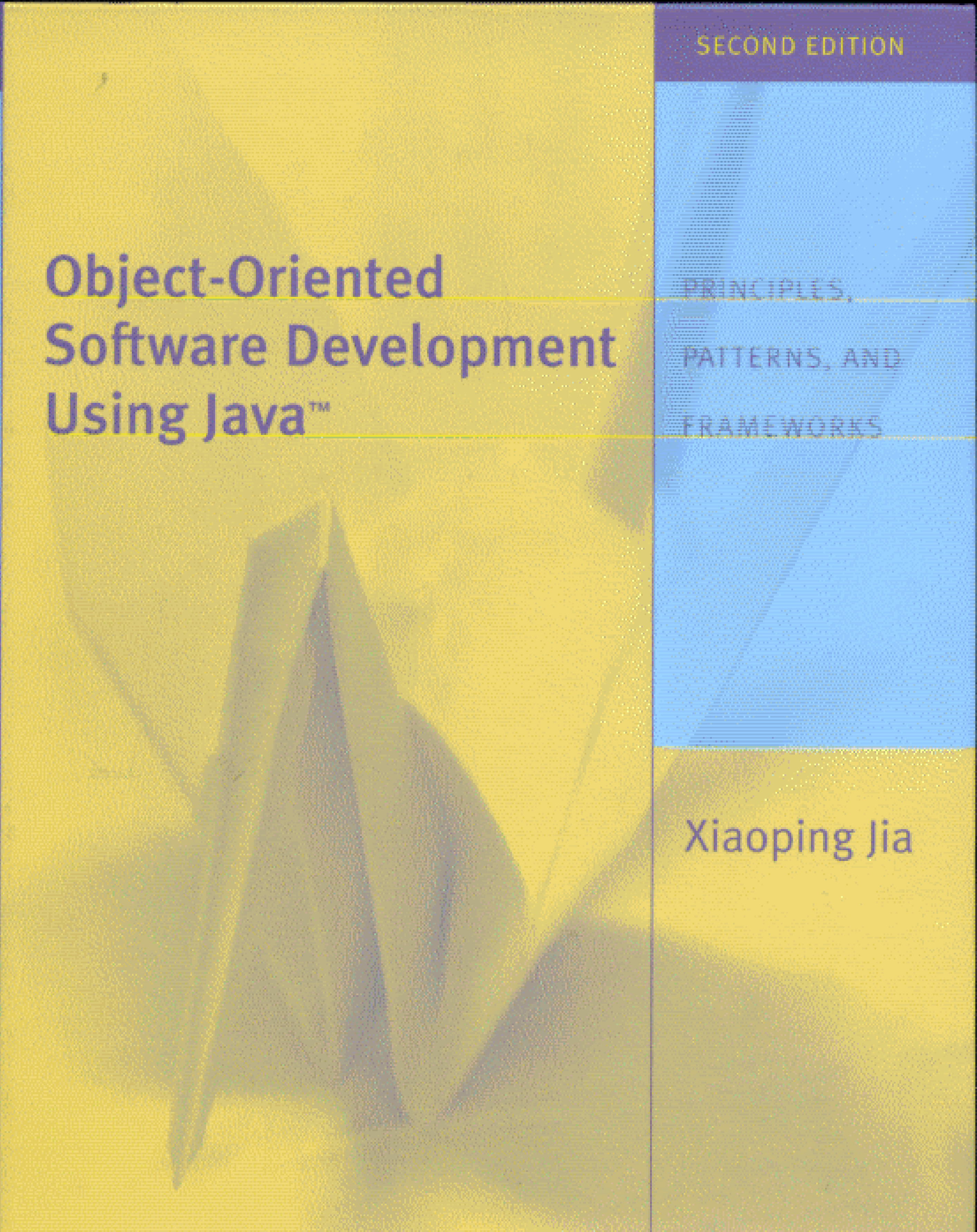
SECOND EDITION

Object-Oriented Software Development Using Java™

PRINCIPLES, PATTERNS, AND FRAMEWORKS

Xiaoping Jia



CONTENTS

Preface xv

CHAPTER 1 Object-Oriented Software Development 1

1.1 The Challenges of Software Development. 2
1.2 An Engineering Perspective 4
1.2.1 Software Development Activities 4
1.2.2 Software Development Processes 5
1.2.3 Desirable Qualities of Software Systems 6
1.2.4 Is Software Development an Engineering Process? 8
1.3 Object Orientation 9
1.3.1 Modeling the Real World 9
1.3.2 Evolution of Programming Models 10
1.3.3 A Brief History 10
1.4 Iterative Development Processes 11
1.4.1 Object-Oriented Development Activities 12
1.4.2 Rational Unified Process 13
1.4.3 Extreme Programming 15
Chapter Summary 16
Further Readings 17
Exercises 18

CHAPTER 2 Object-Oriented Modeling Using UML 19

2.1 Principles and Concepts 19
2.1.1 Objects and Classes 20
2.1.2 Principles 26
2.2 Modeling Relationships and Structures 29
2.2.1 Inheritance 29
2.2.2 Association 32
2.2.3 Aggregation and Composition 34
2.2.4 Dependency 35
2.3 Modeling Dynamic Behavior 36

2.3.1 Sequence Diagram 36
2.3.2 State Diagram 37
2.4 Modeling Requirements with Use Cases 41
2.4.1 Terms and Concepts 41
2.4.2 Use Case Diagrams 42
2.5 Case Study: An E-Bookstore 44
2.5.1 Conceptualization 44
2.5.2 Use Cases 44
2.5.3 Object Models 46
Chapter Summary 51
Further Readings 52
Exercises 52

CHAPTER 3 Introduction to Java 55
3.1 An Overview of the Java 2 Platform 56
3.2 The Java Run-Time Architecture 59
3.2.1 Program Execution Models 60
3.2.2 Java Virtual Machine 61
3.3 Getting Started with Java 65
3.3.1 A Simple Java Application 65
3.3.2 A Java Applet 67
Common Problems and Solutions 72
Chapter Summary 73
Further Readings 73
Exercises 74

CHAPTER 4 Elements of Java 75
4.1 Lexical Elements 76
4.1.1 Character Set 76
4.1.2 Identifiers 77
4.1.3 Primitive Types and Literals 77
4.1.4 Operators and Expressions 80
4.2 Variables and Types 87
4.2.1 Variable Declarations 88
4.2.2 Type Compatibility and Conversion 88
4.2.3 Reference Types 89
4.2.4 Arrays 91
4.3 Statements 93
4.3.1 Expression Statements 94
4.3.2 Statement Blocks 94
4.3.3 Local Variable Declarations 95
4.3.4 The `return` Statement 95
4.3.5 Selection Statements 96

4.3.6 Loop Statements 96
4.3.7 The **break** and **continue** Statements 99
4.4 Class Declarations 101
4.4.1 Syntax of Class Declarations 101
4.4.2 Creating and Initializing Objects 104
4.4.3 Accessing Fields and Methods 106
4.4.4 Method Invocation and Parameter Passing 107
4.4.5 Class (Static) Fields and Methods 110
4.4.6 Object Reference **this** 114
4.4.7 Interfaces and Abstract Classes 117
4.4.8 Strings 118
4.4.9 Wrapper Classes 128
4.5 Packages 134
4.5.1 Using Packages 135
4.5.2 Partitioning the Name Space 136
4.5.3 Packages and the Directory Structure 136
4.5.4 Organization of the Java Class Library 138
4.6 Exceptions 139
4.6.1 Sources of Exceptions 140
4.6.2 Hierarchy of Exceptions 140
4.6.3 Throwing Exceptions 143
4.6.4 Catching and Handling Exceptions 144
4.7 A Simple Animation Applet 148
Chapter Summary 155
Exercises 156
Project 157

CHAPTER 5 Classes and Inheritance 159
5.1 Overloading Methods and Constructors 159
5.2 Extending Classes 163
5.2.1 Constructors of Extended Classes 164
5.2.2 Subtypes and Polymorphism 165
5.2.3 Overriding Methods 171
5.3 Extending and Implementing Interfaces 176
5.3.1 Subtypes Revisited 177
5.3.2 Single Versus Multiple Inheritance 179
5.3.3 Name Collisions among Interfaces 183
5.3.4 Marker Interfaces 184
5.4 Hiding Fields and Class Methods 184
5.5 Applications—Animation Applets 186
5.5.1 Parameters of Applets 186
5.5.2 An Idiom for Animation Applets 188
5.5.3 Double-Buffered Animation 193
5.5.4 Reading Files in Applets 200

Common Problems and Solutions 202
Chapter Summary 202
Exercises 204
Projects 204

CHAPTER 6 From Building Blocks to Projects 207
6.1 Design and Implementation of Classes 207
6.1.1 Public and Helper Classes 207
6.1.2 Class Members 209
6.1.3 Design Guidelines 209
6.1.4 Documenting the Source Code 214
6.2 Contracts and Invariants 216
6.2.1 Contracts of Methods 216
6.2.2 Invaraints of Classes 222
6.2.3 Assertions 224
6.2.4 Design by Contract 226
6.3 The Canonical Form of Classes 227
6.3.1 No-Argument Constructor 228
6.3.2 Object Equality 228
6.3.3 Hash Code of Objects 230
6.3.4 Cloning Objects 231
6.3.5 String Representation of Objects 234
6.3.6 Serialization 234
6.4 Unit Testing 235
6.4.1 Simple Unit Testing 235
6.4.2 JUnit—A Unit-Testing Tool 239
6.4.3 Testing Coverage Criteria 241
6.5 Project Build 243
6.5.1 Ant—A Build Tool 243
Chapter Summary 246
Further Readings 247
Exercises 247

CHAPTER 7 Design by Abstraction 249
7.1 Design Patterns 249
7.1.1 Design Pattern: Singleton 251
7.2 Designing Generic Components 252
7.2.1 Refactoring 252
7.2.2 Design Pattern: Template Method 266
7.2.3 Generalizing 271
7.2.4 Design Pattern: Strategy 275
7.3 Abstract Coupling 276
7.3.1 Enumerating Elements 278

7.3.2 Design Pattern: Iterator 283
7.4 Design Case Study—Animation of Sorting Algorithms 284
7.4.1 The Initial Implementation 285
7.4.2 Separating Algorithms 290
7.4.3 Design Pattern: Factory 296
7.4.4 Separating Display Strategies 296
Chapter Summary 302
Further Readings 304
Exercises 304
Project 304

CHAPTER 8 Object-Oriented Application Frameworks 305
8.1 Application Frameworks 305
8.1.1 Characteristics 306
8.1.2 Design Requirements 307
8.1.3 Specific Frameworks Considered 308
8.2 The Collections Framework 308
8.2.1 Abstract Collections 309
8.2.2 Interfaces of Collections 310
8.2.3 Implementations of Collections 315
8.2.4 Iterators of Collections 319
8.2.5 Ordering and Sorting 324
8.3 The Graphical User Interface Framework—AWT and Swing 333
8.3.1 The GUI Components 333
8.3.2 Design Pattern: Composite 336
8.3.3 Layout Managers 338
8.3.4 Handling Events 348
8.3.5 Frames and Dialogs 359
8.4 The Input/Output Framework 366
8.4.1 Byte Streams 367
8.4.2 Design Pattern: Decorator 380
8.4.3 Character Streams 382
8.4.4 Random Access Files 389
Chapter Summary 392
Further Reading 394
Exercises 394
Projects 395

CHAPTER 9 Design Case Study: A Drawing Pad 397
9.1 Planning 397
9.2 Iteration 1: A Simple Scribble Pad 398
9.2.1 The Scribbling Canvas and Its Listener 399
9.2.2 The Application 402

9.3 Iteration 2: Menus, Options, and Files 403
9.3.1 Strokes 403
9.3.2 The Scribble Canvas 405
9.3.3 The Canvas Listener 408
9.3.4 The Application 409
9.3.5 Choosing Colors 416
9.4 Iteration 3: Refactoring 421
9.4.1 The Shapes 421
9.4.2 The Tools 424
9.4.3 Extending Components 428
9.5 Iteration 4: Adding Shapes and Tools 432
9.5.1 The Shapes 433
9.5.2 The Toolkit 436
9.5.3 Design Pattern: State 438
9.5.4 A Concrete Tool—`TwoEndsTool` 439
9.5.5 Extending Components 442
9.5.6 Design Pattern: Factory Method 447
9.6 Iteration 5: More Drawing Tools 448
9.6.1 Filled Shapes 448
9.6.2 Drawing Filled Shapes 449
9.6.3 The Application 452
9.7 Iteration 6: The Text Tool 453
9.7.1 The Text Shape 454
9.7.2 The Keyboard Input Tool 455
9.7.3 The Font Option Menu 459
Chapter Summary 462
Further Readings 462
Project 463

CHAPTER 10 More Design Patterns 465
10.1 Type-Safe Enumeration Types 465
10.1.1 A Simple Maze Game 465
10.1.2 Enumeration Types 466
10.1.3 Unordered Type-Safe Enumeration Idiom 468
10.1.4 Ordered Type-Safe Enumeration Idiom 469
10.2 Creational Design Patterns 470
10.2.1 A Simple Design of the Maze Game 470
10.2.2 Design Pattern: Abstract Factory 484
10.2.3 Design Pattern: Factory Method 491
10.2.4 Design Pattern: Prototype 495
10.2.5 Design Pattern: Builder 502
10.3 Behavioral Patterns 507
10.3.1 Design Pattern: Command 507

10.3.2 Supporting Undo 509

10.4 Structural Patterns 513

10.4.1 Design Pattern: Adapter 513

10.4.2 Design Pattern: Composite 531

Chapter Summary 544

Further Readings 545

CHAPTER 11 Concurrent Programming 547

11.1 Threads 547

11.1.1 Creation of Threads 548

11.1.2 Controlling Threads 553

11.2 Thread Safety and Liveness 556

11.2.1 Synchronization 557

11.2.2 Cooperation Among Threads 564

11.2.3 Liveness Failures 569

11.3 Design Case Study—Tic-Tac-Toe Game 571

11.3.1 The Game Board 572

11.3.2 The Game 577

11.3.3 The Players 579

11.3.4 Idiom: Taking Turns 582

Chapter Summary 583

Further Reading 584

Exercises 584

Projects 585

CHAPTER 12 Distributed Computing 587

12.1 Socket-Based Communication 588

12.1.1 Server and Client Sockets 588

12.1.2 Servers and Clients Using Sockets 590

12.1.3 Design Case Study—Stock Quotes I 600

12.2 Remote Method Invocation 614

12.2.1 The Architecture 614

12.2.2 Using RMI 616

12.2.3 Design Case Study—Stock Quotes II 620

12.3 Java Database Connectivity 628

12.4 Common Object Request Broker Architecture 640

Chapter Summary 641

Further Readings 642

Exercises 642

Projects 643

APPENDIX A Summary of the APPLET Tag 645

APPENDIX B Summary of Documentation Tags 647

APPENDIX C Summary of Java Naming Conventions 649

Glossary 653

References 663

Index 667