

Contents

Acknowledgments	page xvii
Foreword	xix
Preface	xxi
About the Author	xxv
Chapter 1	
Leading-Edge Software Development	1
1.1 Modern Development Technologies	2
1.1.1 Object Technology	2
1.1.2 Extensible Markup Language (XML)	5
1.1.3 Relational Database (RDB) Technology	6
1.1.4 Web Services	7
1.2 Modern Development Techniques	8
1.2.1 Agile Software Development	9
1.2.2 Unified Modeling Language (UML)	11
1.2.3 The Unified Process (UP)	13
1.2.4 Model-Driven Architecture (MDA)	14
1.2.5 Using Them Together	15
1.3 The Organization of This Book	16
1.4 The Case Studies	19
1.4.1 The Bank Case Study	20
1.5 What You Have Learned	22

Chapter 2	
Understanding the Basics—Object-Oriented Concepts	23
2.1 A Brief Overview of OO Concepts	24
2.2 OO Concepts from a Structured Point of View	27
2.3 The Diagrams of UML 2	28
2.4 Objects and Classes	28
2.5 Attributes and Operations/Methods	32
2.6 Abstraction, Encapsulation, and Information Hiding	34
2.6.1 Abstraction	34
2.6.2 Encapsulation	34
2.6.3 Information Hiding	35
2.6.4 An Example	35
2.7 Inheritance	37
2.7.1 Modeling Inheritance	38
2.7.2 Inheritance Tips and Techniques	39
2.7.3 Single and Multiple Inheritance	40
2.7.4 Abstract and Concrete Classes	43
2.8 Persistence	43
2.9 Relationships	44
2.9.1 Associations	45
2.9.2 Modeling the Unknown	48
2.9.3 How Associations Are Implemented	49
2.9.4 Properties	50
2.9.5 Aggregation and Composition	50
2.9.6 Dependencies	52
2.10 Collaboration	54
2.11 Coupling	57
2.12 Cohesion	58
2.13 Polymorphism	59
2.13.1 An Example: The Poker Game	59
2.13.2 Polymorphism at the University	60
2.14 Interfaces	62
2.15 Components	63
2.16 Patterns	65
2.17 What You Have Learned	66
2.18 Review Questions	67

CONTENTS**ix****Chapter 3****Full Lifecycle Object-Oriented Testing (FLOOT)****68**

3.1 The Cost of Change	69
3.2 Testing Philosophies	74
3.3 Full Lifecycle Object-Oriented Testing (FLOOT)	75
3.4 Regression Testing	78
3.5 Quality Assurance	79
3.6 Testing Your Models	80
3.6.1 Proving It with Code	80
3.6.2 Usage Scenario Testing	81
3.6.3 Prototype Reviews/Walkthroughs	84
3.6.4 User-Interface Testing	85
3.6.5 Model Reviews	85
3.6.6 When to Use Each Technique	87
3.7 Testing Your Code	88
3.7.1 Testing Terminology	88
3.7.2 Testing Tools	89
3.7.3 Traditional Code Testing Concepts	89
3.7.4 Object-Oriented Testing Techniques	92
3.7.5 Code Inspections	94
3.8 Testing Your System in Its Entirety	95
3.9 Testing by Users	96
3.10 Test-Driven Development (TDD)	97
3.11 What You Have Learned	99
3.12 Review Questions	99

Chapter 4**Agile Model-Driven Development (AMDD)****101**

4.1 Modeling Philosophies	102
4.2 Project Stakeholders	106
4.3 What Is Agile Modeling (AM)?	107
4.4 The Values of AM	108
4.5 The Principles of AM	109
4.6 The Practices of AM	109
4.7 Easing into Agile Modeling	109
4.8 Agile Model-Driven Development (AMDD)	118
4.8.1 How is AMDD Different?	120

4.9 Fundamental Information Gathering Skills	121
4.9.1 Interviewing	121
4.9.2 Observation	123
4.9.3 Brainstorming	123
4.10 Agile Documentation	124
4.11 Making Whiteboards Work for Software Development	125
4.12 AMDD and Other Agile Methodologies	129
4.13 Agile Modeling and Test-Driven Development (TDD)	129
4.14 What You Have Learned	132
4.15 Review Questions	132
Chapter 5	
Usage Modeling	134
5.1 Use Case Modeling	134
5.1.1 Starting Agile	136
5.1.2 Essential Use Case Diagrams	139
5.1.3 Identifying Actors	146
5.1.4 Writing an Essential Use Case	149
5.1.5 Identifying Use Cases	151
5.1.6 System Use Case Diagrams	153
5.1.7 System Use Cases	153
5.1.8 Writing Alternate Courses of Action	160
5.1.9 Other Use Case Styles	163
5.1.10 Comparing Essential and System Use Cases	164
5.1.11 Reuse in Use Case Models: <<extend>>, <<include>>, and Inheritance	167
5.1.12 Packages	172
5.1.13 Use Case Modeling Tips	173
5.1.14 Remaining Agile	176
5.2 User Stories	177
5.2.1 What About System User Stories?	180
5.3 Features	180
5.4 What You Have Learned	182
5.5 Review Questions	183

CONTENTS**xi****Chapter 6****User-Interface Development****184**

6.1 Essential User-Interface Prototyping	185
6.2 Traditional User-Interface Prototyping	191
6.3 User-Interface Flow Diagramming	197
6.4 Usability	199
6.5 User-Interface Design Strategies	200
6.6 Agile Stakeholder Documentation	205
6.7 What You Have Learned	207
6.8 Review Questions	207

Chapter 7**Supplementary Requirements****209**

7.1 Business Rules	210
7.2 Technical Requirements	214
7.3 Constraints	215
7.4 Object Constraint Language (OCL)	216
7.5 Glossaries	217
7.6 Supplementary Specifications	218
7.7 What You Have Learned	218
7.8 Review Questions	219

Chapter 8**Conceptual Domain Modeling****220**

8.1 Robustness Diagrams	221
8.2 Object Role Model (ORM) Diagrams	227
8.3 Class Responsibility Collaborator (CRC) Cards	231
8.4 Analysis Class Diagrams	237
8.4.1 Modeling Classes and Responsibilities	241
8.4.2 Modeling Associations	245
8.4.3 Introducing Reuse between Classes via Inheritance	247
8.4.4 Modeling Composition and Associations	249
8.4.5 Modeling Vocabularies	250
8.5 Logical Data Models (LDMs)	251
8.6 Applying Analysis Patterns Effectively	254

8.7 UML Object Diagram	256
8.8 Keeping Conceptual Domain Modeling Agile	258
8.9 What You Have Learned	259
8.10 Review Questions	260
Chapter 9	
Business Process Modeling	262
9.1 Data Flow Diagrams (DFDs)	263
9.2 Flowcharts	268
9.3 UML Activity Diagrams	270
9.4 What You Have Learned	277
9.5 Review Questions	277
Chapter 10	
Agile Architecture	278
10.1 Architecture Techniques and Concepts	280
10.1.1 Put Architectural Decisions Off as Long as Possible	280
10.1.2 Accept That Some Architectural Decisions Are Already Made	281
10.1.3 Prove It with Code	281
10.1.4 Set an Architectural Change Strategy	282
10.1.5 Consider Reuse	283
10.1.6 Roll Up Your Sleeves	284
10.1.7 Be Prepared to Make Trade-offs	285
10.1.8 Consider Adopting the Zachman Framework	285
10.1.9 Apply Architectural Patterns Gently	289
10.2 Looking to the Future with Change Cases	289
10.3 UML Package Diagrams	291
10.3.1 Class Package Diagrams	292
10.3.2 Data Package Diagrams	294
10.3.3 Use Case Package Diagrams	294
10.4 UML Component Diagrams	296
10.4.1 Interfaces and Ports	298
10.4.2 Designing Components	300
10.4.3 Creating Component Models	302
10.4.4 Remaining Agile	306

CONTENTS	xiii
10.5 Free-Form Diagrams	306
10.6 UML Deployment Diagrams	308
10.7 Network Diagrams	313
10.8 Layering Your Architecture	314
10.9 What You Have Learned	317
10.10 Review Questions	317
Chapter 11	
Dynamic Object Modeling	319
11.1 UML Sequence Diagrams	321
11.1.1 Visual Coding Via Sequence Diagrams	332
11.1.2 How to Draw Sequence Diagrams	333
11.1.3 Keeping It Agile	334
11.2 UML Communication Diagrams	334
11.3 UML State Machine Diagrams	337
11.4 UML Timing Diagrams	344
11.5 UML Interaction Overview Diagrams	346
11.6 UML Composite Structure Diagrams	348
11.7 What You Have Learned	350
11.8 Review Questions	350
Chapter 12	
Structural Design Modeling	351
12.1 UML Class Diagrams	351
12.1.1 Modeling Methods during Design	352
12.1.2 Modeling Attributes during Design	360
12.1.3 Inheritance Techniques	366
12.1.4 Association and Dependency Techniques	368
12.1.5 Composition Techniques	372
12.1.6 Introducing Interfaces into Your Model	373
12.1.7 Class Modeling Design Tips	376
12.2 Applying Design Patterns Effectively	380
12.2.1 The <i>Singleton</i> Design Pattern	380
12.2.2 The <i>Façade</i> Design Pattern	381
12.2.3 Tips for Applying Patterns Effectively	382
12.3 Physical Data Modeling with the UML	383
12.4 What You Have Learned	390

12.5 Review Questions	390
12.5.1 The Bank Case Study Six Months Later	391

Chapter 13**Object-Oriented Programming 393**

13.1 Philosophies for Effective Programming	394
13.2 Programming Tips and Techniques for Writing High-Quality Code	397
13.3 Test-Driven Development (TDD)	400
13.3.1 TDD and AMDD	402
13.3.2 Why TDD?	403
13.4 From Object Design to Java Code	404
13.4.1 From UML Sequence Diagrams to Code	404
13.4.2 From UML Class Diagrams to Code	406
13.4.3 Implementing a Class in Java	408
13.4.4 Declaring Instance Attributes in Java	409
13.4.5 Implementing Instance Methods in Java	410
13.4.6 Implementing Static Methods and Attributes in Java	412
13.4.7 Documenting Methods	417
13.4.8 Implementing Constructors	419
13.4.9 Encapsulating Attributes with Accessors	420
13.4.10 Implementing Inheritance in Java	426
13.4.11 Implementing Interfaces in Java	426
13.4.12 Implementing Relationships in Java	429
13.4.13 Implementing Dependencies	438
13.4.14 Implementing Collaborations in Java	439
13.4.15 Implementing Business Rules	439
13.4.16 Iterate, Iterate, Iterate	440
13.5 What You Have Learned	440
13.6 Review Questions	440

Chapter 14**Agile Database Development 442**

14.1 Philosophies for Effective Data Development	444
14.2 Mapping Objects to Relational Databases	445

CONTENTS**xv**

14.2.1	Shadow Information	446
14.2.2	Mapping Inheritance Structures	447
14.2.3	Mapping Relationships	451
14.3	Strategies for Implementing Persistence Code	453
14.4	From Design to Database Code	455
14.4.1	Defining and Modifying Your Persistence Schema	455
14.4.2	Creating, Retrieving, Updating, and Deleting Data	456
14.4.3	Interacting with a Database from Java	458
14.4.4	Implementing Your Mappings	460
14.5	Data-Oriented Implementation Strategies	460
14.5.1	Concurrency Control	462
14.5.2	Transaction Control	464
14.5.3	Shared Logic and Referential Integrity	466
14.5.4	Security Access Control	471
14.5.5	Searching for Objects	473
14.5.6	Reports	476
14.6	Database Refactoring	477
14.7	Legacy Analysis	481
14.7.1	Formalizing Contract Models	482
14.7.2	Common Legacy Challenges	483
14.7.3	Creating Contract Models	484
14.8	What You Have Learned	485
14.9	Review Questions	486
Chapter 15		
Where to Go from Here		487
15.1	Become a Generalizing Specialist	487
15.2	Continuing Your Learning Process	490
15.3	Parting Words	492
Glossary		493
References and Recommended Reading		525
Index		533