

Functional Programming Scala Paul Chiusano

????????????????????C????????C????????????????????C ???

INTRODUCTION xv CHAPTER 1: LANGUAGE FEATURES 1 Static Types and Type Inference 2 Implicit Parameters, Conversions, and Their Resolution 3 Case Class, Tuples, and Case Object 5 Abstract Class, Traits, and Sealed 6 Pattern Matching 8 Statements Are Expressions 9 String Interpolation 9 Scala Collections, immutable and mutable 10 For Comprehension 12 Packages, Companion Objects, Package Objects, and Scoping 13 AnyVal, AnyRef, Any, and the Type Hierarchy 16 Summary 17 CHAPTER 2: FUNCTIONAL PROGRAMMING 19 Immutability 20 Pure Functions 22 Recursion 23 Higher-Order Functions 26 Core Collection Methods 27 Methods Returning a Collection 29 Methods Returning a Value 31 Currying and Partially Applied Functions 32 Null Handling (Option) 34 Strict versus Non-Strict Initialization 35 Summary 36 CHAPTER 3: JAVA COMPATIBILITY 37 Scala and Java Collections 37 Interfaces and Traits 40 Scala/Java Enumerations 42 Summary 43 CHAPTER 4: SIMPLE BUILD TOOL 45 Basic Usage 46 Project Structure 47 Single Project 47 Scopes 49 Custom Tasks 50 Dependencies 50 Resolvers 51 Advanced Usage 52 Advanced Dependencies 53 Testing in the Console 55 Release Management 56 Deploying to Sonatype 56 Packaging with SBT-Native-Packager 58 Creating a Docker Image 59 Common SBT Commands 60 Useful Plugins 61 Summary 62 CHAPTER 5: MAVEN 63 Getting Started with Maven and Scala 64 Introducing scala-maven-plugin 67 Adding Library Dependencies 70 Using the REPL 71 Getting Help 72 Running Tests 72 Joint Compilation with Java 74 Accelerating Compilation with Zinc 76 Summary 77 CHAPTER 6: SCALA STYLE/LINT 79 Scala with Style 79 Scaliform 81 Scapegoat 82 WartRemover 82 Scoverage 84 Summary 84 CHAPTER 7: TESTING 85 ScalaTest 86 Unit Tests 87 Integration Testing 87 Data-Driven Tests 88 Performance Testing 89 Acceptance Testing 90 Mocks 92 Load Testing 93 Summary 94 CHAPTER 8: DOCUMENTING YOUR CODE WITH SCALADOC 95 Why Document Your Code? 96 Revealing the Benefits 96 Bookending the Continuum 96 Choosing What to Document 96 Scaladoc Structure 97 Overall Layout 97 Index Pane 98 Content Pane 100 Invoking the Scaladoc Tool 106 Wiki Syntax 108 Formatting with Inline Wiki Syntax 108 Structuring with Block Elements 110 Linking 113 Locating Scaladoc 117 Tagging 117 Everyday Tagging 117 Tagging for Groups 123 Advanced Tagging 125 Invoking scaladoc: Additional Options 132 Integrating Scaladoc Creation with Your Project 133 Configuring Maven 133 Configuring SBT 134 Publishing Scaladoc 134 Tables and CSS 136 Summary 138 CHAPTER 9: TYPE SYSTEM 139 What Is a Type System? 140 Static versus Dynamic Typing 140 What Static Type Systems Are Good For 141 What Dynamic Type Systems Are Good For 141 Scala's Unified Type System 141 Value Classes 143 Polymorphism 145 Subtype Polymorphism 145 Parametric Polymorphism 146 Ad Hoc Polymorphism 146 Bounds 149 Context Bounds 149 Upper and Lower Bounds 150 Variance 151 Other Niceties 155 Self-Type Annotations 155 Self-Recursive Types 158 Abstract Type Members 159 Dynamic Programming 161 Structural Types 161 Dynamic Trait 162 Summary 164 CHAPTER 10: ADVANCED FUNCTIONAL PROGRAMMING 165 Higher-Kinded Types 165 Functional Design Patterns 167 Functor 167 Applicative Functor 170 Monad 172 Semigroup 173 Monoid 174 Summary 176 CHAPTER 11: CONCURRENCY 179 Synchronize/Atomic Variables 181 Future Composition 184 Parallel Collections 187 Reactive Streams 192 STM 195 Actors (Akka) 198 Spark 200 Summary 202 CHAPTER 12: SCALA.JS 205 Scala.Js and Its Design 205 Getting Started: Scala.Js with SBT 206 Scala.Js Peculiarities 210 Webjars and Dealing with the Frontend Ecosystem 211 Summary 213 INDEX 215

?????:??

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

This full-color booklet contains chapter notes, hints, solutions to exercises, addenda, and errata for the book "Functional Programming in Scala" by Paul Chiusano and Runar Bjarnason. This material is freely available online, but is compiled here as a convenient companion to the book itself. All code is colorfully syntax-highlighted.

????????????

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.

??????“???”????????????????

Helps programmers learn functional programming and apply it to the everyday business of coding. Original.

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

Functional Programming in ScalaManning Publications

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

