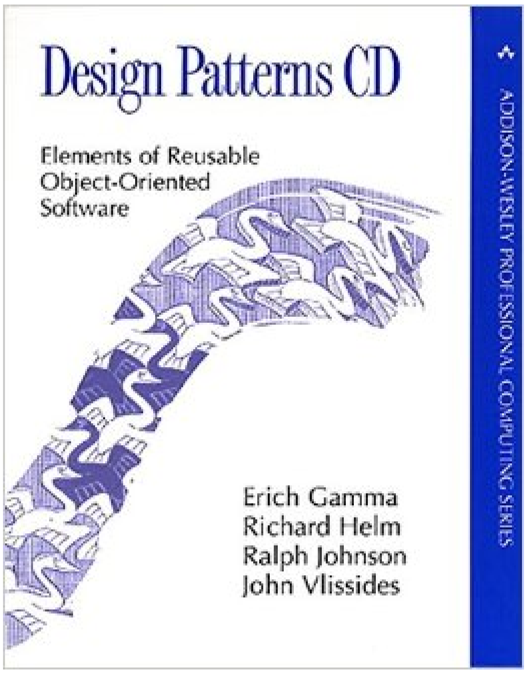
Design Patterns CD: Elements Of Reusable Object-Oriented Software (Professional Computing)


Design Patterns CD
Elements of Reusable Object-Oriented Software
Erich Gamma
Richard Helm
Ralph Johnson
John Vlissides
ADDISON-WESLEY PROFESSIONAL COMPUTING SERIES

Synopsis

Published in 1995, Design Patterns: Elements of Reusable Object-Oriented Software has elicited a great deal of praise from the press and readers. The 23 patterns contained in the book have become an essential resource for anyone developing reusable software designs. In response to a great number of requests from readers of the book and from the object-oriented community as a whole, these designs patterns, along with the entire text of the book, are being made available on CD. This electronic version will enable students to install the patterns directly onto a computer and create an architecture for using and building reusable components. Produced in HTML format, the CD is heavily cross-referenced with numerous links to the online text.

Book Information

CD-ROM: 2 pages

Publisher: Addison-Wesley Professional; 1 edition (May 31, 1998)

Language: English

ISBN-10: 0201634988

ISBN-13: 978-0201634983

Product Dimensions: 7.6 x 0.8 x 9.3 inches

Shipping Weight: 0.3 ounces (View shipping rates and policies)

Average Customer Review: 4.5 out of 5 starsÂ Â See all reviewsÂ (455 customer reviews)

Best Sellers Rank: #1,203,061 in Books (See Top 100 in Books) #11 inÂ Books > Computers & Technology > Programming > Software Design, Testing & Engineering > Software Reuse #416 inÂ Books > Textbooks > Computer Science > Object-Oriented Software Design #1438 inÂ Books > Computers & Technology > Programming > Software Design, Testing & Engineering > Object-Oriented Design

Customer Reviews

This book really changed my way of thinking about object-oriented design. The idea is that when designing a new class hierarchy, though implementation details may differ, you often find yourself using the same kinds of solutions over and over again. Rather than approaching each design task out of context as an individual, isolated problem, the strategy is to study the task and identify the underlying design pattern most likely to be applicable, and follow the class structure outlined by that pattern. It’s a "cookbook" school of design that works amazingly well.There are other advantages to this book. It isolates 23 of the most common patterns and presents them in detail. You wouldn’t think that 23 patterns would be enough, but once you become adept at recognizing patterns, you’ll find that a large fraction of the patterns you use in practice are among these 23. For each pattern, the book carefully presents the intent of the pattern, a motivating example, consequences of using that pattern, implementation considerations and pitfalls, sample code (C++ or Smalltalk), known uses of that pattern in real-world applications, and a list of related patterns.Upon first reading, you will start to recognize these patterns in the frameworks you see. Upon second reading, you'll begin to see how these patterns can help you in your own designs, and may also start to see new patterns not listed in the book. Once you become familiar with the pattern concept, you will be able to originate your own patterns, which will serve you well in the future. One of the most valuable contributions of this book is that it is designed not merely to help you identify patterns, but to give you a sense of which patterns are appropriate in which contexts.

... well, it's over. "Patterns" have not revolutionized the world. Nor does this book need to be "studied" for deep insights.What it seems patterns are actually good for is giving common names to popular solutions to problems, to make them easier to call to mind, and easier to discuss with others. Even this much is overrated. Before the advent of patterns, you could have said "callbacks" and people would have understood. Now you say "the Observer pattern"._Design Patterns_ is none the less valuable, because it is one of those few books that EVERYONE is expected to have read. This is helpful in practice, as you can expect everyone to be familiar with its vocabulary. Few books truly fall into this "required reading" category. The only other that comes to mind is the MIT algorithms text. Many tech pundits claim that every next book is "required reading", and the claim becomes tiring after a while, but this is one of the few that really is.I would not necessarily purchase it, though. The "pattern" schematic is verbose, and requires pages upon pages to describe something that, once you have seen it in practice once or twice, you will recognize immediately. Omitting the appendixes, the book is barely 350 pages, and presents only 23 patterns. Only a handful of the patterns are truly famous: Singleton, Observer, Template Method ... perhaps a few more. A number of them are poorly presented. Chain of Responsibility, for instance, is just one of many ways to define an event framework and does not belong in a book that doesn't present the alternatives. Mediator is another; there must be dozens of ways to create a Mediator, which most people would call an "event registry" or something else, rather than a Mediator.

Download to continue reading...

Design Patterns CD: Elements of Reusable Object-Oriented Software (Professional Computing) Design Patterns: Elements of Reusable Object-Oriented Software Fundamentals of Object-Oriented Design in UML Object Oriented Programming with Swift 2 Wireless Computing in Medicine: From