

# Contents

Foreword to the First Edition xvii

Preface xix

About the Author xxi

## 1 THE BASICS **A1** 1

1.1 The Scala Interpreter 1

1.2 Declaring Values and Variables 4

1.3 Commonly Used Types 5

1.4 Arithmetic and Operator Overloading 6

1.5 More about Calling Methods 8

1.6 The apply Method 9

1.7 Scaladoc 10

Exercises 15

## 2 CONTROL STRUCTURES AND FUNCTIONS **A1** 17

2.1 Conditional Expressions 18

2.2 Statement Termination 19

2.3 Block Expressions and Assignments 20

2.4	Input and Output	21
2.5	Loops	22
2.6	Advanced for Loops	24
2.7	Functions	25
2.8	Default and Named Arguments <b>L1</b>	26
2.9	Variable Arguments <b>L1</b>	26
2.10	Procedures	28
2.11	Lazy Values <b>L1</b>	28
2.12	Exceptions	29
	Exercises	31

### 3 WORKING WITH ARRAYS **A1** 35

3.1	Fixed-Length Arrays	35
3.2	Variable-Length Arrays: Array Buffers	36
3.3	Traversing Arrays and Array Buffers	37
3.4	Transforming Arrays	38
3.5	Common Algorithms	40
3.6	Deciphering Scaladoc	41
3.7	Multidimensional Arrays	42
3.8	Interoperating with Java	43
	Exercises	44

### 4 MAPS AND TUPLES **A1** 47

4.1	Constructing a Map	48
4.2	Accessing Map Values	48
4.3	Updating Map Values	49
4.4	Iterating over Maps	50
4.5	Sorted Maps	50
4.6	Interoperating with Java	50
4.7	Tuples	51
4.8	Zipping	52
	Exercises	52

<b>5</b>	<b>CLASSES</b> <b>A1</b>	<b>55</b>
5.1	Simple Classes and Parameterless Methods	55
5.2	Properties with Getters and Setters	56
5.3	Properties with Only Getters	59
5.4	Object-Private Fields	60
5.5	Bean Properties <b>L1</b>	61
5.6	Auxiliary Constructors	62
5.7	The Primary Constructor	63
5.8	Nested Classes <b>L1</b>	66
	Exercises	68

<b>6</b>	<b>OBJECTS</b> <b>A1</b>	<b>71</b>
6.1	Singletons	71
6.2	Companion Objects	72
6.3	Objects Extending a Class or Trait	73
6.4	The apply Method	73
6.5	Application Objects	74
6.6	Enumerations	75
	Exercises	77

<b>7</b>	<b>PACKAGES AND IMPORTS</b> <b>A1</b>	<b>79</b>
7.1	Packages	80
7.2	Scope Rules	81
7.3	Chained Package Clauses	83
7.4	Top-of-File Notation	83
7.5	Package Objects	83
7.6	Package Visibility	84
7.7	Imports	85
7.8	Imports Can Be Anywhere	85
7.9	Renaming and Hiding Members	86
7.10	Implicit Imports	86
	Exercises	87

- ## 8 INHERITANCE **A1** 91
- 8.1 Extending a Class 91
  - 8.2 Overriding Methods 92
  - 8.3 Type Checks and Casts 93
  - 8.4 Protected Fields and Methods 94
  - 8.5 Superclass Construction 94
  - 8.6 Overriding Fields 95
  - 8.7 Anonymous Subclasses 97
  - 8.8 Abstract Classes 97
  - 8.9 Abstract Fields 97
  - 8.10 Construction Order and Early Definitions **L3** 98
  - 8.11 The Scala Inheritance Hierarchy 100
  - 8.12 Object Equality **L1** 102
  - 8.13 Value Classes **L2** 103
    - Exercises 105
- ## 9 FILES AND REGULAR EXPRESSIONS **A1** 109
- 9.1 Reading Lines 109
  - 9.2 Reading Characters 110
  - 9.3 Reading Tokens and Numbers 111
  - 9.4 Reading from URLs and Other Sources 111
  - 9.5 Reading Binary Files 112
  - 9.6 Writing Text Files 112
  - 9.7 Visiting Directories 112
  - 9.8 Serialization 113
  - 9.9 Process Control **A2** 114
  - 9.10 Regular Expressions 116
  - 9.11 Regular Expression Groups 117
    - Exercises 118
- ## 10 TRAITS **L1** 121
- 10.1 Why No Multiple Inheritance? 121
  - 10.2 Traits as Interfaces 123
  - 10.3 Traits with Concrete Implementations 124

- 10.4 Objects with Traits 125
- 10.5 Layered Traits 125
- 10.6 Overriding Abstract Methods in Traits 127
- 10.7 Traits for Rich Interfaces 127
- 10.8 Concrete Fields in Traits 128
- 10.9 Abstract Fields in Traits 130
- 10.10 Trait Construction Order 130
- 10.11 Initializing Trait Fields 132
- 10.12 Traits Extending Classes 133
- 10.13 SelfTypes **L2** 134
- 10.14 What Happens under the Hood 135
  - Exercises 137

- ## 11 OPERATORS **L1** 141
- 11.1 Identifiers 142
  - 11.2 Infix Operators 143
  - 11.3 Unary Operators 143
  - 11.4 Assignment Operators 144
  - 11.5 Precedence 144
  - 11.6 Associativity 145
  - 11.7 The apply and update Methods 146
  - 11.8 Extractors **L2** 147
  - 11.9 Extractors with One or No Arguments **L2** 149
  - 11.10 The unapplySeq Method **L2** 149
  - 11.11 Dynamic Invocation **L2** 150
    - Exercises 153

- ## 12 HIGHER-ORDER FUNCTIONS **L1** 157
- 12.1 Functions as Values 157
  - 12.2 Anonymous Functions 159
  - 12.3 Functions with Function Parameters 160
  - 12.4 Parameter Inference 160
  - 12.5 Useful Higher-Order Functions 161
  - 12.6 Closures 162

8	12.7	SAM Conversions	163
	12.8	Currying	164
	12.9	Control Abstractions	166
	12.10	The return Expression	167
		Exercises	168
<b>13</b>	<b>COLLECTIONS</b>	<b>A2</b>	<b>171</b>
	13.1	The Main Collections Traits	172
	13.2	Mutable and Immutable Collections	173
	13.3	Sequences	174
	13.4	Lists	175
	13.5	Sets	177
	13.6	Operators for Adding or Removing Elements	178
	13.7	Common Methods	180
	13.8	Mapping a Function	182
	13.9	Reducing, Folding, and Scanning	<b>A3</b> 184
	13.10	Zippping	187
	13.11	Iterators	188
	13.12	Streams	<b>A3</b> 189
	13.13	LazyViews	<b>A3</b> 190
	13.14	Interoperability with Java Collections	191
	13.15	Parallel Collections	193
		Exercises	194
<b>14</b>	<b>PATTERN MATCHING AND CASE CLASSES</b>	<b>A2</b>	<b>197</b>
	14.1	A Better Switch	198
	14.2	Guards	199
	14.3	Variables in Patterns	199
	14.4	Type Patterns	200
	14.5	Matching Arrays, Lists, and Tuples	201
	14.6	Extractors	202
	14.7	Patterns in Variable Declarations	203
	14.8	Patterns in for Expressions	204
	14.9	Case Classes	205

- 14.10 The copy Method and Named Parameters 205
- 14.11 Infix Notation in case Clauses 206
- 14.12 Matching Nested Structures 207
- 14.13 Are Case Classes Evil? 208
- 14.14 Sealed Classes 209
- 14.15 Simulating Enumerations 209
- 14.16 The Option Type 210
- 14.17 Partial Functions **L2** 211
- Exercises 212

## 15

ANNOTATIONS **A2** 215

- 15.1 What Are Annotations? 216
- 15.2 What Can Be Annotated? 216
- 15.3 Annotation Arguments 217
- 15.4 Annotation Implementations 218
- 15.5 Annotations for Java Features 219
  - 15.5.1 Java Modifiers 219
  - 15.5.2 Marker Interfaces 220
  - 15.5.3 Checked Exceptions 220
  - 15.5.4 Variable Arguments 221
  - 15.5.5 JavaBeans 221
- 15.6 Annotations for Optimizations 222
  - 15.6.1 Tail Recursion 222
  - 15.6.2 Jump Table Generation and Inlining 223
  - 15.6.3 Eliding Methods 224
  - 15.6.4 Specialization for Primitive Types 225
- 15.7 Annotations for Errors and Warnings 226
- Exercises 227

## 16

XML PROCESSING **A2** 229

- 16.1 XML Literals 230
- 16.2 XML Nodes 230
- 16.3 Element Attributes 232
- 16.4 Embedded Expressions 233

- 16.5 Expressions in Attributes 234
- 16.6 Uncommon Node Types 235
- 16.7 XPath-like Expressions 235
- 16.8 Pattern Matching 237
- 16.9 Modifying Elements and Attributes 238
- 16.10 Transforming XML 239
- 16.11 Loading and Saving 239
- 16.12 Namespaces 242
  - Exercises 243
- 17 FUTURES A2 247**
  - 17.1 Running Tasks in the Future 248
  - 17.2 Waiting for Results 250
  - 17.3 The Try Class 251
  - 17.4 Callbacks 251
  - 17.5 Composing Future Tasks 252
  - 17.6 Other Future Transformations 255
  - 17.7 Methods in the Future Object 256
  - 17.8 Promises 258
  - 17.9 Execution Contexts 260
    - Exercises 260
- 18 TYPE PARAMETERS L2 265**
  - 18.1 Generic Classes 266
  - 18.2 Generic Functions 266
  - 18.3 Bounds for Type Variables 266
  - 18.4 View Bounds 268
  - 18.5 Context Bounds 268
  - 18.6 The ClassTag Context Bound 269
  - 18.7 Multiple Bounds 269
  - 18.8 Type Constraints L3 269
  - 18.9 Variance 271
  - 18.10 Co- and Contravariant Positions 272



- 18.11 Objects Can't Be Generic 274
- 18.12 Wildcards 275
- Exercises 275

## 19 ADVANCED TYPES **L2** 279

- 19.1 Singleton Types 280
- 19.2 Type Projections 281
- 19.3 Paths 282
- 19.4 Type Aliases 283
- 19.5 Structural Types 283
- 19.6 Compound Types 284
- 19.7 Infix Types 285
- 19.8 Existential Types 286
- 19.9 The Scala Type System 287
- 19.10 Self Types 288
- 19.11 Dependency Injection 289
- 19.12 Abstract Types **L3** 291
- 19.13 Family Polymorphism **L3** 293
- 19.14 Higher-Kinded Types **L3** 296
- Exercises 299

## 20 PARSING **A3** 303

- 20.1 Grammars 304
- 20.2 Combining Parser Operations 305
- 20.3 Transforming Parser Results 307
- 20.4 Discarding Tokens 308
- 20.5 Generating Parse Trees 309
- 20.6 Avoiding Left Recursion 310
- 20.7 More Combinators 311
- 20.8 Avoiding Backtracking 314
- 20.9 Packrat Parsers 314
- 20.10 What Exactly Are Parsers? 315
- 20.11 Regex Parsers 316

- 20.12 Token-Based Parsers 317
- 20.13 Error Handling 319
  - Exercises 320

## 21 IMPLICITS **L3** 323

- 21.1 Implicit Conversions 324
- 21.2 Using Implicits for Enriching Existing Classes 324
- 21.3 Importing Implicits 325
- 21.4 Rules for Implicit Conversions 326
- 21.5 Implicit Parameters 328
- 21.6 Implicit Conversions with Implicit Parameters 329
- 21.7 Context Bounds 329
- 21.8 Type Classes 331
- 21.9 Evidence 333
- 21.10 The `@implicitNotFound` Annotation 334
- 21.11 `CanBuildFrom` Demystified 334
  - Exercises 336
- Index 338