Contents

Preface xiii

I The Language 1

1 Getting Started 3

1.1 Chunks 4
1.2 Global Variables 5
1.3 Some Lexical Conventions 6
1.4 The Stand-Alone Interpreter 7

2 Types and Values 9

2.1 Nil 10
2.2 Booleans 10
2.3 Numbers 10
2.4 Strings 10
2.5 Tables 13
2.6 Functions 16
2.7 Userdata and Threads 16

3 Expressions 17

3.1 Arithmetic Operators 17
3.2 Relational Operators 17
3.3 Logical Operators 18
3.4 Concatenation 19
3.5 Precedence 19
3.6 Table Constructors 20

4 Statements 23

4.1 Assignment 23
4.2 Local Variables and Blocks 24
4.3 Control Structures 26
4.4 **break** and **return** 29

5 Functions 31
5.1 Multiple Results 32
5.2 Variable Number of Arguments 35
5.3 Named Arguments 37

6 More about Functions 39
6.1 Closures 41
6.2 Non-Global Functions 44
6.3 Proper Tail Calls 46

7 Iterators and the Generic for 49
7.1 Iterators and Closures 49
7.2 The Semantics of the Generic **for** 51
7.3 Stateless Iterators 52
7.4 Iterators with Complex State 54
7.5 True Iterators 55

8 Compilation, Execution, and Errors 57
8.1 The `require` Function 60
8.2 C Packages 61
8.3 Errors 62
8.4 Error Handling and Exceptions 64
8.5 Error Messages and Tracebacks 65

9 Coroutines 67
9.1 Coroutine Basics 67
9.2 Pipes and Filters 70
9.3 Coroutines as Iterators 73
9.4 Non-Preemptive Multithreading 75

10 Complete Examples 81
10.1 Data Description 81
10.2 Markov Chain Algorithm 84

II Tables and Objects 89

11 Data Structures 91
11.1 Arrays 91
11.2 Matrices and Multi-Dimensional Arrays 92
11.3 Linked Lists 93
11.4 Queues and Double Queues 94
11.5 Sets and Bags 95
11.6 String Buffers 96

12 Data Files and Persistence 99
12.1 Serialization 101

13 Metatables and Metamethods 107
13.1 Arithmetic Metamethods 107
13.2 Relational Metamethods 110
13.3 Library-Defined Metamethods 111
13.4 Table-Access Metamethods 112

14 The Environment 119
14.1 Accessing Global Variables with Dynamic Names 119
14.2 Declaring Global Variables 121
14.3 Non-Global Environments 122

15 Packages 125
15.1 The Basic Approach 125
15.2 Privacy 127
15.3 Packages and Files 128
15.4 Using the Global Table 129
15.5 Other Facilities 131

16 Object-Oriented Programming 135
16.1 Classes 137
16.2 Inheritance 138
16.3 Multiple Inheritance 140
16.4 Privacy 142
16.5 The Single-Method Approach 144

17 Weak Tables 147
17.1 Memoize Functions 149
17.2 Object Attributes 150
17.3 Revisiting Tables with Default Values 151

III The Standard Libraries 153

18 The Mathematical Library 155

19 The Table Library 157
19.1 Array Size 157
19.2 Insert and Remove 158
19.3 Sort 159

20 The String Library 161
20.1 Pattern-Matching Functions 163
20.2 Patterns 164

20.3 Captures 168
20.4 Tricks of the Trade 172

21 The I/O Library 179
21.1 The simple I/O model 179
21.2 The Complete I/O Model 182
21.3 Other Operations on Files 186

22 The Operating System Library 187
22.1 Date and Time 187
22.2 Other System Calls 189

23 The Debug Library 191
23.1 Introspective Facilities 191
23.2 Hooks 195
23.3 Profiles 196

IV The C API 199

24 An Overview of the C API 201
24.1 A First Example 202
24.2 The Stack 204
24.3 Error Handling with the C API 210

25 Extending your Application 213
25.1 Table Manipulation 215
25.2 Calling Lua Functions 218
25.3 A Generic Call Function 220

26 Calling C from Lua 223
26.1 C Functions 223
26.2 C Libraries 225

27 Techniques for Writing C Functions 229
27.1 Array Manipulation 229
27.2 String Manipulation 231
27.3 Storing State in C functions 233

28 User-Defined Types in C 239
28.1 Userdata 239
28.2 Metatables 242
28.3 Object-Oriented Access 244
28.4 Array Access 246
28.5 Light Userdata 247

29 Managing Resources 249
29.1 A Directory Iterator 249
29.2 An XML Parser 252

Index 261