

# Read Book Functional And Reactive Domain Modeling Pdf For Free

Functional and Reactive Domain Modeling Functional and Reactive Domain Modeling Implementing Domain-driven Design Hands-On Reactive Programming in Spring 5 Domain Modeling Made Functional Reactive Messaging Patterns with the Actor Model Patterns, Principles, and Practices of Domain-Driven Design Hands-On Spring Security 5 for Reactive Applications Domain-Driven Design Distilled Reactive Application Development Modern Java in Action Functional Reactive Programming Reactive Design Patterns Architecture Patterns with Python Domain-driven Design Reactive Programming with RxJava Feedstock Technology for Reactive Metal Injection Molding Hands-On Reactive Programming with Clojure DSLs in Action Mastering Shiny Domain-Driven Design and Microservices Hands-On Domain-Driven Design with .NET Core Scala Reactive Programming Real-World Functional Programming Reactive Systems in Java Reactive Web Applications Protein Sensors and Reactive Oxygen Species, Part B: Thiol Enzymes and Proteins Introduction to Reactive Gas Dynamics Reactive Transport in Natural and Engineered Systems Building Evolutionary Architectures Practical Domain-Driven Design in Enterprise Java Applied Akka Patterns Learning for Adaptive and Reactive Robot Control Domain Storytelling Reactive Streams in Java Protein Sensors and Reactive Oxygen Species Secure by Design Feedback Systems Antioxidants and Reactive Oxygen Species in Plants Site Reliability Engineering

**Protein Sensors and Reactive Oxygen Species, Part B: Thiol Enzymes and Proteins** Feb 11 2021 This volume of Methods in Enzymology is a companion to Volume 347 and addresses direct sensing of reactive oxygen species and related free radicals by thiol enzymes and proteins.

*Reactive Design Patterns* Apr 27 2022 Summary Reactive Design Patterns is a clearly written guide for building message-driven distributed systems that are resilient, responsive, and elastic. In this book you'll find 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. Foreword by Jonas Bonér. Purchase of the print book includes a free eBook in PDF, Kindle, and ePub formats from Manning Publications. About the Technology Modern web applications serve potentially vast numbers of users - and they need to keep working as servers fail and new ones come online, users overwhelm limited resources, and information is distributed globally. A Reactive application adjusts to partial failures and varying loads, remaining responsive in an ever-changing distributed environment. The secret is message-driven architecture - and design patterns to organize it. About the Book Reactive Design Patterns presents the principles, patterns, and best practices of Reactive application design. You'll learn how to keep one slow component from bogging down others with the Circuit Breaker pattern, how to shepherd a many-staged transaction to completion with the Saga pattern, how to divide datasets by Sharding, and more. You'll even see how to keep your source code readable and the system testable despite many potential interactions and points of failure. What's Inside The definitive guide to the Reactive Manifesto Patterns for flow control, delimited consistency, fault tolerance, and much more Hard-won lessons about what doesn't work Architectures that scale under tremendous load About the Reader Most examples use Scala, Java, and Akka. Readers should be familiar with distributed systems. About the Author Dr. Roland Kuhn led the Akka team at Lightbend and coauthored the Reactive Manifesto. Brian Hanafée and Jamie Allen are experienced distributed systems architects. Table of Contents PART 1 - INTRODUCTION Why Reactive? A walk-through of the Reactive Manifesto Tools of the trade PART 2 - THE PHILOSOPHY IN A NUTSHELL Message passing Location transparency Divide and conquer Principled failure handling Delimited consistency Nondeterminism by need Message flow PART 3 - PATTERNS Testing reactive applications Fault tolerance and recovery patterns Replication patterns Resource-management patterns Message flow patterns Flow control patterns State management and persistence patterns

*Antioxidants and Reactive Oxygen Species in Plants* Jan 31 2020 Reactive oxygen species (ROS) are produced during the interaction of metabolism with oxygen. As ROS have the potential to cause oxidative damage by reacting with biomolecules, research on ROS has concentrated on the oxidative damage that results from exposure to environmental stresses and on the role of ROS in defence against pathogens. However, more recently, it has become apparent that ROS also have important roles as signalling molecules. A complex network of enzymatic and small molecule antioxidants controls the concentration of ROS and repairs oxidative damage, and research is revealing the complex and subtle interplay between ROS and antioxidants in controlling plant growth, development and response to the environment. This book covers these new developments, generally focussing on molecular and biochemical details and providing a point of entry to the detailed literature. It is directed at researchers and professionals in plant molecular biology, biochemistry and cell biology, in both the academic and industrial sectors.

*Reactive Application Development* Jul 31 2022 Summary Reactive Application Development is a hands-on guide that teaches you how to build reliable enterprise applications using reactive design patterns. Purchase of the print book includes a free eBook in PDF, Kindle, and ePub formats from Manning Publications. Foreword by Jonas Bonér, Creator of Akka About the Technology Mission-critical applications have to respond instantly to changes in load, recover gracefully from failure, and satisfy exacting requirements for performance, cost, and reliability. That's no small task! Reactive designs make it easier to meet these demands through modular, message-driven architecture, innovative tooling, and cloud-based infrastructure. About the Book Reactive Application Development teaches you how to build reliable enterprise applications using reactive design patterns. This hands-on guide begins by exposing you to the reactive mental model, along with a survey of core technologies like the Akka actors framework. Then, you'll build a proof-of-concept system in Scala, and learn to use patterns like CQRS and Event Sourcing. You'll master the principles of reactive design as you implement elasticity and resilience, integrate with traditional architectures, and learn powerful testing techniques. What's Inside Designing elastic domain models Building fault-tolerant systems Efficiently handling large data volumes Examples can be built in Scala or Java About the Reader Written for Java or Scala programmers familiar with distributed application designs. About the Author Duncan DeVore, Sean Walsh, and Brian Hanafee are seasoned architects with experience building and deploying reactive systems in production. Table of Contents PART 1 - FUNDAMENTALS What is a reactive application? Getting started with Akka Understanding Akka PART 2 - BUILDING A REACTIVE APPLICATION Mapping from domain to toolkit Domain-driven design Using remote actors Reactive streaming CQRS and Event Sourcing A reactive interface Production readiness

*Functional Reactive Programming* May 29 2022 Summary Functional Reactive Programming teaches the concepts and applications of FRP. It offers a careful walk-through of core FRP operations and introduces the concepts and techniques you'll need to use FRP in any language. Purchase of the print book includes a free eBook in PDF, Kindle, and ePub formats from Manning Publications. About the Technology Today's software is shifting to more asynchronous, event-based solutions. For decades, the Observer pattern has been the go-to event infrastructure, but it is known to be bug-prone. Functional reactive programming (FRP) replaces Observer, radically improving the quality of event-based code. About the Book Functional Reactive Programming teaches you how FRP works and how to use it. You'll begin by gaining an understanding of what FRP is and why it's so powerful. Then, you'll work through greenfield and legacy code as you learn to apply FRP to practical use cases. You'll find examples in this book from many application domains using both Java and JavaScript. When you're finished, you'll be able to use the FRP approach in the systems you build and spend less time fixing problems. What's Inside Think differently about data and events FRP techniques for Java and JavaScript Eliminate Observer one listener at a time Explore Sodium, RxJS,

and Kefir.js FRP systems About the Reader Readers need intermediate Java or JavaScript skills. No experience with functional programming or FRP required. About the Authors Stephen Blackheath and Anthony Jones are experienced software developers and the creators of the Sodium FRP library for multiple languages. Foreword by Heinrich Apfelmus. Illustrated by Duncan Hill. Table of Contents Stop listening! Core FRP Some everyday widget stuff Writing a real application New concepts FRP on the web Switch Operational primitives Continuous time Battle of the paradigms Programming in the real world Helpers and patterns Refactoring Adding FRP to existing projects Future directions

**Reactive Web Applications** Mar 15 2021 Summary Reactive Web Applications teaches web developers how to benefit from the reactive application architecture and presents hands-on examples using the Play framework. Purchase of the print book includes a free eBook in PDF, Kindle, and ePub formats from Manning Publications. About the Technology Reactive applications build on top of components that communicate asynchronously as they react to user and system events. As a result, they become scalable, responsive, and fault-tolerant. Java and Scala developers can use the Play Framework and the Akka concurrency toolkit to easily implement reactive applications without building everything from scratch. About the Book Reactive Web Applications teaches web developers how to benefit from the reactive application architecture and presents hands-on examples using Play, Akka, Scala, and Reactive Streams. This book starts by laying out the fundamentals required for writing functional and asynchronous applications and quickly introduces Play as a framework to handle the plumbing of your application. The book alternates between chapters that introduce reactive ideas (asynchronous programming with futures and actors, managing distributed state with CQRS) and practical examples that show you how to build these ideas into your applications. What's Inside Reactive application architecture Basics of Play and Akka Examples in Scala Functional and asynchronous programming About Reader Description For readers comfortable programming with a higher-level language such as Java or C#, and who can read Scala code. No experience with Play or Akka needed. About the Author Manuel Bernhardt is a passionate engineer, author, and speaker. As a consultant, he guides companies through the technological and organizational transformation to distributed computing. Table of Contents PART 1 GETTING STARTED WITH REACTIVE WEB APPLICATIONS Did you say reactive? Your first reactive web application Functional programming primer Quick introduction to Play PART 2 CORE CONCEPTS Futures Actors Dealing with state Responsive user interfaces PART 3 ADVANCED TOPICS Reactive Streams Deploying reactive Play applications Testing reactive web applications

**Introduction to Reactive Gas Dynamics** Jan 13 2021 In high energy gas flows, at high velocities and high temperatures, physical and chemical processes such as molecular vibrational excitation, dissociation, ionisation or various reactions take place and deeply influence the structure of the flows. The characteristic times of these processes have the same order of magnitude as

aerodynamic characteristic times, so that these reactive media are generally in thermodynamic and chemical non-equilibrium. This book presents a general introductory study of these media. In the first part their fundamental statistical aspects are described, starting from their discrete structure and taking into account the interactions between elementary particles: transport phenomena, relaxation and kinetics as well as their coupling are analysed and illustrated by many examples. The second part deals with the macroscopic re-entry bodies. Finally, the experimental aspects of these flows, their simulations in shock tubes and shock tunnels are described, as well as their application, particularly in the aerospace domain. This book is intended for students that have acquired a basic knowledge in thermodynamics, statistical physics and fluid mechanics. It will also be of interest to engineers in research and industry, in particular in the aerospace industry, and more generally to all researchers trying to simulate and calculate complex reactive flows.

**Feedstock Technology for Reactive Metal Injection Molding** Dec 24 2021 Feedstock Technology for Reactive Metal Injection Molding: Process, Design, and Application provides an authoritative guide on the basics of feedstock technology and the latest developments in binders for titanium metal injection moulding and their potential implications. In addition, the book presents challenges that MIM technology of reactive metals is currently facing and potential solutions for commercial success. As both commercial growth and research development are fundamentally driven by the economics of manufacture, this book presents the problems and potential solutions regarding reactive metals, making it a valuable resource for engineers intending to utilize MIM in commercial product design. Provides comprehensive details and case studies on the feedstocks currently under extensive development, in research and in the commercial domain Discusses the most recent developments of binder chemistry and design, along with the most critical challenges in MIM technology Includes comprehensive evaluations with regard to feedstock characterization and impurity control

*Hands-On Reactive Programming with Clojure* Nov 22 2021 Learn how to use RxClojure to deal with stateful computations  
Key FeaturesLeverage the features of Functional Reactive Programming using ClojureCreate dataflow-based systems that are the building blocks of Reactive ProgrammingUse different Functional Reactive Programming frameworks, techniques, and patterns to solve real-world problemsBook Description Reactive Programming is central to many concurrent systems, and can help make the process of developing highly concurrent, event-driven, and asynchronous applications simpler and less error-prone. This book will allow you to explore Reactive Programming in Clojure 1.9 and help you get to grips with some of its new features such as transducers, reader conditionals, additional string functions, direct linking, and socket servers. Hands-On Reactive Programming with Clojure starts by introducing you to Functional Reactive Programming (FRP) and its formulations, as well as showing you how it inspired Compositional Event Systems (CES). It then guides you in understanding Reactive

Programming as well as learning how to develop your ability to work with time-varying values thanks to examples of reactive applications implemented in different frameworks. You'll also gain insight into some interesting Reactive design patterns such as the simple component, circuit breaker, request-response, and multiple-master replication. Finally, the book introduces microservices-based architecture in Clojure and closes with examples of unit testing frameworks. By the end of the book, you will have gained all the knowledge you need to create applications using different Reactive Programming approaches. What you will learn

- Understand how to think in terms of time-varying values and event streams
- Create, compose, and transform observable sequences using Reactive extensions
- Build a CES framework from scratch using core.async as its foundation
- Develop a simple ClojureScript game using Reagi
- Integrate Om and RxJS in a web application
- Implement a reactive API in Amazon Web Services (AWS)
- Discover helpful approaches to backpressure and error handling
- Get to grips with futures and their applications

Who this book is for If you're interested in using Reactive Programming to build asynchronous and concurrent applications, this is the book for you. Basic knowledge of Clojure programming is necessary to understand the concepts covered in this book.

*Reactive Transport in Natural and Engineered Systems* Dec 12 2020 Open system behavior is predicated on a fundamental relationship between the timescale over which mass is transported and the timescale over which it is chemically transformed. This relationship describes the basis for the multidisciplinary field of reactive transport (RT). In the 20 years since publication of *Review in Mineralogy and Geochemistry* volume 34: *Reactive Transport in Porous Media*, RT principles have expanded beyond early applications largely based in contaminant hydrology to become broadly utilized throughout the Earth Sciences. RT is now employed to address a wide variety of natural and engineered systems across diverse spatial and temporal scales, in tandem with advances in computational capability, quantitative imaging and reactive interface characterization techniques. The present volume reviews the diversity of reactive transport applications developed over the past 20 years, ranging from the understanding of basic processes at the nano- to micrometer scale to the prediction of Earth global cycling processes at the watershed scale. Key areas of RT development are highlighted to continue advancing our capabilities to predict mass and energy transfer in natural and engineered systems.

**Hands-On Reactive Programming in Spring 5** Feb 06 2023 Explore the reactive system and create efficient microservices with Spring Boot 2.1 and Spring Cloud Key Features Understand the kind of system modern businesses require with Spring Gain deeper insights into reactive programming with Reactor and Spring Cloud Get in-depth knowledge on asynchronous and nonblocking communication with Spring 5 WebFlux Book Description These days, businesses need a new type of system that can remain responsive at all times. This is achievable with reactive programming; however, the development of these kinds of systems is a complex task, requiring a deep understanding of the domain. In order to develop highly responsive systems, the

developers of the Spring Framework came up with Project Reactor. Hands-On Reactive Programming in Spring 5 begins with the fundamentals of Spring Reactive programming. You'll explore the endless possibilities of building efficient reactive systems with the Spring 5 Framework along with other tools such as WebFlux and Spring Boot. Further on, you'll study reactive programming techniques and apply them to databases and cross-server communication. You will advance your skills in scaling up Spring Cloud Streams and run independent, high-performant reactive microservices. By the end of the book, you will be able to put your skills to use and get on board with the reactive revolution in Spring 5.1! What you will learn Discover the difference between a reactive system and reactive programming Explore the benefits of a reactive system and understand its applications Get to grips with using reactive programming in Spring 5 Gain an understanding of Project Reactor Build a reactive system using Spring 5 and Project Reactor Create a highly efficient reactive microservice with Spring Cloud Test, monitor, and release reactive applications Who this book is for This book is for Java developers who use Spring to develop their applications and want to build robust and reactive applications that can scale in the cloud. Basic knowledge of distributed systems and asynchronous programming will help you understand the concepts covered in this book.

Reactive Streams in Java Jun 05 2020 Get an easy introduction to reactive streams in Java to handle concurrency, data streams, and the propagation of change in today's applications. This compact book includes in-depth introductions to RxJava, Akka Streams, and Reactor, and integrates the latest related features from Java 9 and 11, as well as reactive streams programming with the Android SDK. Reactive Streams in Java explains how to manage the exchange of stream data across an asynchronous boundary—passing elements on to another thread or thread-pool—while ensuring that the receiving side is not forced to buffer arbitrary amounts of data which can reduce application efficiency. After reading and using this book, you'll be proficient in programming reactive streams for Java in order to optimize application performance, and improve memory management and data exchanges. What You Will Learn Discover reactive streams and how to use them Work with the latest features in Java 9 and Java 11 Apply reactive streams using RxJava Program using Akka Streams Carry out reactive streams programming in Android Who This Book Is For Experienced Java programmers.

*Reactive Systems in Java* Apr 15 2021 Reactive systems and event-driven architecture are becoming indispensable to application design, and companies are taking note. Reactive systems ensure that applications are responsive, resilient, and elastic no matter what failures or errors may be occurring, while event-driven architecture offers a flexible and composable option for distributed systems. This practical book helps Java developers bring these approaches together using Quarkus 2.x, the Kubernetes-native Java framework. Clement Escoffier and Ken Finnigan show you how to take advantage of event-driven and reactive principles to build robust distributed systems, reducing latency and increasing throughput, particularly in microservices and serverless

applications. You'll also get a foundation in Quarkus to help you create true Kubernetes-native applications for the cloud. Understand the fundamentals of reactive systems and event-driven architecture Learn how to use Quarkus to build reactive applications Combine Quarkus with Apache Kafka or AMQP to build reactive systems Develop microservices that utilize messages with Quarkus for use in event-driven architectures Learn how to integrate external messaging systems, such as Apache Kafka, with Quarkus Build applications with Quarkus using reactive systems and reactive programming concepts

*Domain Modeling Made Functional* Jan 05 2023 You want increased customer satisfaction, faster development cycles, and less wasted work. Domain-driven design (DDD) combined with functional programming is the innovative combo that will get you there. In this pragmatic, down-to-earth guide, you'll see how applying the core principles of functional programming can result in software designs that model real-world requirements both elegantly and concisely - often more so than an object-oriented approach. Practical examples in the open-source F# functional language, and examples from familiar business domains, show you how to apply these techniques to build software that is business-focused, flexible, and high quality. Domain-driven design is a well-established approach to designing software that ensures that domain experts and developers work together effectively to create high-quality software. This 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](http://fsharp.org).

**Site Reliability Engineering** Jan 01 2020 The overwhelming majority of a software system's lifespan is spent in use, not in design or implementation. So, why does conventional wisdom insist that software engineers focus primarily on the design and development of large-scale computing systems? In this collection of essays and articles, key members of Google's Site Reliability Team explain how and why their commitment to the entire lifecycle has enabled the company to successfully build,



deploy, monitor, and maintain some of the largest software systems in the world. You'll learn the principles and practices that enable Google engineers to make systems more scalable, reliable, and efficient—lessons directly applicable to your organization. This book is divided into four sections: Introduction—Learn what site reliability engineering is and why it differs from conventional IT industry practices Principles—Examine the patterns, behaviors, and areas of concern that influence the work of a site reliability engineer (SRE) Practices—Understand the theory and practice of an SRE's day-to-day work: building and operating large distributed computing systems Management—Explore Google's best practices for training, communication, and meetings that your organization can use

**Domain Storytelling** Jul 07 2020 Storytelling is at the heart of human communication--why not use it to overcome costly misunderstandings when designing software? By telling and visualising stories, domain experts and team members make business processes and domain knowledge tangible. Domain Storytelling enables everyone to understand the relevant people, activities, and work items. With this guide, the method's inventors explain how domain experts and teams can work together to capture insights with simple pictographs, show their work, solicit feedback, and get everyone on the same page. Stefan Hofer and Henning Schwentner introduce the methods easy pictographic language, scenario-based modeling techniques, workshop format, and relationship to other modeling methods. Using step-by-step case studies, they guide you through solving many common problems: Fully align all project participants and stakeholders, both technical and business-focused Master a simple set of symbols and rules for modeling any process or workflow Use workshop-based collaborative modeling to find better solutions faster Draw clear boundaries to organise your domain, software, and teams Transform domain knowledge into requirements, embedded naturally into an agile process Move your models from diagrams and sticky notes to code Gain better visibility into your IT landscape so you can consolidate or optimise it This guide is for everyone who wants more effective software--from developers, architects, and team leads to the domain experts, product owners, and executives who rely on it every day.

*Real-World Functional Programming* May 17 2021 Functional programming languages like F#, Erlang, and Scala are attracting attention as an efficient way to handle the new requirements for programming multi-processor and high-availability applications. Microsoft's new F# is a true functional language and C# uses functional language features for LINQ and other recent advances. *Real-World Functional Programming* is a unique tutorial that explores the functional programming model through the F# and C# languages. The clearly presented ideas and examples teach readers how functional programming differs from other approaches. It explains how ideas look in F#-a functional language-as well as how they can be successfully used to solve programming problems in C#. Readers build on what they know about .NET and learn where a functional approach makes the most sense and how to apply it effectively in those cases. The reader should have a good working knowledge of C#. No prior

exposure to F# or functional programming 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.

**Domain-Driven Design and Microservices** Aug 20 2021 Domain-Driven Design (DDD) concept was introduced by first Eric Evans in 2003. The concept of microservices did not exist at that time. So basically DDD was introduced to solve the problem of a large monolithic code base. In the monolithic world, once the codebase starts growing with the growth of the business, it becomes difficult to maintain the code organized and structured as it was originally designed. Monolithic applications designed using MVC architecture have good separation between the business layer and the presentation layer. But in the absence of the strict architectural guidelines, the business layer does not provide specific rules to maintain responsibility boundaries between different modules and classes. That's why as the code base grows it increases the risk of logic breakdown, responsibility leakage between the different components of the application.

**Learning for Adaptive and Reactive Robot Control** Aug 08 2020 Methods by which robots can learn control laws that enable real-time reactivity using dynamical systems; with applications and exercises. This book presents a wealth of machine learning techniques to make the control of robots more flexible and safe when interacting with humans. It introduces a set of control laws that enable reactivity using dynamical systems, a widely used method for solving motion-planning problems in robotics. These control approaches can replan in milliseconds to adapt to new environmental constraints and offer safe and compliant control of forces in contact. The techniques offer theoretical advantages, including convergence to a goal, non-penetration of obstacles, and passivity. The coverage of learning begins with low-level control parameters and progresses to higher-level competencies composed of combinations of skills. Learning for Adaptive and Reactive Robot Control is designed for graduate-level courses in robotics, with chapters that proceed from fundamentals to more advanced content. Techniques covered include learning from demonstration, optimization, and reinforcement learning, and using dynamical systems in learning control laws, trajectory planning, and methods for compliant and force control . Features for teaching in each chapter: • applications, which range from arm manipulators to whole-body control of humanoid robots; • pencil-and-paper and programming exercises; • lecture videos, slides, and MATLAB code examples available on the author's website . • an eTextbook platform website offering protected material[EPS2] for instructors including solutions.

**Reactive Programming with RxJava** Jan 25 2022 In today's app-driven era, when programs are asynchronous and responsiveness is so vital, reactive programming can help you write code that's more reliable, easier to scale, and better-performing. With this practical book, Java developers will first learn how to view problems in the reactive way, and then build programs that leverage the best features of this exciting new programming paradigm. Authors Tomasz Nurkiewicz and Ben

Christensen include concrete examples that use the RxJava library to solve real-world performance issues on Android devices as well as the server. You'll learn how RxJava leverages parallelism and concurrency to help you solve today's problems. This book also provides a preview of the upcoming 2.0 release. Write programs that react to multiple asynchronous sources of input without descending into "callback hell" Get to that aha! moment when you understand how to solve problems in the reactive way Cope with Observables that produce data too quickly to be consumed Explore strategies to debug and to test programs written in the reactive style Efficiently exploit parallelism and concurrency in your programs Learn about the transition to RxJava version 2

Modern Java in Action Jun 29 2022 Summary Manning's bestselling Java 8 book has been revised for Java 9! In Modern Java in Action, you'll build on your existing Java language skills with the newest features and techniques. Purchase of the print book includes a free eBook in PDF, Kindle, and ePub formats from Manning Publications. About the Technology Modern applications take advantage of innovative designs, including microservices, reactive architectures, and streaming data. Modern Java features like lambdas, streams, and the long-awaited Java Module System make implementing these designs significantly easier. It's time to upgrade your skills and meet these challenges head on! About the Book Modern Java in Action connects new features of the Java language with their practical applications. Using crystal-clear examples and careful attention to detail, this book respects your time. It will help you expand your existing knowledge of core Java as you master modern additions like the Streams API and the Java Module System, explore new approaches to concurrency, and learn how functional concepts can help you write code that's easier to read and maintain. What's inside Thoroughly revised edition of Manning's bestselling Java 8 in Action New features in Java 8, Java 9, and beyond Streaming data and reactive programming The Java Module System About the Reader Written for developers familiar with core Java features. About the Author Raoul-Gabriel Urma is CEO of Cambridge Spark. Mario Fusco is a senior software engineer at Red Hat. Alan Mycroft is a University of Cambridge computer science professor; he cofounded the Raspberry Pi Foundation. Table of Contents PART 1 - FUNDAMENTALS Java 8, 9, 10, and 11: what's happening? Passing code with behavior parameterization Lambda expressions PART 2 - FUNCTIONAL-STYLE DATA PROCESSING WITH STREAMS Introducing streams Working with streams Collecting data with streams Parallel data processing and performance PART 3 - EFFECTIVE PROGRAMMING WITH STREAMS AND LAMBDAS Collection API enhancements Refactoring, testing, and debugging Domain-specific languages using lambdas PART 4 - EVERYDAY JAVA Using Optional as a better alternative to null New Date and Time API Default methods The Java Module System PART 5 - ENHANCED JAVA CONCURRENCY Concepts behind CompletableFuture and reactive programming CompletableFuture: composable asynchronous programming Reactive programming PART 6 - FUNCTIONAL PROGRAMMING AND FUTURE

JAVA EVOLUTION Thinking functionally Functional programming techniques Blending OOP and FP: Comparing Java and Scala Conclusions and where next for Java

Functional and Reactive Domain Modeling Apr 08 2023 Summary Functional and Reactive Domain Modeling teaches you how to think of the domain model in terms of pure functions and how to compose them to build larger abstractions. Purchase of the print book includes a free eBook in PDF, Kindle, and ePub formats from Manning Publications. About the Technology Traditional distributed applications won't cut it in the reactive world of microservices, fast data, and sensor networks. To capture their dynamic relationships and dependencies, these systems require a different approach to domain modeling. A domain model composed of pure functions is a more natural way of representing a process in a reactive system, and it maps directly onto technologies and patterns like Akka, CQRS, and event sourcing. About the Book Functional and Reactive Domain Modeling teaches you consistent, repeatable techniques for building domain models in reactive systems. This book reviews the relevant concepts of FP and reactive architectures and then methodically introduces this new approach to domain modeling. As you read, you'll learn where and how to apply it, even if your systems aren't purely reactive or functional. An expert blend of theory and practice, this book presents strong examples you'll return to again and again as you apply these principles to your own projects. What's Inside Real-world libraries and frameworks Establish meaningful reliability guarantees Isolate domain logic from side effects Introduction to reactive design patterns About the Reader Readers should be comfortable with functional programming and traditional domain modeling. Examples use the Scala language. About the Author Software architect Debasish Ghosh was an early adopter of reactive design using Scala and Akka. He's the author of DSLs in Action, published by Manning in 2010. Table of Contents Functional domain modeling: an introduction Scala for functional domain models Designing functional domain models Functional patterns for domain models Modularization of domain models Being reactive Modeling with reactive streams Reactive persistence and event sourcing Testing your domain model Summary - core thoughts and principles *Hands-On Spring Security 5 for Reactive Applications* Oct 02 2022 Secure your Java applications by integrating the Spring Security framework in your code Key Features Provide authentication, authorization and other security features for Java applications. Learn how to secure microservices, cloud, and serverless applications easily Understand the code behind the implementation of various security features Book Description Security is one of the most vital concerns for any organization. The complexity of an application is compounded when you need to integrate security with existing code, new technology, and other frameworks. This book will show you how to effectively write Java code that is robust and easy to maintain. Hands-On Spring Security 5 for Reactive Applications starts with the essential concepts of reactive programming, Spring Framework, and Spring Security. You will then learn about a variety of authentication mechanisms and how to integrate them easily with the

Spring MVC application. You will also understand how to achieve authorization in a Spring WebFlux application using Spring Security. You will be able to explore the security configurations required to achieve OAuth2 for securing REST APIs and integrate security in microservices and serverless applications. This book will guide you in integrating add-ons that will add value to any Spring Security module. By the end of the book, you will be proficient at integrating Spring Security in your Java applications. What you will learn: Understand how Spring Framework and Reactive application programming are connected; Implement easy security configurations with Spring Security expressions; Discover the relationship between OAuth2 and OpenID; Connect Secure microservices and serverless applications with Spring Integrate add-ons, such as HDIV, Crypto Module, and CORS support; Apply Spring Security 5 features to enhance your Java reactive applications. Who this book is for: If you are a Java developer who wants to improve application security, then this book is for you. A basic understanding of Spring, Spring Security framework, and reactive applications is required to make the most of the book.

**Domain-driven Design** Feb 23 2022 "Domain-Driven Design" incorporates numerous examples in Java-case studies taken from actual projects that illustrate the application of domain-driven design to real-world software development.

**Implementing Domain-driven Design** Mar 07 2023 Vaughn Vernon presents concrete and realistic domain-driven design (DDD) techniques through examples from familiar domains, such as a Scrum-based project management application that integrates with a collaboration suite and security provider. Each principle is backed up by realistic Java examples, and all content is tied together by a single case study of a company charged with delivering a set of advanced software systems with DDD.

Feedback Systems Mar 03 2020 The essential introduction to the principles and applications of feedback systems—now fully revised and expanded. This textbook covers the mathematics needed to model, analyze, and design feedback systems. Now more user-friendly than ever, this revised and expanded edition of Feedback Systems is a one-volume resource for students and researchers in mathematics and engineering. It has applications across a range of disciplines that utilize feedback in physical, biological, information, and economic systems. Karl Åström and Richard Murray use techniques from physics, computer science, and operations research to introduce control-oriented modeling. They begin with state space tools for analysis and design, including stability of solutions, Lyapunov functions, reachability, state feedback observability, and estimators. The matrix exponential plays a central role in the analysis of linear control systems, allowing a concise development of many of the key concepts for this class of models. Åström and Murray then develop and explain tools in the frequency domain, including transfer functions, Nyquist analysis, PID control, frequency domain design, and robustness. Features a new chapter on design principles and tools, illustrating the types of problems that can be solved using feedback. Includes a new chapter on fundamental

limits and new material on the Routh-Hurwitz criterion and root locus plots Provides exercises at the end of every chapter Comes with an electronic solutions manual An ideal textbook for undergraduate and graduate students Indispensable for researchers seeking a self-contained resource on control theory

**Hands-On Domain-Driven Design with .NET Core** Jul 19 2021 Solve complex business problems by understanding users better, finding the right problem to solve, and building lean event-driven systems to give your customers what they really want Key Features Apply DDD principles using modern tools such as EventStorming, Event Sourcing, and CQRS Learn how DDD applies directly to various architectural styles such as REST, reactive systems, and microservices Empower teams to work flexibly with improved services and decoupled interactions Book Description Developers across the world are rapidly adopting DDD principles to deliver powerful results when writing software that deals with complex business requirements. This book will guide you in involving business stakeholders when choosing the software you are planning to build for them. By figuring out the temporal nature of behavior-driven domain models, you will be able to build leaner, more agile, and modular systems. You'll begin by uncovering domain complexity and learn how to capture the behavioral aspects of the domain language. You will then learn about EventStorming and advance to creating a new project in .NET Core 2.1; you'll also and write some code to transfer your events from sticky notes to C#. The book will show you how to use aggregates to handle commands and produce events. As you progress, you'll get to grips with Bounded Contexts, Context Map, Event Sourcing, and CQRS. After translating domain models into executable C# code, you will create a frontend for your application using Vue.js. In addition to this, you'll learn how to refactor your code and cover event versioning and migration essentials. By the end of this DDD book, you will have gained the confidence to implement the DDD approach in your organization and be able to explore new techniques that complement what you've learned from the book. What you will learn Discover and resolve domain complexity together with business stakeholders Avoid common pitfalls when creating the domain model Study the concept of Bounded Context and aggregate Design and build temporal models based on behavior and not only data Explore benefits and drawbacks of Event Sourcing Get acquainted with CQRS and to-the-point read models with projections Practice building one-way flow UI with Vue.js Understand how a task-based UI conforms to DDD principles Who this book is for This book is for .NET developers who have an intermediate level understanding of C#, and for those who seek to deliver value, not just write code. Intermediate level of competence in JavaScript will be helpful to follow the UI chapters.

**Mastering Shiny** Sep 20 2021 Master the Shiny web framework—and take your R skills to a whole new level. By letting you move beyond static reports, Shiny helps you create fully interactive web apps for data analyses. Users will be able to jump between datasets, explore different subsets or facets of the data, run models with parameter values of their choosing, customize

visualizations, and much more. Hadley Wickham from RStudio shows data scientists, data analysts, statisticians, and scientific researchers with no knowledge of HTML, CSS, or JavaScript how to create rich web apps from R. This in-depth guide provides a learning path that you can follow with confidence, as you go from a Shiny beginner to an expert developer who can write large, complex apps that are maintainable and performant. Get started: Discover how the major pieces of a Shiny app fit together Put Shiny in action: Explore Shiny functionality with a focus on code samples, example apps, and useful techniques Master reactivity: Go deep into the theory and practice of reactive programming and examine reactive graph components Apply best practices: Examine useful techniques for making your Shiny apps work well in production

*Functional and Reactive Domain Modeling* May 09 2023 Summary Functional and Reactive Domain Modeling teaches you how to think of the domain model in terms of pure functions and how to compose them to build larger abstractions. Purchase of the print book includes a free eBook in PDF, Kindle, and ePub formats from Manning Publications. About the Technology Traditional distributed applications won't cut it in the reactive world of microservices, fast data, and sensor networks. To capture their dynamic relationships and dependencies, these systems require a different approach to domain modeling. A domain model composed of pure functions is a more natural way of representing a process in a reactive system, and it maps directly onto technologies and patterns like Akka, CQRS, and event sourcing. About the Book Functional and Reactive Domain Modeling teaches you consistent, repeatable techniques for building domain models in reactive systems. This book reviews the relevant concepts of FP and reactive architectures and then methodically introduces this new approach to domain modeling. As you read, you'll learn where and how to apply it, even if your systems aren't purely reactive or functional. An expert blend of theory and practice, this book presents strong examples you'll return to again and again as you apply these principles to your own projects. What's Inside Real-world libraries and frameworks Establish meaningful reliability guarantees Isolate domain logic from side effects Introduction to reactive design patterns About the Reader Readers should be comfortable with functional programming and traditional domain modeling. Examples use the Scala language. About the Author Software architect Debasish Ghosh was an early adopter of reactive design using Scala and Akka. He's the author of DSLs in Action, published by Manning in 2010. Table of Contents Functional domain modeling: an introduction Scala for functional domain models Designing functional domain models Functional patterns for domain models Modularization of domain models Being reactive Modeling with reactive streams Reactive persistence and event sourcing Testing your domain model Summary - core thoughts and principles

**Applied Akka Patterns** Sep 08 2020 When it comes to big data processing, we can no longer ignore concurrency or try to add it in after the fact. Fortunately, the solution is not a new paradigm of development, but rather an old one. With this hands-on guide, Java and Scala developers will learn how to embrace concurrent and distributed applications with the open source Akka

toolkit. You'll learn how to put the actor model and its associated patterns to immediate and practical use. Throughout the book, you'll deal with an analogous workforce problem: how to schedule a group of people across a variety of projects while optimizing their time and skillsets. This example will help you understand how Akka uses actors, streams, and other tools to stitch your application together. Model software that reflects the real world with domain-driven design Learn principles and practices for implementing individual actors Unlock the real potential of Akka with patterns for combining multiple actors Understand the consistency tradeoffs in a distributed system Use several Akka methods for isolating and dealing with failures Explore ways to build systems that support availability and scalability Tune your Akka application for performance with JVM tools and dispatchers

**Domain-Driven Design Distilled** Sep 01 2022 Domain-Driven Design (DDD) software modeling delivers powerful results in practice, not just in theory, which is why developers worldwide are rapidly moving to adopt it. Now, for the first time, there's an accessible guide to the basics of DDD: What it is, what problems it solves, how it works, and how to quickly gain value from it. Concise, readable, and actionable, *Domain-Driven Design Distilled* never buries you in detail—it focuses on what you need to know to get results. Vaughn Vernon, author of the best-selling *Implementing Domain-Driven Design*, draws on his twenty years of experience applying DDD principles to real-world situations. He is uniquely well-qualified to demystify its complexities, illuminate its subtleties, and help you solve the problems you might encounter. Vernon guides you through each core DDD technique for building better software. You'll learn how to segregate domain models using the powerful Bounded Contexts pattern, to develop a Ubiquitous Language within an explicitly bounded context, and to help domain experts and developers work together to create that language. Vernon shows how to use Subdomains to handle legacy systems and to integrate multiple Bounded Contexts to define both team relationships and technical mechanisms. *Domain-Driven Design Distilled* brings DDD to life. Whether you're a developer, architect, analyst, consultant, or customer, Vernon helps you truly understand it so you can benefit from its remarkable power. Coverage includes What DDD can do for you and your organization—and why it's so important The cornerstones of strategic design with DDD: Bounded Contexts and Ubiquitous Language Strategic design with Subdomains Context Mapping: helping teams work together and integrate software more strategically Tactical design with Aggregates and Domain Events Using project acceleration and management tools to establish and maintain team cadence [Building Evolutionary Architectures](#) Nov 10 2020 The software development ecosystem is constantly changing, providing a constant stream of new tools, frameworks, techniques, and paradigms. Over the past few years, incremental developments in core engineering practices for software development have created the foundations for rethinking how architecture changes over time, along with ways to protect important architectural characteristics as it evolves. This practical guide ties those parts



together with a new way to think about architecture and time.

**Architecture Patterns with Python** Mar 27 2022 As Python continues to grow in popularity, projects are becoming larger and more complex. Many Python developers are now taking an interest in high-level software design patterns such as hexagonal/clean architecture, event-driven architecture, and the strategic patterns prescribed by domain-driven design (DDD). But translating those patterns into Python isn't always straightforward. With this hands-on guide, Harry Percival and Bob Gregory from MADE.com introduce proven architectural design patterns to help Python developers manage application complexity—and get the most value out of their test suites. Each pattern is illustrated with concrete examples in beautiful, idiomatic Python, avoiding some of the verbosity of Java and C# syntax. Patterns include: Dependency inversion and its links to ports and adapters (hexagonal/clean architecture) Domain-driven design's distinction between entities, value objects, and aggregates Repository and Unit of Work patterns for persistent storage Events, commands, and the message bus Command-query responsibility segregation (CQRS) Event-driven architecture and reactive microservices

*Scala Reactive Programming* Jun 17 2021 Build fault-tolerant, robust, and distributed applications in Scala Key Features - Understand and use the concepts of reactive programming to build distributed systems running on multiple nodes. - Learn how reactive architecture reduces complexity throughout the development process. - Get to grips with functional reactive programming and Reactive Microservices. Book Description Reactive programming is a scalable, fast way to build applications, and one that helps us write code that is concise, clear, and readable. It can be used for many purposes such as GUIs, robotics, music, and others, and is central to many concurrent systems. This book will be your guide to getting started with Reactive programming in Scala. You will begin with the fundamental concepts of Reactive programming and gradually move on to working with asynchronous data streams. You will then start building an application using Akka Actors and extend it using the Play framework. You will also learn about reactive stream specifications, event sourcing techniques, and different methods to integrate Akka Streams into the Play Framework. This book will also take you one step forward by showing you the advantages of the Lagom framework while working with reactive microservices. You will also learn to scale applications using multi-node clusters and test, secure, and deploy your microservices to the cloud. By the end of the book, you will have gained the knowledge to build robust and distributed systems with Scala and Akka. What you will learn Understand the fundamental principles of Reactive and Functional programming Develop applications utilizing features of the Akka framework Explore techniques to integrate Scala, Akka, and Play together Learn about Reactive Streams with real-time use cases Develop Reactive Web Applications with Play, Scala, Akka, and Akka Streams Develop and deploy Reactive microservices using the Lagom framework and ConductR Who this book is for This book is for Scala developers who would like to build fault-tolerant, scalable

distributed systems. No knowledge of Reactive programming is required.

**Reactive Messaging Patterns with the Actor Model** Dec 04 2022 USE THE ACTOR MODEL TO BUILD SIMPLER SYSTEMS WITH BETTER PERFORMANCE AND SCALABILITY Enterprise software development has been much more difficult and failure-prone than it needs to be. Now, veteran software engineer and author Vaughn Vernon offers an easier and more rewarding method to succeeding with Actor model. Reactive Messaging Patterns with the Actor Model shows how the reactive enterprise approach, Actor model, Scala, and Akka can help you overcome previous limits of performance and scalability, and skillfully address even the most challenging non-functional requirements. Reflecting his own cutting-edge work, Vernon shows architects and developers how to translate the longtime promises of Actor model into practical reality. First, he introduces the tenets of reactive software, and shows how the message-driven Actor model addresses all of them—making it possible to build systems that are more responsive, resilient, and elastic. Next, he presents a practical Scala bootstrap tutorial, a thorough introduction to Akka and Akka Cluster, and a full chapter on maximizing performance and scalability with Scala and Akka. Building on this foundation, you'll learn to apply enterprise application and integration patterns to establish message channels and endpoints; efficiently construct, route, and transform messages; and build robust systems that are simpler and far more successful. Coverage Includes How reactive architecture replaces complexity with simplicity throughout the core, middle, and edges The characteristics of actors and actor systems, and how Akka makes them more powerful Building systems that perform at scale on one or many computing nodes Establishing channel mechanisms, and choosing appropriate channels for each application and integration challenge Constructing messages to clearly convey a sender's intent in communicating with a receiver Implementing a Process Manager for your Domain-Driven Designs Decoupling a message's source and destination, and integrating appropriate business logic into its router Understanding the transformations a message may experience in applications and integrations Implementing persistent actors using Event Sourcing and reactive views using CQRS Find unique online training on Domain-Driven Design, Scala, Akka, and other software craftsmanship topics using the [for{comprehension}](http://forcomprehension.com) website at [forcomprehension.com](http://forcomprehension.com).

Practical Domain-Driven Design in Enterprise Java Oct 10 2020 See how Domain-Driven Design (DDD) combines with Jakarta EE MicroProfile or Spring Boot to offer a complete suite for building enterprise-grade applications. In this book you will see how these all come together in one of the most efficient ways to develop complex software, with a particular focus on the DDD process. Practical Domain-Driven Design in Enterprise Java starts by building out the Cargo Tracker reference application as a monolithic application using the Jakarta EE platform. By doing so, you will map concepts of DDD (bounded contexts, language, and aggregates) to the corresponding available tools (CDI, JAX-RS, and JPA) within the Jakarta EE platform. Once you have

completed the monolithic application, you will walk through the complete conversion of the monolith to a microservices-based architecture, again mapping the concepts of DDD and the corresponding available tools within the MicroProfile platform (config, discovery, and fault tolerance). To finish this section, you will examine the same microservices architecture on the Spring Boot platform. The final set of chapters looks at what the application would be like if you used the CQRS and event sourcing patterns. Here you'll use the Axon framework as the base framework. What You Will Learn Discover the DDD architectural principles and use the DDD design patterns Use the new Eclipse Jakarta EE platform Work with the Spring Boot framework Implement microservices design patterns, including context mapping, logic design, entities, integration, testing, and security Carry out event sourcing Apply CQRS Who This Book Is For Junior developers intending to start working on enterprise Java; senior developers transitioning from monolithic- to microservices-based architectures; and architects transitioning to a DDD philosophy of building applications.

**Secure by Design** Apr 03 2020 Summary Secure by Design teaches developers how to use design to drive security in software development. This book is full of patterns, best practices, and mindsets that you can directly apply to your real world development. You'll also learn to spot weaknesses in legacy code and how to address them. About the technology Security should be the natural outcome of your development process. As applications increase in complexity, it becomes more important to bake security-mindedness into every step. The secure-by-design approach teaches best practices to implement essential software features using design as the primary driver for security. About the book Secure by Design teaches you principles and best practices for writing highly secure software. At the code level, you'll discover security-promoting constructs like safe error handling, secure validation, and domain primitives. You'll also master security-centric techniques you can apply throughout your build-test-deploy pipeline, including the unique concerns of modern microservices and cloud-native designs. What's inside Secure-by-design concepts Spotting hidden security problems Secure code constructs Assessing security by identifying common design flaws Securing legacy and microservices architectures About the reader Readers should have some experience in designing applications in Java, C#, .NET, or a similar language. About the author Dan Bergh Johnsson, Daniel Deogun, and Daniel Sawano are acclaimed speakers who often present at international conferences on topics of high-quality development, as well as security and design.

**Protein Sensors and Reactive Oxygen Species** May 05 2020 This volume of Methods in Enzymology is a companion to Volume 347 and addresses direct sensing of reactive oxygen species and related free radicals by thiol enzymes and proteins.

**Patterns, Principles, and Practices of Domain-Driven Design** Nov 03 2022 Methods for managing complex software construction following the practices, principles and patterns of Domain-Driven Design with code examples in C# This book

presents the philosophy of Domain-Driven Design (DDD) in a down-to-earth and practical manner for experienced developers building applications for complex domains. A focus is placed on the principles and practices of decomposing a complex problem space as well as the implementation patterns and best practices for shaping a maintainable solution space. You will learn how to build effective domain models through the use of tactical patterns and how to retain their integrity by applying the strategic patterns of DDD. Full end-to-end coding examples demonstrate techniques for integrating a decomposed and distributed solution space while coding best practices and patterns advise you on how to architect applications for maintenance and scale. Offers a thorough introduction to the philosophy of DDD for professional developers Includes masses of code and examples of concept in action that other books have only covered theoretically Covers the patterns of CQRS, Messaging, REST, Event Sourcing and Event-Driven Architectures Also ideal for Java developers who want to better understand the implementation of DDD

*DSLs in Action* Oct 22 2021 Your success—and sanity—are closer at hand when you work at a higher level of abstraction, allowing your attention to be on the business problem rather than the details of the programming platform. Domain Specific Languages—"little languages" implemented on top of conventional programming languages—give you a way to do this because they model the domain of your business problem. *DSLs in Action* introduces the concepts and definitions a developer needs to build high-quality domain specific languages. It provides a solid foundation to the usage as well as implementation aspects of a DSL, focusing on the necessity of applications speaking the language of the domain. After reading this book, a programmer will be able to design APIs that make better domain models. For experienced developers, the book addresses the intricacies of domain language design without the pain of writing parsers by hand. The book discusses DSL usage and implementations in the real world based on a suite of JVM languages like Java, Ruby, Scala, and Groovy. It contains code snippets that implement real world DSL designs and discusses the pros and cons of each implementation. 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. What's Inside Tested, real-world examples How to find the right level of abstraction Using language features to build internal DSLs Designing parser/combinator-based little languages

[digitaltutorials.jrn.columbia.edu](http://digitaltutorials.jrn.columbia.edu)