

Review of the book

*"A Concise and Practical Introduction to Programming Algorithms in  
Java"*

by Frank Nielsen  
Springer, 2009

ISBN: 978-1-84882-338-9

Luigi Lo Iacono

## 1 What the book is about

The book at hand by Frank Nielsen is a textbook mainly targeted to undergraduate students as a very first course in programming. As the title of book says, the programming language used is Java(TM). Following the demands of the targeted audience, the book introduces the topics programming and algorithms without requiring prior knowledge. More advanced topics and concepts such as for example object orientation are intentionally omitted in order to stay focused with the book's goal.

The book is organized into ten chapters split over two parts as follows:

- The first part is related to programming. Based on the programming language Java(TM), the foundations are introduced to get ready to program elementary tasks. After the presentation of the basic skeleton of a Java program and the required steps to edit, compile and run the compulsive *Hello World*, chapter 1 explains the fundamental notions of expressions, variables and assignments with type checking. Arithmetic operations and their priority order as well as primitives types and their implicit casting rules are targeted in addition to other fundamentals such as commenting and indenting of Java(TM) programs as well as I/O, debugging and IDEs. Chapter 2 is devoted to the control of program and instruction flows. The instruments for conditional branching (simple and multiple choices) and loops are introduced together with related aspects such as boolean expressions and relational operators. The concept of structuring and organizing code in subroutines is explained in chapter 3. As mentioned earlier, this topic is dealt with from a procedural programming standpoint keeping out the additional complexity of object orientation. Rather the syntax of functions in Java(TM) is described and its two essential merits are emphasized, namely modularity and code re-use. Additionally, this chapter deals with the local and global variables, the Java(TM) function call stack and recursions. The first part of the book closes with chapter 4, which provides explanations on the management and processing of many data items using arrays. It describes the usage of single- and multi-dimensional arrays from the declaration and instantiation of arrays to the determination of the length of an arrays, the passing of arrays as function arguments, printing of arrays and finally memory-related aspects.
- Building on the foundations of the first part, the second part of the book focuses on more advanced data structures and algorithms. To introduce the elementary topics, the first part of the book omitted the object orientation for complexity reasons. For the more advanced subjects addressed in the second part, basic principles and notations used in object oriented programming are required and are therefore introduced in chapter 5. From a programming perspective, the declaration of classes and their instantiation to objects are discussed. Further concepts, such as object references, cloning and equality testing are treated as well. Chapter 5 closes by emphasizing the general explanations in terms of the `String` class.

Chapter 6 goes into searching and sorting. As searching algorithms, the author introduces the sequential and bisection search strategies. Besides demonstrating the implementation of these algorithms, their complexity is discussed. Since the bisection search strategy requires a fully sorted array of objects, sorting algorithms are presented next. The targeted sort strategies are selection and quick sort. Again, implementation examples as well as complexity discussions are given.

An important data structure takes the center stage of chapter 7. The concept of linked lists is explained and the implementation of this abstract data structure is illustrated step-wise. Necessary aspects including the insertion of elements to and the deletion of elements from as linked lists as well as traversing, sorting, copying and pretty printing of linked lists are covered.

Chapter 8 continues with three more most fundamental data structures: queues, heaps and stacks. In conformance with the all the other chapters, the underlying theoretical concepts are explained first, before the implementation of these algorithms in Java(TM) is explained. All basic functions related to these abstract data structures (mainly inserting and fetching) together with their implementation are discussed.

Fundamental optimization problems are considered in chapter 9. It starts by motivating, that solving certain types of problems with exhaustive search strategies—in exploring the full configuration space to retain the solution—, may not be feasible. The knapsack problem is used for this discussion. Algorithms presented for solving optimization problems are backtracking, greedy algorithms based on heuristics to solve the 0-1 knapsack and set cover problems.

The last chapter leaves the scope of this book and talks more general about the history of the digital world and gives a short outlook on how the future of computing in the 21st century might look like.

Each chapter ends with a section containing exercises to practice and deepen the introduced contents self-contained. As an appendix, the author provides an example exam (including an example solution), for a 2-hour test reviewing the main concepts of programming in Java(TM).

## 2 What is the book like

The book is written in a clear and fluent way. The general style is concise and to the point. The structure is well-organized and although the chapters build on each other, the author managed to write them in a way that also allows to read them selectively.

Mainly targeting undergraduate students, the book gives a good understandable introduction to programming based on the Java(TM) programming language. Most importantly, this book does not make any assumptions on prior knowledge in neither programming nor computer science. It is therefore not only suited for undergraduate students in the computer science field, but rather also for students from other domains requiring programming skills. The way the book is organized and written, it can be used by students for self-studies or practical hands-on training as well as by lecturers as a textbook or a source for exercises. In the appendix of the book, an example exam including an example solution is given which might serve students as a preparation for their exam but also as an guideline for junior lecturer to prepare their exam.

The author finds an appropriate balance between explaining the underlying theoretical concepts and showing their implementation using the Java(TM) programming language. Based on a quite large amount of source code, the author provides a strong guidance for novice programmer. In addition to the topics themselves, the author also presents a variety of related examples in order to give further explanations and to deepen the content, including Euclid's algorithm to compute the greatest common divisor, Koch's mathematical recursive snowflakes and Sierpinski's triangle fractal.

Another positive aspects of the book is, that the discussed data structures are all explained from an empty space viewpoint. Although most of these elements can be found in the standard Java(TM) runtime environment, the author decided to leave them out of the scope. The author only provides references to the according Java APIs, for further readings. This strategy fits very well to the overall target of the book. Again, the overhead coming with these frameworks contained in the Java(TM) runtime environment is sacrificed for the sake of staying focused on the main topic of learning programming.

The main disadvantage of the book comes from the fact, that the author does not want the targeted audience to get lost in object oriented details. This is a clear design decision for this book and in fact, it is a good and right decision especially when taking into account that the target reader is a programming freshman. However, as a consequence, the source code in this book consists of almost only class variables and class method with the according `static` keyword being omnipresent. This may lead to the adoption of a poor programming style and bad programming habits. Although chapter 5 introduces basic object oriented principles, notations and mechanisms, these instruments are not used afterwards.

It does not get clear, why chapter 10 is part of this book, since it includes a general and very short discussion on the history of computing and not more then a glimpse into the importance of computer sciences in the 21st century. Regarding the real topic of the book and the length and depth of the present discussions, omitting this chapter does not seem to hurt at all.

### 3 Recommendation

I would certainly recommend this book for the audience it is targeting. It provides a good starting point for novice programmers, regardless of whether they are working in the computer science field or for any other discipline in which programming skills are required. When learning programming, gaining practical experiences is very important. With the numerous examples and exercises included in the book the author respects exactly this requirement and supports the reader with a large amount of material for hands-on training. Henceforth, this book is not only a valuable source for undergraduate students but also for lecturer who can benefit from this book in terms of a source for many programming examples and exercises.

*The reviewer is a senior researcher with NEC Laboratories Europe.*