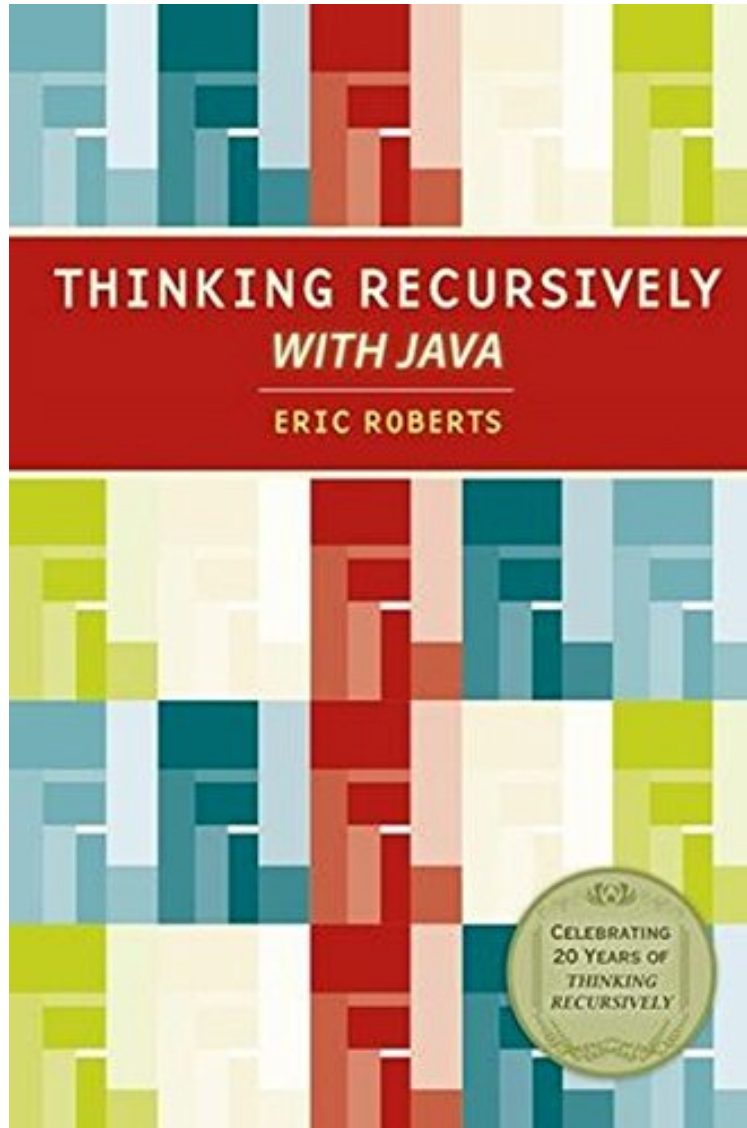


[DOWNLOAD] Thinking Recursively with Java

# Thinking Recursively with Java

*Eric S. Roberts*

*DOC | \*audiobook | ebooks | Download PDF | ePub*



#819160 in Books 2005-11-09Ingredients: Example IngredientsFormat: Deluxe EditionOriginal language:EnglishPDF # 1 9.20 x .40 x 6.10l, .55 #File Name: 0471701467188 pages | File size: 41.Mb

**Eric S. Roberts : Thinking Recursively with Java** before purchasing it in order to gage whether or not it would be worth my time, and all praised Thinking Recursively with Java:

5 of 5 people found the following review helpful. SuperbBy JacobeanEraI don't program in Java, mainly c++. If you program in any c-style language you will find this book insightful in Thinking about recursion. The examples are great and the book is written in a very nice style. I came upon this book by chance. I owned a second hand copy of the original with included examples in Pascal. I much prefer this edition. To get the most out of this book, I'd recommend

you attempt and think about the questions at the end of each of the chapters. The mathematical induction related chapter was eye opening and very enjoyable. I'd recommend any text that Eric Roberts has written, he is obviously gifted in his ability to educate and make it so rewarding in the process. Thank you Gary11 of 11 people found the following review helpful. A good book on recursion implemented in Java By calvinme In mathematics and computer science, recursion specifies a class of objects or methods by defining a few very simple base cases or methods, and then defining rules to break down complex cases into simpler cases. Of the books out there on recursion, this really is a very good one. This is the 20th anniversary edition of the author's classic book "Thinking Recursively", which was published in 1986, with all code illustrations now done in the Java programming language. In fact, this book has the exact same number of chapters with the exact same names as the original edition with improved and expanded material. The author does a good job of comparing the procedural approach to the recursive approach while showing an example of a producing a "context-free grammar" in which the procedural approach fails. This is not only a good discussion of the shortcomings of the procedural approach, it has some concrete examples that explain the concept of the context-free grammar better than programming language textbooks that are dedicated to the cause. Next the book offers up the recursive solution for the familiar and classic Tower of Hanoi problem. This section uses index cards to illustrate the solution. Necessary tasks for each subgoal are listed on the index card and are gradually marked out to indicate progress in moving the disks from one tower to the other. There are also discussions on permutations and sorting that are best solved by recursion. Of course, you can find this information in most good algorithm textbooks, but you can't usually find in those textbooks the code examples that show the complete solution to the problem that you find in this book. Where the book is particularly excellent and unique is in later chapters when it includes recursive solutions to graphical applications and even introduces the GPen class, which is defined in the acm.graphics package developed by the Association of Computing Machinery (ACM), for this purpose. Drawings of fractals are used as an application to show how one carries out this recursive drawing. It is also explained that Java's graphical classes, though extensive, are not suited to recursive drawing. This book contains quite a bit of mathematics as well as Java code, so the reader should already be proficient in Java programming as well as discrete mathematics. This book could serve as a textbook since besides its many examples it includes exercises with solutions. 3 of 3 people found the following review helpful. rapid read By W Boudville This little book is a minor classic. First published 20 years ago, it gave an extended explanation of recursion, which is a vital concept in computer science. Of course, Java did not exist back then, so that edition used Pascal for the example code. But Pascal has undergone a severe downturn. Hence this second edition has code in Java. Classic pedagogic examples like the Towers of Hanoi are shown to yield to a recursive assault. The code examples are concise. Not atypical of recursive methods. If you find the entire idea of recursion to be a little weird, you can focus on the text's examples. Unlike code for, say, GUI building, which is often voluminous, recursion is subtler. And far more elegant, if you appreciate this type of abstraction. Roberts also brings up fractals. Another trendy topic. He shows that recursion and fractals are a very natural fit. The concept of self-similarity that underpins fractals is so easy to express in a recursive routine. If you understand recursion by this point in the text, a bonus may be the nice insight this gives into fractals. The only minor dissenting point is that there seems to be no discussion about when you should not code a recursive solution, even if such a method is possible. If you have a large data set, that triggers a stack or heap overflow, due to repeated, recursive method calls, where each call pushes return address information for that method onto a stack. I have had to rewrite sections of my own code, that were originally recursive, due to this.

Updated and revised to include the use of Java for programming examples, this book provides readers with a thorough and clear introduction to the difficult concept of recursion. Uses a broad range of examples to illustrate the principles used in recursion and how to apply them to programming. Features imaginative examples along with various exercises and their solutions.

From the Back Cover To understand recursion, you first have to understand recursion. If reading the statement above gives you a mild headache, you're not alone. Recursion is not only one of the most important concepts in the computer science curriculum; it's also one of the most challenging to understand. Now in *Thinking Recursively with Java*, author and award-winning teacher Eric Roberts, demystifies this often-frustrating topic by equipping you with effective problem-solving strategies that enable you to "think recursively." Based on Roberts's now classic text, *Thinking Recursively* (Wiley 1986), this 20th anniversary edition now uses Java, making recursion even more relevant to today's students. Features Learn how to apply recursive techniques, so you can succeed in advanced CS courses that depend on the use of recursive strategies. The book's ample selection of examples and exercises (more than the typical general text) enables you to work through as many problems as you need to master recursive techniques. Examples now use Java, making the book compatible with modern approaches to introductory computer science. Expanded chapters on recursive backtracking and graphical applications support interesting examples enabled by current technology. The code in the book is fully compatible with the libraries produced by the ACM Java Task Force, as well as with other standard approaches to teaching Java. About the Author Eric Roberts is Professor of Computer

Science and John and Cynthia Gunn University Fellow in Undergraduate Education at Stanford University. He is widely recognized as an expert teacher and has won numerous teaching awards at Stanford. He was the principal author of the ACM/IEEE-CS report on Computing Curricula 2001 and received the ACM-SIGCSE Award for Outstanding Contributions to Computer Science Education in 2003. About the Author Eric Roberts is Professor of Computer Science and John and Cynthia Gunn University Fellow in Undergraduate Education at Stanford University. He is widely recognized as an expert teacher and has won numerous teaching awards at Stanford. He was the principal author of the ACM/IEEE-CS report on Computing Curricula 2001 and received the ACM-SIGCSE Award for Outstanding Contributions to Computer Science Education in 2003.