Reviewed by Andrew Binstock

July 6, 2020

It's no secret that many developers today are excessively reliant on solving programming language issues by searching on the web or fishing for answers on sites such as Stack Overflow. If you use the web for problem-solving, you have surely found that the experience can be frustrating and occasionally counterproductive. I often come across recommendations that have defects, exhibit code smells, or are written without fully understanding the problem. And sometimes, accurate answers are undermined by invalid objections. Despite occasional helpful nuggets, the web can be an unrewarding place for "copy pasta" coders.

If this describes your experience—which I think we've all had when we've needed a quick and dirty solution—consider a novel approach: going to the experts as your first source of solutions. The ideal expert source is a recipe book such as the present volume. The idea of "recipes" in software development means getting a step-by-step explanation of how to solve a specific problem. Good recipe books also give plenty of background information so that if you need to customize the sample code for a specific situation, you understand what to do and how to do it.

There are quite a few Java recipe books available, so it's important to choose a volume that is both comprehensive and written by a true Java expert. A good rule of thumb is to favor the works of Java Champions. If you do this with Java SE titles, you'll find two excellent options. Earlier, I reviewed Modern Java Recipes, by Ken Kousen, which covers Java 8 and 9. The second is the present volume by Ian Darwin, which goes through Java 14. Both volumes are excellent, although the Darwin volume is significantly more comprehensive. [Darwin is a frequent contributor to Java Magazine. —Ed.]

This volume starts with installation of Java, traipses through the grammar of various language features, collections, and then into I/O. The latter coverage includes file I/O (which is an endless topic for recipes, in my experience) and goes on to all forms of network and web I/O. The latter topic is explored both from the client and server sides.

Additional sections cover reflection, data handling, and the packaging of applications. Darwin also provides considerable
coverage of secondary topics, such as interfacing Java with other languages, using Java in data science with R, functional programming in Java, and so on.

A particularly interesting choice Darwin makes for some topics is to show solutions from multiple points of view, which adds considerable depth. For example, the section on processing JSON data has recipes showing solutions with plain Java, Jackson, org.json, and JSON-B. Likewise, logging enjoys coverage of the major frameworks.

As to the actual recipes, I am impressed by how many topics are covered, including many less-common problems. For example (one of a multitude I could choose from), there is a three-page explanation of how to duplicate an I/O stream as it is being written to. I choose this example because it illustrates the kind of material you'll find throughout: a deep explanation illustrated with plenty of code, which has its own set of embedded comments. And unlike several books I've reviewed recently, the code is properly formatted and never wraps to the next line. The explanations also benefit from extensive cross-referencing to other recipes.

This is the fourth edition of this book and you can see that the cumulative editions have resulted in a very well-thought-through and diligently presented set of recipes.

I have only a few minor complaints. The discussions of tools are far too short, and I think those pages would have been better used for additional code recipes. Also, some topics I'd expect to find received scant coverage. For example, the topic of generics, which is surely one of the trickiest grammatical topics, barely gets any mention at all. By the same token, I confess, a book that covered every topic in Java to the depth Darwin shows here would be vast indeed.

I highly recommend this book and suggest that you get the ebook version so you can quickly search for the topic you want and solve programming problems quickly and correctly.

Andrew Binstock

Andrew Binstock (@platypusguy) was formerly the editor in chief of Java Magazine. Previously, he was the editor of Dr. Dobb's Journal. He co-founded the company behind the open-source iText PDF library, which was acquired in 2015. His book on algorithm implementation in C went through 16 printings before joining the long tail. Previously, he was the editor in chief of UNIX Review and, earlier, the founding editor of the C Gazette. He lives in Silicon Valley with his wife. When not coding or editing, he studies piano.