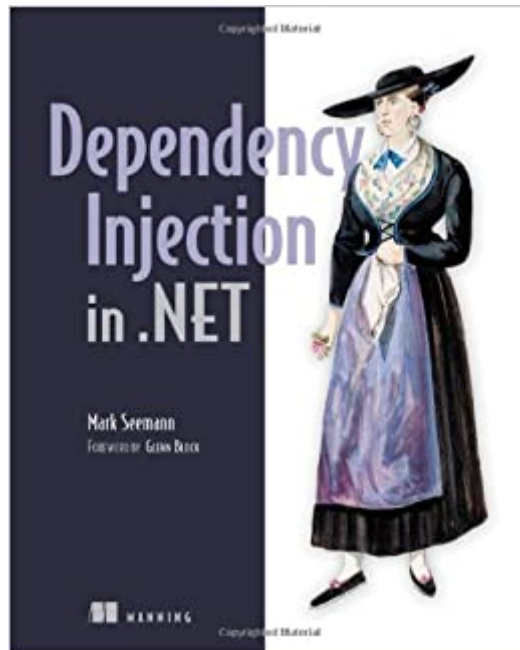The book was found

Dependency Injection In .NET

Synopsis

SummaryDependency Injection in .NET, winner of the 2013 Jolt Awards for Productivity, presents core DI patterns in plain C#, so you'll fully understand how DI works, covers integration with standard Microsoft technologies like ASP.NET MVC, and teaches you to use DI frameworks like Structure Map, Castle Windsor, and Unity.About the TechnologyDependency Injection is a great way to reduce tight coupling between software components. Instead of hard-coding dependencies, such as specifying a database driver, you inject a list of services that a component may need. The services are then connected by a third party. This technique enables you to better manage future changes and other complexity in your software.About this BookDependency Injection in .NET introduces DI and provides a practical guide for applying it in .NET applications. The book presents the core patterns in plain C#, so you'll fully understand how DI works. Then you'll learn to integrate DI with standard Microsoft technologies like ASP.NET MVC, and to use DI frameworks like StructureMap, Castle Windsor, and Unity. By the end of the book, you'll be comfortable applying this powerful technique in your everyday .NET development.This book is written for C# developers. No previous experience with DI or DI frameworks is required. Purchase of the print book comes with an offer of a free PDF, ePub, and Kindle eBook from Manning. Also available is all code from the book. Winner of 2013 Jolt Awards: The Best BooksÃ¢â ¬â •one of five notable books every serious programmer should read.What's InsideMany C#-based examplesA catalog of DI patterns and anti-patternsUsing both Microsoft and open source DI frameworksTabel of ContentsPART 1 PUTTING DEPENDENCY INJECTION ON THE MAPA Dependency Injection tasting menuA comprehensive exampleDI ContainersPART 2 DI CATALOGDI patternsDI anti-patternsDI refactoringsPART 3 DIY DIObject CompositionObject LifetimeInterceptionPART 4 DI CONTAINERSCastle WindsorStructureMapSpring.NETAutofacUnityMEF

Book Information

Paperback: 584 pages

Publisher: Manning Publications; 1 edition (September 28, 2011)

Language: English

ISBN-10: 1935182501

ISBN-13: 978-1935182504

Product Dimensions: 7.4 x 1.3 x 9.2 inches

Shipping Weight: 2.3 pounds (View shipping rates and policies)

Average Customer Review: 4.4 out of 5 stars   75 customer reviews

Best Sellers Rank: #123,305 in Books (See Top 100 in Books)   #15 inÃ Â Books > Computers & Technology > Business Technology > Microsoft Project   #22 inÃ Â Books > Computers & Technology > Programming > Software Design, Testing & Engineering > Tools   #24 inÃ Â Books > Computers & Technology > Programming > Microsoft Programming > .NET

Customer Reviews

Mark Seemann is a professional software developer and architect living in Copenhagen, Denmark. He has been working with software since 1995 and TDD since 2003, including six years with Microsoft as a consultant, developer and architect. These days he's more into best-of-breed technologies and methodologies such as Continuous Integration, Continuous Delivery, Distributed Version Control Systems, Domain Specific Languages for unit testing and whatever else catches his fancy.

The Short Story: This is the best software development book I have ever read. By miles. Or, in deference to Mr. Seeman, in kilometers. Stop. Buy this book. It will change how you think, how you reason, how you sleep at night. See you in a year, we will both be smarter.The Long Story: I bought this book last year. I can't remember exactly why I bought it, but I suspect that it had something to do with intellectual intimidation and the frightening title. I come from a non-OO background, but I am too young to get through the next 20 years without dealing with the reality of OO prevalence in small (i.e. numerous) projects. I had 2 choices ... start at the bottom or start at the top. Believe me, I chose the latter with this book.I'm not going to explain the content in every chapter, simply because other reviewers have already done the job as I would have. Read Mr. T. Anderson's fine review if you need that kind of detail. Instead, I will talk about the effect that this book had had on how I think.Chapter 2 is the velvet sledgehammer in the face. I read along with the case study, nodding my head and exercising my (in retrospect, tiny) brain as Seeman describes how "Mary" and "Jens" go about building a layered application. I'm thinking, yes Mary and Jens, this is what the magazines, blog articles, and dime-a-dozen gurus are saying regarding the construction of layered application. Seeman then dissects the "layered" application. Actually, he doesn't dissect it; he tears it to shreds and stamps all over it. Brilliantly. It's truly scary to read this chapter. You will feel like a complete novice at the end of it. You then have two choices ... (1) reject this stuff as abstract, ivory tower nonsense, or (2) put on your big-boy-pants.Having lived with this book for almost a year, this is how I suggest you use it:1. Read from Part 1 through Part 3.2. Stop. Think. Cry at how embarrassed you feel at the end of Part 1 Chapter 2.3. Read Part 2 again.4. Stop. Think. Cry at how happy you feel

now that you've lifted yourself beyond 99% of .NET developers. Beyond 99% of software professionals, period.5. Do some "poor-man's DI" exercises.6. Read only the introductory sections of each chapter within Part 4.7. Pick a DI container in Part 4 that appeals to you based on the previous bullet, read its Part 4 chapter, and spend 1-2 months playing with it.8. Revisit Parts 2 and 3 on a regular basis as references.9. Don't think about sauces or Fowlerisms.Criticisms: Very, very, few. I do not think it's too "wordy" at all. Any repetition is done with the realization that this stuff is really, really, hard for most of us. But yes, any mention of nonsense about "Anemic Domain Models" normally sets my blood boiling and warrants an immediate docking of at least 1 star. I can't dock a star from this book. It's that good. It's written by a mildly (but not offensively) dogmatic Danish guy who talks about sauces in every chapter. I like my Filets Mignons dry, thank you very much. And still, at the end of the day ... this is best software development book I have ever read. By far. By miles. By kilometers ...

This is the only technical book that I've purchased multiple times. I keep loaning it out to teammates encouraging them to read it, losing track of it (or, more likely, them not wanting to give it back), needing it again, and then the repurchasing cycle continues. The principles have really altered my approach to programming and problem solving in general, and I've been able to save incredible amounts of time and effort by implementing dependency injection and utilizing the flexibility that comes with it. I've used these principles over the last 5+ years on multiple projects and keep coming back to this book and finding new nuggets of helpful information. Mark explains the ideas and motivations incredibly well. This book should be required reading for every developer, and is worth its weight in gold.That may seem like hyperbole, but it's really not - I would not have been able to meet tight deadlines for many of the projects without- being able to easily test code to find bugs early (the later in the process that a bug is found the more expensive it is to fix) (TDD and Liskov Substitution Principle)- being able to build up an toolbox of small chunks of code that do a single thing well and can be composed into different things quickly depending on what I need at the time- being able to quickly react to changing requirements by swapping out different dependencies (paying attention to where the composition root should be)- being able to add cross-cutting functionality without modifying existing code modules (decorator pattern)- being able to "make things fit" (adapter pattern)- being able to manage project risk by hiding dependencies on other people's code (whether open-source, another team in the company, etc) behind seams/interfaces in the right place

This book is great. His writing style and explanation are clear and engaging. His careful considerations of how most people don't quite "get it" when they attempt a 3-tier architecture was illuminating. I'm a much better software designer after reading this book. I'd recommend it for any serious .Net professional who can write code that "works", but wants to make the jump into writing elegant, maintainable, loosely coupled code.I would agree with a review I've read (somewhere...can't remember if it was on or not...) that there are some prerequisites to reading this book, such as a good grasp of MVC.Net. I definitely got a lot out of the examples, but couldn't help but feel sorry for readers who don't have that background. I'm still giving it 5 stars though. My word of caution to readers before purchasing this book would be to go through some MVC.Net tutorials and at least get familiar with the paradigm itself, the Global.asax file, Entity Framework basics, Controller Factories, and interfaces/abstract classes. You'll get more out of it if you have a basic understanding of these concepts. If you don't, some of the content won't seem relevant.Lastly, I'd recommend reading the author's blog and Stack Overflow articles. He is one sharp dude. LEARN FROM HIM!

Plastic Injection Molding: Product Design & Material Selection Fundamentals (Vol II: Fundamentals of Injection Molding) (Fundamentals of injection molding series) Plastic Injection Molding: Mold Design and Construction Fundamentals (Fundamentals of Injection Molding) (2673) (Fundamentals of injection molding series) Dependency Injection in .NET Pro ASP.NET MVC 5 (Expert's Voice in ASP.Net) Concepts of Chemical Dependency (Book Only) Addiction Treatment (SW 393R 23-Treatment of Chemical Dependency) The Road Back to Me: Healing and Recovering from Co-Dependency, Addiction, Enabling, and Low Self Esteem Chemical Dependency: A Family Affair (Substance Abuse) Chemical Dependency: A Systems Approach (4th Edition) Primal Endurance: Escape chronic cardio and carbohydrate dependency and become a fat burning beast! Dual Disorders: Counseling Clients with Chemical Dependency and Mental Illness Politics of Favoritism in Public Procurement in Turkey: Reconfigurations of Dependency Networks in the AKP Era (Reform and Transition in the Mediterranean) 12 Steps to Raw Foods: How to End Your Dependency on Cooked Food Social Change and Development: Modernization, Dependency and World-System Theories (SAGE Library of Social Research) Substance Abuse Counseling (SW 393R 23-Treatment of Chemical Dependency) Chemical Dependency Counseling: A Practical Guide Addiction Treatment: A Strengths Perspective (SW 393R 23-Treatment of Chemical Dependency) Injection Techniques in Musculoskeletal Medicine: A Practical Manual for Clinicians in Primary and Secondary Care, 4e Foot and Ankle Injection Techniques: A Practical Guide, 1e

Stop-Motion Puppet Sculpting: A Manual of Foam Injection, Build-Up, and Finishing Techniques