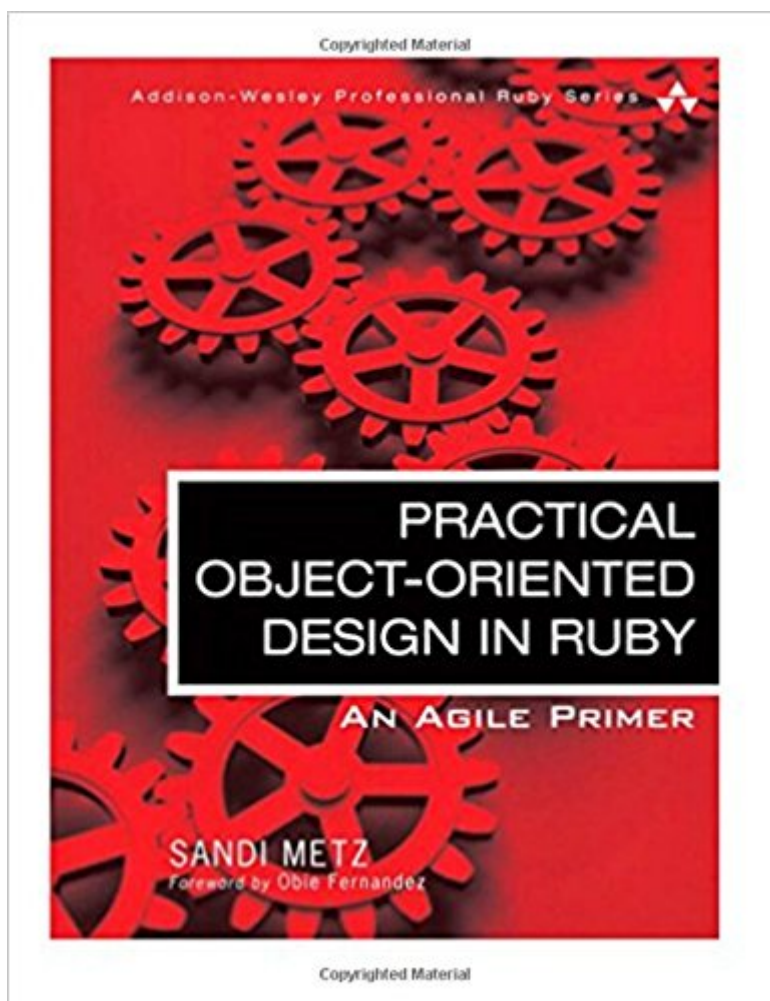The book was found

Practical Object-Oriented Design In Ruby: An Agile Primer (Addison-Wesley Professional Ruby)

Synopsis

The Complete Guide to Writing More Maintainable, Manageable, Pleasing, and Powerful Ruby Applications Rubyâ ™s widely admired ease of use has a downside: Too many Ruby and Rails applications have been created without concern for their long-term maintenance or evolution. The Web is awash in Ruby code that is now virtually impossible to change or extend. This text helps you solve that problem by using powerful real-world object-oriented design techniques, which it thoroughly explains using simple and practical Ruby examples. Sandi Metz has distilled a lifetime of conversations and presentations about object-oriented design into a set of Ruby-focused practices for crafting manageable, extensible, and pleasing code. She shows you how to build new applications that can survive success and repair existing applications that have become impossible to change. Each technique is illustrated with extended examples, all downloadable from the companion Web site, poodr.info. The first title to focus squarely on object-oriented Ruby application design, Practical Object-Oriented Design in Ruby will guide you to superior outcomes, whatever your previous Ruby experience. Novice Ruby programmers will find specific rules to live by; intermediate Ruby programmers will find valuable principles they can flexibly interpret and apply; and advanced Ruby programmers will find a common language they can use to lead development and guide their colleagues. This guide will help you Understand how object-oriented programming can help you craft Ruby code that is easier to maintain and upgrade Decide what belongs in a single Ruby class Avoid entangling objects that should be kept separate Define flexible interfaces among objects Reduce programming overhead costs with duck typing Successfully apply inheritance Build objects via composition Design cost-effective tests Solve common problems associated with poorly designed Ruby code Â

Book Information

Series: Addison-Wesley Professional Ruby

Paperback: 272 pages

Publisher: Addison-Wesley Professional; 1 edition (September 15, 2012)

Language: English

ISBN-10: 0321721330

ISBN-13: 978-0321721334

Product Dimensions: 7 x 0.6 x 9.1 inches

Shipping Weight: 1.2 pounds (View shipping rates and policies)

Average Customer Review: 4.7 out of 5 stars   173 customer reviews

Best Sellers Rank: #50,985 in Books (See Top 100 in Books)   #2 inÂ Books > Computers & Technology > Programming > Languages & Tools > Ruby   #26 inÂ Books > Textbooks > Computer Science > Object-Oriented Software Design   #46 inÂ Books > Computers & Technology > Databases & Big Data > Data Processing

Customer Reviews

Â â œThis is great stuff! Your descriptions are so vibrant and vivid that Iâ ™m rediscovering the truth buried in OO principles that are otherwise so internalized that I forget to explore them. Your thoughts on design and knowing the future are especially eloquent.â • â ”Ian McFarland, President, New Context, Inc. Â â œAs a self-taught programmer, this was an extremely helpful dive into some OOP concepts that I could definitely stand to become better acquainted with! And, Iâ ™m not alone: thereâ ™s a sign posted at work that reads, â ˆWWSMD? â “ What Would Sandi Metz Do?â ™â • â ”Jonathan Mukai, Pivotal in NYC Â â œMeticulously pragmatic and exquisitely articulate, Practical Object Oriented Design in Ruby makes otherwise elusive knowledge available to an audience which desperately needs it. The prescriptions are appropriate both as rules for novices and as guidelines for experienced professionals.â • â ”Katrina Owen, developer, Bengler Â â œI do believe this will be the most important Ruby book of 2012. Not only is the book 100% on-point, Sandi has an easy writing style with lots of great analogies that drive every point home.â • â ”Avdi Grimm, Author of Exceptional Ruby and Objects on Rails   â œWhile Ruby is an object-oriented language, little time is spent in the documentation on what OO truly means or how it should direct the way we build programs. Here Metz brings it to the fore, covering most of the key principles of OO development and design in an engaging, easy-to-understand manner. This is a must for any respectable Ruby bookshelf.â • â “Peter Cooper, editor, Ruby Weekly Â â œSo good, I couldnâ ™t put it down! This is a must-read for anyone wanting to do object-oriented programming in any language, not to mention it has completely changed the way I approach testing.â • â “Charles Max Wood, video and audio show host, TeachMeToCode.com Â â œDistilling scary OO design practices with clear-cut examples and explanations makes this a book or novices and experts alike. It is well worth the study by anyone interested in OO design being done right and â ˆlight.â ™ I thoroughly enjoyed this book.â • â “Manuel Pais, editor, InfoQ.com Â â œIf you call yourself a Ruby programmer, you should read this book. Itâ ™s jam-packed with great nuggets of practical advice and coding techniques that you can start applying immediately in your projects.â • â “Ylan Segal, San Diego Ruby User Group Â â œThis is the best OO book Iâ ™ve ever read. Itâ ™s short, sweet, but potent. It slowly moves from simple techniques to more advanced, each example improving on the

last. The ideas it presents are useful not just in Ruby but in static languages like C# too. Highly recommended!â • â "Kevin Berridge, software engineering manager, Pointe Blank Solutions, and organizer, Burning River Developers Meetup Â â œThe book is just perfect! The elegance of Ruby shines but it also works as an A to Z of object-oriented programming in general.â • â "Emil Rondahl, C# & .NET consultant Â â œThis is an exceptional Ruby book, in which Metz offers a practical look at writing maintainable, clean, idiomatic code in Ruby. Absolutely fantastic, recommended for my Ruby hacker friends.â • â "Zachary â œZeeâ • Spencer, freelancer & coach Â â œThis is the best programming book Iâ ™ve read in ages. Sandi talks about basic principles, but these are things weâ ™re probably still doing wrong and she shows us why and how. The book has the perfect mix of code, diagrams, and words. I canâ ™t recommend it enough and if youâ ™re serious about being a better programmer, youâ ™ll read it and agree. â "Derick Hitchcock, senior developer, SciMed Solutions Â â œI predict this will become a classic. I have an uncomfortable familiarity with programming literature, and this book is on a completely different level. I am astonished when I find a book that offers new insights and ideas, and even more surprised when it can do so, not just once, but throughout the pages. This book is excellently written, well-organized, with lucid explanations of technical programming concepts.â • â "Han S. Kang, software engineer and member of the LA Rubyists Â â œYou should read this book if you write software for a living. The future developers who inherit your code will thank you.â • â "Jose Fernandez, senior software engineer at New Relic Â â œMetzâ ™s take on the subject is rooted strongly in theory, but the explanation always stays grounded in real world concerns, which helped me to internalize it. The book is clear and concise, yet achieves a tone that is more friendly than terse.â • â "Alex Strasheim, network administrator, Ensemble Travel Group Â â œThis is an amazing book about just how to do object-oriented thinking when youâ ™re programming in Ruby. Although there are some chapters that are more Ruby-specific, this book could be a great resource for developers in any language. All in all, I canâ ™t recommend this book enough.â • â "James Hwang, thriceprime.com Â â œWhether youâ ™re just getting started in your software development career, or youâ ™ve been coding for years (like I have), itâ ™s likely that youâ ™ll learn a lot from Ms. Metzâ ™s book. She does a fantastic job of explaining the whys of well-designed software along with the hows.â • â "Gabe Hollombe, software craftsman, avantbard.com Â â œIn short, this is in my top five programming books Iâ ™ve ever read. I believe that in twenty years this will be considered one of the definitive works on object-oriented programming. I plan to re-read it at least once a year to keep my skills from falling into atrophy. If youâ ™re a relatively new, intermediate, or even somewhat advanced OO developer in any language, purchasing this book is the best way I know to level up your OO

design skills.â • â "Brandon Hays, freelance software developer

Sandi Metz has thirty years of experience working on projects that survived to grow and change. She now writes code every day as a software architect at Duke University, where her team solves real problems for customers who have large object-oriented applications that have been evolving for more than fifteen years. She has spoken at Ruby Nation and speaks regularly at the Gotham Ruby Users Conference. Â

This is a great, well rounded book on object oriented principles using the Ruby language. It is now one of my favorite programming books that I will recommend to any that wish to master their craft of software development.

I have programmed in Ruby for a number of years now and still found this book of value. I would not recommend it for absolute beginners because it assumes knowledge of the language, though you will learn general principles. I would recommend it for anyone wanting to get a better grasp of writing good object oriented code, especially in Ruby. I found it helpful to think of Messages sent instead of just the class hierarchy.

I hate bronchitis! Reading "Practical Object-Oriented Design in Ruby" was a blast. Reading it while ill was frustrating! Lots of thought provoking ideas written in an easy to read style. I kept falling asleep due to sickness and my brain kept yelling "This is cool! Stay awake!" Still, Sandi achieved her goal even through my fog of exhaustion; I now see objects very differently and will happily refactor my code to make it better.The only real ding to the book is Chapter 8's references to Enumerable and Forwardable. Everything else in the book was either easy to read or well explained. These two characters seem to have slipped in without fanfare or explanation. I don't understand them and thus probably missed some of the book's goodness. Perhaps I'll get it later. I still rate this book a 5 star buy!

This is a wonderful book on object-oriented design. The topics that Sandy covers are the keys to success for any practicing object-oriented developer regardless of the language that you write in (I'm a javascript single page web app developer): Designing classes with single responsibility, managing dependencies, creating flexible interfaces, testing, inheritance and composition, delegation you name it, explained with passion and experience. It not only nails everything a great,

practical software book should nail - clear, concise explanations, simple yet practical examples that illustrate the concepts being explained, smooth, logical flow from chapter to chapter, true to its title and goals - but, while doing so, it is filled with gems of object-oriented insight/principles/patterns/guidance. Here are a few of the excerpts that I noted while reading:"This quality of easy changeability reveals the craft of programming." (p15)"Most of the dependencies are unnecessary; they are a side-effect of the coding style." (p37)"..depend on things that change less than you do." (p53)"Your goal is to write code that works today, that can easily be reused, and that can be adapted for unexpected use in the future." (p76)"It is your interfaces more than all of your tests and any of your code, that define your application and determine its future." (p76)And of course my favorite:"Perfection is elusive, perhaps even unreachable; this should not impede your desire to achieve it. Persist. Practice. Experiment. Imagine. Do your best work and all else will follow." (p241)Thank you Sandy for my one and only Ruby book!

I found this to be a great reminder of how to approach finding the right abstraction. A rare balance of theory and practical application of oop concepts. I found the techniques for reuse and design to be refreshers as I've been exposed to design patterns, Gang of Four, clean code, etc. The emphasis on message passing followed by concrete examples using modules and roles to demonstrate coding to an interface were very well thought through. It was easy for me to absorb the authors intent.I found the uml example surprising and useful. I normally wouldn't use that as a cheap diagram to view the seams between objects. I'm going to experiment with that over pen/paper or a whiteboard.Overall I think this is a much needed piece of literature in the ruby community at all levels. For those newer to oo design some benefits will still escape you. That said, I'm not sure the foundation could be laid out better to get going down the right path with the least friction.

I've read a great many books on object-oriented programming over the years. I have my favorites, but "Practical Object-Oriented Design in Ruby" (POODR) by Sandi Mentz has just shot to the top of the list. I can't say enough good things about this fantastic contribution to the code craft literature. Sandi has a wonderful writing style and a true gift for explanation. Her examples are imaginative and brilliant vehicles for illustrating each concept. At just 241 pages (not counting the extensive index) this book is easily readable in just a few afternoons, yet it contains surprisingly complete coverage of both classic and contemporary software design principles and practices without ever even coming close to feeling academic or stuffy.Don't be mislead by the title. This is not a book about Ruby. This is a book about object oriented design. Its example code happens to be written in

Ruby - a language I've never programmed in but found very easy to comprehend nonetheless. The author demonstrates some practices that take advantage of Ruby's dynamically-typed nature. This is particularly valuable to JavaScript developers who will be able to apply these same techniques in their own applications.I started reading this book through my Safari Bookshelf subscription but found the content and delivery so compelling that I ended up buying the Kindle edition so that I could add notes and highlights.When asked to recommend a single book to enhance the skills of an object-oriented programmer new or old, I will be recommending POODR without hesitation.

Download to continue reading...

Practical Object-Oriented Design in Ruby: An Agile Primer (Addison-Wesley Professional Ruby) Agile Project Management: QuickStart Guide - The Simplified Beginners Guide To Agile Project Management (Agile Project Management, Agile Software Development, Agile Development, Scrum) Agile Project Management: Agile Revolution, Beyond Software Limits: A Practical Guide to Implementing Agile Outside Software Development (Agile Business Leadership, Book 4) Coaching Agile Teams: A Companion for ScrumMasters, Agile Coaches, and Project Managers in Transition (Addison-Wesley Signature Series (Cohn)) Essential Scrum: A Practical Guide to the Most Popular Agile Process (Addison-Wesley Signature Series (Cohn)) Agile Product Management with Scrum: Creating Products that Customers Love (Addison-Wesley Signature Series (Cohn)) Agile : Agile Project Management, A QuickStart Beginners 's Guide To Mastering Agile Project Management ! Agile Product Management: Product Owner: 27 Tips To Manage Your Product And Work With Scrum Teams (scrum, scrum master, agile development, agile software development) Managing Software Requirements (paperback): A Use Case Approach (Addison-Wesley Object Technology (Paperback)) Object-Oriented Software Engineering: An Agile Unified Methodology (Irwin Computer Science) The Object Primer: Agile Model-Driven Development with UML 2.0 The Go Programming Language (Addison-Wesley Professional Computing Series) Using Econometrics: A Practical Guide (6th Edition) (Addison-Wesley Series in Economics) Metaprogramming Ruby 2: Program Like the Ruby Pros (Facets of Ruby) Agile Testing: A Practical Guide for Testers and Agile Teams Systems Analysis and Design: An Object-Oriented Approach with UML Design Patterns: Elements of Reusable Object-Oriented Software Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development (3rd Edition) An Object-Oriented Approach to Programming Logic and Design Object-Oriented Modeling and Design with UML (2nd Edition)

Privacy

FAQ & Help