



The book was found

Design Patterns: Elements Of Reusable Object-Oriented Software



Synopsis

These texts cover the design of object-oriented software and examine how to investigate requirements, create solutions and then translate designs into code, showing developers how to make practical use of the most significant recent developments. A summary of UML notation is included.

Book Information

Hardcover: 395 pages

Publisher: Addison-Wesley Professional; 1 edition (November 10, 1994)

Language: English

ISBN-10: 0201633612

ISBN-13: 978-0201633610

Product Dimensions: 7.4 x 1 x 9.4 inches

Shipping Weight: 1.9 pounds (View shipping rates and policies)

Average Customer Review: 4.5 out of 5 stars 481 customer reviews

Best Sellers Rank: #4,695 in Books (See Top 100 in Books) #1 in Books > Computers & Technology > Programming > Software Design, Testing & Engineering > Software Reuse #2 in Books > Computers & Technology > Programming > Software Design, Testing & Engineering > Object-Oriented Design #2 in Books > Textbooks > Computer Science > Object-Oriented Software Design

Customer Reviews

Design Patterns is a modern classic in the literature of object-oriented development, offering timeless and elegant solutions to common problems in software design. It describes patterns for managing object creation, composing objects into larger structures, and coordinating control flow between objects. The book provides numerous examples where using composition rather than inheritance can improve the reusability and flexibility of code. Note, though, that it's not a tutorial but a catalog that you can use to find an object-oriented design pattern that's appropriate for the needs of your particular application--a selection for virtuoso programmers who appreciate (or require) consistent, well-engineered object-oriented designs.

This book isn't an introduction to object-oriented technology or design. Many books already do a good job of that...this isn't an advanced treatise either. It's a book of design patterns that describe simple and elegant solutions to specific problems in object-oriented software design....Once you

understand the design patterns and have had an "Aha!" (and not just a "Huh?" experience with them, you won't ever think about object-oriented design in the same way. You'll have insights that can make your own designs more flexible, modular, reusable, and understandable--which is why you're interested in object-oriented technology in the first place, right? -- From the Preface This is one of the best written and wonderfully insightful books that I have read in a great long while...this book establishes the legitimacy of patterns in the best way: not by argument, but by example. -- C++ Report

This book is a must-have, both for reference purposes and for developing technical skill in any language. It could be used as a teacher's aid, for lesson ideas, or for any software project at all. Extremely useful - literally chock full of great ideas and well-tested, proven patterns that work. A classic that any developer should consider a standard go-to reference. The uses for the ideas and patterns in this book are virtually unlimited. Clearly great design, made easily available and usable.

As someone who is new to CS (I am an electrical engineer by education and trade), it is helpful to have a reference of various design patterns. I have found myself using it as a reference when I start new projects. It is organized well, with similar patterns grouped into sections.

This book was recommended by a project manager of mine. So long after college I bought a textbook and ended up learning a great deal.

Great book! Must have for anyone that likes/does coding of anykind

I teach OOP and OOD for several semesters now. I knew, ofcourse, about design patterns and the GoF, but somehow never had the chance to read this book. Finally I've decided I have to read it, so I bought it and it blew my mind. Clear, sophisticated and yet so important. It was such a great pleasure to read it, to think about every pattern, to try to understand deeply the solutions - one of the best books I've ever read. My advice for whom it may concern - don't hurry. Lots of the patterns seem identical, and in a brief look you might not get the differences between them. Don't give up - try to understand the solution, try to invent a design problem of your own, it's worthy!

This is a great book. Even if you aren't working in Object Oriented languages, it can be helpful to read through the concepts behind the patterns. If you really learn this book well, your own designs

will be more flexible and extensible even when you aren't using a specific pattern.

What you've heard is true: this is the most important book on programming of the last __ years (at least 10, for me maybe of all time). I wanted to just put a few thoughts in that I didn't see in the other reviews:1. I read an article one time where John Vlissides (one of the authors) was saying he spoke somewhere and asked how many people had read the book and almost everyone raised their hands, then he asked, who would like to come up and explain how to implement the Composite pattern and suddenly only a couple hands were raised. Though this book is a fount of great ideas, it really will be most useful to you if you become CONVERSATIONAL with EACH of the patterns. That's the whole idea from Alexander anyway so consider it a mandate from on high.2. One easy way I've used to explain to people what patterns are about is that mere object oriented training leaves people with an idea of how to model things as objects, but so many times I've done reviews of programmers code and they got that far and then as soon as work needed to be done that required more than one class, one of two things happens: they start passing data all over the place (back to the structured world we go), or they start binding their objects into deadly embraces. This book teaches you how to have some 'tricks' in your bag for modeling just such situations. Now, that said, there is some work you will have to do to map it into the newer programming world we're living in, for instance distributed Java makes some services available (like EJBs/container services/messaging) that changes some of the implementation ideas considerably.3. If you are using Java, you can start to learn patterns and their application to your chosen platform by looking at a vast wealth of work that's already been done. The JDK uses the Observer pattern for its event model. Some of the more advanced frameworks, like BEA's Theory Center, are loaded w/Design Patterns (Chain of Responsibility and Strategy). And many products (Together/J) use patterns in their APIs (Visitor). Finally, whenever I'm interviewing programmers now I ask them if they know what patterns are, then if they get past that, if they can give me an example of a recent use of a pattern and how it worked. I had a guy a couple of weeks ago looking for 6 figures who told me he'd heard of the book but hadn't gotten around to opening it. Der, that's like a doctor saying 'I've heard milk might not be good for ulcers, but take it because I haven't had time to review the research yet.' More writers need to put out CD editions that are this good, dang it!!

every software engineer should read this.

[Download to continue reading...](#)

Design Patterns: Elements of Reusable Object-Oriented Software Object-Oriented and Classical

Software Engineering Object-Oriented Software Engineering: An Agile Unified Methodology (Irwin Computer Science) Object Oriented Software Development Using Java (2nd Edition) Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development (3rd Edition) Object-Oriented Design and Patterns Software Engineering: The Current Practice (Chapman & Hall/CRC Innovations in Software Engineering and Software Development Series) Systems Analysis and Design: An Object-Oriented Approach with UML An Object-Oriented Approach to Programming Logic and Design Practical Object-Oriented Design in Ruby: An Agile Primer (Addison-Wesley Professional Ruby) Object-Oriented Modeling and Design with UML (2nd Edition) Head First Object-Oriented Analysis and Design Tools For Structured and Object-Oriented Design (7th Edition) Object-Oriented Analysis and Design with Applications (3rd Edition) Object Oriented Systems Analysis and Design Object-Oriented Analysis and Design with the Unified Process (Available Titles CengageNOW) Object Lessons for a Year: 52 Talks for the Children's Sermon Time (Object Lesson Series) The Object-Oriented Thought Process (4th Edition) (Developer's Library) Introduction to Programming with Greenfoot: Object-Oriented Programming in Java with Games and Simulations (2nd Edition) C++ and Object-Oriented Numeric Computing for Scientists and Engineers

[Contact Us](#)

[DMCA](#)

[Privacy](#)

[FAQ & Help](#)