Java 9 modularity. The writing is clear and easy to understand, and the authors do not expect the reader to know much more than how to program in Java; low-level details (such as how classloaders work) are explained on the fly. This book was explicitly recommended at DevOxx in November by members of the core Java team, and I fully agree with their assessment.



Sponsored Content

Contact

US Sales: +1.800.633.0738

Global Contacts

Support Directory

About Us

Careers

Communities

Company Information

Downloads and Trials

Java for Developers

Java Runtime Download

Software Downloads

News and Events

Acquisitions

Blogs

Events

