

Contents

1	An Introduction to Functional Programming Languages	1
1.1	Main Characteristics of Functional Programming Languages	2
1.2	Some Functional Programming Languages	3
1.2.1	LISP	3
1.2.2	FP	3
1.2.3	Standard ML (SML)	4
1.2.4	Haskell	4
1.3	Scala	4
1.4	Running Scala	4
2	The Basics of the Language	7
2.1	Data Types	7
2.1.1	Strings	9
2.2	Statements and Expressions	11
2.3	Statement Separator and Blocks	11
2.4	Comments	12
2.5	Declarations	12
2.5.1	Composite Types: Cartesian Products	13
2.5.2	Nested Declarations	13
2.6	Functions	13
2.6.1	Alternative Ways to Define Types in Functions	15
2.6.2	Type Inference in Scala	15
2.6.3	Signature	16
2.6.4	Referentially Transparent	17
2.6.5	Higher-Order Functions	18
2.6.6	Currification	19
2.6.7	Recursive Functions	20
2.6.8	Functions and Non Functional Programming	22
2.7	Lists	22
2.7.1	Recursion on Lists	24
2.8	Pattern Matching	26
2.8.1	Pattern Matching on Lists	27
2.9	Collections and Their Higher Order Functions	29
2.9.1	Mutable and Immutable Data Structures	29
2.9.2	Mutable and Immutable Collections	30
2.9.3	Some Imperative Construction on Collections	31
2.9.4	Higher-Order Functions for Collections	32
2.10	List Comprehension	36
3	Lazy and Eager Evaluation	37
3.1	Parameter Passing	39
3.2	Lazy Val	40

3.3	Streams and Other Infinite Data Structures	41
3.4	Stream of Even Numbers	44
3.5	Stream of Odd Numbers	45
3.6	The Fibonacci Numbers	46
3.7	The Prime Numbers	46
3.8	Exercises with Streams	47
4	Object-Oriented Programming in Scala	51
4.1	Class Hierarchy	53
4.2	Definition of a Class	54
4.2.1	Notation	57
4.3	Value Classes	58
4.4	Case Classes	59
4.5	Abstract Classes	59
4.6	Singleton Objects	60
4.7	Companion Objects	61
4.8	Traits	63
4.8.1	Inheritance	63
4.8.2	Multiple Inheritance	64
4.8.3	Name Clashes in Traits	65
4.9	Packages	66
4.10	Some Additional Issues	67
5	Types and Classes Revisited: Polymorphism	69
5.1	Classes with Polymorphic Types	70
5.2	Monoids, Functors, and Monads	72
5.2.1	Monoids	73
5.2.2	Functors	73
5.2.3	Monads	74
6	Scala: OOL and FP	77
6.1	Tail-Recursive Functions	77
6.1.1	Some Scala Technicalities	79
6.1.2	Additional Examples of Tail-Recursive Functions	80
6.2	Functions in Scala and Object-Oriented Programming	81
6.3	Defining Functions Revisited: val and def	83
6.4	Data Types and Efficiency	84
7	Algebraic Data Types	87
7.1	Definition of Algebraic Data Types in Standard ML	87
7.2	Algebraic Data Types in Scala	90
7.3	Data Types and Efficiency Revisited	92
8	Parallelism	93
8.1	Collections	94
8.2	Actors	97

8.2.1	Definition	98
8.2.2	Receive and React, ! and !?	103
8.2.3	Futures and !!.	105
8.2.4	Others	107
8.2.5	Akka's Actor Model	107
9	Solutions	111
	References	119
	Index	121

Scala: From a Functional Programming Perspective
An Introduction to the Programming Language

Torra, V.

2016, XIII, 124 p. 7 illus., Softcover

ISBN: 978-3-319-46480-0