

# A Beginner's Guide to Scala, Object Orientation and Functional Programming

John Hunt

# A Beginner's Guide to Scala, Object Orientation and Functional Programming

Second Edition

 Springer

John Hunt  
Midmarsh Technology Ltd  
Bath, Wiltshire  
UK

ISBN 978-3-319-75770-4      ISBN 978-3-319-75771-1 (eBook)  
<https://doi.org/10.1007/978-3-319-75771-1>

Library of Congress Control Number: 2018932535

1st edition: © Springer International Publishing Switzerland 2014

2nd edition: © Springer International Publishing AG 2018

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

The publisher, the authors and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, express or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Printed on acid-free paper

This Springer imprint is published by Springer Nature

The registered company is Springer International Publishing AG

The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

*This book is dedicated to my wife Denise; you  
are a constant source of inspiration to me.*



# Contents

<b>1</b>	<b>Introduction</b>	1
1.1	Introduction	1
1.2	What Is Scala?	1
1.3	Why Scala?	2
1.4	Java to Scala Quick Comparison	3
1.5	Scala Versions	5
1.6	Is This Book for You?	5
1.7	Approach Taken by This Book	5
<b>2</b>	<b>Elements of Object Orientation</b>	7
2.1	Introduction	7
2.2	Terminology	7
2.3	Types of Hierarchy	9
2.4	The Move to Object Technology	12
2.5	Summary	12
2.6	Exercises	13
2.7	Further Reading	13
<b>3</b>	<b>Why <i>Object Orientation</i>?</b>	15
3.1	Introduction	15
3.2	The Procedural Approach	15
3.2.1	A Naked Data Structure	16
3.2.2	Procedures for the Data Structure	16
3.2.3	Packages	17
3.3	Does Object Orientation Do Better?	17
3.3.1	Packages Versus Classes	17
3.3.2	Inheritance	19
3.4	Summary	20

<b>4</b>	<b>Constructing an Object-Oriented System</b>	21
4.1	Introduction	21
4.2	The Application: Windscreen Wipe Simulation	21
4.3	Where Do We Start?	22
4.4	Identifying the Objects	23
4.5	Identifying the Services or Methods	24
4.6	Refining the Objects	26
4.7	Bringing It All Together	26
4.8	Where Is the Structure?	29
4.9	Summary	31
4.10	Exercises	31
4.11	Further Reading	32
	References	33
<b>5</b>	<b>Functional Programming</b>	35
5.1	Introduction	35
5.2	What Is Functional Programming?	35
5.3	Advantages to Functional Programming	37
5.4	Disadvantages of Functional Programming	39
5.5	Scala and Functional Programming	40
<b>6</b>	<b>Scala Background</b>	41
6.1	Introduction	41
6.2	The Class Person	41
6.3	Functional Programming	43
6.4	A Hybrid Language	44
<b>7</b>	<b>A Little Scala</b>	47
7.1	Introduction	47
7.2	The Scala Environment	47
7.3	The Scala Shell	48
7.4	The Scala IDE	49
	7.4.1 Creating an IntelliJ Project	52
	7.4.2 Inside IntelliJ	53
	7.4.3 Creating a Module	53
7.5	Implementing the Object	56
7.6	Running the Application	56
	7.6.1 Scala Interpreter Console	57
7.7	Scala Classpath	58
7.8	Compiling and Executing Scala	58
7.9	Memory Management	60
	7.9.1 Why Have Automatic Memory Management?	60
	7.9.2 Memory Management in Scala	61

- 7.9.3 When Is Garbage Collection Performed? . . . . . 61
- 7.9.4 Checking the Available Memory . . . . . 61
- 8 Scala Building Blocks . . . . . 65**
  - 8.1 Introduction . . . . . 65
  - 8.2 Apps and Applications . . . . . 65
  - 8.3 The Basics of the Language . . . . . 67
    - 8.3.1 Some Terminology . . . . . 67
    - 8.3.2 The Message Passing Mechanism . . . . . 68
    - 8.3.3 The Statement Terminator . . . . . 68
- 9 Scala Classes . . . . . 69**
  - 9.1 Introduction . . . . . 69
  - 9.2 Classes . . . . . 69
    - 9.2.1 Class Definitions . . . . . 69
    - 9.2.2 Developing a Class Definition . . . . . 70
    - 9.2.3 Classes and Messages . . . . . 72
    - 9.2.4 Instances and Instance Variables . . . . . 72
    - 9.2.5 Classes and Inheritance . . . . . 73
    - 9.2.6 Instance Creation . . . . . 76
    - 9.2.7 Constructors . . . . . 77
    - 9.2.8 Auxiliary Constructors . . . . . 79
    - 9.2.9 Class Initialisation Behaviour . . . . . 80
    - 9.2.10 Review of Classes and Constructors . . . . . 81
  - 9.3 Case Classes . . . . . 82
    - 9.3.1 A Sample Class . . . . . 83
- 10 Scala Methods . . . . . 87**
  - 10.1 Introduction . . . . . 87
  - 10.2 Method Definitions . . . . . 87
    - 10.2.1 Method Parameters . . . . . 89
    - 10.2.2 Comments . . . . . 91
    - 10.2.3 The Local Variables Section . . . . . 93
    - 10.2.4 The Statements Section . . . . . 94
    - 10.2.5 The Return Operator . . . . . 94
    - 10.2.6 An Example Method . . . . . 95
    - 10.2.7 Overriding toString . . . . . 96
    - 10.2.8 Defining Property Methods . . . . . 98
  - 10.3 Named Parameters . . . . . 100
- 11 Packages and Encapsulation . . . . . 103**
  - 11.1 Introduction . . . . . 103
  - 11.2 Packages . . . . . 103

11.2.1	Declaring a Package . . . . .	104
11.2.2	Additional Package Definitions Options . . . . .	104
11.2.3	An Example Package . . . . .	109
11.2.4	Accessing Package Elements . . . . .	110
11.2.5	An Example of Using a Package . . . . .	111
11.3	Import Options . . . . .	112
11.4	Additional Import Features . . . . .	113
11.5	Package Objects . . . . .	114
11.6	Key Scala Packages . . . . .	115
11.7	Default Imports . . . . .	115
11.8	Encapsulation . . . . .	116
11.8.1	Scala Visibility Modifiers . . . . .	116
11.8.2	Private Modified . . . . .	117
11.8.3	Protected Modifier . . . . .	118
<b>12</b>	<b>Building a Class . . . . .</b>	<b>121</b>
12.1	Introduction . . . . .	121
12.2	Create a New Module . . . . .	121
12.3	Create a New Package . . . . .	123
12.4	Create a New Class . . . . .	124
12.5	Defining the Class . . . . .	125
12.6	Adding Behaviour . . . . .	126
12.7	Test Application . . . . .	126
12.8	Override ToString . . . . .	128
12.9	Extras . . . . .	130
<b>13</b>	<b>Classes, Inheritance and Abstraction . . . . .</b>	<b>131</b>
13.1	Introduction . . . . .	131
13.1.1	What Are Classes for? . . . . .	131
13.2	Inheritance Between Types . . . . .	132
13.3	Inheritance Between Classes . . . . .	133
13.3.1	The Role of a Subclass . . . . .	134
13.3.2	Capabilities of Classes . . . . .	135
13.3.3	Overriding Behaviour . . . . .	136
13.3.4	Protected Members . . . . .	137
13.4	Restricting a Subclass . . . . .	138
13.5	Abstract Classes . . . . .	139
13.6	The Super Keyword . . . . .	142
13.7	Scala Type Hierarchy . . . . .	143
13.8	Polymorphism . . . . .	144
<b>14</b>	<b>Objects and Instances . . . . .</b>	<b>147</b>
14.1	Introduction . . . . .	147
14.2	Singleton Objects . . . . .	147

- 14.3 Companion Objects . . . . . 149
  - 14.3.1 Companion Object Behaviour . . . . . 151
  - 14.3.2 A Object or an Instance . . . . . 152
- 15 Value Classes . . . . . 155**
  - 15.1 Introduction . . . . . 155
  - 15.2 Value Classes . . . . . 155
  - 15.3 Simple Value Type Example . . . . . 156
  - 15.4 Additional Value Class Concepts . . . . . 157
  - 15.5 Negating Value Classes . . . . . 158
- 16 Scala Constructs . . . . . 159**
  - 16.1 Introduction . . . . . 159
  - 16.2 Numbers and Numeric Operators . . . . . 159
    - 16.2.1 Numeric Values . . . . . 159
    - 16.2.2 Arithmetic Operators . . . . . 160
  - 16.3 Characters and Strings . . . . . 161
    - 16.3.1 Characters . . . . . 161
    - 16.3.2 Strings . . . . . 161
  - 16.4 Assignments . . . . . 162
  - 16.5 Variables . . . . . 164
    - 16.5.1 Temporary Variables . . . . . 164
    - 16.5.2 Pseudo-Variables . . . . . 165
    - 16.5.3 Variable Scope . . . . . 165
    - 16.5.4 Option, Some and None . . . . . 165
    - 16.5.5 Boolean Values . . . . . 167
    - 16.5.6 Literals . . . . . 167
  - 16.6 Messages and Message Selectors . . . . . 168
    - 16.6.1 Invoking Methods . . . . . 168
    - 16.6.2 Precedence . . . . . 168
  - 16.7 Summary . . . . . 169
- 17 Control and Iteration . . . . . 171**
  - 17.1 Introduction . . . . . 171
  - 17.2 Control Structures . . . . . 171
    - 17.2.1 The if Statement . . . . . 171
    - 17.2.2 If Returns a Value . . . . . 173
  - 17.3 Iteration . . . . . 174
    - 17.3.1 For Loops . . . . . 174
    - 17.3.2 For Until . . . . . 175
    - 17.3.3 For Loop with a Filter . . . . . 175
    - 17.3.4 Longhand for Loop . . . . . 176

- 17.3.5 For-Yield Loop . . . . . 176
- 17.3.6 While Loops . . . . . 177
- 17.3.7 Do Loops . . . . . 178
- 17.3.8 An Example of Loops . . . . . 179
- 17.4 Equality . . . . . 180
- 17.5 Recursion . . . . . 180
  - 17.5.1 The Match Expression . . . . . 183
- 18 Traits . . . . . 187**
  - 18.1 Introduction . . . . . 187
  - 18.2 What Are Traits? . . . . . 187
  - 18.3 Defining a Trait . . . . . 189
  - 18.4 Using a Trait . . . . . 190
  - 18.5 Abstract Trait Members . . . . . 193
  - 18.6 Dynamic Binding of Traits . . . . . 193
  - 18.7 Sealed Traits . . . . . 195
  - 18.8 Marker Traits . . . . . 195
  - 18.9 Trait Dependencies . . . . . 196
  - 18.10 To Trait or not to Trait . . . . . 198
- 19 Further Traits . . . . . 199**
  - 19.1 Introduction . . . . . 199
  - 19.2 Stackable Modifications . . . . . 199
  - 19.3 Fat Versus Thin Traits . . . . . 204
  - 19.4 Universal Traits . . . . . 205
  - 19.5 Traits for a Data Type . . . . . 207
  - 19.6 Single Abstract Method (SAM) Traits . . . . . 208
- 20 Arrays . . . . . 211**
  - 20.1 Introduction . . . . . 211
  - 20.2 Arrays . . . . . 211
    - 20.2.1 Arrays of Objects . . . . . 213
    - 20.2.2 Ragged Arrays . . . . . 215
  - 20.3 Creating Square Arrays . . . . . 217
  - 20.4 Looping Through Arrays . . . . . 218
  - 20.5 The Main Method Revisited . . . . . 219
- 21 Tuples . . . . . 223**
  - 21.1 Introduction . . . . . 223
  - 21.2 Tuples . . . . . 223
  - 21.3 Tuple Characteristics . . . . . 224
  - 21.4 Tuple Classes . . . . . 224
  - 21.5 Creating a Tuple . . . . . 224

- 21.6 Working with Tuples . . . . . 225
- 21.7 Iterating Over a Tuple . . . . . 226
- 21.8 Element Extraction . . . . . 227
- 22 Functional Programming in Scala . . . . . 229**
  - 22.1 Introduction . . . . . 229
  - 22.2 Scala as a Functional Language . . . . . 229
  - 22.3 Defining Scala Functions . . . . . 230
  - 22.4 Class, Objects, Methods and Functions . . . . . 233
  - 22.5 Lifting a Method . . . . . 235
  - 22.6 Single Abstract Method Traits . . . . . 235
  - 22.7 Closure . . . . . 237
  - 22.8 Referential Transparency . . . . . 239
- 23 Higher-Order Functions . . . . . 241**
  - 23.1 Introduction . . . . . 241
  - 23.2 Higher-Order Function Concepts . . . . . 241
  - 23.3 Scala Higher-Order Functions . . . . . 242
  - 23.4 Using Higher-Order Functions . . . . . 244
  - 23.5 Higher-Order Functions in Scala Collections . . . . . 245
- 24 Partially Applied Functions and Currying . . . . . 247**
  - 24.1 Introduction . . . . . 247
  - 24.2 Partially Applied Functions . . . . . 248
  - 24.3 Currying . . . . . 250
    - 24.3.1 Introduction to Currying . . . . . 250
    - 24.3.2 Defining Multiple Parameter List Functions . . . . . 250
    - 24.3.3 Using Curried Functions . . . . . 251
    - 24.3.4 Building Domain-Specific Languages . . . . . 252
- 25 Scala Collections Framework . . . . . 255**
  - 25.1 Introduction . . . . . 255
  - 25.2 What Are Collections? . . . . . 255
  - 25.3 Scala Collections . . . . . 256
    - 25.3.1 Package Scala.Collection . . . . . 258
    - 25.3.2 Common Seq Behaviour . . . . . 259
    - 25.3.3 Common Set Behaviour . . . . . 260
    - 25.3.4 Common Map Behaviour . . . . . 260
- 26 Immutable Lists and Maps . . . . . 261**
  - 26.1 Introduction . . . . . 261
  - 26.2 The Immutable List Collection . . . . . 261
    - 26.2.1 List Creation . . . . . 261
    - 26.2.2 List Concatenation . . . . . 262

26.2.3	List Operations . . . . .	264
26.2.4	List Processing . . . . .	265
26.2.5	Further List Processing . . . . .	265
26.2.6	Pattern Matching . . . . .	269
26.2.7	Converting to a List . . . . .	270
26.2.8	Lists of User Defined Types . . . . .	270
26.3	The Immutable Map Type . . . . .	272
<b>27</b>	<b>Immutable and Mutable Collection Packages . . . . .</b>	<b>275</b>
27.1	Introduction . . . . .	275
27.2	Package <code>Scala.Collection.Immutable</code> . . . . .	275
27.2.1	Sequences . . . . .	275
27.2.2	Sets . . . . .	278
27.2.3	Maps . . . . .	279
27.3	Package <code>Scala.Collection.Mutable</code> . . . . .	279
27.3.1	<code>ArrayBuffer</code> . . . . .	280
27.3.2	<code>ListBuffer</code> . . . . .	281
27.3.3	<code>LinkedList</code> . . . . .	282
27.3.4	<code>Stack</code> . . . . .	282
27.3.5	<code>HashMap</code> . . . . .	283
27.4	Generic Collections . . . . .	284
27.5	Summary . . . . .	285
<b>28</b>	<b>Type Parameterisation . . . . .</b>	<b>287</b>
28.1	Introduction . . . . .	287
28.2	The Set Class . . . . .	287
28.3	Adding Type Parameterisation . . . . .	288
28.3.1	The <code>MyQueue Mutable Class</code> . . . . .	288
28.3.2	The <code>Immutable Queue Class</code> . . . . .	290
28.4	Variance . . . . .	292
28.5	Lower and Upper Bounds . . . . .	293
28.6	Combining Variance and Bounds . . . . .	293
<b>29</b>	<b>Further Language Constructs . . . . .</b>	<b>295</b>
29.1	Introduction . . . . .	295
29.2	Implicit Conversions . . . . .	295
29.3	Implicit Parameters . . . . .	297
29.4	Implicit Objects . . . . .	299
29.5	Implicit Classes . . . . .	301
29.6	Scala Annotations . . . . .	303
29.7	Type Declarations . . . . .	305
29.8	Enumerations . . . . .	306
29.9	Lazy Evaluation . . . . .	311

- 30 Exception Handling** . . . . . 313
  - 30.1 Introduction . . . . . 313
  - 30.2 What Is an Exception? . . . . . 313
  - 30.3 What Is Exception Handling? . . . . . 315
  - 30.4 Throwing an Exception . . . . . 316
  - 30.5 Catching an Exception . . . . . 317
  - 30.6 Try Block Returns a Value . . . . . 320
  - 30.7 Defining an Exception . . . . . 321
  - 30.8 A More Functional Approach . . . . . 322
  - 30.9 The Try Type . . . . . 325
  
- 31 Akka Actors** . . . . . 327
  - 31.1 Introduction . . . . . 327
  - 31.2 The Actor Model . . . . . 327
  - 31.3 Some Terminology . . . . . 328
  - 31.4 Scala Threads . . . . . 329
  - 31.5 Akka Scala Actor Library . . . . . 330
  - 31.6 Concurrent Hello World . . . . . 332
  - 31.7 Concurrent Actors . . . . . 335
  - 31.8 The Akka Actor API . . . . . 336
  - 31.9 Actor Lifecycle . . . . . 337
  - 31.10 Akka Configuration . . . . . 339
  - 31.11 Actor DSL . . . . . 343
  
- 32 Further Akka Actors** . . . . . 345
  - 32.1 Introduction . . . . . 345
  - 32.2 Generating a Result from an Actor . . . . . 345
  - 32.3 Futures . . . . . 348
  - 32.4 Dispatchers . . . . . 351
  - 32.5 Actor Hierarchies . . . . . 354
  - 32.6 Actor Supervision . . . . . 355
  - 32.7 Good Practices . . . . . 360
  
- 33 Scala and JDBC Database Access** . . . . . 361
  - 33.1 Introduction . . . . . 361
  - 33.2 Why JDBC? . . . . . 361
  - 33.3 What Is JDBC? . . . . . 362
  - 33.4 Working with JDBC . . . . . 364
  - 33.5 The Database Driver . . . . . 365
  - 33.6 Registering Drivers . . . . . 366
  - 33.7 Setting Up MySQL . . . . . 367
  - 33.8 Setting Up the Database . . . . . 368
    - 33.8.1 Starting/Stopping/Connecting to MySQL . . . . . 368
  - 33.9 Creating a Database . . . . . 369

- 33.9.1 Adding a User . . . . . 369
- 33.9.2 Selecting to Work with a Database . . . . . 370
- 33.9.3 Creating a Table . . . . . 370
- 33.9.4 Adding Data to a Table . . . . . 371
- 33.10 Opening a Connection . . . . . 372
- 33.11 Inserting into a Table . . . . . 374
- 33.12 Obtaining Data from a Database . . . . . 376
- 33.13 Update an Existing Row . . . . . 378
- 33.14 Deleting from a Table . . . . . 379
- 33.15 Creating a Table . . . . . 379
- 33.16 Stored Procedures . . . . . 381
- 33.17 JDBC Data Sources . . . . . 381
- 33.18 Connection Pooling . . . . . 384
- 33.19 JDBC Metadata . . . . . 386
  - 33.19.1 DatabaseMetaData . . . . . 387
  - 33.19.2 ResultSetMetaData . . . . . 388
- 34 Scala Style Database Access . . . . . 389**
  - 34.1 Introduction . . . . . 389
  - 34.2 Slick . . . . . 389
  - 34.3 Querulus . . . . . 390
  - 34.4 Squeryl . . . . . 391
  - 34.5 O/R Broker . . . . . 392
- 35 Slick: Functional Relational Mapping for Scala . . . . . 395**
  - 35.1 Introduction . . . . . 395
  - 35.2 Obtaining Slick . . . . . 395
  - 35.3 Connecting to a Database . . . . . 397
  - 35.4 Mapping Tables to Scala Types . . . . . 398
  - 35.5 Table Query Interface . . . . . 400
  - 35.6 Creating Tables . . . . . 401
  - 35.7 Inserting Data . . . . . 401
  - 35.8 Composing Database I/O Actions . . . . . 402
  - 35.9 Example Setting Up a Database . . . . . 402
  - 35.10 Querying for Data . . . . . 404
  - 35.11 Updating, Upserting and Deleting Data . . . . . 407
- 36 Testing . . . . . 409**
  - 36.1 Introduction . . . . . 409
  - 36.2 Types of Testing . . . . . 409
  - 36.3 What Should Be Tested? . . . . . 410
  - 36.4 Types of Testing . . . . . 411
    - 36.4.1 Unit Testing . . . . . 412
    - 36.4.2 Integration Testing . . . . . 413

- 36.4.3 System Testing . . . . . 413
- 36.4.4 Installation Testing . . . . . 413
- 36.4.5 Smoke Tests . . . . . 414
- 36.5 Automating Testing . . . . . 414
- 37 Scala Testing . . . . . 415**
- 37.1 Introduction . . . . . 415
- 37.2 Scala Runtime Test Facilities . . . . . 415
  - 37.2.1 Validation Checks . . . . . 415
  - 37.2.2 Using Require and Assert . . . . . 416
- 37.3 Test Libraries in Scala . . . . . 417
  - 37.3.1 ScalaTest . . . . . 417
  - 37.3.2 Spec . . . . . 417
  - 37.3.3 ScalaCheck . . . . . 418
- 37.4 Scalatest Testing Framework . . . . . 418
  - 37.4.1 Setting up Your Scala Project . . . . . 418
  - 37.4.2 ScalaTest and JUnit . . . . . 419
  - 37.4.3 Scala Test and Functional Test Suites . . . . . 424
- 37.5 Scalatest and Feature Tests . . . . . 425
- 37.6 Test-Driven Development . . . . . 427
  - 37.6.1 The TDD Cycle . . . . . 427
  - 37.6.2 Test Complexity . . . . . 428
  - 37.6.3 Refactoring . . . . . 429
- 38 Play Framework . . . . . 431**
- 38.1 Introduction . . . . . 431
- 38.2 Introduction to Play . . . . . 431
  - 38.2.1 Working with a Web Application . . . . . 432
  - 38.2.2 How Play Changes the Stack . . . . . 432
- 38.3 Starting with Play . . . . . 434
  - 38.3.1 Download and Install Play . . . . . 434
  - 38.3.2 Using a Play Starter Project . . . . . 434
  - 38.3.3 Starting the Application . . . . . 436
- 38.4 Examining the Structure of the Application . . . . . 438
- 38.5 Editing the Application . . . . . 439
  - 38.5.1 Working with Your IDE . . . . . 439
  - 38.5.2 Importing into IntelliJ . . . . . 439
  - 38.5.3 Working with the Application . . . . . 441
- 38.6 Model–View–Controller . . . . . 442
- 38.7 Exploring the Play Application . . . . . 444
- 39 RESTful Services . . . . . 447**
- 39.1 Introduction . . . . . 447
- 39.2 RESTful Services . . . . . 447

- 39.3 A RESTful API . . . . . 448
- 39.4 Creating the RESTful Web Application . . . . . 449
- 39.5 JavaScript and jQuery . . . . . 454
- 39.6 The jQuery Client . . . . . 455
  - 39.6.1 Obtaining jQuery . . . . . 455
  - 39.6.2 Adding jQuery to the Application . . . . . 456
- 40 Scalaz** . . . . . 463
  - 40.1 Introduction . . . . . 463
  - 40.2 Obtaining Scalaz . . . . . 463
  - 40.3 Scalaz Overview . . . . . 464
  - 40.4 Some Useful Typeclasses . . . . . 465
    - 40.4.1 Equals Typeclass . . . . . 465
    - 40.4.2 Order Typeclass . . . . . 465
  - 40.5 Standard Class Extensions . . . . . 466
    - 40.5.1 Extensions to Option . . . . . 466
    - 40.5.2 Boolean Extensions . . . . . 468
    - 40.5.3 Extensions to List . . . . . 469
    - 40.5.4 Extensions for Map . . . . . 470
    - 40.5.5 Extensions to String . . . . . 471
  - 40.6 The Other Either . . . . . 471
  - 40.7 Tagging . . . . . 472
- 41 GUIs in Scala Swing** . . . . . 475
  - 41.1 Introduction . . . . . 475
  - 41.2 Windows as Objects . . . . . 475
  - 41.3 Windows in Scala . . . . . 476
  - 41.4 Scala Swing . . . . . 477
  - 41.5 Scala Swing Packages . . . . . 478
  - 41.6 Swing Scala Worked Examples . . . . . 480
    - 41.6.1 Simple Hello World UI . . . . . 480
    - 41.6.2 Panels and UI Layout . . . . . 481
    - 41.6.3 Working with a BorderLayout . . . . . 482
    - 41.6.4 Working with a BoxPanel . . . . . 484
    - 41.6.5 Displaying a Table . . . . . 485
- 42 User Input in Scala Swing** . . . . . 489
  - 42.1 Introduction . . . . . 489
  - 42.2 Handling User Input . . . . . 489
    - 42.2.1 Scala Swing Actions . . . . . 492
    - 42.2.2 Working with Menus . . . . . 494
  - 42.3 A Simple GUI Example . . . . . 496

- 43 Scala Build Tools** . . . . . 501
  - 43.1 Introduction . . . . . 501
  - 43.2 Why We Need a Build Tool . . . . . 501
  - 43.3 Maven . . . . . 502
    - 43.3.1 Maven Repositories . . . . . 504
    - 43.3.2 The Maven POM . . . . . 504
    - 43.3.3 Scala and Maven . . . . . 506
    - 43.3.4 Maven Lifecycle Commands . . . . . 508
  - 43.4 SBT . . . . . 508
    - 43.4.1 Creating an SBT Project . . . . . 510
    - 43.4.2 SBT Lifecycle Commands . . . . . 512
  
- 44 Scala & Java Interoperability** . . . . . 515
  - 44.1 Introduction . . . . . 515
  - 44.2 A Simple Example . . . . . 515
  - 44.3 Inheritance . . . . . 517
  - 44.4 Issues . . . . . 517
    - 44.4.1 Scala Objects . . . . . 518
    - 44.4.2 Companion Modules . . . . . 519
    - 44.4.3 Traits . . . . . 521
  - 44.5 Functions . . . . . 524
  - 44.6 Collection Classes . . . . . 528
  - 44.7 Implementing a Java Interface . . . . . 530