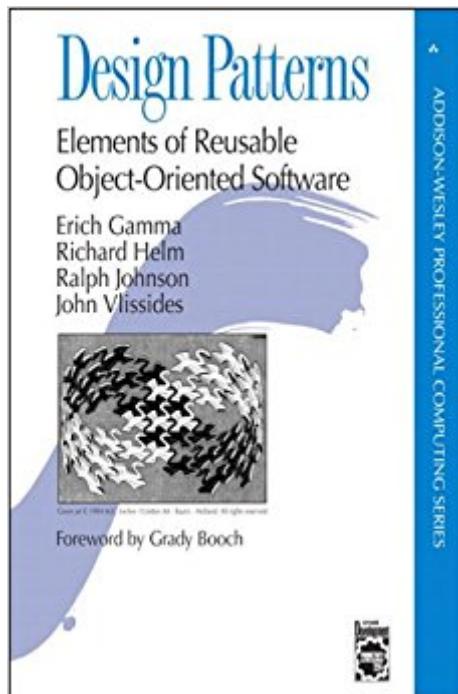


The book was found

# Design Patterns: Elements Of Reusable Object-Oriented Software (Adobe Reader)



## Synopsis

Capturing a wealth of experience about the design of object-oriented software, four top-notch designers present a catalog of simple and succinct solutions to commonly occurring design problems. Previously undocumented, these 23 patterns allow designers to create more flexible, elegant, and ultimately reusable designs without having to rediscover the design solutions themselves. The authors begin by describing what patterns are and how they can help you design object-oriented software. They then go on to systematically name, explain, evaluate, and catalog recurring designs in object-oriented systems. With *Design Patterns* as your guide, you will learn how these important patterns fit into the software development process, and how you can leverage them to solve your own design problems most efficiently. Each pattern describes the circumstances in which it is applicable, when it can be applied in view of other design constraints, and the consequences and trade-offs of using the pattern within a larger design. All patterns are compiled from real systems and are based on real-world examples. Each pattern also includes code that demonstrates how it may be implemented in object-oriented programming languages like C++ or Smalltalk.

## Book Information

File Size: 19069 KB

Print Length: 395 pages

Simultaneous Device Usage: Up to 5 simultaneous devices, per publisher limits

Publisher: Addison-Wesley Professional; 1 edition (October 31, 1994)

Publication Date: October 31, 1994

Sold by: Digital Services LLC

Language: English

ASIN: B000SEIBB8

Text-to-Speech: Enabled

X-Ray: Enabled

Word Wise: Not Enabled

Lending: Not Enabled

Enhanced Typesetting: Enabled

Best Sellers Rank: #60,765 Paid in Kindle Store (See Top 100 Paid in Kindle Store) #2 in Books > Computers & Technology > Programming > Software Design, Testing & Engineering > Software Reuse #10 in Books > Computers & Technology > Computer Science > AI & Machine Learning >

## 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: Elements of Reusable Object-Oriented Software (Adobe Reader) Reusable Software : The Base Object-Oriented Component Libraries (Prentice Hall Object-Oriented Series) Design Patterns CD: Elements of Reusable Object-Oriented Software (Professional Computing) Design Patterns: Elements of Reusable Object-Oriented Software Object Success : A Manager's Guide to Object-Oriented Technology And Its Impact On the Corporation (Object-Oriented Series) Object-oriented software development: Engineering software for reuse Object-Oriented Software Engineering: Practical Software Development Using UML and Java Object-Oriented Software Engineering Using UML, Patterns, and Java (3rd Edition) [Economy Edition] Object-Oriented Software Engineering: Using UML, Patterns and Java (2nd Edition) Visual Object-Oriented Programming Using Delphi With CD-ROM (SIGS: Advances in Object Technology) Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development (3rd Edition) Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and the Unified Process (2nd Edition) Analysis Patterns: Reusable Object Models Analysis Patterns: Reusable Object Models (paperback) 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 Learn Adobe Illustrator CC for Graphic Design and Illustration: Adobe Certified Associate Exam Preparation (Adobe Certified Associate (ACA)) Growing Object-Oriented Software, Guided by Tests Object-Oriented and Classical Software Engineering Object-Oriented Reengineering Patterns

[Dmca](#)