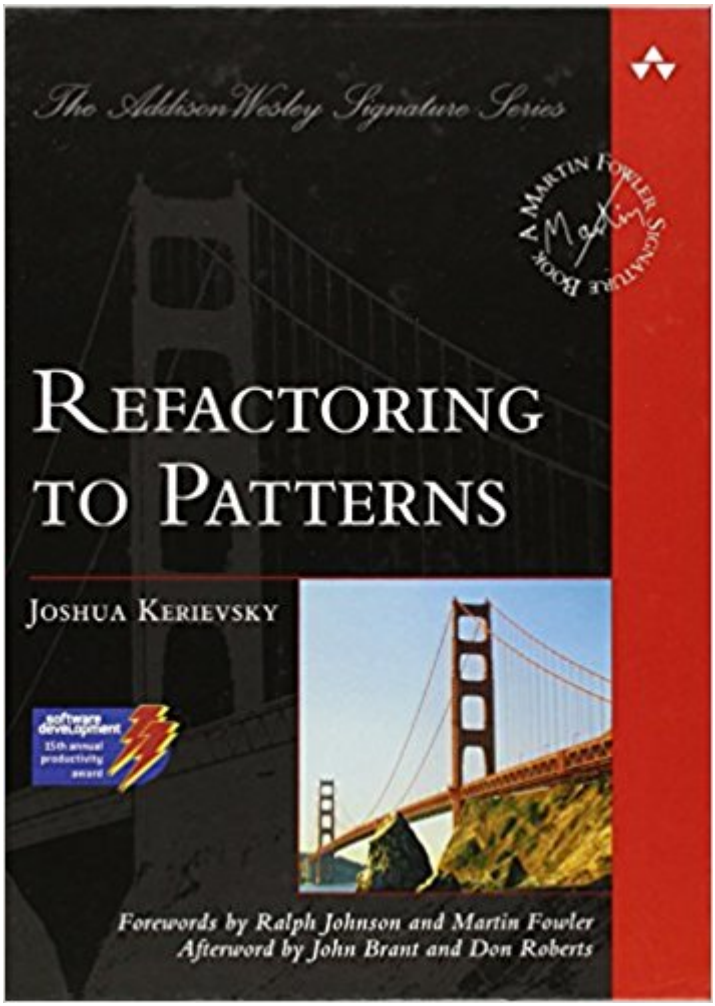
Ebook Directory
the best source of ebook

The book was found

Refactoring To Patterns



PDF
Adobe
DOWNLOAD EBOOK

Synopsis

In 1994, DESIGN PATTERNS changed the landscape of object-oriented development by introducing classic Solutions to recurring design problems. In 1999, REFACTORING revolutionized design by introducing an effective process for improving code. With the highly-anticipated REFACTORING TO PATTERNS, Joshua Kerievsky has changed our approach to design by forever uniting patterns with the evolutionary process of refactoring. This book introduces the theory and practice of pattern-directed refactorings: sequences of low-level refactorings that allow designers to safely move designs to, towards, or away from pattern implementations. Using code from real-world projects, Kerievsky documents the thinking and steps underlying over two dozed pattern-based design transformations. Along the way he offers insights into pattern differences and how to implement patterns in the simplest possible ways.

Book Information

Hardcover: 400 pages

Publisher: Addison-Wesley Professional; 1 edition (August 15, 2004)

Language: English

ISBN-10: 0321213351

ISBN-13: 978-0321213358

Product Dimensions: 7 x 1 x 9.3 inches

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

Average Customer Review: 4.1 out of 5 stars 60 customer reviews

Best Sellers Rank: #348,331 in Books (See Top 100 in Books) #18 inÃ Â Books > Crafts, Hobbies & Home > Antiques & Collectibles > Antiques Care & Reference > Care & Restoration #136 inÃ Â Books > Textbooks > Computer Science > Object-Oriented Software Design #481 inÃ Â Books > Computers & Technology > Programming > Software Design, Testing & Engineering > Object-Oriented Design

Customer Reviews

In 1994, Design Patterns changed the landscape of object-oriented development by introducing classic solutions to recurring design problems. In 1999, Refactoring revolutionized design by introducing an effective process for improving code. With the highly anticipated Refactoring to Patterns, Joshua Kerievsky has changed our approach to design by forever uniting patterns with the evolutionary process of refactoring. This book introduces the theory and practice of pattern-directed refactorings: sequences of low-level refactorings that allow designers to safely move designs to,

towards, or away from pattern implementations. Using code from real-world projects, Kerievsky documents the thinking and steps underlying over two dozen pattern-based design transformations. Along the way he offers insights into pattern differences and how to implement patterns in the simplest possible ways. Coverage includes: A catalog of twenty-seven pattern-directed refactorings, featuring real-world code examples Descriptions of twelve design smells that indicate the need for this book’s refactorings General information and new insights about patterns and refactoring Detailed implementation mechanics: how low-level refactorings are combined to implement high-level patterns Multiple ways to implement the same pattern—and when to use each Practical ways to get started even if you have little experience with patterns or refactoring Refactoring to Patterns reflects three years of refinement and the insights of more than sixty software engineering thought leaders in the global patterns, refactoring, and agile development communities. Whether you’re focused on legacy or "greenfield" development, this book will make you a better software designer by helping you learn how to make important design changes safely and effectively.

In 1994, "Design Patterns" changed the landscape of object-oriented development by introducing classic solutions to recurring design problems. In 1999, "Refactoring" revolutionized design by introducing an effective process for improving code. With the highly anticipated "Refactoring to Patterns," Joshua Kerievsky has changed our approach to design by forever uniting patterns with the evolutionary process of refactoring.This book introduces the theory and practice of pattern-directed refactorings: sequences of low-level refactorings that allow designers to safely move designs to, towards, or away from pattern implementations. Using code from real-world projects, Kerievsky documents the thinking and steps underlying over two dozen pattern-based design transformations. Along the way he offers insights into pattern differences and how to implement patterns in the simplest possible ways.Coverage includes: A catalog of twenty-seven pattern-directed refactorings, featuring real-world code examplesDescriptions of twelve design smells that indicate the need for this book’s refactoringsGeneral information and new insights about patterns and refactoringDetailed implementation mechanics: how low-level refactorings are combined to implement high-level patternsMultiple ways to implement the same pattern--and when to use eachPractical ways to get started even if you have little experience with patterns or refactoring"Refactoring to Patterns" reflects three years of refinement and the insights of more than sixty software engineering thought leaders in the global patterns, refactoring, and agile development communities. Whether you’re focused on legacy or "greenfield" development, this book will make

you a better software designer by helping you learn how to make important design changes safely and effectively.

The emergence of Design Patterns has certainly been one of the most important things that has ever happened to object-oriented programming. We are more likely to end up with much cleaner and more testable systems when using design patterns as they are based upon tried and tested formulas for solving common problems. In reality though, we're often faced with different challenges. We need to work on legacy systems that often contain a lot of poorly structured and untested code. In the real world we often know that patterns would help a lot to clean up that mess, but adding them to an existing system is ... hard.This book attempts to show you how to cut through undergrowth of various code smells and how to patch the systems with appropriate patterns. The book contains a lot of bite-sized chapters, so it is okay to grab it for 15 minutes, read a bit and get back to it some time later. Each refactoring contains a theoretical part (explaining the symptoms of said smell and a suggested refactoring), a "mechanics" part (how to perform the refactoring) and a practical example.As of 2016, I didn't find step-by-step mechanics very useful (but this book was written back in 2004 and modern IDEs we use nowadays were nowhere as smart as they are today), but reading the original (unrefactored) code and comparing it with the final one (refactored) is definitely worth it.

I have just finished this book. All i can tell is that this one clearly is the next book you should read after knowing basic pattern design.there is only one thing i wish any pattern book should have is that "at the end of each chapter, the author should include the COMPLETE refactored code so we can spend less time to understand what is going on". That's why i won't give this one 5 stars because it makes us spending more neccessary time to understand what is going on.anyway, if you have no knowledge about pattern, you will need to spend more time to understand this book. Whenever you start a new chapter, you need to go online to search for a better understanding of what the pattern is about before diving into each chapter. if not, you will get lost for sure if you just concentrate on the mechanics section. The mechanics section is just the steps helping you reach the final form of a pattern . So if you know the structure of the pattern already, it will help you understand the mechanics section a lot faster.

I heard a lot of praise regarding this book, so I decided to buy it.Overall it's a good book but I was kind of hoping for something more. The author does a lot of references to "Refactoring" by Martin

Fowler and to THE Design Patterns book (Gamma ...). In my view this book is more like a "how to understand and utilize" those previous two books and might increase the understanding for when to apply design patterns and when to avoid it.I'd say that the examples used in this book is very good and they certainly do feel like non-fabricated examples. The Author's reasoning to when and how to refactor code in the examples are well described and easy to follow.I think that if you studied Design Patterns & Refactoring and had problems understanding when or how to utilize what you learned during those books, then this book might be something for you. This might also be interesting if you wonder how to (as the title suggest) refactor towards a pattern. However, I think that where this book fails is that in a lot of situations refactoring toward patterns is to overcomplicate a code base when there are much easier solutions at hand.Finally I'd like to say that this is certainly a good book, however I wouldn't rate it as one of those "must read"-books.I'd much rather recommend the following books:Clean Code - Robert C. MartinRefactoring - Martin FowlerDesign Patterns - 4 authors : Gamma ...The Pragmatic Programmer - Andrew Hunt and David Thomas

This book is an important contribution to the design patterns library, particularly for practicing programmers as opposed to theoreticians. Most of the design books that we read deal with patterns in a vacuum, constructing toy examples which demonstrate the pattern at hand but have little in common with the kinds of software contexts that we are likely to encounter as serious programmers. Typically we are not designing systems from scratch; normally there is some combination of new and legacy code that must be made to cooperate in some coherent way, and we may perceive that some pattern would be desirable, but executing the transformation of the existing systems to express that pattern well typically involves significant risk of regression. That is a central focus of this book: how to break up this transformation into sequences of safe re-factorings.Another important theme of the book is that the application of patterns typically entails various trade-offs. A well-known pattern may better communicate the relationships and responsibilities of the various software components we are working with, but on the other hand the pattern may introduce extra complexity if the problem being solved is amenable to some simple and minimal approach. Using the approaches of this book it is possible to introduce patterns if needed later in the development cycle, and in many cases it may turn out that the patterns are in fact not needed. In these situations, the lengthy application of patterns upfront would have been a premature optimization.And this brings me to the last point that I want to make about this book: its content is strongly based on real world experience, and presented in a way which fits what we need as real world engineers. I thought it was interesting to see a negative review of this book where the complaint was that the

author was talking about his own experiences too much. The relating of the theory presented to actual past problems solved is one of the most valuable aspects of the book, in my opinion.

Download to continue reading...

Refactoring to Patterns AntiPatterns: Refactoring Software, Architectures, and Projects in Crisis Refactoring: Improving the Design of Existing Code His Forge Burns Hot for Mosaic Damascus: Knife Patterns & Techniques: Damascus pattern making & techniques. Learn how to make mosaic Damascus patterns ... techniques for making Damascus patterns. 11 Crochet Shawl Patterns: Crochet Poncho Patterns, Free Easy Crochet Patterns and More Bavarian Crochet: Learn About Bavarian Stitches and Patterns and Make 15 Cute and Easy Projects: (Crochet Patterns, Crochet for Beginners) (Crochet Books Patterns, Cute And Easy Crochet) Crochet: Crochet Books: 30 Crochet Patterns In 30 Days With The Ultimate Crochet Guide! (crochet patterns on kindle free, crochet patterns, crochet books, ... crocheting, crochet magazine Book 1) Dress Up Dolls Amigurumi Crochet Patterns: 5 big dolls with clothes, shoes, accessories, tiny bear and big carry bag patterns (Sayjai's Amigurumi Crochet Patterns) (Volume 3) Dress Up Dolls Amigurumi Crochet Patterns: 5 big dolls with clothes, shoes, accessories, tiny bear and big carry bag patterns (Sayjai's Amigurumi Crochet Patterns Book 3) Knitting For Beginners: The A-Z Guide to Have You Knitting in 3 Days (Includes 15 Knitting Patterns) (Knitting Patterns in Black&White) Detailed Designs and Beautiful Patterns (Sacred Mandala Designs and Patterns Coloring Books for Adults) (Volume 28) Drawing: Drawing and Sketching,Doodling,Shapes,Patterns,Pictures and Zen Doodle (drawing, zentangle, drawing patterns, drawing shapes, how to draw, doodle, creativity) 1,000 Tangles, Patterns & Doodled Designs: Hundreds of tangles, designs, borders, patterns and more to inspire your creativity! Patterns for Meditation Coloring Books for Adults: An Adult Coloring Book Featuring 35+ Geometric Patterns and Designs (Jenean Morrison Adult Coloring Books) Pixel Craft with Perler Beads: More Than 50 Super Cool Patterns: Patterns for Hama, Perler, Pyssla, Nabbi, and Melty Beads! Crochet Patterns for Babies (2nd Edition): 41 Adorable Patterns for Baby Hats, Blankets, & Clothes! Plant Patterns (Finding Patterns) EQUIGLAS Equine Suncatcher Patterns (EQUIGLAS Stained Glass Patterns) (Volume 1) Black Background Designs: Stress Relieving Mandalas and Patterns (Adult Coloring Patterns) (Volume 35) Crochet: 365 Days of Crochet: 365 Crochet Patterns for 365 Days (Crochet, Crochet Patterns, DIY Crochet, Crochet Books, Crochet for Beginners, Crochet Afghans, Crochet Christmas, Holiday Crochet)

Contact Us

DMCA

Privacy

FAQ & Help