Dsls In Action Debasish Ghosh

Thank you very much for downloading **Dsls In Action Debasish Ghosh**. Maybe you have knowledge that, people have search numerous times for their chosen readings like this Dsls In Action Debasish Ghosh, but end up in malicious downloads.
Rather than enjoying a good book with a cup of coffee in the afternoon, instead they cope with some infectious virus inside their laptop.

Dsls In Action Debasish Ghosh is available in our digital library an online access to it is set as public so you can download it instantly.
Our books collection spans in multiple locations, allowing you to get the most less latency time to download any of our books like this one.
Merely said, the Dsls In Action Debasish Ghosh is universally compatible with any devices to read

`Scala in Depth` "O'Reilly Media, Inc."
```
There is an easier way to build Hadoop applications. With this hands-on book, you'll learn how to use
Cascading, the open source abstraction framework for Hadoop that lets you easily create and manage powerful
enterprise-grade data processing applications—without having to learn the intricacies of MapReduce. Working
with sample apps based on Java and other JVM languages, you'll quickly learn Cascading's streamlined approach
to data processing, data filtering, and workflow optimization. This book demonstrates how this framework can
help your business extract meaningful information from large amounts of distributed data. Start working on
Cascading example projects right away Model and analyze unstructured data in any format, from any source Build
and test applications with familiar constructs and reusable components Work with the Scalding and Cascalog
Domain-Specific Languages Easily deploy applications to Hadoop, regardless of cluster location or data size
Build workflows that integrate several big data frameworks and processes Explore common use cases for
Cascading, including features and tools that support them Examine a case study that uses a dataset from the
Open Data Initiative
```
Elements of Information Organization and Dissemination Packt Publishing Ltd
Summary A fully revised edition that covers the new features available in Clojure 1.6. Purchase of the print book includes a free eBook in PDF, Kindle, and ePub formats from Manning Publications. About the Technology Clojure is a modern Lisp for the JVM. It has the strengths you expect: first-class functions, macros, and Lisp's clean programming style. It supports functional programming, making it ideal for concurrent programming and for creating domain-specific languages. Clojure lets you solve harder problems, make faster changes, and end up with a smaller code base. It's no wonder that there are so many Clojure success stories. About the Book Clojure in Action, Second Edition is an expanded and improved version that's been updated to cover the new features of Clojure 1.6. The book gives you a rapid introduction to the Clojure language, moving from abstract theory to practical examples. You'll start by learning how to use Clojure as a general-purpose language. Next, you'll explore Clojure's efficient concurrency model, based on the database concept of Software Transactional Memory (STM). You'll gain a new level of productivity through Clojure DSLs that can run on the JVM. Along the way, you'll learn countless tips, tricks, and techniques for writing smaller, safer, and faster code. What's Inside Functional programming basics Metaprogramming with Clojure's macros Interoperating with Java Covers Clojure 1.6 About the Reader Assumes readers are familiar with a programming language like C, Java, Ruby, or Python. Table of Contents Introducing Clojure Clojure elements: Data structures and functions Building blocks of Clojure Multimethod polymorphism Exploring Clojure and Java interop State and the concurrent world Evolving Clojure through macros More on functional programming Protocols, records, and types Test-driven development and more More macros and DSL
.NET Design Patterns Manning Publications
Functional and Reactive Domain Modeling teaches readers how to think of the domain model in terms of pure functions and how to compose them to build larger abstractions. It begins with the basics of functional programming and gradually progresses to the advanced concepts and patterns needed to implement complex domain models. The book demonstrates how advanced FP patterns like algebraic data types, typeclass based design, and isolation of side-effects can make models compose for readability and verifiability. On the subject of reactive modeling, the book focuses on higher order concurrency patterns like actors and futures. It uses the Akka framework as the reference implementation and demonstrates how advanced architectural patterns like event sourcing and CQRS can be put to great use in implementing scalable models. It offers techniques that are radically different from the standard RDBMS based applications that are based on mutation of records. It also shares important patterns like using asynchronous messaging for interaction based on non blocking concurrency and model persistence, which delivers the speed of in- memory processing along with suitable guarantees of reliability.
Realm of Racket "O'Reilly Media, Inc."
Provides a guide to using Scala and Clojure to solve in-depth programming problems.
Advances in Computer, Communication and Control Basic Books
"Dean Wampler, Java expert and author of Programming Scala (O'Reilley), shows you how to apply 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 the code you write."--From p. [4] of cover.
Functional and Reactive Domain Modeling "O'Reilly Media, Inc."
With the rapid rise in the ubiquity and sophistication of Internet technology and the accompanying growth in the number of network attacks, network intrusion detection has become increasingly important. Anomaly-based network intrusion detection refers to finding exceptional or nonconforming patterns in network traffic data compared to normal behavior. Finding these anomalies has extensive applications in areas such as cyber security, credit card and insurance fraud detection, and military surveillance for enemy activities. Network Anomaly Detection: A Machine Learning Perspective presents machine learning techniques in depth to help you more effectively detect and counter network intrusion. In this book, you'll learn about: Network anomalies and vulnerabilities at various layers The pros and cons of various machine learning techniques and algorithms A taxonomy of attacks based on their characteristics and behavior Feature selection algorithms How to assess the accuracy, performance, completeness, timeliness, stability, interoperability, reliability, and other dynamic aspects of a network anomaly detection system Practical tools for launching attacks, capturing packet or flow traffic, extracting features, detecting attacks, and evaluating detection performance Important unresolved issues and research challenges that need to be overcome to provide better protection for networks Examining numerous attacks in detail, the authors look at the tools that intruders use and show how to use this knowledge to protect networks. The book also provides material for hands-on development, so that you can code on a testbed to implement detection methods toward the development of your own intrusion detection system. It offers a thorough introduction to the state of the art in network anomaly detection using machine learning approaches and systems.

Reconfigurable Architectures and Design Automation Tools for Application-Level Network Security Simon and Schuster
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
Camel in Action Simon and Schuster
The book discusses the recent research trends in various sub-domains of computing, communication and control. It includes research papers presented at the First International Conference on Emerging Trends in Engineering and Science. Focusing on areas such as optimization techniques, game theory, supply chain, green computing, 5g networks, Internet of Things, social networks, power electronics and robotics, it is a useful resource for academics and researchers alike.
Domain-driven Design "O'Reilly Media, Inc."
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
Metaprogramming in .NET Addison-Wesley
Racket is a descendant of Lisp, a programming language renowned for its elegance, power, and challenging learning curve. But while Racket retains the functional goodness of Lisp, it was designed with beginning programmers in mind. Realm of Racket is your introduction to the Racket language. In Realm of Racket, you'll learn to program by creating increasingly complex games. Your journey begins with the Guess My Number game and coverage of some basic Racket etiquette. Next you'll dig into syntax and semantics, lists, structures, and conditionals, and learn to work with recursion and the GUI as you build the Robot Snake game. After that it's on to lambda and mutant structs (and an Orc Battle), and fancy loops and the Dice of Doom. Finally, you'll explore laziness, AI, distributed games, and the Hungry Henry game. As you progress through the games, chapter checkpoints and challenges help reinforce what you've learned. Offbeat comics keep things fun along the way. As you travel through the Racket realm, you'll: –Master the quirks of Racket's syntax and semantics –Learn to write concise and elegant functional programs –Create a graphical user interface using the 2htdp/image library –Create a server to handle true multiplayer games Realm of Racket is a lighthearted guide to some serious programming. Read it to see why Racketeers have so much fun!
Applying Domain-Driven Design and Patterns No Starch Press
The notion that "thinking about computing is one of the most exciting things the human mind can do" sets both The Little Schemer (formerly known as The Little LISPer) and its new companion volume, The Seasoned Schemer, apart from other books on LISP. The authors' enthusiasm for their subject is compelling as they present abstract concepts in a humorous and easy-to-grasp fashion. Together, these books will open new doors of thought to anyone who wants to find out what computing is really about. The Little Schemer introduces computing as an extension of arithmetic and algebra; things that everyone studies in grade school and high school. It introduces programs as recursive functions and briefly discusses the limits of what computers can

do. The authors use the programming language Scheme, and interesting foods to illustrate these abstract ideas. The Seasoned Schemer informs the reader about additional dimensions of computing: functions as values, change of state, and exceptional cases. The Little LISPer has been a popular introduction to LISP for many years. It had appeared in French and Japanese. The Little Schemer and The Seasoned Schemer are worthy successors and will prove equally popular as textbooks for Scheme courses as well as companion texts for any complete introductory course in Computer Science.

Functional Programming Patterns in Scala and Clojure CRC Press

The relevance of the Internet has dramatically grown in the past decades. However, the enormous financial impact attracts many types of criminals. Setting up proper security mechanisms (e.g., Intrusion Detection Systems (IDS)) has therefore never been more important than today. To further compete with today's data transfer rates (10 to 100 Gbit/s), dedicated hardware accelerators have been proposed to offload compute intensive tasks from general purpose processors. As one key technology, reconfigurable hardware architectures, e.g., the Field Programmable Gate Array (FPGA), are of particular interest to this end. This work addresses the use of such FPGAs in the context of interactive communication applications, which goes beyond the regular packet level operations often seen in this area. To support rapid prototyping, a novel FPGA platform (NetStage) has been designed and developed, which provides a communication core for Internet communication and a flexible connection bus for attaching custom applications modules. A hardware honeypot (the MalCoBox) has been set up as a proof-of-concept application. Furthermore, to address the ongoing issue of hardware programming complexity, the domain-specific Malacoda language for abstractly formulating honeypot packet communication dialogs is presented and discussed. An associated compiler translates Malacoda into high-performance hardware modules for NetStage. Together, NetStage and Malacoda address some of the productivity deficiencies often recognized as major hindrances for the more widespread use of reconfigurable computing in communications applications. Finally, the NetStage platform has been evaluated in a real production environment.

Object Thinking Springer

The second international conference on INformation Systems Design and Intelligent Applications (INDIA – 2015) held in Kalyani, India during January 8-9, 2015. The book covers all aspects of information system design, computer science and technology, general sciences, and educational research. Upon a double blind review process, a number of high quality papers are selected and collected in the book, which is composed of two different volumes, and covers a variety of topics, including natural language processing, artificial intelligence, security and privacy, communications, wireless and sensor networks, microelectronics, circuit and systems, machine learning, soft computing, mobile computing and applications, cloud computing, software engineering, graphics and image processing, rural engineering, e-commerce, e-governance, business computing, molecular computing, nano-computing, chemical computing, intelligent computing for GIS and remote sensing, bio-informatics and bio-computing. These fields are not only limited to computer researchers but also include mathematics, chemistry, biology, bio-chemistry, engineering, statistics, and all others in which computer techniques may assist.

Enterprise Data Workflows with Cascading Simon and Schuster

Summary The Joy of Clojure, Second Edition is a deep look at the Clojure language. Fully updated for Clojure 1.6, this new edition goes beyond just syntax to show you the "why" of Clojure and how to write fluent Clojure code. You'll learn functional and declarative approaches to programming and will master the techniques that make Clojure so elegant and efficient. Purchase of the print book includes a free eBook in PDF, Kindle, and ePub formats from Manning Publications. About the Technology The Clojure programming language is a dialect of Lisp that runs on the Java Virtual Machine and JavaScript runtimes. It is a functional programming language that offers great performance, expressive power, and stability by design. It gives you built-in concurrency and the predictable precision of immutable and persistent data structures. And it's really, really fast. The instant you see long blocks of Java or Ruby dissolve into a few lines of Clojure, you'll know why the authors of this book call it a "joyful language." It's no wonder that enterprises like Staples are betting their infrastructure on Clojure. About the Book The Joy of Clojure, Second Edition is a deep account of the Clojure language. Fully updated for Clojure 1.6, this new edition goes beyond the syntax to show you how to write fluent Clojure code. You'll learn functional and declarative approaches to programming and will master techniques that make Clojure elegant and efficient. The book shows you how to solve hard problems related to concurrency, interoperability, and performance, and how great it can be to think in the Clojure way. Appropriate for readers with some experience using Clojure or common Lisp. What's Inside Build web apps using ClojureScript Master functional programming techniques Simplify concurrency Covers Clojure 1.6 About the Authors Michael Fogus and Chris Houser are contributors to the Clojure and ClojureScript programming languages and the authors of various Clojure libraries and language features. Table of Contents PART 1 FOUNDATIONS Clojure philosophy Drinking from the Clojure fire hose Dipping your toes in the pool PART 2 DATA TYPES On scalars Collection types PART 3 FUNCTIONAL PROGRAMMING Being lazy and set in your ways Functional programming PART 4 LARGE-SCALE DESIGN Macros Combining data and code Mutation and concurrency Parallelism PART 5 HOST SYMBIOSIS Java.next Why ClojureScript? PART 6 TANGENTIAL CONSIDERATIONS Data-oriented programming Performance Thinking programs Clojure changes the way you think

Scala for Java Developers Simon and Schuster

Software engineering for complex systems requires abstraction, multi-domain expertise, separation of concerns, and reuse. Domain experts rarely are software engineers and should formulate solutions using their domain's vocabulary instead of general purpose programming languages (GPLs). Successful integration of domain-specific languages (DSLs) into a software system requires a separation of concerns between domain issues and integration issues while retaining a loose enough coupling to support DSL reuse in different contexts. Component-based software engineering (CBSE) increases reuse and separation of concerns by encapsulating functionalities in components. Components are GPL artifacts, which raises accidental complexities. Model-driven engineering (MDE) abstracts from GPLs by lifting models to primary development artifacts. Models can be abstract and better comprehensible by using domain vocabulary instead of a GPL. They can be platform-independent and translated into GPLs for different target platforms. Component & connector (C&C) architecture description languages (ADLs) combine CBSE and MDE to compose of architectures from component models. We present concepts for engineering software systems with exchangeable component behavior languages. The concepts are realized in a software architecture modeling infrastructure that comprises modeling languages to develop applications based on C&C software architectures with exchangeable component behavior DSLs. It supports transformations from platform-independent to platform-specific software architectures and compositional code generation. With this, it enables domain experts to (re-)use the most appropriate component behavior DSL and facilitates composition of domain solutions through encapsulation in components.

Functional and Reactive Domain Modeling MIT Press

Get up to speed on Scala--the JVM, JavaScript, and natively compiled language that offers all the benefits of functional programming, a modern object model, 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. You'll learn why Scala is ideal for building today's highly scalable, data-centric applications while maximizing developer productivity. While Java remains popular and Kotlin has become popular, Scala hasn't been sitting still. This third edition covers the new features in Scala 3 with updates throughout the book. Programming Scala is ideal for beginning to advanced developers who want a complete understanding of Scala's design philosophy and features with a thoroughly practical focus. Program faster with Scala's succinct and flexible syntax Dive into basic and advanced functional programming techniques Build killer big data and distributed apps using Scala's functional combinators and tools like Spark and Akka Create concise solutions to challenging design problems with the sophisticated type system, mixin composition with traits, pattern matching, and more

Network Anomaly Detection Simon and Schuster

Most people are baffled by how computers work and assume that they will never understand them. What they don't realize—and what Daniel Hillis's short book brilliantly demonstrates—is that computers' seemingly complex operations can be broken down into a few simple parts that perform the same simple procedures over and over again. Computer wizard Hillis offers an easy-to-follow explanation of how data is processed that makes the operations of a computer seem as straightforward as those of a bicycle.Avoiding technobabble or discussions of advanced hardware, the lucid explanations and colorful anecdotes in The Pattern on the Stone go straight to the heart of what computers really do. Hillis proceeds from an outline of basic logic to clear descriptions of programming languages, algorithms, and memory. He then takes readers in simple steps up to the most exciting developments in computing today—quantum computing, parallel computing, neural networks, and self-organizing systems.Written clearly and succinctly by one of the world's leading computer scientists, The Pattern on the Stone is an indispensable guide to understanding the workings of that most ubiquitous and important of machines: the computer.

Implementing Domain-Specific Languages with Xtext and Xtend Academic Press

Summary Scala in Depth is a unique new book designed to help you integrate Scala effectively into your development process. By presenting the emerging best practices and designs from the Scala community, it guides you through dozens of powerful techniques example by example. About the Book Scala is a powerful JVM language that blends the functional and OO programming models. You'll have no trouble getting introductions to Scala in books or online, but it's hard to find great examples and insights from experienced practitioners. You'll find them in Scala in Depth. There's little heavy-handed theory here—just dozens of crisp, practical techniques for coding in Scala. Written for readers who know Java, Scala, or another OO language. 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 Concise, expressive, and readable code style How to integrate Scala into your existing Java projects Scala's 2.8.0 collections API How to use actors for concurrent programming Mastering the Scala type system Scala's OO features—type member inheritance, multiple inheritance, and composition Functional concepts and patterns—immutability, applicative functors, and monads ==?========== Table of Contents Scala—a blended language The core rules Modicum of style—coding conventions Utilizing object orientation Using implicits to write expressive code The type system Using implicits and types together Using the right collection Actors Integrating Scala with Java Patterns in functional programming

Programming Scala Pearson Education

Object Thinking blends historical perspective, experience, and visionary insight - exploring how developers can work less like the computers they program and more like problem solvers.

The Seasoned Schemer, second edition Packt Publishing Ltd

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