Reusable software

The Base object-oriented component libraries

Bertrand Meyer

ISE, Santa Barbara
SOL, Paris
Contents

Preface: A Linnaean reconstruction of software fundamentals vii

Techniques and principles viii
Classes and features ix
Doubts and certainties ix
Language independence x
From reuse consumer to reuse producer xi
Acknowledgments xii

Contents xiii

PART A TECHNIQUES AND PRINCIPLES 1

Chapter 1 Introduction to the Base libraries 3
1.1 Overview 3
1.2 Other libraries 4
1.3 Contents of the libraries and order of presentation 6
1.4 Two notes on the Kernel library 9

Chapter 2 Building libraries: techniques 11
2.1 Overview 11
2.2 The modular structures 11
2.3 The run-time model 15
2.4 Information hiding 21
2.5 Assertions and design by contract 23
Chapter 2 Principles of library design

2.6 Interface documentation
2.7 Genericity, inheritance and typing
2.8 Constants and shared information
2.9 Library management
2.10 Environment support

Chapter 3 Principles of library design

3.1 Overview
3.2 The library challenge
3.3 Some general principles
3.4 Design by Contract
3.5 Documenting library classes
3.6 The shopping list approach: how big should a library class be?
3.7 The size of feature interfaces
3.8 Naming issues
3.9 Other conventions
3.10 Using inheritance
3.11 The ideal library developer
3.12 Further reading

PART B PRESENTATION OF THE LIBRARIES

Chapter 4 Abstract container structures: the taxonomy

4.1 Overview
4.2 The three criteria
4.3 Container operations
4.4 The access hierarchy
4.5 The storage hierarchy
4.6 The traversal hierarchy

Chapter 5 Mathematical properties

5.1 Overview
5.2 Class usage
5.3 Partial and total order relations
5.4 Numeric objects
5.5 Hashable objects
Chapter 6 Linear structures: sequences, chains, lists, circular chains

6.1 Overview 139
6.2 The inheritance structure 141
6.3 Higher-level traversal classes 141
6.4 Sequences and chains 147
6.5 Lists and circular structures 149
6.6 Implementations 152
6.7 Sorted linear structures 157

Chapter 7 Dispenser structures: stacks, queues, priority lists

7.1 Overview 159
7.2 The notion of dispenser 160
7.3 Stacks 161
7.4 Types of queues 162
7.5 Priority queues 162

Chapter 8 Trees

8.1 Basic terminology 163
8.2 Recursive trees 165
8.3 Binary trees 167
8.4 Cursor trees 169

Chapter 9 Sets and hash tables

9.1 Overview 173
9.2 Hash tables 173
9.3 Sets 178

Chapter 10 Iteration

10.1 Overview 181
10.2 The notion of iterator 181
10.3 Simple examples 183
10.4 A discussion 187
10.5 Using the iteration library 189
10.6 Building and using iterators 195

Chapter 11 Lexical analysis: the Lex library

11.1 Overview 201
11.2 Aims and scope of the Lex library 202
11.3 Tokens 204
11.4 Building and using lexical analyzers 204
11.5 Lexical grammar files and class SCANNING 205
11.6 Analyzing input 207
11.7 Regular expressions 209
11.8 Using METALEX to build a lexical analyzer 214
11.9 Building a lexical analyzer with LEX_BUILDER 216

Chapter 12 Parsing: The Parse Library 219
12.1 Overview 219
12.2 Why use the Parse library? 220
12.3 Aim and scope of the Parse library 221
12.4 Library classes 222
12.5 Examples 223
12.6 Constructs and productions 223
12.7 Parsing concepts 226
12.8 Preparing grammars 228
12.9 Interface to lexical analysis 231
12.10 Specifying the semantics 233
12.11 How parsing works 236
12.12 Building a document processor 238
12.13 Future work 240

Chapter 13 The Kernel Library 243
13.1 Overview 243
13.2 Clusters of the Kernel library 243
13.3 Universal classes and their features 244
13.4 Language-related facilities 248
13.5 Files, input, output 252
13.6 Persistence, storage and retrieval 254
13.7 Access to internal properties 256

PART C CLASS REFERENCE 261

Chapter 14 Classes for abstract container structures 263
14.1 Overview 263
14.2 Class CONTAINER 265
14.3 Class BOX 266
14.4 Class FINITE 267
14.5 Class BOUNDED 268
14.6 Class FIXED 269
14.7 Class RESIZABLE 270
14.8 Class INFINITE 271
14.9 Class COLLECTION 272
14.10 Class BAG 274
14.11 Class TABLE 275
14.12 Class INDEXABLE 277
14.13 Class ACTIVE 278
14.14 Class CURSOR_STRUCTURE 279
14.15 Class DISPENSER 281
14.16 Class TRAVERSABLE 283
14.17 Class HIERARCHICAL 284
14.18 Class LINEAR 285

Chapter 15 Classes for mathematical properties 287
15.1 Overview 287
15.2 Class PART_COMPARABLE 287
15.3 Class COMPARABLE 288
15.4 Class NUMERIC 289
15.5 Class HASHABLE 290

Chapter 16 Classes for linear structures 291
16.1 Overview 291
16.2 Class BILINEAR 293
16.3 Class SEQUENCE 294
16.4 Classes CHAIN and LIST 295
16.5 Classes LINKED_LIST and TWO_WAY_LIST 299
16.6 Class ARRAYED_LIST 304
16.7 Class MULTI_ARRAY_LIST 307
16.8 Class FIXED_LIST 309
16.9 Class PART_SORTED_LIST 311
16.10 Classes SORTED_LIST and SORTED_TWO_WAY_LIST 312
16.11 Class CIRCULAR 318
16.12 Classes LINKED_CIRCULAR and TWO_WAY_CIRCULAR 322
16.13 Class ARRAYED_CIRCULAR 324
16.14 Classes SORTED_STRUCT and COMPARABLE_STRUCT 326
16.15 Class COUNTABLE_SEQUENCE 327
| 16.16 Class | LINKABLE | 329 |
| 16.17 Class | BI_LINKABLE | 330 |

**Chapter 17 Dispenser classes**

| 17.1 Overview | 331 |
| 17.2 Class | STACK | 333 |
| 17.3 Class | ARRAYED_STACK | 335 |
| 17.4 Class | BOUNDED_STACK | 337 |
| 17.5 Class | LINKED_STACK | 339 |
| 17.6 Class | QUEUE | 340 |
| 17.7 Class | ARRAYED_QUEUE | 342 |
| 17.8 Class | BOUNDED_QUEUE | 344 |
| 17.9 Class | LINKED_QUEUE | 346 |
| 17.10 Classes | HEAP_PRIORITY_QUEUE, LINKED_PRIORITY_QUEUE and PRIORITY_QUEUE | 347 |

**Chapter 18 Tree classes**

| 18.1 Overview | 349 |
| 18.2 Class | TREE | 350 |
| 18.3 Classes | LINKED_TREE and TWO_WAY_TREE | 354 |
| 18.4 Class | ARRAYED_TREE | 361 |
| 18.5 Class | FIXED_TREE | 363 |
| 18.6 Class | BINARY_TREE | 365 |
| 18.7 Class | BINARY_SEARCH_TREE | 368 |
| 18.8 Class | CURSOR_TREE | 369 |
| 18.9 Classes | LINKED_CURSOR_TREE and TWO_WAY_CURSOR_TREE | 372 |
| 18.10 Class | COMPACT_CURSOR_TREE | 376 |

**Chapter 19 Set and hash table classes**

| 19.1 Overview | 379 |
| 19.2 Class | HASH_TABLE | 381 |
| 19.3 Class | SET | 383 |
| 19.4 Class | SUBSET | 385 |
| 19.5 Class | LINKED_SET | 386 |
| 19.6 Class | COMPARABLE_SET | 388 |
19.7 Class `PART_SORTED_SET` 389
19.8 Classes `BINARY_SEARCH_TREE_SET` and `TWO WAY_SORTED_SET` 391

**Chapter 20 Iteration classes** 393
20.1 Overview 393
20.2 Class `ITERATOR` 393
20.3 Class `LINEAR_ITERATOR` 396
20.4 Class `TWO WAY_CHAIN_ITERATOR` 398
20.5 Class `CURSOR_TREE_ITERATOR` 400

**Chapter 21 Lexical analysis classes** 403
21.1 Overview 403
21.2 Class `SCANNING` 403
21.3 Class `LEXICAL` 405
21.4 Class `TOKEN` 406
21.5 Class `METALEX` 407
21.6 Class `LEX.Builder` 408

**Chapter 22 Parsing classes** 413
22.1 Overview 413
22.2 Class `CONSTRUCT` 413
22.3 Class `AGGREGATE` 417
22.4 Class `CHOICE` 418
22.5 Class `REPLICATION` 419
22.6 Class `TERMINAL` 420
22.7 Class `KEYWORD` 421
22.8 Class `L_INTERFACE` 422
22.9 Class `INPUT` 423

**Chapter 23 Kernel library classes** 425
23.1 Overview 425
23.2 Classes `ANY`, `GENERAL` and `PLATFORM` 425
23.3 Class `ARRAY` 428
23.4 Class `STRING` 430
23.5 Class `SPECIAL` 434
23.6 Class `BASIC_ROUTINES` 435
23.7 Classes `DOUBLE_MATH` and `SINGLE_MATH` 436
23.8 Class `PRIMES` 437
23.9 Class RANDOM
23.10 Class ASCII
23.11 Class FILE
23.12 Class UNIX_FILE
23.13 Class UNIX_FILE_INFO
23.14 Class UNIX_STD
23.15 Class DIRECTORY
23.16 Class STD_FILES
23.17 Class STORABLE
23.18 Class EXCEPTIONS
23.19 Class UNIX_SIGNALS
23.20 Class MEMORY
23.21 Class MEM_INFO
23.22 Class GC_INFO
23.23 Class ARGUMENTS
23.24 Class INTERNAL
23.25 Class BOOLEAN
23.26 Class CHARACTER
23.27 Class INTEGER
23.28 Classes REAL and DOUBLE

Bibliography

Index