# CONTENTS


Preface xiii

CHAPTER 1 INTRODUCTION 1
Why Patterns? 1
Why Design Patterns? 2
Why Java? 3
UML 4
Challenges 4
The Organization of This Book 5
Welcome to Oozinoz! 6
Summary 7

PART I INTERFACE PATTERNS 9

CHAPTER 2 INTRODUCING INTERFACES 11
Interfaces and Abstract Classes 11
Interfaces and Obligations 13
Summary 15
Beyond Ordinary Interfaces 16

CHAPTER 3 ADAPTER 17
Adapting to an Interface 17
Class and Object Adapters 21
Adapting Data for a JTable 25
Identifying Adapters 30
Summary 31

CHAPTER 4 FACADE 33
Facades, Utilities, and Demos 33
Refactoring to FACADE 35
Summary 46

| | | |
|---|---|---|
| CHAPTER 17 | ABSTRACT FACTORY | 175 |
| | A Classic Example: GUI Kits | 175 |
| | Abstract Factories and Factory Method | 180 |
| | Packages and Abstract Factories | 184 |
| | Summary | 185 |
| CHAPTER 18 | PROTOTYPE | 187 |
| | Prototypes as Factories | 187 |
| | Prototyping with Clones | 189 |
| | Summary | 192 |
| CHAPTER 19 | MEMENTO | 193 |
| | A Classic Example: Using Memento for Undo | 193 |
| | Memento Durability | 201 |
| | Persisting Mementos Across Sessions | 201 |
| | Summary | 205 |
| **PART IV** | **OPERATION PATTERNS** | **207** |
| CHAPTER 20 | INTRODUCING OPERATIONS | 209 |
| | Operations and Methods | 209 |
| | Signatures | 211 |
| | Exceptions | 212 |
| | Algorithms and Polymorphism | 213 |
| | Summary | 214 |
| | Beyond Ordinary Operations | 215 |
| CHAPTER 21 | TEMPLATE METHOD | 217 |
| | A Classic Example: Sorting | 217 |
| | Completing an Algorithm | 221 |
| | TEMPLATE METHOD Hooks | 224 |
| | Refactoring to TEMPLATE METHOD | 225 |
| | Summary | 228 |
| CHAPTER 22 | STATE | 229 |
| | Modeling States | 229 |
| | Refactoring to STATE | 233 |
| | Making States Constant | 238 |
| | Summary | 240 |

| | | |
|---|---|---|
| CHAPTER 23 | STRATEGY | 241 |
| | Modeling Strategies | 241 |
| | Refactoring to STRATEGY | 244 |
| | Comparing STRATEGY and STATE | 248 |
| | Comparing STRATEGY and TEMPLATE Method | 249 |
| | Summary | 250 |
| CHAPTER 24 | COMMAND | 251 |
| | A Classic Example: Menu Commands | 251 |
| | Using COMMAND to Supply a Service | 254 |
| | COMMAND Hooks | 255 |
| | COMMAND in Relation to Other Patterns | 257 |
| | Summary | 259 |
| CHAPTER 25 | INTERPRETER | 261 |
| | An INTERPRETER Example | 261 |
| | Interpreters, Languages, and Parsers | 274 |
| | Summary | 275 |
| PART V | EXTENSION PATTERNS | 277 |
| CHAPTER 26 | INTRODUCING EXTENSIONS | 279 |
| | Principles of Object-Oriented Design | 279 |
| | The Liskov Substitution Principle | 280 |
| | The Law of Demeter | 281 |
| | Removing Code Smells | 283 |
| | Beyond Ordinary Extensions | 283 |
| | Summary | 285 |
| CHAPTER 27 | DECORATOR | 287 |
| | A Classic Example: Streams and Writers | 287 |
| | Function Wrappers | 295 |
| | DECORATOR in Relation to Other Patterns | 303 |
| | Summary | 303 |
| CHAPTER 28 | ITERATOR | 305 |
| | Ordinary Iteration | 305 |
| | Thread-Safe Iteration | 307 |
| | Iterating over a Composite | 313 |
| | Summary | 324 |

| | | |
|---|---|---|
| CHAPTER 29 | VISITOR | 325 |
| | VISITOR Mechanics | 325 |
| | An Ordinary VISITOR | 327 |
| | VISITOR Cycles | 333 |
| | VISITOR Risks | 338 |
| | Summary | 340 |
| **PART VI** | **APPENDIXES** | **341** |
| APPENDIX A | DIRECTIONS | 343 |
| | Get the Most from This Book | 343 |
| | Understand the Classics | 344 |
| | Weave Patterns into Your Code | 344 |
| | Keep Learning | 345 |
| APPENDIX B | SOLUTIONS | 347 |
| APPENDIX C | OOZINOZ SOURCE | 427 |
| | Acquiring and Using the Source | 427 |
| | Building the Oozinoz Code | 427 |
| | Testing the Code with JUnit | 428 |
| | Finding Files Yourself | 428 |
| | Summary | 429 |
| APPENDIX D | UML AT A GLANCE | 431 |
| | Classes | 432 |
| | Class Relationships | 433 |
| | Interfaces | 435 |
| | Objects | 436 |
| | States | 437 |
| Glossary | | 439 |
| Bibliography | | 447 |
| Index | | 449 |