

## Functional Programming Patterns In Scala And Clojure Write Lean Programs For The Jvm Michael Bevilacqua Linn

In large projects, programmers tend to get overwhelmed by their complexity. It can be hard to keep track of all the interdependencies in the code-base and how its state changes on runtime. The solution to these problems is Functional Programming, a paradigm specifically designed to deal with the complexity of software development. Mastering ... 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 Debashish 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

The official book on the Rust programming language, written by the Rust development team at the Mozilla Foundation, fully updated for Rust 2018. The Rust Programming Language is the official book on Rust: an open source systems programming language that helps you write faster, more reliable software. Rust offers control over low-level details (such as memory usage) in combination with high-level ergonomics, eliminating the hassle traditionally associated with low-level languages. The authors of The Rust Programming Language, members of the Rust Core Team, share their knowledge and experience to show you how to take full advantage of Rust's features--from installation to creating robust and scalable programs. You'll begin with basics like creating functions, choosing data types, and binding variables and then move on to more advanced concepts, such as: • Ownership and borrowing, lifetimes, and traits • Using Rust's memory safety guarantees to build fast, safe programs • Testing, error handling, and effective refactoring • Generics, smart pointers, multithreading, trait objects, and advanced pattern matching • Using Cargo, Rust's built-in package manager, to build, test, and document your code and manage dependencies • How best to use Rust's advanced compiler with compiler-led programming techniques You'll find plenty of code examples throughout the book, as well as three chapters dedicated to building complete projects to test your learning: a number guessing game, a Rust implementation of a command line tool, and a multithreaded server. New to this edition: An extended section on Rust macros, an expanded chapter on modules, and appendices on Rust development tools and editions.

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.

Tackle Software Complexity with Domain-Driven Design and F#

Domain Modeling Made Functional

Functional Programming in Kotlin

Build scalable apps with patterns in multithreading, synchronization, and functional programming

Patterns for Practical Reuse and Design

Modern Systems Programming with Scala Native

A book for intermediate to advanced Scala developers. Aimed at those who understand functional effects, referential transparency and the benefits of functional programming to some extent but who are missing some pieces to put all these concepts together to build a large application in a time-constrained manner.Throughout the chapters we will design, architect and develop a complete stateful application serving an API via HTTP, accessing a database and dealing with cached data, using the best practices and best functional libraries available in the Cats ecosystem such as Cats Effect, Fs2, Http4s, Skunk, Refined and others.You will also learn about common design patterns such as managing state, error handling and anti-patterns, all accompanied by clear examples. Furthermore, in the Bonus Chapter, we will dive into some advanced concepts such as MTL and Optics, and will explore Fs2 streams with a few interesting examples.A digital version is also available on LeanPub.

Scala is now an established programming language developed by Martin Oderskey and his team at the EPFL. The name Scala is derived from Sca(lable) La(nguage). Scala is a multi-paradigm language, incorporating object oriented approaches with functional programming. Although some familiarity with standard computing concepts is assumed (such as the idea of compiling a program and executing this compiled from etc.) and with basic procedural language concepts (such as variables and allocation of values to these variables) the early chapters of the book do not assume any familiarity with object orientation nor with functional programming These chapters also step through other concepts with which the reader may not be familiar (such as list processing). From this background, the book provides a practical introduction to both object and functional approaches using Scala. These concepts are introduced through practical experience taking the reader beyond the level of the language syntax to the philosophy and practice of object oriented development and functional programming. Students and those actively involved in the software industry will find this comprehensive introduction to Scala invaluable.

Functional programming languages like F#, Erlang, and Scala are attractingattention as an efficient way to handle the new requirements for programmingmulti-processor and high-availability applications. Microsoft's new F# is a truefunctional language and C# uses functional language features for LINQ andother recent advances. Real-World Functional Programming is a unique tutorial that explores thefunctional programming model through the F# and C# languages. The clearlypresented ideas and examples teach readers how functional programming differsfrom other approaches. It explains how ideas look in F#-a functionallanguage-as well as how they can be successfully used to solve programmingproblems in C# Readers build on what they know about .NET and learn wherea functional approach makes the most sense and how to apply it effectively inthose cases. The reader should have a good working knowledge of C#. No prior exposure toF# 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.

Take your Haskell and functional programming skills to the next level by exploring new idioms and design patterns About This Book Explore Haskell on a higher level through idioms and patterns Get an in-depth look into the three strongholds of Haskell: higher-order functions, the Type system, and Lazy evaluation Expand your understanding of Haskell and functional programming, one line of executable code at a time Who This Book Is For If you're a Haskell programmer with a firm grasp of the basics and ready to move more deeply into modern idiomatic Haskell programming, then this book is for you. What You Will Learn Understand the relationship between the "Gang of Four" OOP Design Patterns and Haskell Try out three ways of Streaming I/O: imperative, Lazy, and Iteratee based Explore the pervasive pattern of Composition: from function composition through to high-level composition with Lenses Synthesize Functor, Applicative, Arrow and Monad in a single conceptual framework Follow the grand arc of Fold and Map on lists all the way to their culmination in Lenses and Generic Programming Get a taste of Type-level programming in Haskell and how this relates to dependently-typed programming Retrace the evolution, one key language extension at a time, of the Haskell Type and Kind systems Place the elements of modern Haskell in a historical framework In Detail Design patterns and idioms can widen our perspective by showing us where to look, what to look at, and ultimately how to see what we are looking at. At their best, patterns are a shorthand method of communicating better ways to code (writing less, more maintainable, and more efficient code). This book starts with Haskell 98 and through the lens of patterns and idioms investigates the key advances and programming styles that together make "modern Haskell". Your journey begins with the three pillars of Haskell. Then you'll experience the problem with Lazy I/O, together with a solution. You'll also trace the hierarchy formed by Functor, Applicative, Arrow, and Monad. Next you'll explore how Fold and Map are generalized by Foldable and Traversable, which in turn is unified in a broader context by functional Lenses. You'll delve more deeply into the Type system, which will prepare you for an overview of Generic programming. In conclusion you go to the edge of Haskell by investigating the Kind system and how this relates to Dependently-typed programming. Style and approach Using short pieces of executable code, this guide gradually explores the broad pattern landscape of modern Haskell. Ideas are presented in their historical context and arrived at through intuitive derivations, always with a focus on the problems they solve.

The Science of Functional Programming (draft version)

Hands-on Scala Programming: Learn Scala in a Practical, Project-Based Way

Haskell Design Patterns

Build Your Functional Skills One Idea at a Time

Write Lean Programs for the JVM

Exploring Clojure, Elixir, Haskell, Scala, and Swift

*This practically-focused textbook presents a concise tutorial on data structures and algorithms using the object-functional language Scala. The material builds upon the foundation established in the title Programming with Scala: Language Exploration by the same author, which can be treated as a companion text for those less familiar with Scala. Topics and features: discusses data structures and algorithms in the form of design patterns; covers key topics on arrays, lists, stacks, queues, hash tables, binary trees, sorting, searching, and graphs; describes examples of complete and running applications for each topic; presents a functional approach to implementations for data structures and algorithms (excepting arrays); provides numerous challenge exercises (with solutions), encouraging the reader to take existing solutions and improve upon them; offers insights from the author's extensive industrial experience; includes a glossary, and an appendix supplying an overview of discrete mathematics. Highlighting the techniques and skills necessary to quickly derive solutions to applied problems, this accessible text will prove invaluable to time-pressured students and professional software engineers.*

*Save time and trouble when using Scala to build object-oriented, functional, and concurrent applications. With more than 250 ready-to-use recipes and 700 code examples, this comprehensive cookbook covers the most common problems you'll encounter when using the Scala language, libraries, and tools. It's ideal not only for experienced Scala developers, but also for programmers learning to use this JVM language. Author Alvin Alexander (creator of DevDaily.com) provides solutions based on his experience using Scala for highly scalable, component-based applications that support concurrency and distribution. Packed with real-world scenarios, this book provides recipes for: Strings, numeric types, and control structures Classes, methods, objects, traits, and packaging Functional programming in a variety of situations Collections covering Scala's wealth of classes and methods Concurrency, using the Akka Actors library Using the Scala REPL and the Simple Build Tool (SBT) Web services on both the client and server sides Interacting with SQL and NoSQL databases Best practices in Scala development*

*Hands-on Scala teaches you how to use the Scala programming language in a practical, project-based fashion. This book is designed to quickly teach an existing programmer everything needed to go from "hello world" to building production applications like interactive websites, parallel web crawlers, and distributed systems in Scala. In the process you will learn how to use the Scala language to solve challenging problems in an elegant and intuitive manner.*

*Why learn Scala? You don't need to be a data scientist or distributed computing expert to appreciate this object-oriented functional programming language. This practical book provides a comprehensive yet approachable introduction to the language, complete with syntax diagrams, examples, and exercises. You'll start with Scala's core types and syntax before diving into higher-order functions and immutable data structures. Author Jason Swartz demonstrates why Scala's concise and expressive syntax make it an ideal language for Ruby or Python developers who want to improve their craft, while its type safety and performance ensures that it's stable and fast enough for any application. Learn about the core data types, literals, values, and variables Discover how to think and write in expressions, the foundation for Scala's syntax Write higher-order functions that accept or return other functions Become familiar with immutable data structures and easily transform them with type-safe and declarative operations Create custom infix operators to simplify existing operations or even to start your own domain-specific language Build classes that compose one or more traits for full reusability, or create new functionality by mixing them in at instantiation*

*The Rust Programming Language (Covers Rust 2018)*

*Write Lean, High-Performance Code without the JVM*

*A Beginner's Guide to Scala, Object Orientation and Functional Programming*

*Build real world projects using popular Scala frameworks like Play, Akka, and Spark*

*Paradigm Over Syntax*

*Scala for Java Developers*

A step-by-step guide in building high-performance scalable applications with the latest features of Scala. Key FeaturesDevelop a strong foundation in functional programming and Scala's Standard Library (STL)Get a detailed coverage of Lightbend Lagom—the latest microservices framework from LightbendUnderstand the Akka framework and learn event-based Programming with Scala Book Description The second version of Scala has undergone multiple changes to support features and library implementations. Scala 2.13, with its main focus on modularizing the standard library and simplifying collections, brings with it a host of updates. Learn Scala Programming addresses both technical and architectural changes to the redesigned standard library and collections, along with covering in-depth type systems and first-level support for functions. You will discover how to leverage implicits as a primary mechanism for building type classes and look at different ways to test Scala code. You will also learn about abstract building blocks used in functional programming, giving you sufficient understanding to pick and use any existing functional programming library out there. In the concluding chapters, you will explore reactive programming by covering the Akka framework and reactive streams. By the end of this book, you will have built microservices and learned to implement them with the Scala and Lagom framework. What you will learnAcquaint yourself with the new standard library of Scala 2.13Get to grips with the Grok functional paradigmsGet familiar with type system to express domain constraintsUnderstand the actor model and different Akka librariesGrasp the concept of building microservices using Lagom frameworkDeep dive into property-based testing and its practical applicationsWho this book is for This book is for beginner to intermediate level Scala developers who would like to advance and gain knowledge of the intricacies of the Scala language, expand their functional programming tools, and explore actor-based concurrency models.

Functional Programming Patterns in Scala and ClojureWrite Lean Programs for the JVM

Write efficient, clean, and reusable code with Scala About This Book Unleash the power of Scala and apply it in the real world Increase your efficiency by leveraging the power of Creational, Structural, Behavioural, and Functional design patterns Build object oriented and functional applications quickly and effectively Who This Book Is For If you want to increase your understanding of Scala and apply it to real-life application development, then this book is for you. We've also designed the book to be used as a quick reference guide while creating applications. Previous Scala programming knowledge is expected. What You Will Learn Immerse yourself in industry-standard design patterns—structural, creational, and behavioral—to create extraordinary applications Feel the power of traits and their application in Scala Implement abstract and self types and build clean design patterns Build complex entity relationships using structural design patterns Create applications faster by applying functional design patterns In Detail Scala has become increasingly popular in many different IT sectors. The language is exceptionally feature-rich which helps developers write less code and get faster results. Design patterns make developer's lives easier by helping them write great software that is easy to maintain, runs efficiently and is valuable to the company or people concerned. You will learn about the various features of Scala and be able to apply well-known, industry-proven design patterns in your work. The book starts off by focusing on some of the most interesting features of Scala while using practical real-world examples. We will also cover the popular "Gang of Four" design patterns and show you how to incorporate functional patterns effectively. By the end of this book, you will have enough knowledge and understanding to quickly assess problems and come up with elegant solutions. Style and approach The design patterns in the book will be explained using real-world, step-by-step examples. For each design pattern, there will be hints about when to use it and when to look for something more suitable. This book can also be used as a practical guide, showing you how to leverage design patterns effectively.

Presents an introduction to the Scala programming language which is an abbreviated version of object-orientated programming combined with the power of concurrency capable of running on the Java Virtual Machine.

Data Structures and Algorithms with Scala

A Practitioner's Approach with Emphasis on Functional Programming

Scala in Depth

Learn Scala Programming

Functional Programming Patterns in Scala and Clojure

A comprehensive guide covering functional and reactive programming with Scala 2.13, Akka, and Lagom

Scala is a new and exciting programming language that is a hybrid between object oriented languages such as Java and functional languages such as Haskell. As such it has its own programming idioms and development styles. Scala Design Patterns looks at how code reuse can be successfully achieved in Scala. A major aspect of this is the design patterns in terms of Scala and its language structures (that is the use of Traits, Classes, Objects and Functions). It includes an exploration of functional design patterns and considers how these can be interpreted in Scala's uniquely hybrid style. A key aspect of the book is the many code examples that accompany each design pattern the design pattern but also to explore powerful and flexible Scala language features. Including numerous source code examples, this book will be of value to professionals and practitioners working in the field of software engineering.

Packed with examples and exercises. Get Programming with Scala is perfect starting point for developers with some OO knowledge who want to learn this multi-style programming language for the JVM, and pick up a few FP skills along the way. Master Scala, and you'll be well-equipped to match your programming approach to the type of examples and exercises, Get Programming with Scala is perfect starting point for developers with some OO knowledge who want to learn this multi-style programming language for the JVM, and pick up a few FP skills along the way. Master Scala, and you'll be well-equipped to match your programming approach to the type of problem you

Scala teaches you the core skills you'll need to code with Scala. You'll start by reviewing OOP concepts in the Scala language. Then, you'll gradually open up the world of functional programming. You'll explore functions and types and learn how to combine them to create powerful, flexible abstractions. Scala can be daunting at first, especially when you're new to the language. Fortunately, with the examples and exercises in this book, you'll get over the initial learning hump quickly and start doing interesting projects before you know it! Purchase of the print book includes a free eBook in PDF, Kindle, and ePub formats from Manning Publications.

Summary Scala in Action is a comprehensive tutorial that introduces Scala through clear explanations and numerous hands-on examples. Because Scala is a rich and deep language, it can be daunting to absorb all the new concepts at once. This book takes a "how-to" approach, explaining language concepts as you explore familiar programming concepts and apply them to day work. About the Technology Scala runs on the JVM and combines object-orientation with functional programming. It's designed to produce succinct, type-safe code, which is crucial for enterprise applications. Scala implements Actor-based concurrency through the amazing Akka framework, so you can avoid Java's messy threading while still getting the performance benefits of the JVM. This book is a comprehensive tutorial that introduces the language through clear explanations and numerous hands-on examples. It takes a "how to" approach, explaining language concepts as you explore familiar programming tasks. You'll tackle concurrent programming in Akka, learn to work with Scala and Spring, and learn how to use Scala's powerful productivity tools. You'll learn both the language and how to use it. Experience with Java is helpful but not required. Ruby and Python programmers will also find this book accessible. What's Inside A Scala tutorial How to use Java and Scala open source libraries How to use SBT Test-driven development Debugging Updated for Scala 2.10 Purchase of the print book includes a free eBook in PDF, Kindle, and ePub formats from Manning Publications. About the Author Nilanjan Raychaudhuri is a skilled developer, speaker, and an avid polyglot programmer who works with Scala on production systems. Table of Contents PART 1 SCALA: THE BASICS Why Scala? Getting started OOP in Scala Having fun with functional data structures and Akka PART 2 WORKING WITH SCALA Building web applications in functional style Connecting to a database Building scalable and extensible components Concurrency programming in Scala Building confidence with testing PART 3 ADVANCED STEPS Interoperability between Scala and Java Scalable and distributed applications using Akka and Spring Access the power of bare-metal systems programming with Scala Native, an ahead-of-time Scala compiler. Without the baggage of legacy frameworks and virtual machines, Scala Native lets you re-imagine how your programs interact with your operating system. Compile Scala code down to native machine instructions; seamlessly invoke operations like networking and IO; control pointers, arrays, and other memory management techniques for extreme performance; and enjoy instant start-up times. Skip the JVM and improve your code performance by getting close to the metal. Developers generally build systems on top of the work of those who came before, accumulating layer upon layer of complexity. Scala Native offers a rare opportunity to remove layers. Without the JVM, Scala Native uses POSIX and ANSI C APIs to build concise, expressive programs that run unusually close to bare metal. Scala Native compiles Scala code down to native machine instructions instead of JVM bytecode. It starts up fast, without the sluggish warm-up phase that's common for JVM-based programs. Scala Native programs can seamlessly invoke operating system APIs for low-level networking and IO. And Scala Native lets you control pointers, arrays, and other memory layout types for extreme performance. Write practical, bare-metal code with Scala Native, step by step. Understand the foundations of systems programming, including pointers, arrays, and sockets. Use the UNIX socket API to write network client and server programs without the sort of frameworks higher-level languages rely on. Put all the pieces together to design and implement a modern, asynchronous microservice-style HTTP framework from scratch. Take advantage of Scala Native's clean, modern syntax to write lean, high-performance code. Purchase of the print book includes a free eBook in PDF, Kindle, and ePub formats from Manning Publications. About the Author Need: A modern Windows, Mac OS, or Linux system capable of running Docker. All code examples in the book are designed to run on a portable Docker-based build environment that runs anywhere. If you don't have Docker yet, see the Appendix for instructions on how to get it.

Programming in Scala

Scala Cookbook

Programming Scala

Mastering Functional Programming

Harnessing the Power Of Java 8 Lambda Expressions

Scala in Action

Software development today is embracing functional programming (FP), whether it's for writing concurrent programs or for managing Big Data. Where does that leave Java developers? This concise book offers a pragmatic, approachable introduction to FP for Java developers or anyone who uses an object-oriented language. Dean Wampler, Java expert and author of Programming Scala (O'Reilly), shows you how to apply FP principles such as immutability, avoidance of side-effects, and higher-order functions to your Java code. Each chapter provides exercises to help you practice what you've learned. Once you grasp the benefits of functional programming, you'll discover that it improves all of the code you write. Learn basic FP principles and apply them to object-oriented programming Discover how FP is more concise and modular than OOP Get useful FP lessons for your Java type design—such as avoiding nulls Design data structures and algorithms using functional programming principles Write concurrent programs using the Actor model and software transactional memory Use functional libraries and frameworks for Java—and learn where to go next to deepen your functional programming skills

This book is intended for the intermediate Scala programmer who is interested in functional programming and works mainly on the web service backend side. Ideally she has experience with libraries like Akka HTTP and Slick which are in heavy use in that area. However maybe you have wondered if we can't do better even though aforementioned projects are battle tested and proven. The answer to this can be found in this book which is intended to be read from cover to cover in the given order. Within the book the following libraries will be used: Cats, Cats Effect, http4s, Doobie, Refined, fs2, tapir, Monocle and probably others. -) This edition includes a chapter about migrating the project to Scala 3. Which includes all the nasty issues that we tend to run into if we touch code after a longer time. Code and book source can be found in the author's github account.

Presents an introduction to the new programming language for the Java Platform.

Discover unique features and powerful capabilities of Scala Programming as you build projects in a wide range of domains Key FeaturesDevelop a range of Scala projects from web applications to big data analysisLeverage full power of modern web programming using Play FrameworkBuild real-time data pipelines in Scala with a Bitcoin transaction analysis appBook Description Scala is a type-safe JVM language that incorporates object-oriented and functional programming (OOP and FP) aspects. This book gets you started with essentials of software development by guiding you through various aspects of Scala programming, helping you bridge the gap between learning and implementing. You will learn about the unique features of Scala through diverse applications and experience simple yet powerful approaches for software development. Scala Programming Projects will help you build a number of applications, beginning with simple projects, such as a financial independence calculator, and advancing to other projects, such as a shopping application and a Bitcoin transaction analyzer. You will be able to use various Scala features, such as its OOP and FP capabilities, and learn how to write concise, reactive, and concurrent applications in a type-safe manner. You will also learn how to use top-notch libraries such as Akka and Play and integrate Scala apps with Kafka, Spark, and Zeppelin, along with deploying applications on a cloud platform. By the end of the book, you will not only know the ins and outs of Scala, but you will also be able to apply it to solve a variety of real-world problems What you will learnBuild, test, and package code using Scala Build ToolDecompose code into functions, classes, and packages for maintainabilityImplement the functional programming capabilities of ScalaDevelop a simple CRUD REST API using the Play frameworkAccess a relational database using SlickDevelop a dynamic web UI using Scala.jsSource streaming data using Spark Streaming and write a Kafka producerUse Spark and Zeppelin to analyze dataWho this book is for If you are an amateur programmer who wishes to learn how to use Scala, this book is for you. Knowledge of Java will be beneficial, but not necessary, to understand the concepts covered in this book.

Build scalable, functional reactive microservices with Akka, Play, and Lagom

Recipes for Object-Oriented and Functional Programming

Refactoring to Patterns

Learning Scala

Tools for Better Concurrency, Abstraction, and Agility

Scalability = Functional Programming + Objects

**Behind every programming language lies a vision of how programs should be built. The vision behind Clojure is of a radically simple language framework holding together a sophisticated collection of programming features. Learning Clojure involves much more than just learning the mechanics of the language. To really get Clojure you need to understand the ideas underlying this structure of framework and features. You need this book: an accessible introduction to Clojure that focuses on the ideas behind the language as well as the practical details of writing code. Clojure attracts developers on the cutting edge and is arguably the best language for learning to program in the functional style without compromise. But this comes with a steep learning curve. Getting Clojure directly addresses this by teaching you how to think functionally as it teaches you the language. You'll learn about Clojure's powerful data structures and high-level functions, but you'll also learn what it means for a language to be functional, and how to think in Clojure's functional way. Each chapter of Getting Clojure takes a feature or two or three from the language, explains the syntax and the mechanics behind that feature so that you can make it work before digging into the deeper questions: What is the thinking behind the feature? And how does it fit in with the rest of the language? In Getting Clojure you'll learn Clojure's very simple syntax, but you'll also learn why that syntax is integral the way the language is constructed. You'll discover that most data structures in Clojure are immutable, but also why that leads to more reliable programs. And you'll see how easy it is to write Clojure functions and also how you can use those functions to build complex and capable systems. With real-world examples of how working Clojure programmers use the language, Getting Clojure will help you see the challenges of programming through the eye of experienced Clojure developers. What You Need: You will need to some background in programming. To follow along with the examples in the book, you will need Java 6 or new, Clojure 1.8 or 1.9, and Leiningen 2.**

**A definitive guide to mastering and implementing concurrency patterns in your applications Key FeaturesBuild scalable apps with patterns in multithreading, synchronization, and functional programmingExplore the parallel programming and multithreading techniques to make the code run fasterEfficiently use the techniques outlined to build reliable applicationsBook Description Selecting the correct concurrency architecture has a significant impact on the design and performance of your applications. This book explains how to leverage the different characteristics of parallel architecture to make your code faster and more efficient. To start with, you'll understand the basic concurrency concepts and explore patterns around explicit locking, lock free programming, futures & actors. Then, you'll get insights into different concurrency models and parallel algorithms and put them to practice in different scenarios to realize your application's true potential. We'll take you through multithreading design patterns, such as master, slave, leader, follower, map-reduce, and monitor, also helping you to learn hands-on coding using these patterns. Once you've grasped all of this, you'll move on to solving problems using synchronizer patterns. You'll discover the rationale for these patterns in distributed & parallel applications, followed by studying how future composition, immutability and the monadic flow help create more robust code. Toward the end of the book, you'll learn about the actor paradigm and actor patterns - the message passing concurrency paradigm. What you will learnExplore parallel architecture Get acquainted with concurrency models Internalize design themes by implementing multithreading patterns Get insights into concurrent design patterns Discover design principles behind many java threading abstractions Work with functional concurrency patternsWho this book is for This is a must-have guide for developers who want to learn patterns to build scalable and high-performing apps. It's assumed that you already have a decent level of programming knowledge.**

**Summary Functional Programming in Java teaches Java developers how to incorporate the most powerful benefits of functional programming into new and existing Java code. You'll learn to think functionally about coding tasks in Java and use FP to make your applications easier to understand, optimize, maintain, and scale. Purchase of the print book includes a free eBook in PDF, Kindle, and ePub formats from Manning Publications. About the Technology Here's a bold statement: learn functional programming and you'll be a better Java developer. Fortunately, you don't have to master every aspect of FP to get a big payoff. If you take in a few core principles, you'll see an immediate boost in the scalability, readability, and maintainability of your code. And did we mention that you'll have fewer bugs? Let's get started! About the Book Functional Programming in Java teaches you how to incorporate the powerful benefits of functional programming into new and existing Java code. This book uses easy-to-grasp examples, exercises, and illustrations to teach core FP principles such as referential transparency, immutability, persistence, and laziness. Along the way, you'll discover which of the new functionally inspired features of Java 8 will help you most. What's Inside Writing code that's easier to read and reason about Safer concurrent and parallel programming Handling errors without exceptions Java 8 features like lambdas, method references, and functional interfaces About the Reader Written for Java developers with no previous FP experience. About the Author Pierre-Yves Saumont is a seasoned Java developer with three decades of experience designing and building enterprise software. He is an R&D engineer at Alcatel-Lucent Submarine Networks. Table of Contents What is functional programming? Using functions in Java Making Java more functional Recursion, coreursion, and memoization Data handling with lists Dealing with optional data Handling errors and exceptions Advanced list handling Working with laziness More data handling with trees Solving real problems with advanced trees Handling state mutation in a functional way Functional input/output Sharing mutable state with actors Solving common problems functionally**

**Functional Programming in Kotlin is a serious tutorial for programmers looking to learn FP and apply it to the everyday business of coding. Based on the bestselling Functional Programming in Scala, this book guides intermediate Java and Kotlin programmers from basic techniques to advanced topics in a logical, concise, and clear progression. In this authoritative guide, you'll take on the challenge of learning functional programming from first principles, and start writing Kotlin code that's easier to read, easier to reuse, better for concurrency, and less prone to bugs and errors. Functional Programming in Kotlin is a serious tutorial for programmers looking to learn FP and apply it to the everyday business of coding. Based on the bestselling Functional Programming in Scala, this book guides intermediate Java and Kotlin programmers from basic techniques to advanced topics in a logical, concise, and clear progression. In it, you'll find concrete examples and exercises that open up the world of functional programming. The book will deliver practical mastery of FP using Kotlin and a valuable perspective on program design that you can apply to other languages. Purchase of the print book includes a free eBook in PDF, Kindle, and ePub formats from Manning Publications.**

**With examples in F# and C#**

**Get Programming with Scala**

**Concurrent Patterns and Best Practices**

**Practical Functional Programming for the JVM**

**Pure functional HTTP APIs in Scala**

**Functional techniques for sequential and parallel programming with Scala**

**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. Provides a guide to using Scala and Clojure to solve in-depth programming problems.**

**Get up to speed on Scala, the JVM language that offers all the benefits of a modern object model, functional programming, and an advanced type system. Packed with code examples, this comprehensive book shows you how to be productive with the language and ecosystem right away, and explains why Scala is ideal for today's highly scalable, data-centric applications that support concurrency and distribution. This second edition covers recent language features, with new chapters on pattern matching, comprehensions, and advanced functional programming. You'll also learn about Scala's command-line tools, third-party tools, libraries, and language-aware plugins for editors and IDEs. This book is ideal for beginning and advanced Scala developers alike. Program faster with Scala's succinct and flexible syntax Dive into basic and advanced functional programming (FP) techniques Build killer big-data apps, using Scala's functional combinators Use traits for mixin composition and pattern matching for data extraction Learn the sophisticated type system that combines FP and object-oriented programming concepts Explore Scala-specific concurrency tools, including Akka Understand how to develop rich domain-specific languages Learn good design techniques for building scalable and robust Scala applications**

**Save time and trouble building object-oriented, functional, and concurrent applications with Scala. The latest edition of this comprehensive cookbook is packed with more than 250 ready-to-use recipes and 1,000 code examples to help you solve the most common problems when working with Scala 3 and its popular libraries. Scala changes the way you think about programming--and that's a good thing. Whether you're working on web, big data, or distributed applications, this cookbook provides recipes based on real-world scenarios for both experienced Scala developers and programmers just learning to use this JVM language. Author Alvin Alexander includes practical solutions from his experience using Scala for component-based, highly scalable applications that support concurrency and distribution. Recipes cover: Strings, numbers, and control structures Classes, methods, objects, traits, packaging, and imports Functional programming techniques Scala's wealth of collections classes and methods Building and publishing Scala applications with sbt Actors and concurrency with Scala Future and Akka Typed Popular libraries, including Spark, Scala.js, Play Framework, and GraalVM Types, such as variance, givens, intersections, and unions Best practices, including pattern matching, modules, and functional error handling**

**Getting Clojure**

*A Practical Primer*

*Real-World Functional Programming*

*Scala Programming Projects*

*Scala Reactive Programming*

Intermediate level, for programmers fairly familiar with Java, but new to the functional style of programming and lambda expressions. Get ready to program in a whole new way. Functional Programming in Java will help you quickly get on top of the new, essential Java 8 language features and the functional style that will change and improve your code. This short, targeted book will help you make the paradigm shift from the old imperative way to a less error-prone, more elegant, and concise coding style that's also a breeze to parallelize. You'll explore the syntax and semantics of lambda expressions, method and constructor references, and functional interfaces. You'll design and write applications better using the new standards in Java 8 and the JDK. Lambda expressions are lightweight, highly concise anonymous methods backed by functional interfaces in Java 8. You can use them to leap forward into a whole new world of programming in Java. With functional programming capabilities, which have been around for decades in other languages, you can now write elegant, concise, less error-prone code using standard Java. This book will guide you though the paradigm change, offer the essential details about the new features, and show you how to transition from your old way of coding to an improved style. In this book you'll see popular design patterns, such as decorator, builder, and strategy, come to life to solve common design problems, but with little ceremony and effort. With these new capabilities in hand, Functional Programming in Java will help you pick up techniques to implement designs that were beyond easy reach in earlier versions of Java. You'll see how you can reap the benefits of tail call optimization, memoization, and effortless parallelization techniques. Java 8 will change the way you write applications. If you're eager to take advantage of the new features in the language, this is the book for you. What you need: Java 8 with support for lambda expressions and the JDK is required to make use of the concepts and the examples in this book.

If you're familiar with functional programming basics and want to gain a much deeper understanding, this in-depth guide takes you beyond syntax and demonstrates how you need to think in a new way. Software architect Neal Ford shows intermediate to advanced developers how functional coding allows you to step back a level of abstraction so you can see your programming problem with greater clarity. Each chapter shows you various examples of functional thinking, using numerous code examples from Java 8 and other JVM languages that include functional capabilities. This book may bend your mind, but you'll come away with a much better grasp of functional programming concepts. Understand why many imperative languages are adding functional capabilities Compare functional and imperative solutions to common problems Examine ways to cede control of routine chores to the runtime Learn how memoization and laziness eliminate hand-crafted solutions Explore functional approaches to design patterns and code reuse View real-world examples of functional thinking with Java 8, and in functional architectures and web frameworks Learn the pros and cons of living in a paradigmatically richer world If you're new to functional programming, check out Josh Backfield's book *Becoming Functional*.

Master the fundamentals of Scala and understand its emphasis on functional programming that sets it apart from Java. This book will help you translate what you already know in Java to Scala to start your functional programming journey. Learn Scala is split into four parts: a tour of Scala, a comparison between Java and Scala, Scala-specific features and functional programming idioms, and finally a discussion about adopting Scala in existing Java teams and legacy projects. After reading and using this tutorial, you'll come away with the skills in Scala to kick-start your productivity with this growing popular language. What You'll Learn Tour Scala and learn the basic syntax, constructs, and how to use the REPL Translate Java syntax that you already know into Scala Learn what Scala offers over and above Java Become familiar with functional programming concepts and idioms Gain tips and advice useful when transitioning existing Java projects to Scala Who This Book Is For Java developers looking to transition to Scala. No prior experience necessary in Scala.

Summary Functional Programming in Scala is a serious tutorial for programmers looking to learn FP and apply it to the everyday business of coding. The book guides readers from basic techniques to advanced topics in a logical, concise, and clear progression. In it, you'll find concrete examples and exercises that open up the world of functional programming. Purchase of the print book includes a free eBook in PDF, Kindle, and ePub formats from Manning Publications. About the Technology Functional programming (FP) is a style of software development emphasizing functions that don't depend on program state. Functional code is easier to test and reuse, simpler to parallelize, and less prone to bugs than other code. Scala is an emerging JVM language that offers strong support for FP. Its familiar syntax and transparent interoperability with Java make Scala a great place to start learning FP. About the Book Functional Programming in Scala is a serious tutorial for programmers looking to learn FP and apply it to their everyday work. The book guides readers from basic techniques to advanced topics in a logical, concise, and clear progression. In it, you'll find concrete examples and exercises that open up the world of functional programming. This book assumes no prior experience with functional programming. Some prior exposure to Scala or Java is helpful. What's Inside Functional programming concepts The whys and hows of FP How to write multicore programs Exercises and checks for understanding About the Authors Paul Chiusano and Rúnar Bjarnason are recognized experts in functional programming with Scala and are core contributors to the Scalaz library. Table of Contents PART 1 INTRODUCTION TO FUNCTIONAL PROGRAMMING What is functional programming? Getting started with functional programming in Scala Functional data structures Handling errors without exceptions Strictness and laziness Purely functional state PART 2 FUNCTIONAL DESIGN AND COMBINATOR LIBRARIES Purely functional parallelism Property-based testing Parser combinators PART 3 COMMON STRUCTURES IN FUNCTIONAL DESIGN Monoids Monads Applicative and traversable functors PART 4 EFFECTS AND I/O External effects and I/O Local effects and mutable state Stream processing and incremental I/O

How functional techniques improve your Java programs

Discover the pure functional side of HTTP API programming in Scala.

Functional Programming in Java

Practical FP in Scala: a Hands-On Approach (2nd Edition)

Functional and Reactive Domain Modeling

Functional Programming in Scala

Grok and perform effective functional programming in Scala About This Book Understand functional programming patterns by comparing them with the traditional object-oriented design patterns Write robust, safer, and better code using the declarative programming paradigm An illustrative guide for programmers to create functional programming patterns with Scala Who This Book Is For If you have done Java programming before and have a basic knowledge of Scala and its syntax, then this book is an ideal choice to help you to understand the context, the traditional design pattern applicable, and the Scala way. Having previous knowledge of design patterns will help, though it is not strictly necessary. What You Will Learn Get to know about functional programming and the value Scala's FP idioms bring to the table Solve day-to-day programming problems using functional programming idioms Cut down the boiler-plate and express patterns simply and elegantly using Scala's concise syntax Tame system complexity by reducing the moving parts Write easier to reason about concurrent code using the actor paradigm and the Akka library Apply recursive thinking and understand how to create solutions without mutation Reuse existing code to compose new behavior Combine the object-oriented and functional programming approaches for effective programming using Scala In Detail Scala is used to construct elegant class hierarchies for maximum code reuse and extensibility and to implement their behavior using higher-order functions. Its functional programming (FP) features are a boon to help you design “easy to reason about” systems to control the growing software complexities. Knowing how and where to apply the many Scala techniques is challenging. Looking at Scala best practices in the context of what you already know helps you grasp these concepts quickly, and helps you see where and why to use them. This book begins with the rationale behind patterns to help you understand where and why each pattern is applied. You will discover what tail recursion brings to your table and will get an understanding of how to create solutions without mutations. We then explain the concept of memorization and infinite sequences for on-demand computation. Further, the book takes you through Scala's stackable traits and dependency injection, a popular technique to produce loosely-coupled software systems. You will also explore how to currying favors to your code and how to simplify it by de-construction via pattern matching. We also show you how to do pipeline transformations using higher order functions such as the pipes and filters pattern. Then we guide you through the increasing importance of concurrent programming and the pitfalls of traditional code concurrency. Lastly, the book takes a paradigm shift to show you the different techniques that functional programming brings to your plate. This book is an invaluable source to help you understand and perform functional programming and solve common programming problems using Scala's programming patterns. Style and approach This is a hands-on guide to Scala's game-changing features for programming. It is filled with many code examples and figures that illustrate various Scala idioms and best practices.

Explore functional programming and discover new ways of thinking about code. You know you need to master functional programming, but learning one functional language is only the start. In this book, through articles drawn from PragPub magazine and articles written specifically for this book, you'll explore functional thinking and functional style and idioms across languages. Led by expert guides, you'll discover the distinct strengths and approaches of Clojure, Elixir, Haskell, Scala, and Swift and learn which best suits your needs. Contributing authors: Rich Hickey, Stuart Halloway, Aaron Bedra, Michael Bevilacqua-Linn, Venkat Subramaniam, Paul Callaghan, Jose Valim, Dave Thomas, Natasha Murashev, Tony Hillerson, Josh Chisholm, and Bruce Tate. Functional programming is on the rise because it lets you write simpler, cleaner code, and its emphasis on immutability makes it ideal for maximizing the benefits of multiple cores and distributed solutions. So far nobody's invented the perfect functional language – each has its unique strengths. In Functional Programming: A PragPub Anthology, you'll investigate the philosophies, tools, and idioms of five different functional programming languages. See how Swift, the development language for iOS, encourages you to build highly scalable apps using functional techniques like map and reduce. Discover how Scala allows you to transition gently but deeply into functional programming without losing the benefits of the JVM, while with Lisp-based Clojure, you can plunge fully into the functional style. Learn about advanced functional concepts in Haskell, a pure functional language making powerful use of the type system with type inference and type classes. And see how functional programming is becoming more elegant and friendly with Elixir, a new functional language built on the powerful Erlang base.The industry has been embracing functional programming more and more, driven by the need for concurrency and parallelism. This collection of articles will lead you to mastering the functional approach to problem solving. So put on your explorer's hat and prepare to be surprised. The goal of exploration is always discovery. What You Need:

Familiarity with one or more programming languages.

In 1994, Design Patterns changed the landscape of object-oriented development by introducing classic solutions to recurring design problems. In 1999, Refactoring revolutionized design by introducing an effective process for improving code. With the highly anticipated Refactoring to Patterns , Joshua Kerievsky has changed our approach to design by forever uniting patterns with the evolutionary process of refactoring. This book introduces the theory and practice of pattern-directed refactorings: sequences of low-level refactorings that allow designers to safely move designs to, towards, or away from pattern implementations. Using code from real-world projects, Kerievsky documents the thinking and steps underlying over two dozen pattern-based design transformations. Along the way he offers insights into pattern differences and how to implement patterns in the simplest possible ways. Coverage includes: A catalog of twenty-seven pattern-directed refactorings, featuring real-world code examples Descriptions of twelve design smells that indicate the need for this book's refactorings General information and new insights about patterns and refactoring Detailed implementation mechanics: how low-level refactorings are combined to implement high-level patterns Multiple ways to implement the same pattern—and when to use each Practical ways to get started even if you have little experience with patterns or refactoring Refactoring to Patterns reflects three years of refinement and the insights of more than sixty software engineering thought leaders in the global patterns, refactoring, and agile development communities. Whether you're focused on legacy or “greenfield” development, this book will make you a better software designer by helping you learn how to make important design changes safely and effectively.

Scala Design Patterns

Scala Functional Programming Patterns

Functional Programming: A PragPub Anthology

Functional Thinking

Functional Programming for Java Developers