CLOJURE FOR THE BRAVE AND TRUE

learn the ultimate language and become a better programmer

Daniel Higginbotham

CONTENTS IN DETAIL

FOREWORD by Alan Dipert xvii

ACKNOWLEDGMENTS xix

INTRODUCTION xxi

Learning a New Programming Language: A Journey Through the Four Labyrinths. xxii
How This Book Is Organized xxii
Part I: Environment Setup xxii
Part II: Language Fundamentals. xxiii
Part III: Advanced Topics xxiv
The Code xxv
The Journey Begins! xxv

PART I: ENVIRONMENT SETUP

1
BUILDING, RUNNING, AND THE REPL 3

First Things First: What Is Clojure? 4
Leiningen 5
Creating a New Clojure Project 5
Running the Clojure Project 6
Building the Clojure Project 7
Using the REPL 7
Clojure Editors 9
Summary 9

2
HOW TO USE EMACS, AN EXCELLENT CLOJURE EDITOR 11

Installation. 12
Configuration 13
Emacs Escape Hatch 14
Emacs Buffers 14
Working with Files 15
Key Bindings and Modes 17
Emacs Is a Lisp Interpreter 17
Modes 18
Installing Packages 19
Core Editing Terminology and Key Bindings 19
Point 20
Movement 20
Selection with Regions 20
Killing and the Kill Ring 21
Editing and Help 22
Using Emacs with Clojure 23
Fire Up Your REPL! 23
Interlude: Emacs Windows and Frames 24

A Cornucopia of Useful Key Bindings . . . 25
How to Handle Errors . . . 27
Paredit . . . 28
Continue Learning . . . 30
Summary . . . 31

PART II: LANGUAGE FUNDAMENTALS

3
DO THINGS: A CLOJURE CRASH COURSE 35

Syntax . . . 36
Forms . . . 36
Control Flow . . . 37
Naming Values with def . . . 40
Data Structures . . . 41
Numbers . . . 42
Strings . . . 42
Maps . . . 43
Keywords . . . 44
Vectors . . . 45
Lists . . . 45
Sets . . . 46
Simplicity . . . 48
Functions . . . 48
Calling Functions . . . 48
Function Calls, Macro Calls, and Special Forms . . . 50
Defining Functions . . . 51
Anonymous Functions . . . 57
Returning Functions . . . 58
Pulling It All Together . . . 59
The Shire's Next Top Model . . . 59
let . . . 61
loop . . . 63
Regular Expressions . . . 64
Symmetrizer . . . 65
Better Symmetrizer with reduce . . . 66
Hobbit Violence . . . 67
Summary . . . 69
Exercises . . . 69

4
CORE FUNCTIONS IN DEPTH 71

Programming to Abstractions . . . 72
Treating Lists, Vectors, Sets, and Maps as Sequences . . . 73
first, rest, and cons . . . 74
Abstraction Through Indirection . . . 77
Seq Function Examples . . . 79
map . . . 79
reduce . . . 80
take, drop, take-while, and drop-while . . . 81

filter and some . . . 83
sort and sort-by . . . 84
concat . . . 84
Lazy Seqs . . . 84
Demonstrating Lazy Seq Efficiency . . . 84
Infinite Sequences . . . 87
The Collection Abstraction . . . 88
into . . . 88
conj . . . 90
Function Functions . . . 90
apply . . . 91
partial . . . 91
complement . . . 92
A Vampire Data Analysis Program for the FWPD . . . 93
Summary . . . 96
Exercises . . . 96

5
FUNCTIONAL PROGRAMMING 97

Pure Functions: What and Why . . . 98
Pure Functions Are Referentially Transparent . . . 98
Pure Functions Have No Side Effects . . . 99
Living with Immutable Data Structures . . . 100
Recursion Instead of for/while . . . 100
Function Composition Instead of Attribute Mutation . . . 103
Cool Things to Do with Pure Functions . . . 105
comp . . . 105
memoize . . . 107
Peg Thing . . . 108
Playing . . . 108
Code Organization . . . 110
Creating the Board . . . 111
Moving Pegs . . . 117
Rendering and Printing the Board . . . 120
Player Interaction . . . 121
Summary . . . 124
Exercises . . . 124

6
ORGANIZING YOUR PROJECT: A LIBRARIAN'S TALE 125

Your Project as a Library . . . 126
Storing Objects with def . . . 127
Creating and Switching to Namespaces . . . 129
refer . . . 130
alias . . . 132
Real Project Organization . . . 133
The Relationship Between File Paths and Namespace Names . . . 133
Requiring and Using Namespaces . . . 134
The ns Macro . . . 138
To Catch a Burglar . . . 140
Summary . . . 144

7
CLOJURE ALCHEMY: READING, EVALUATION, AND MACROS 147

An Overview of Clojure's Evaluation Model 148
The Reader 153
Reading 153
Reader Macros 154
The Evaluator 155
These Things Evaluate to Themselves 156
Symbols 156
Lists 159
Macros 160
Syntactic Abstraction and the -> Macro 163
Summary 164
Exercises 164

8
WRITING MACROS 165

Macros Are Essential 166
Anatomy of a Macro 167
Building Lists for Evaluation 168
Distinguishing Symbols and Values 168
Simple Quoting 169
Syntax Quoting 171
Using Syntax Quoting in a Macro 173
Refactoring a Macro and Unquote Splicing 174
Things to Watch Out For 176
Variable Capture 176
Double Evaluation 178
Macros All the Way Down 179
Brews for the Brave and True 180
Validation Functions 180
if-valid 182
Summary 184
Exercises 184

PART III: ADVANCED TOPICS

9
THE SACRED ART OF CONCURRENT AND PARALLEL PROGRAMMING 189

Concurrency and Parallelism Concepts 190
Managing Multiple Tasks vs. Executing Tasks Simultaneously 190
Blocking and Asynchronous Tasks 191
Concurrent Programming and Parallel Programming 191
Clojure Implementation: JVM Threads 191
What's a Thread? 192
The Three Goblins: Reference Cells, Mutual Exclusion, and Dwarven Berserkers 193

Futures, Delays, and Promises 196
Futures 196
Delays 198
Promises 200
Rolling Your Own Queue 202
Summary 205
Exercises 206

10
CLOJURE METAPHYSICS:
ATOMS, REFS, VARS, AND CUDDLE ZOMBIES 207

Object-Oriented Metaphysics 208
Clojure Metaphysics 210
Atoms 212
Watches and Validators 215
Watches 215
Validators 217
Refs 218
Modeling Sock Transfers 218
commute 221
Vars 223
Dynamic Binding 223
Altering the Var Root 227
Stateless Concurrency and Parallelism with pmap 228
Summary 232
Exercises 232

11
MASTERING CONCURRENT PROCESSES WITH CORE.ASYNC 233

Getting Started with Processes 234
Buffering 236
Blocking and Parking 237
thread 238
The Hot Dog Machine Process You've Been Longing For 239
alts!! 241
Queues 243
Escape Callback Hell with Process Pipelines 244
Additional Resources 245
Summary 245

12
WORKING WITH THE JVM 247

The JVM 248
Writing, Compiling, and Running a Java Program 250
Object-Oriented Programming in the World's Tiniest Nutshell 250
Ahoy, World 251
Packages and Imports 253
JAR Files 255
clojure.jar 255
Clojure App JARs 257

Java Interop 257
Interop Syntax 258
Creating and Mutating Objects 259
Importing 260
Commonly Used Java Classes 261
The System Class 261
The Date Class 262
Files and Input/Output 262
Resources 264
Summary 264

13
CREATING AND EXTENDING ABSTRACTIONS WITH MULTIMETHODS, PROTOCOLS, AND RECORDS 265

Polymorphism 266
Multimethods 266
Protocols 269
Records 272
Further Study 275
Summary 275
Exercises 275

A
BUILDING AND DEVELOPING WITH LEININGEN 277

The Artifact Ecosystem 277
Identification 278
Dependencies 278
Plug-Ins 279
Summary 280

B
BOOT, THE FANCY CLOJURE BUILD FRAMEWORK 281

Boot's Abstractions 282
Tasks 282
The REPL 284
Composition and Coordination 285
Handlers and Middleware 285
Tasks Are Middleware Factories 287
Filesets 288
Next Steps 289

FAREWELL! 291

INDEX 293