Reactive Programming With Swift 4 Build Asynchronous Reactive Applications With Easy To Maintain And Clean Code Using Rxswift And Xcode 9

Best practices to adapt and bottlenecks to avoid About This Book Tackle all kinds of performance-related issues and streamline your development Master the new features and new APIs of Java 9 to implement highly efficient and reliable codes Gain an in-depth knowledge of Java application performance and obtain best results from performance testing Who This Book Is For This book is for Java developers who would like to build reliable and high-performance applications. Prior Java programming knowledge is assumed. What You Will Learn Work with JIT compilers Understand the usage of profiling tools Generate JSON with code examples Leverage the command-line tools to speed up application development Build microservices in Java 9 Explore the use of APIs to improve application code Speed up your application with reactive programming and concurrency In Detail Finally, a book that focuses on the practicalities rather than theory of Java application performance tuning. This book will be your one-stop guide to optimize the performance of your Java applications. We will begin by understanding the new features and APIs of Java 9. You will then be taught the practicalities of Java application performance tuning, how to make the best use of garbage collector, and find out how to optimize code with microbenchmarking. Moving ahead, you will be introduced to multithreading and learning about concurrent programming with Java 9 to build highly concurrent and efficient applications. You will learn how to fine tune your Java code for best results. You will discover techniques on how to benchmark performance and reduce various bottlenecks in your applications. We'll also cover best practices of Java programming that will help you improve the quality of your codebase. By the end of the book, you will be armed with the knowledge to build and deploy efficient, scalable, and concurrent applications in Java. Style and approach This step-by-step guide provides real-world examples to give you a hands-on experience.

Learn reactive programming using Java and its functional aspects, sometimes called RxJava. This book shows you how to solve "callback hell" with RxJava and shows you how to write thread-safe code without hanging onto state variables which comes in handy for cloud computing software-as-a-service issues, especially when dealing with big data processes through streaming. Reactive Java Programming includes unique coverage of reactive Android programming, growing more and more popular in mobile development with the Cloud. After reading this guide to reactive programming, you'll be able to apply it to your own big data cloud applications that use Java. What You'll Learn Use and map observables Filter and combine events Employ subjects, schedulers, and backpressure Handle reactive patterns Test your RxJava code Write your own operators Carry out reactive Android programming Who This Book Is For Experienced Java programmers new to reactive programming and those who may have some experience with reactive programming new to Java.

This book presents a selection of papers from the 2017 World Conference on Information Systems and Technologies (WorldCIST'17), held between the 11st and 13th of April 2017 at Porto Santo Island, Madeira, Portugal. WorldCIST is a global forum for researchers and practitioners to present and discuss recent results and innovations, current trends, professional experiences and challenges involved in modern Information Systems and Technologies research, together with technological developments and applications. The main topics covered are: Information and Knowledge Management; Organizational Models and Information Systems; Software and Systems Modeling; Software Systems, Architectures, Applications and Tools; Multimedia Systems and Applications; Computer Networks, Mobility and Pervasive Systems; Intelligent and Decision Support Systems; Big Data Analytics and Applications; Human–Computer Interaction; Ethics, Computers & Security; Health Informatics; Information Technologies in Education; and Information Technologies in Radiocommunications.

This book constitutes the refereed proceedings of the 8th International RuleML Symposium, RuleML 2014, co-located with the 21st European Conference on Artificial Intelligence, ECAI 2014, held in Prague, Czech Republic, in August 2014. The 17 full and 6 short papers presented together with 3 keynote talks were carefully reviewed and selected from 48 submissions. The papers cover the following topics: semantic web rule languages and standards, rule engines, formal and operational semantics and rule-based systems, the relation between natural language and rules, automation of business rules generation from existing data, and aspects related to legal rules and norms for web and corporate environments.

Bring the power of functional programming to Swift to develop clean, smart, scalable and reliable applications. About This Book Written for the latest version of Swift, this is a comprehensive guide that introduces iOS, Web and macOS developers to the all-new world of functional programming that has so far been alien to them Get familiar with using functional programming alongside existing OOP techniques so you can get the best of both worlds and develop clean, robust, and scalable code Develop a case study on example backend API with Swift and Vapor Framework and an iOS application with Functional Programming, Protocol-Oriented Programming, Functional Reactive Programming, and Object-Oriented Programming techniques Who This Book Is For Meant for a reader who knows object-oriented programming, has some experience with Objective-C/Swift programming languages and wants to further enhance his skills with functional programming techniques with Swift 3.x. What You Will Learn Understand what functional programming is and why it matters Understand custom operators, function composition, currying, recursion, and memoization Explore algebraic data types, pattern matching, generics, associated type protocols, and type erasure Get acquainted with higher-kinded types and higher-order functions using practical examples Get familiar with functional and non-functional ways to deal with optionals Make use of functional data structures such as semigroup, monoid, binary search tree, linked list, stack, and lazy list Understand the importance of immutability, copy constructors, and lenses Develop a backend API with Vapor Create an iOS app by combining FP, OOP, FRP, and POP paradigms In Detail Swift is a multi-paradigm programming language enabling you to tackle different problems in various ways. Understanding each paradigm and knowing when and how to utilize and combine them can lead to a better code base. Functional programming (FP) is an important paradigm that empowers us with declarative development and makes applications more suitable for testing, as well as performant and elegant. This book aims to simplify the FP paradigms, making them easily understandable and usable, by showing you how to solve many of your day-to-day development problems using Swift FP. It starts with the basics of FP, and you will go through all the core concepts of Swift and the building blocks of FP. You will also go through important aspects, such as function composition and currying, custom operator definition, monads, functors, applicative functors,memoization, lenses, algebraic data types, type erasure, functional data structures, functional reactive programming (FRP), and protocol-oriented programming(POP). You will then learn to combine those techniques to develop a fully functional iOS application from scratch Style and approach An easy-to-follow guide that is full of hands-on coding examples of real-world applications. Each topic is explained sequentially and placed in context, and for the more inquisitive, there are more details of the concepts used. It introduces the Swift language basics and functional programming techniques in simple, non-mathematical vocabulary with examples in Swift.

This book constitutes the refereed proceedings of the 13th Pacific Rim Conference on Artificial Intelligence, PRICAI 2014, held in Gold Coast, Queensland, Australia, in December 2014. The 74 full papers and 20 short papers presented in this volume were carefully reviewed and selected from 203 submissions. The topics include inference; reasoning; robotics; social intelligence. AI foundations; applications of

AI; agents; Bayesian networks; neural networks; Markov networks; bioinformatics; cognitive systems; constraint satisfaction; data mining and knowledge discovery; decision theory; evolutionary computation; games and interactive entertainment; heuristics; knowledge acquisition and ontology; knowledge representation, machine learning; multimodal interaction; natural language processing; planning and scheduling; probabilistic.

This book constitutes the refereed proceedings of the 11th Portuguese Conference on Artificial Intelligence, EPIA 2003, held in Beja, Portugal in December 2003. The 29 revised full papers and 20 revised short papers presented were carefully reviewed and selected from a total of 119 submissions. In accordance with the five constituting workshops, the papers are organized in topical sections on artificial life and evolutionary algorithms, constraint and logic programming systems, extraction of knowledge from databases, multi-agent systems and AI for the Internet, and natural language processing and text retrieval.

The use of mathematical logic as a formalism for artificial intelligence was recognized by John McCarthy in 1959 in his paper on Programs with Common Sense. In a series of papers in the 1960's he expanded upon these ideas and continues to do so to this date. It is now 41 years since the idea of using a formal mechanism for AI arose. It is therefore appropriate to consider some of the research, applications and implementations that have resulted from this idea. In early 1995 John McCarthy suggested to me that we have a workshop on Logic-Based Artificial Intelligence (LBAI). In June 1999, the Workshop on Logic-Based Artificial Intelligence was held as a consequence of McCarthy's suggestion. The workshop came about with the support of Ephraim Glinert of the National Science Foundation (IIS-9S2013S), the American Association for Artificial Intelligence who provided support for graduate students to attend, and Joseph JaJa, Director of the University of Maryland Institute for Advanced Computer Studies who provided both manpower and financial support, and the Department of Computer Science. We are grateful for their support. This book consists of refereed papers based on presentations made at the Workshop. Not all of the Workshop participants were able to contribute papers for the book. The common theme of papers at the workshop and in this book is the use of logic as a formalism to solve problems in AI.

This Festschrift volume, published in honor of Michael Gelfond on the occasion of his 65th birthday, contains a collection of papers written by his closest friends and colleagues. Several of these papers were presented during the Symposium on Constructive Mathematics in Computer Science, held in Lexington, KY, USA on October 25-26, 2010. The 27 scientific papers included in the book focus on answer set programming. The papers are organized in sections named “Foundations: ASP and Theories of LP, KR, and NMR”, “ASP and Dynamic Domains”, and “ASP – Applications and Tools”.

This seminal book of Computer Science is the most cited reference on the subject of programming in logic. Originally published in 1979, this now classic text was the first comprehensive attempt to define the scope of logic for problem solving. In this extended edition, Robert Kowalski revisits his classic text in the light of subsequent developments in a substantial commentary of fifty pages. This work investigates the application of logic to problem-solving and computer programming. It assumes no previous knowledge of these fields, and may be appropriate therefore as an introduction to logic, the theory of problem-solving, and computer programming. At the focal point is Computational Logic. It centers around the famous slogan: Algorithm = Logic + Control, which was coined by the author and is explained in this book. According to this view, an algorithm consists of a problem description (the logic part) and a strategy to perform useful computations on this description (the control part). This separation of concerns ideally leads to declarative programs that are simple to develop, clear to understand and easy to maintain.

Deep Dive Into Swift!Swift is a rich language with a plethora of features to offer. Reading the official documentation or entry-level books is important, but it's not enough to grasp the true power of the language.Expert Swift is here to help, by showing you how to harness the full power of Swift. You'll learn about advanced usages of protocols, generics, functional reactive programming, API design and more.Who This Book is ForThis book is for intermediate Swift developers who already know the basics of Swift and are looking to deepen their knowledge and understanding of the language.Topics Covered in Expert SwiftProtocols and Generics: Learn how protocols and generics work, and how you can leverage them in your code to produce clean, long-lasting and easy-to-refactor APIs.Sequences and Collections: Learn how to use Sequences and Collections to write generic algorithms that operate across type families.Unsafe: Understand the memory layout of types and how to use typed and untyped pointers.Functional Reactive Programming: Explore the most important and refined concepts of functional reactive programming and how you can apply these concepts to your apps.Objective-C Interoperability: Learn how to expose Objective-C code to Swift and vice versa.Library and API Design: Enhancing your skill set and intuition for designing great APIs.One thing you can count on: after reading this book, you'll be prepared to use the advanced features of Swift and improve your existing code with the knowledge you'll acquire.

"The Reactive approach will help you to write clean, cohesive, resilient, scalable, and maintainable code.Rx Swift belongs to a large family of Rx implementations in different programming languages that all share an almost identical syntax and semantics. We will introduce you to the world of Reactive programming, primarily focusing on mobile platforms. We tell you how you can benefit from using Rx Swift in your projects, existing or new. We are going to build a simple application that allows people to look up any movie and add it to a favourites list. With this app we will be able to utilize RxSwift to react in real-time to any business logic that could be done through server-side with Google Firebase. The course will demonstrate how unbelievably easy it is to configure asynchronous behavior and other app aspects that are traditionally considered to be hard to implement and maintain. It explains what Rx is made of, and how to switch to the Reactive way of thinking to get the most out of it."--Resource description page.

This book constitutes the thoroughly refereed post-proceedings of the Second International Workshop on Radical Agent Concepts, WRAC 2005, held in Greenbelt, MD, USA in September 2005. The 27 full papers presented are fully revised to incorporate reviewers' comments and discussions at the workshop. Topics addressed are social aspects of agents, agent architectures, autonomic systems, agent communities, and agent intelligence.

???????EJB 3?????????????,??EJB 3?????EJB 3???????Java???API?EJB 3??????????????,???

ThepromiseoftheSemanticWeb,atits most expansive, is to allow knowledge to be freely accessed and exchanged by software. It is now recognized that if the SemanticWebis to containdeepknowledge,theneedfornewrepresentationand reasoning techniques is going to be critical. These techniques need to ?nd the righttrade-o?betweenexpressiveness,scalabilityandrobustnesstodealwiththe inherently incomplete, contradictory and uncertain nature of knowledge on the Web. The International Conference on Web Reasoning and Rule Systems (RR) was founded to address these needs and has grown into a major international forum in this area. The third RR conference was held during October 25–26, 2009 in Chantilly, Virginia, co-located with the International Semantic Web Conference (ISWC 2009). This year 41 papers were submitted from authors in 21 countries. The P- gram Committee performed outstandingly to ensure that each paper submitted to RR 2009 was thoroughly reviewed by at least three referees in a short - riod of time. The resulting conference presented papers of high quality on many of the key issues for reasoning on the Semantic Web. RR 2009 was fortunate to have two distinguished invited speakers. Robert Kowalski, in his talk “- tegrating Logic Programming and Production Systems with Abductive Logic Programming Agents” addressed some of the fundamental considerations - hind reasoning about evolving systems. Benjamin Grossof’s talk “SILK: Higher Level Rules with Defaults and Semantic Scalability” described the design of a major next-generation rule system. The invited tutorial “Uncertainty Reas- ing for the Semantic Web” by Thomas Lukasiewicz provided perspectives on a central issue in this area.

This volume contains the proceedings of the ninth international workshop on logic-based program synthesis and transformation (LOPSTR’99) which was held in Venice (Italy), September 22-24, 1999. LOPSTRistheannualworkshopandforumforresearchersinthelogic-based program development stream of computational logic. The main focus used to be on synthesis and transformation of logic programs, but the workshop is open to contributions on logic-based program development in any paradigm. Previous workshops were held in Manchester, UK (1991, 1992), Louvain-la-Neuve, B- gium (1993), Pisa, Italy (1994), Arnhem, The Netherlands (1995), Stockholm, Sweden (1996), Leuven, Belgium (1997), and Manchester, UK (1998). LOPSTR is a real workshop in the sense that it is a friendly and lively forum for presenting recent and current research as well as discussing future trends. Formal proceedings of the workshop are produced only after the workshop and contain only those papers selected by the program committee after a second refereeing process. The program committee of LOPSTR’99 accepted 20 extended abstracts for presentation at the workshop; then selected 14 papers for inclusion in the po- workshop proceedings. Selected papers cover all the main streams of LOPSTR’s topics: synthesis, specialization, transformation, analysis, and veri?cation. Ve- ?cation, transformation, and specialization methods are applied to functional, constraint, logic, and imperative programming.

????:Richard Helm,Ralph Johnson,John Vlissides ????:???,??,???

This book constitutes the proceedings of the 19th International Conference on Logic for Programming, Artificial Intelligence and Reasoning, LPAR-19, held in December 2013 in Stellenbosch, South Africa. The 44 regular papers and 8 tool descriptions and experimental papers included in this volume were carefully reviewed and selected from 152 submissions. The series of International Conferences on Logic for Programming, Artificial Intelligence and Reasoning (LPAR) is a forum where year after year, some of the most renowned researchers in the areas of logic, automated reasoning, computational logic, programming languages and their applications come to present cutting-edge results, to discuss advances in these fields and to exchange ideas in a scientifically emerging part of the world.

?????:?????

From learning about the most sought-after design patterns to a comprehensive coverage of architectural patterns and code testing, this book is all you need to write clean, reusable code Key Features Write clean, reusable and maintainable code, and make the most of the latest Swift version. Analyze case studies of some of the popular open source projects and give your workflow a huge boost Choose patterns such as MVP, MVC, and MVVM depending on the application being built Book Description Swift keeps gaining traction not only amongst Apple developers but also as a server-side language. This book demonstrates how to apply design patterns and best practices in real-life situations, whether that's for new or already existing projects. You'll begin with a quick refresher on Swift, the compiler, the standard library, and the foundation, followed by the Cocoa design patterns - the ones at the core of many cocoa libraries - to follow up with the creational, structural, and behavioral patterns as defined by the GoF. You'll get acquainted with application architecture, as well as the most popular architectural design patterns, such as MVC and MVVM, and learn to use them in the context of Swift. In addition, you'll walk through dependency injection and functional reactive programming. Special emphasis will be given to techniques to handle concurrency, including callbacks, futures and promises, and reactive programming. These techniques will help you adopt a test-driven approach to your workflow in order to use Swift Package Manager and integrate the framework into the original code base, along with Unit and UI testing. By the end of the book, you'll be able to build applications that are scalable, faster, and easier to maintain. What you will learn Work efficiently with Foundation and Swift Standard library Understand the most critical GoF patterns and use them efficiently Use Swift 4.2 and its unique capabilities (and limitations) to implement and improve GoF patterns Improve your application architecture and optimize for maintainability and performance Write efficient and clean concurrent programs using futures and promises, or reactive programming techniques Use Swift Package Manager to refactor your program into reusable components Leverage testing and other techniques for writing robust code Who this book is for This book is for intermediate developers who want to apply design patterns with Swift to structure and scale their applications. You are expected to have basic knowledge of iOS and Swift.

This book is the Proceedings of the 19th Annual RoboCup International Symposium, held in Hefei, China, in July 2015. The book contains 20 papers presented at the Symposium, carefully selected from 39 submissions. Additionally the book contains 11 champion team papers and one paper from the Workshop on Benchmarking Service Robots. The papers present current research in robotics, artificial intelligence, computer vision, multiagent systems, simulation, and other areas.

????

Learn how to solve blocking user experience and build event based reactive applications with Swift. Key Features Build fast and scalable apps with RxSwift Apply reactive programming to solve complex problems and build efficient programs with reactive user interfaces Take expressiveness, scalability, and maintainability of your Swift code to the next level with this practical guide Book Description RxSwift belongs to a large family of Rx implementations in different programming languages that share almost identical syntax and semantics. Reactive approach will help you to write clean, cohesive, resilient, scalable, and maintainable code with highly configurable behavior. This book will introduce you to the world of reactive programming, primarily focusing on mobile platforms. It will tell how you can benefit from using RxSwift in your projects, existing or new. Further on, the book will demonstrate the unbelievable ease of configuring asynchronous behavior and other aspects of the app that are traditionally considered to be hard to implement and maintain. It will explain what Rx is made of, and how to switch to reactive way of thinking to get the most out of it. Also, test production code using RxTest and the red/ green approach. Finally, the book will dive into real-world recipes and show you how to build a real-world app by applying the reactive paradigm. By the end of the book, you’ll be able to build a reactive swift application by leveraging all the concepts this book takes you through. What you will learn Understand the practical benefits of Rx on a mobile platform Explore the building blocks of Rx, and Rx data flows with marble diagrams Learn how to convert an existing code base into RxSwift code base Learn how to debug and test your Rx Code Work with Playgrounds to transform sequences by filtering them using map, flatmap and other operators Learn how to combine different operators to work with Events in a more controlled manner. Discover RxCocoa and convert your simple UI elements to Reactive components Build a complete RxSwift app using MVVM as design pattern Who this book is for This book is for the developers who are familiar with Swift and iOS application development and are looking out to reduce the complexity of their apps. Prior experience of reactive programming is not necessary.

Declarative languages have long promised the ability to rapidly create easily maintainable software for complex applications. The International Symposium of Practical Aspects of Declarative Languages (PADL) provides a yearly - rum for presenting results on the principles the implementations and especially the applications of declarative languages. The PADL symposium held January 19–20, 2009 in Savannah, Georgia was the 11th in this series. This year 48 papers were submitted from authors in 17 countries. The P- gram Committee performed outstandingly to ensure that each of these papers submitted to PADL 2009 was thoroughly reviewed by at least three referees in a short period of time. The resulting symposium presented a microcosm of how the current generation of declarative languages are being used to address real applications, along with on-going work on the languages themselves. The program also included two invited talks, “Inspecting and Preferring Abductive Models” by Luis Moniz Pereira and “Applying Declarative Languages to C- mercial Hardware Design” by Je? Lewis. Regular papers presented a variety of applications, including distributed applications over networks, network veri?- tion, user interfaces, visualization in astrophysics, nucleotide sequence analysis and planning under incomplete information. PADL 2009 also included ongoing work on the declarative languages themselves. Multi-threaded and concurrent Prolog implementation was addressed in several papers, as were innovations for tabling in Prolog and functional arraysin Haskell. Recent applications have also sparked papers on meta-predicates in Prolog and a module system for ACL2.

This book presents selected papers from the 10th International Conference on Information Science and Applications (ICISA 2019), held on December 16–18, 2019, in Seoul, Korea, and provides a snapshot of the latest issues regarding technical convergence and convergences of security technologies. It explores how information science is at the core of most current research as well as industrial and commercial activities. The respective chapters cover a broad range of topics, including ubiquitous computing, networks and information systems, multimedia and visualization, middleware and operating systems, security and privacy, data mining and artificial intelligence, software engineering and web technology, as well as applications and problems related to technology convergence, which are reviewed and illustrated with the aid of case studies. Researchers in academia, industry, and at institutes focusing on information science and technology will gain a deeper understanding of the current state of the art in information strategies and technologies for convergence security. ?

This book constitutes the refereed proceedings of the 6th International Conference on Database Theory, ICDT '97, held in Delphi, Greece, in January 1997. The 29 revised full papers presented in the volume were carefully selected from a total of 118 submissions. Also included are invited papers by Serge Abiteboul and Jeff Ullman as well as a tutorial on data mining by Heikki Mannila. The papers are organized in sections on conjunctive queries in heterogeneous databases, logic and databases, active databases, new applications, concurrency control, unstructured data, object-oriented databases, access methods, and spatial and bulk data.

Beginning Reactive Programming with SwiftUsing RxSwift, Amazon Web Services, and JSON with iOS and macOSApress

Modern distributed applications must deliver near-realtime performance while simultaneously managing big data and high user loads spread across environments ranging from cloud systems to mobile devices. Unlike traditional enterprise applications which focus on decoupling their internal components by defining programming interfaces, reactive applications go one step further and decouple their components also at runtime. This makes it possible to react effectively and efficiently to failures, varying user demands, and changes in the application's execution environment. The resulting systems are highly concurrent and fault-tolerant, with minimal dependencies among individual system components. Reactive Design Patterns is a clearly-written guide for building message-driven distributed systems that are resilient, responsive, and elastic. It contains patterns for messaging, flow control, resource management, and concurrency, along with practical issues like test-friendly designs. All patterns include concrete examples using Scala and Akka—in some cases, Java, JavaScript, and Erlang. Software engineers and architects will learn patterns that address day-to-day distributed development problems in a fault-tolerant and scalable way. Project leaders and CTOs will gain a deeper understanding of the reactive design philosophy. Purchase of the print book includes a free eBook in PDF, Kindle, and ePub formats from Manning Publications.

The simplest way to create world-class apps Have a unique app idea but worried you don’t quite have the coding skills to build it? Good news: You can stop fretting about someone beating you to market with the same idea and start work right now using SwiftUI. SwiftUI is a gateway app development framework that has become one of the best ways for fledgling developers to get iOS apps off the ground without having to become a coding expert overnight. SwiftUI For Dummies makes that process even faster, providing a friendly introduction to the SwiftUI and Swift programming language and helping you feel right at home creating and building with playgrounds. The book also covers the frameworks and APIs that make it so easy to create smooth, intuitive interfaces—just dive right in and have fun! Combine projects into workspaces Employ Xcode editing tools Use constants and variables Test your code on iOS Simulator Time is of the essence, and with SwiftUI For Dummies, it’s also on your side. Get going with this friendly guide today, and you’ll be celebrating the successful launch of your app way before you thought possible!

This book constitutes the strictly refereed proceedings of the 9th International Conference on Computer Aided Verification, CAV '97, held in Haifa, Israel, in June 1997. The volume presents 34 revised full papers selected from a total of 84 submissions. Also included are 7 invited contributions as well as 12 tool descriptions. The volume is dedicated to the theory and practice of computer aided formal methods for software and hardware verification, with an emphasis on verification tools and algorithms and the techniques needed for their implementation. The book is a unique record documenting the recent progress in the area.

Learn the basics of reactive programming and how it makes apps more responsive. This book shows you how to incorporate reactive programming into existing development products and cycles using RXSwift and RXCocoa on iOS and Mac. As we move away from the traditional paradigm of typing or touching one step at a time to interact with programs, users expect apps to adapt and not need constant hand-holding. People today expect their devices to do much more than just follow commands. They expect devices to react and adapt. Reactive programming, a new term for asynchronous processing, requires new app architectures, and you'll learn how these are already built into iOS and macOS in many places. As part of this more complex environment, you'll move beyond Cocoa and Cocoa Touch to incorporate data from Amazon Web Services (AWS), JavaScript Object Notation (JSON), and other formats, and standards. Together with the concepts of reactive programming and RxSwift, these tools help you build more powerful and useful apps that have wide appeal and use. What You'll Learn Work with tools such as Darwin microkernel, RxSwift, and RxCocoa Use Git repositiories and other resoucrces to get into coding Create apps that adapt to gestures and UI interaction as well as what's happening in and around the environment of the app itself. Who This Book Is For This book is for Swift programmers interested in learning to create reactive apps with RxSwift.

This book is dedicated to Marek Sergot, Professor in Computational Logic at Imperial College London, on the occasion of his 60th birthday. Professor Sergot’s scientific contributions range over many different fields. He has developed a series of novel ideas and formal methods bridging areas including artificial intelligence, computational logic, philosophical logic, legal theory, artificial intelligence and law, multi-agent systems and bioinformatics. By combining his background in logic and computing with his interest in the law, deontic logic, action, and related areas, and applying to all his capacity to understand the subtleties of social interaction and normative reasoning, Professor Sergot has opened up new directions of research, and has been a reference, an inspiration, and a model for many researchers in the fields to which he has contributed. The Festschrift includes several reminiscences and introductory essays describing Professor Sergot's achievements, followed by a series of articles on logic programming, temporal reasoning and action languages, artificial intelligence and law, deontic logic and norm-governed systems, and logical approaches to policies.

This book presents revised full papers from the 10th International Workshop on Logic-Based Program Synthesis and Transformation, LOPSTR 2000, held in London, UK, in July 2000 as part of the International Conference on Computational Logic. The 10 revised full papers presented have gone through two rounds of reviewing, selection and revision. The book is divided in topical sections on synthesis, transformation, analysis, specialization, and abstract interpretation.

????????Servlet?JSP,???????????(????????????),????JSP?????,JSP??????,??????????,????????,?????????????????

Copyright: 69a25f8a55da8456ed0d430964381976