1. Record Nr. UNINA9910300642103321

Autore	Layka Vishal
Titolo	Beginning Scala / / by Vishal Layka, David Pollak
Pubbl/distr/stampa	Berkeley, CA : , : Apress : , : Imprint : Apress, , 2015
ISBN	9781484202326 1484202325
Edizione	[2nd ed. 2015.]
Descrizione fisica	1 online resource (246 p.)
Collana	The expert's voice in Java
Disciplina	004
Soggetti	Java (Computer program language) Software engineering Java Software Engineering/Programming and Operating Systems
Lingua di pubblicazione	Inglese
Formato	Materiale a stampa
Livello bibliografico	Monografia
Note generali	Includes index.
Nota di bibliografia	Includes bibliographical references and index.
Nota di contenuto	Contents at a Glance; Introduction; Chapter 1: Getting Started with Scala; Why Scala?; A Concise Language; Lightweight Syntax; Multiparadigm Language; Object-Oriented Language; Functional Language; Interoperability and Seamless Integration with Java; Language for the Java Virtual Machine; Installing Scala; Scala at the Command Line and Scala Scripts; Interactive Scala; Scala Scripts; Compiling Scala Programs; Your First Scala Programs; Hello World; Printing Some Numbers; Summary; Chapter 2: Basics of Scala; Variables; Scala Type Hierarchy; Any, AnyVal and AnyRef Types; Numeric Types Boolean TypeChar Type; Unit type; Nothing and Null Types; Strings; String Interpolation; Functions; Function without Parameter; Function with Parameters; Arrays, Lists, Ranges, and Tuples; Arrays; Lists; Ranges; Tuples; Built-in Control Structures; If Expressions; While Loops; For Comprehension; Basic for Expression; Filters; Variable Binding; Yielding; try expressions; Match Expressions; Comments; Summary; Chapter 3: Object Orientation in Scala; Classes and Objects; Constructors; Constructors with Parameters; Parameter Declared as a val; Parameter Declared as a var Parameter Declared as a private val or varParameter Declared without

val or var; Auxiliary Constructor; Method Declaration; Code Blocks; Call-by-Name; Method Invocation; Objects; Singleton Objects; Companion Objects; Packaging and Imports; Inheritance; Extending Class; Traits; Case Classes; Value Classes; Scala versus Java versus Ruby; Classes and Instances; Traits, Interfaces, and Mixins; Object, Static, and Singletons; Functions, Anonymous Inner Classes, and Lambdas/Procs; Summary; Chapter 4: Functional Programming in Scala; Expression-Oriented Programming; A Pure Function
Referential TransparencyFunction Literal/Anonymous Function; First Class Function and Higher Order Function; Function as Variable; Function as Parameter; Returning a Function; Closure; Partially Applied Function; Curried Function; Function Composition; Tail Calls and Tail Call Optimization; Call-by-Name, Call-by-Value, and General Laziness; Summary; Chapter 5: Pattern Matching; Basic Pattern Matching; Matching Any Type; Testing Data Types; Pattern Matching in Lists; Pattern Matching and Lists; Pattern Matching and Case Classes; Nested Pattern Matching in Case Classes
Pattern Matching As FunctionsObject-Oriented and Functional Tensions; Shape Abstractions; Summary; Chapter 6: Scala Collections; Scala Collection Hierarchy; package scala.collection; Sequences; Sets; Map; package scala.collection.immutable; Immutable Sequence; Immutable Set; Immutable Map; package scala.collection.mutable; Sec11; Buffer; Using Immutable Collection Classes; Vector; List[T]; Getting Functional; Transformation; Reduxio; Look Ma, No Loops; Range; Stream; Tuples; Map[K, V]; Mutable Collections; Sec25; Mutable Queue; Mutable Stack; Summary; Chapter 7: Traits; Using Traits as Mixins
Traits and Class Hierarchies

Sommario/riassunto

Beginning Scala, Second Edition takes a down-to-earth approach to teaching Scala that leads you through simple examples that can be combined to build complex, scalable systems and applications. This book introduces you to the Scala programming language, its object-oriented and functional programming characteristics, and then guides you through Scala constructs and libraries that allow you to assemble small components into high-performance, scalable systems. You will learn why Scala is judiciously used for critical business applications by leading companies such as Twitter, LinkedIn, Foursquare, the Guardian, Morgan Stanley, Credit Suisse, UBS, and HSBC. Scala is a multi-paradigm programming language that combines both functional and object-oriented features. Moreover, this highly scalable language lends itself well to building cloud-based/deliverable Software as a Service (SaaS) online applications.