Book review

Introduction to the art of programming using Scala, by Mark C. Lewis, Chapman and Hall/CRC Press, 2012, £46.99 (paperback) ISBN-10:1439896666
doi: 10.1017/S0956796814000252

This book appeared in very interesting times; people are now getting tired of the only paradigm that has been legal in mainstream programming – OOP – and have started looking around for new options. Scala is the language that, like a gateway drug, allows us to transition from the old world of Enterprise Javabeans, patterns, Springs and hibernates to the brave new world of folds, mapreduce, monads, lenses, and Kleisli categories. This world, before Scala, was only available to those who mastered Haskell or a similar language. Now that Scala has opened new horizons, Java programmers don't have to be defensive; they can gradually move over to the "higher-order" programming style, which brings a lot of joy, and a lot of job opportunities, too. So far being a Scala programmer on the job market means leisurely choosing the job you expect most fun from. But even if you are not a Java programmer, but decided to start from the scratch, this book is the right one: having spent enough time and effort, you can become a regular member of the programming community, indeed one of its best branches, the Scala programming community. Scala communities have grown around the world lately; estimates include thousands in San Francisco Bay Area and New York Area, fifteen hundred in London, and hundreds in other European cities. It makes sense to read this book if you plan to become one of them.

The book is a perfect textbook. It contains enough exercises and projects, so that students can walk through it in something like a year, and cover both computing practices in general (if not science) and the Scala language in particular. If I were teaching programming in general by this book, I'd use most of the chapters right away. Alternatively, this book can be used to teach several different courses, so that the instructor could choose one track or another, depending on their goals.

The book contains enough material to cover a moderate (and traditional) computer science course. It includes standard data structures - lists, queues, stacks, trees, spatial trees, priority queues; it covers sorting and searching, recursion, concurrency (as it is done in Java), actors (concurrency, Erlang style), regular expressions and context-free grammar parsing. For the practical side of our profession, it goes into Java graphics, GUI building, XML processing, file handling and network operations. It also covers the areas that programmers usually learn by practice: editors, version control, refactoring and object-oriented concepts. One can only imagine a beginning programmer who has never opened a file for editing, and in their program never opened a file for reading and writing … at least, in Scala.

One area is definitely missing, or not covered enough, however: that is functional programming. The art of programming, as presented in this book, is represented according to the OOP tradition. The now common concepts of map, fold, filter, collect and partition are absent; so, no functional programming, and no monads, either. This may be the right approach if the target audience is the beginner, but it is open to discussion. Some authors suggest to omit totally the "imperative" programming style in teaching new students, and instead just show them lambdas, pure functions, and then move on to monadic operations - but that would be a different book. These days students are hardly ever beginners; they do

have some experience, and so expect something more traditional (or contemporary), matching the knowledge they already have, and this book is the right one.

The goal of this book does not seem to be to teach students the highest level of programming known to the world, but to provide them with the tools that are widespread among the literate part of the community. On the other hand, if you compare it with "Programming in Scala", by Odersky, Spoon and Venners, you will discover that, having the same size, this book helps readers to learn much more material. It does not focus much on the subtleties of Scala; but the thing is, subtleties of Scala tend to change with time, while data structures remain the same, and the problems we are trying to solve seem to be remaining the same, too; so even if you want to just learn Scala, this is the right book.

The author's approach in presenting software development tools is interesting. He starts with vim and notepad, so the students will be able to edit something without having to master sophisticated IDEs, like IntelliJ or Eclipse. Eclipse is introduced much later. Along with Eclipse comes SVN, and these choices are rather questionable. Git is the more modern version control system, and github is definitely what you discover if you are looking for solutions, code samples, libraries. Stackoverflow points to it all the time; but Stackoverflow is also not mentioned in the book. Probably the existence of the internet outside should be covered more in the next editions of the book.

GUI development, even the rather advanced Scala version of Swing library, is not what people usually do these days; instead, web is everywhere. In the same vein, when talking about XML, it might make sense to dedicate more time specifically to HTML. To show that one can write a primitive web server in about 20 lines of Scala code may be a great exercise; simpler and more contemporary than walking through GUI libraries with their strange and contradictory ideas. Also, I would definitely dedicate at least a chapter to the concepts of functional programming. Oh, and of course, mobile device programming. I am sure that teaching the reader to write something simple for Android will make them so happy. Android is a great platform for JVM-based languages; and Scala for Android is practiced by many already.

Another issue is that the book may be *too* practical. Trees, lists, even sets are introduced not as abstract concepts, but as their implementations. It would make more sense to separate ideas and their implementations - as is done with priority queues, which are introduced in Chapter 26, while their implementation as binary heaps is postponed until Chapter 32.

One of the best parts of the book is the accompanying website, http://www.programmingusingscala.net/, where all chapters are also presented in video form; watching graphics show up and move is much more convincing and engaging than just reading about it. Unfortunately, I could not find any link to the code repository; some pieces of code in the book take more than one page, just Scala code; grabbing them from github or from gist would make more sense than just retyping.

In any case, I would recommend this book as the best one for an introductory course of programming; the fact that it is in Scala makes the course much more practically useful. If I were teaching programming, I'd definitely use this book; and if you are a beginner, or a medium level programmer, you will find a lot of interesting and useful knowledge here.

VLAD PATRYSHEV
HealthExpense Inc, San Jose, California, USA