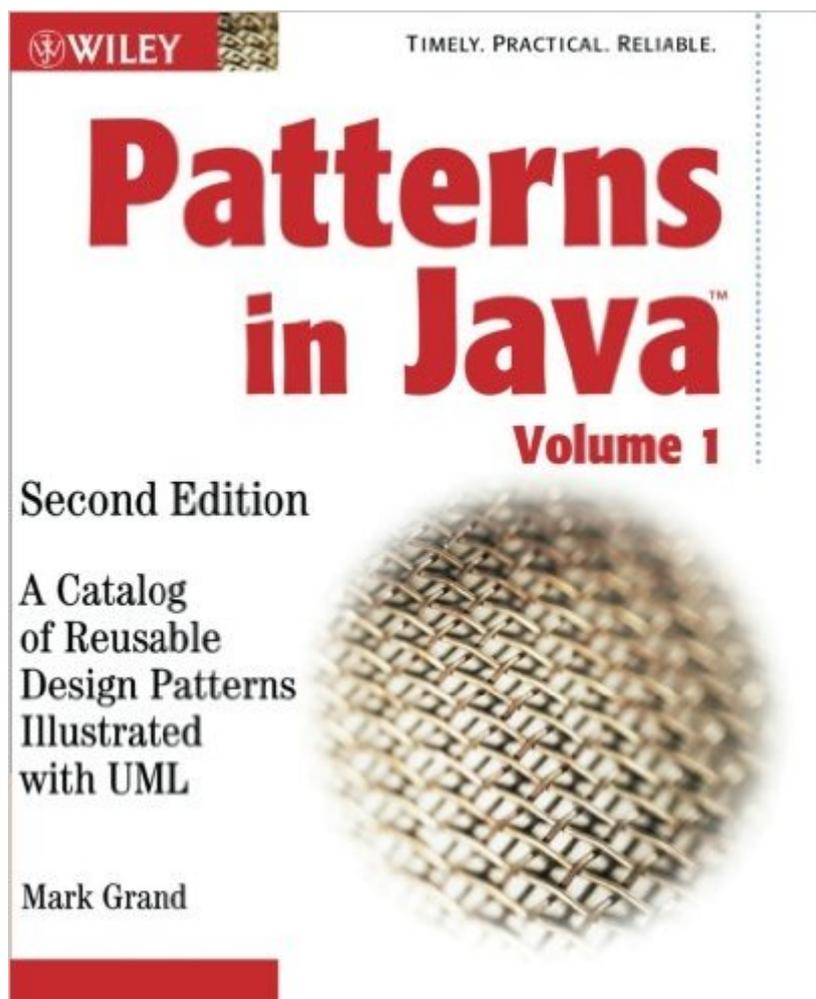# Patterns In Java: A Catalog Of Reusable Design Patterns Illustrated With UML, 2nd Edition, Volume 1

Synopsis

"This is the best book on patterns since the Gang of Four's Design Patterns. The book manages to be a resource for three of the most important trends in professional programming: Patterns, Java, and UML." â "Larry O'Brien, Founding Editor, Software Development Magazine Since the release of Design Patterns in 1994, patterns have become one of the most important new technologies contributing to software design and development. In this volume Mark Grand presents 41 design patterns that help you create more elegant and reusable designs. He revisits the 23 "Gang of Four" design patterns from the perspective of a Java programmer and introduces many new patterns specifically for Java. Each pattern comes with the complete Java source code and is diagrammed using UML. Patterns in Java, Volume 1 gives you: 11 Behavioral Patterns, 9 Structural Patterns, 7 Concurrency Patterns, 6 Creational Patterns, 5 Fundamental Design Patterns, and 3 Partitioning Patterns Real-world case studies that illustrate when and how to use the patterns Introduction to UML with examples that demonstrate how to express patterns using UML The CD-ROM contains: Java source code for the 41 design patterns Trial versions of Together/J Whiteboard Edition from Object International (www.togetherj.com); Rational Rose 98 from Rational Software (www.rational.com); System Architect from Popkin Software (www.popkin.com); and OptimizeIt from Intuitive Systems, Inc.

Book Information

Paperback: 544 pages

Publisher: Wiley; 2 edition (September 17, 2002)

Language: English

ISBN-10: 0471227293

ISBN-13: 978-0471227298

Product Dimensions: 7.5 x 1.3 x 9.2 inches

Shipping Weight: 2.2 pounds (View shipping rates and policies)

Average Customer Review: 3.0 out of 5 starsÂ Â See all reviewsÂ (65 customer reviews)

Best Sellers Rank: #1,567,394 in Books (See Top 100 in Books) #12 inÂ Books > Computers & Technology > Programming > Software Design, Testing & Engineering > Software Reuse #117 inÂ Books > Computers & Technology > Programming > Software Design, Testing & Engineering > UML #517 inÂ Books > Textbooks > Computer Science > Object-Oriented Software Design

Customer Reviews

There appears to be a flame war in a bookshop about this book, with everybody giving it either 1

star or 5. I believe the truth is somewhere in between.I'm using the book as a course text for a final year undergraduate course I'm teaching which focusses on patterns. It's far from ideal, but there's nothing out there better as far as I know. There are many typos and thing which could be explained better, but I disagree with those reviewers who claim that the author doesn't understand the subject - in my opinion he clearly does. With one exception (the bizzare characterisation of Marker Interface as a fundamental design pattern) I don't believe there's anything fundamentally wrong.I'd like to encourage those people who are complaining that it's rubbish to either write a better book, or contribute detailed comments to the author, so he can produce an improved second edition (I'll be doing the latter). It has the potential to be a very good book.I agree with those who are saying that that volume 2 is very disappointing, but reviews of that shouldn't be contributing to the "score" of volume 1.

Having written several Java books myself and being a self confessed design pattern addict, I was looking forward to reading this book. Unfortunately I was immensely disappointed with the content. I agree with most of the negative comments written here about both Vols 1 and 2 and only add my own voice to the crowd to ensure the weight of numbers prevails.The big problem with this book is that it is _so_ inaccurate, both syntactically and semantically, you cannot _trust_ the content.Some of the text is accurate: for example the description of the Visitor pattern is semantically fairly accurate although there are numerous typos and diagramming errors. However, the accompanying code is not a Visitor pattern. Since the key benefit of this book over other design pattern books is that the code is in Java, the usefulness of the book is lost.The net effect is that the beginner will not learn design patterns correctly.Given that a major benefit of design patterns is the common understanding of certain coding idioms, this is a very damaging book. It is like learning to play the piano the wrong way - once the damage is done to the technique it can take years of hard work to repair. You are much better off learning to play properly from the beginning. In the context of design patterns, this means reading the GoF, Siemens, and Doug Lea books.

I was really excited for a Java-slanted version of the famous Design Patterns book. This should have been an easy home run but Grand let us down on the details. I went into this book knowing some about Patterns and was eager to learn more. However, after wasting my time hacking my way through incorrect diagrams and inconsistencies between code and text I am about ready to through the book out, learn C and read Gamma's book. I don't know who edited this book but they obviously didn't know much about UML or Code.You want detail examples: Chapter Eight (Chain of

Responsibility GoF95). A pretty simple pattern made difficult because of the incorrect UML diagram in the context (association arrows going the wrong direction), and the incorrect text conflicting with the code. I figured the pattern out by comparing it to Gamma's example, it is really quite simple.The there are many more examples, especially frustrating on some of the more abstract patterns. Grand should have hired a better editor.If you buy this book and know little about patterns I suggest you also get Gamma's book and refer to it often.

Like others, I eagerly anticipated an encyclopedia of design patterns implemented in Java. Alas...I've hosted a bimonthly design pattern discussion group for just over a year. We recently began studying patterns from Grand's book which we had not yet covered. I acknowledge that I hold my opinions more strongly and am more vocal than most, and I would never try to speak on behalf of others. Even so, I think it would be fair to say that our study circle's concensus is "Patterns in Java, Vol 1" is a poor book.I, of course, have a much stronger opinion: I think "Patterns in Java" is actually detrimental to both the design pattern and Java communities. The writing is terrifically bad. The errors are countless. The intepetation, context, and presentation of the patterns are misguided at best. The example implementations are remarkably sophomoric. As one wag put it, "This book is an excellent source of anti-patterns." An item by item critique of "Patterns in Java" would be roughly twice as long as the book itself; certainly not a task I would relish.I've been advised to take the good with the bad. Sure, no book is perfect. For instance, our circle was less than satisfied with the presentation of both Flyweight and Intepreter in "Design Patterns" by Gamma et al. But I think we all agree that "Design Patterns" is an excellent book overall. I also accept that patterns appearing in the PLoP books are sometimes formative. Schucks, I even value most pattern articles and features in the various magazines.Having said all that, I have found NOTHING whatsoever of value in "Patterns in Java". I strongly recommend that you NOT buy this book. All of the patterns presented in "Patterns in Java" are covered better elsewhere. And if you are new to design patterns, "Patterns in Java" will only serve to misguide you.

Patterns in Java: A Catalog of Reusable Design Patterns Illustrated with UML, 2nd Edition, Volume 1 Patterns in Java, Volume 1, A Catalog of Reusable Design Patterns Illustrated with UML Java: The Ultimate Guide to Learn Java and Python Programming (Programming, Java, Database, Java for dummies, coding books, java programming) (HTML, ... Developers, Coding, CSS, PHP) (Volume 3) JAVA: JAVA in 8 Hours, For Beginners, Learn Java Fast! A Smart Way to Learn Java, Plain & Simple, Learn JAVA Programming Language in Easy Steps, A Beginner's Guide, Start Coding

Today! Java: The Simple Guide to Learn Java Programming In No Time (Programming,Database, Java for dummies, coding books, java programming) (HTML,Javascript,Programming,Developers,Coding,CSS,PHP) (Volume 2) Object-Oriented Software Engineering: Using UML, Patterns and Java (2nd Edition) Framework Design Guidelines: Conventions, Idioms, and Patterns for Reusable .NET Libraries (2nd Edition) Object-Oriented Software Engineering Using UML, Patterns, and Java (3rd Edition) [Economy Edition] Standard Catalog of Smith & Wesson (Standard Catalog of Smith and Wesson) Standard Catalog of World Coins, 1601-1700 (Standard Catalog of World Coins 17th Centuryedition 1601-1700) Standard Catalog of United States Paper (Standard Catalog of United States Paper Money, 14th ed. ed By Robert F. Lemke) Yu-Gi-Oh! Official Card Catalog (Prima Official Card Catalog) Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and the Unified Process (2nd Edition) Real Time UML: Advances in the UML for Real-Time Systems (3rd Edition) UML 2.0 in Action: A project-based tutorial: A detailed and practical walk-through showing how to apply UML to real world development projects Java Programming for Kids: Learn Java Step By Step and Build Your Own Interactive Calculator for Fun! (Java for Beginners) Framework Design Guidelines: Conventions, Idioms, and Patterns for Reusable .NET Libraries Design Patterns CD: Elements of Reusable Object-Oriented Software (Professional Computing) Design Patterns: Elements of Reusable Object-Oriented Software Design Patterns: Elements of Reusable Object-Oriented Software (Adobe Reader)