

book is the first to combine DDD with techniques from statically typed functional programming. This book is perfect for newcomers to DDD or functional programming - all the techniques you need will be introduced and explained. Model a complex domain accurately using the F# type system, creating compilable code that is also readable documentation---ensuring that the code and design never get out of sync. Encode business rules in the design so that you have "compile-time unit tests," and eliminate many potential bugs by making illegal states unrepresentable. Assemble a series of small, testable functions into a complete use case, and compose these individual scenarios into a large-scale design. Discover why the combination of functional programming and DDD leads naturally to service-oriented and hexagonal architectures. Finally, create a functional domain model that works with traditional databases, NoSQL, and event stores, and safely expose your domain via a website or API. Solve real problems by focusing on real-world requirements for your software. What You Need: The code in this book is designed to be run interactively on Windows, Mac and Linux. You will need a recent version of F# (4.0 or greater), and the appropriate .NET runtime for your platform. Full installation instructions for all platforms at fsharp.org.

This book presents the state of the art, challenges and future trends in automotive software engineering. The amount of automotive software has grown from just a few lines of code in the 1970s to millions of lines in today's cars. And this trend seems destined to continue in the years to come, considering all the innovations in electric/hybrid, autonomous, and connected cars. Yet there are also concerns related to onboard software, such as security, robustness, and trust. This book covers all essential aspects of the field. After a general introduction to the topic, it addresses automotive software development, automotive software reuse, E/E architectures and safety, C-ITS and security, and future trends. The specific topics discussed include requirements engineering for embedded software systems, tools and methods used in the automotive industry, software product lines, architectural frameworks, various related ISO standards, functional safety and safety cases, cooperative intelligent transportation systems, autonomous vehicles, and security and privacy issues. The intended audience includes researchers from academia who want to learn what the fundamental challenges are and how they are being tackled in the industry, and practitioners looking for cutting-edge academic findings. Although the book is not written as lecture notes, it can also be used in advanced master's-level courses on software and system engineering. The book also includes a number of case studies that can be used for student projects.

In the era of technological ubiquity and online interaction, the importance of proper computer training cannot be understated. Following established standards and practices boosts the value of communication in digital environments for all users. The Handbook of Research on Interactive Information Quality in Expanding Social Network Communications examines the strategic elements involved in ICT training within the context of online networks. Combining scientific, theoretical, and practical perspectives on the importance of communicability in such networks, this book is an essential reference source for researchers, students, teachers, designers, ICT specialists, engineers, and computer programmers interested in social networking technologies.

Web 2.0 Web, Ajax, CSS, JavaScript, Flash

In this well-illustrated book the authors, Sinan Kanbir, Ken Clements, and Nerida Ellerton, tackle a persistent, and universal, problem in school mathematics—why do so many middle-school and secondary-school students find it difficult to learn algebra well? What makes the book important are the unique features which comprise the design-research approach that the authors adopted in seeking a solution to the problem. The first unique feature is that the authors offer an overview of the history of school algebra. Despite the fact that algebra has been an important component of secondary-school mathematics for more than three centuries, there has never been a comprehensive historical analysis of factors influencing the teaching and learning of that component. The authors identify, through historical analysis, six purposes of school algebra: (a) algebra as a body of knowledge essential to higher mathematical and scientific studies, (b) algebra as generalized arithmetic, (c) algebra as a prerequisite for entry to higher studies, (d) algebra as offering a language and set of procedures for modeling real-life problems, (e) algebra as an aid to describing structural properties in elementary mathematics, and (f) algebra as a study of variables. They also raise the question whether school algebra represents a unidimensional trait. Kanbir, Clements and Ellerton offer an unusual hybrid theoretical framework for their intervention study (by which seventh-grade students significantly improved their elementary algebra knowledge and skills). Their theoretical frame combined Charles Sanders Peirce's triadic signifier-interpretant-signified theory, which is in the realm of semiotics, with Johann Friedrich Herbart's theory of apperception, and Ken Clements' and Gina Del Campo's theory relating to the need to expand modes of communications in mathematics classrooms so that students engage in receptive and expressive modes. Practicing classroom teachers formed part of the research team. This book appears in Springer's series on the "History of Mathematics Education." Not only does it include an important analysis of the history of school algebra, but it also adopts a theoretical frame which relies more on "theories from the past," than on contemporary theories in the field of mathematics education. The results of the well-designed classroom intervention are sufficiently impressive that the study might have created and illuminated a pathway for future researchers to take.

Java professionals long for the productivity of a framework like Ruby on Rails provides without having to leave the Java platform. The new open source lightweight Grails Framework provides the missing link by using the best aspects of the Java-based Groovy scripting language as well as the Java platform itself. This Rails-like framework gives dynamic Java a boost; Java developers now have a viable Java-based solution instead of the non-Java-based Ruby on Rails, which can create more problems than it solves. Learn all about this in The Definitive Guide to Grails, written by the Grails project founder and lead. Grails is a fast-emerging area of much interest.

This book provides a hands-on, application-oriented guide to the language and methodology of both SystemVerilog Assertions and Functional Coverage. Readers will benefit

from the step-by-step approach to learning language and methodology nuances of both SystemVerilog Assertions and Functional Coverage, which will enable them to uncover hidden and hard to find bugs, point directly to the source of the bug, provide for a clean and easy way to model complex timing checks and objectively answer the question 'have we functionally verified everything'. Written by a professional end-user of ASIC/SoC/CPU and FPGA design and Verification, this book explains each concept with easy to understand examples, simulation logs and applications derived from real projects. Readers will be empowered to tackle the modeling of complex checkers for functional verification and exhaustive coverage models for functional coverage, thereby drastically reducing their time to design, debug and cover. This updated third edition addresses the latest functional set released in IEEE-1800 (2012) LRM, including numerous additional operators and features. Additionally, many of the Concurrent Assertions/Operators explanations are enhanced, with the addition of more examples and figures. - Covers in its entirety the latest IEEE-1800 2012 LRM syntax and semantics; - Covers both SystemVerilog Assertions and SystemVerilog Functional Coverage languages and methodologies; - Provides practical applications of the what, how and why of Assertion Based Verification and Functional Coverage methodologies; - Explains each concept in a step-by-step fashion and applies it to a practical real life example; - Includes 6 practical LABs that enable readers to put in practice the concepts explained in the book.

????????????????????,?:????;????;????;????????

Drawing Programs: The Theory and Practice of Schematic Functional Programming describes a diagrammatic (schematic) approach to programming. It introduces a sophisticated tool for programmers who would rather work with diagrams than with text. The language is a complete functional language that has evolved into a representation scheme that is unique. The result is a simple coherent description of the process of modelling with the computer. The experience of using this tool is introduced gradually with examples, small projects and exercises. The new computational theory behind the tool is interspersed between these practical descriptions so that the reasons for the activity can be understood and the activity, in turn, illustrates some elements of the theory Access to the tool, its source code and a set of examples that range from the simple to the complex is free (see www.springer.com/978-1-84882-617-5). A description of the tool's construction and how it may be extended is also given. The authors' experience with undergraduates and graduates who have the understanding and skill of a functional language learnt through using schema have also shown an enhanced ability to program in other computer languages. Readers are provided with a set of concepts that will ensure a good robust program design and, what is more important, a path to error free programming.

??,????????????????????,????????Java???C#??????.

??????????????,????,??,????????????????????,??.

??????,??????????,????????????????????;??????????????,??;????????????,????????????????????

[Copyright: 9b7e4fcca9d0445d1a5fe23828fb7ea2](http://www.springer.com/978-1-84882-617-5)