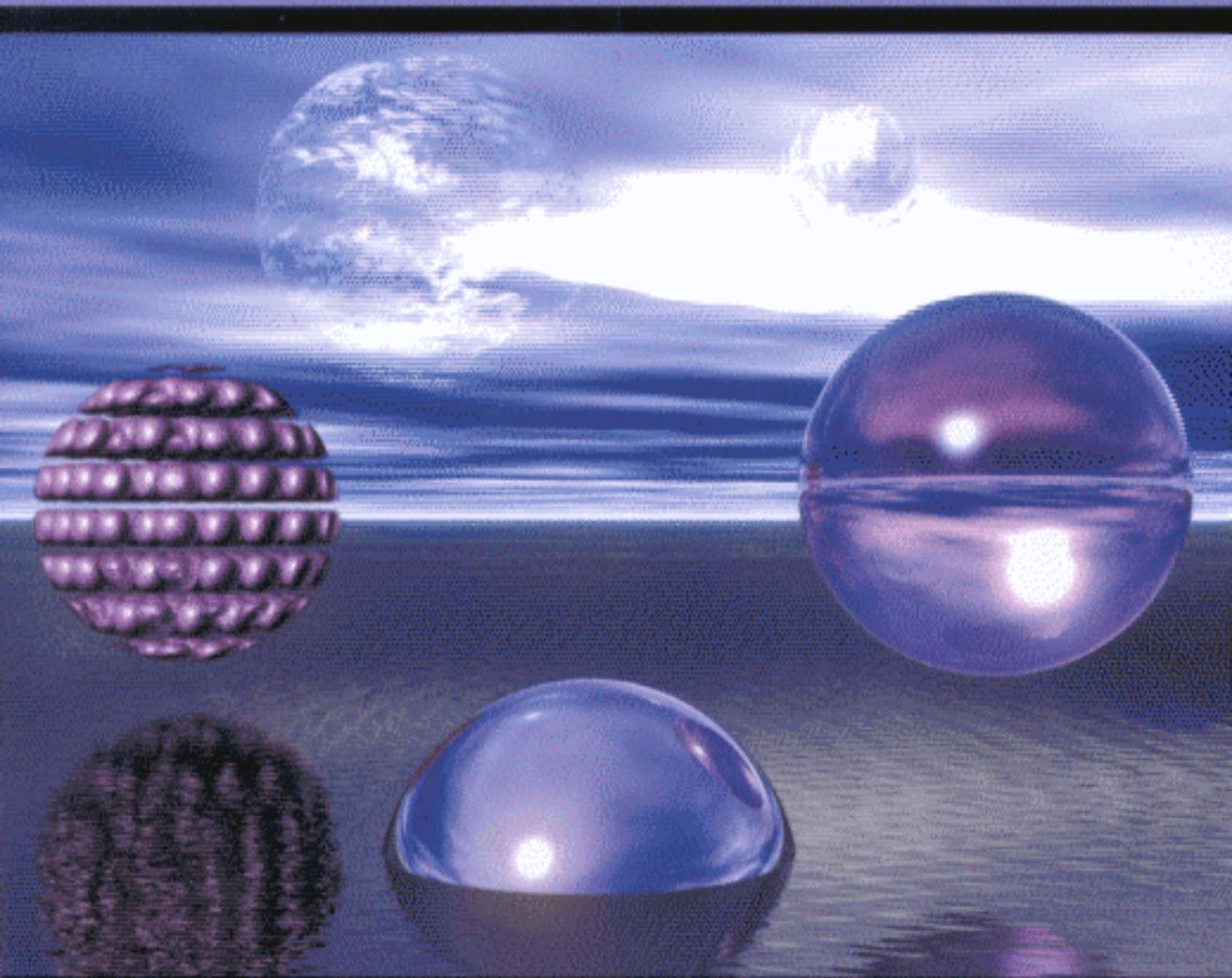


# **PRACTICAL OBJECT-ORIENTED DEVELOPMENT WITH UML AND JAVA™**



Richard C. Lee  
William M. Tepfenhart

# Contents

<b>Preface</b>	<b>xvii</b>
<b>Object-Oriented Technology</b>	<b>xvii</b>
<b>Why Unified Modeling Language?</b>	<b>xviii</b>
<b>Why Java?</b>	<b>xix</b>
<b>Our Approach to Object-Oriented Technology</b>	<b>xx</b>
<b>Organization of the Book</b>	<b>xxi</b>
<b>Using This Book</b>	<b>xxii</b>
<b>Acknowledgments</b>	<b>xxv</b>
<b>1 Managing Complexity with Abstraction</b>	<b>1</b>
<b>Complex Systems</b>	<b>3</b>
System Development Is Modeling	4
A Strategy for Modeling	5
<b>Abstraction Mechanisms</b>	<b>6</b>
Basic Data Types	7
Functions	7
Modules	8
Abstract Data Types	9
Objects	9
Class	10
Generalization/Specialization	10
Polymorphism	11
Interface	11
Reflection	12
<b>Service Activation Abstractions</b>	<b>13</b>
Function Call	13
Event Processing (Asynchronous Communication)	13
Message Passing (Synchronous Communication)	13
Subscription (Asynchronous Communication)	14
<b>Processing Control Abstractions</b>	<b>14</b>
Single Program Execution	15

Multitasking	15
Sequential Execution	15
Multithreading	16
<b>Relationships</b>	<b>16</b>
Associations	16
Aggregation	18
Behavior	19
Static Behavior	20
Dynamic Behavior	20
<b>Rules</b>	<b>21</b>
<b>2 The Object-Oriented Paradigm</b>	<b>24</b>
<b>The Object-Oriented Paradigm</b>	<b>24</b>
What Is an Object?	25
What Is a Class?	27
<b>Principles of Object-Orientation</b>	<b>28</b>
Principle 1. Encapsulation	28
Principle 2. Information Hiding	29
Principle 3. Message Passing	29
Principle 4. Late Binding	30
Principle 5. Delegation	32
Principle 6. Class/Instance/Object	32
Principle 7. Generalization/Specialization without Polymorphism	33
Principle 8. Generalization/Specialization with Polymorphism	33
Principle 9. Relationships	34
Principle 10. Interface/Instance/Object	34
Principle 11. Generalization/Specialization of Interfaces	34
Principle 12. Reflection	35
Principle 13. Multithreading	35
<b>Object-Oriented Model of Computation</b>	<b>35</b>
<b>3 Building a Specification Model</b>	<b>46</b>
<b>Introduction to Use Cases</b>	<b>47</b>
System	48
Actors	49
Use Cases	49
Use-Case Bundles	55
<b>Documenting Use Cases</b>	<b>57</b>
Use-Case Diagram	57
Sequence Diagram: Documenting the Details	58
Textual Description	59
<b>Guidelines for Developing Use Cases</b>	<b>60</b>
Avoiding Analysis Paralysis	60

Identifying Actors	61
Identifying High-Level and Essential Use Cases	63
Establishing Use-Case Bundles	65
Developing Use-Case Details	65
Identifying Supporting Use Cases	67
<b>Contracts</b>	<b>68</b>
<b>Recommended Approach</b>	<b>70</b>
<b>4 Finding the Objects</b>	<b>79</b>
<b>Object-Oriented Analysis: Model of an Application Domain</b>	<b>80</b>
<b>Building the Objected-Oriented Model</b>	<b>80</b>
<b>Identification of Objects, Classes, and Interfaces</b>	<b>82</b>
<b>Current Techniques</b>	<b>84</b>
Using the Things to Be Modeled	84
Using the Definitions of Objects, Categories, and Types	84
Using Object Decomposition	84
Using Generalization	85
Using Subclasses	85
Using Object-Oriented Domain Analysis	86
Reusing an Application Framework	86
Reusing Class Hierarchies	87
Reusing Individual Objects and Classes	88
Using Subassemblies	88
Using Personal Experience	89
<b>Traditional Techniques</b>	<b>89</b>
Using Nouns	89
Using Traditional Data Flow Diagrams	90
Using Class-Responsibility-Collaboration (CRC) Cards	92
<b>Recommended Approaches</b>	<b>94</b>
<b>5 Identifying Responsibilities</b>	<b>101</b>
<b>What Is an Object?</b>	<b>101</b>
<b>What Is an Attribute?</b>	<b>102</b>
Descriptive Attributes	103
Naming Attributes	103
<b>What Is a Service?</b>	<b>104</b>
<b>What Is a Method?</b>	<b>104</b>
<b>Identifying Attributes</b>	<b>104</b>
<b>Specifying Attributes</b>	<b>106</b>
<b>Identifying Services</b>	<b>108</b>
<b>Specifying Services</b>	<b>109</b>
<b>Recommended Approach</b>	<b>109</b>

<b>6</b>	<b>Specifying Static Behavior</b>	<b>115</b>
	<b>What Is Behavior?</b>	<b>115</b>
	<b>Java Services (Operations) That Affect the Behavior Specification</b>	<b>117</b>
	<b>Techniques for Specifying Static Behavior</b>	<b>118</b>
	<b>Techniques for Specifying Control</b>	<b>120</b>
	<b>Techniques for Documenting Control</b>	<b>121</b>
	Activity Diagrams	121
	Collaboration Diagram	122
	Sequence Diagram	123
	<b>Techniques for Documenting Static Behavior</b>	<b>124</b>
	Pre-conditions and Post-conditions	124
	Flowcharting	124
	Data Flow Diagrams	125
	Structured English	125
	<b>Recommended Approach</b>	<b>125</b>
<b>7</b>	<b>Dynamic Behavior</b>	<b>131</b>
	<b>Introduction</b>	<b>132</b>
	<b>Techniques for Identifying Dynamic Behavior</b>	<b>133</b>
	Common Lifecycle Forms	134
	Models for Capturing Lifecycle	135
	<b>Identifying and Specifying Events</b>	<b>137</b>
	Use Case and Scenario	137
	Sequence Diagram	137
	<b>Specifying Dynamic Behavior</b>	<b>142</b>
	Event List	142
	State Transition Table	144
	<b>Documenting Dynamic Behavior</b>	<b>148</b>
	State Diagrams	148
	<b>Recommended Approach</b>	<b>152</b>
<b>8</b>	<b>Identifying Relationships</b>	<b>155</b>
	<b>Accessing Another Object's Services</b>	<b>155</b>
	<b>Relationships</b>	<b>156</b>
	<b>Generalization</b>	<b>157</b>
	<b>Identifying and Specifying Generalization/Specialization</b>	<b>160</b>
	<b>Object Aggregation</b>	<b>161</b>
	<b>Classification of Aggregation</b>	<b>162</b>
	Assembly-Parts (Component-Integral) Composition	163
	Material-Object Composition	164
	Portion-Object Composition	165
	Place-Area Composition	166
	Collection-Members Composition	166

Container Content (Member-Bunch) Composition	167
Member-Partnership Composition	167
Objects and Aggregation Relationships	168
<b>Links Between Objects</b>	<b>168</b>
<b>Identifying and Specifying Links and Aggregations</b>	<b>170</b>
<b>Managing Relationships</b>	<b>171</b>
<b>Documenting Relationships</b>	<b>173</b>
<b>Recommended Approach</b>	<b>175</b>
<b>9 Rules</b>	<b>178</b>
<b>Introduction</b>	<b>178</b>
Rules	180
<b>Identifying Declarative Statements</b>	<b>180</b>
<b>Specifying and Documenting Rules</b>	<b>181</b>
<b>Mapping Rules to the Proper Object-Oriented Concepts</b>	<b>183</b>
<b>Documenting the Rules Using UML</b>	<b>184</b>
<b>Implementing Rules</b>	<b>184</b>
<b>Recommended Approach</b>	<b>185</b>
<b>10 The Model</b>	<b>187</b>
<b>Concepts</b>	<b>187</b>
<b>Concepts and Object-Oriented Model</b>	<b>189</b>
Class	189
Interface	189
Association	190
Class Aggregation	190
Generalization/Specialization of Classes	191
Polymorphism	192
Instantiation	192
<b>Documenting Concepts Using UML</b>	<b>193</b>
Class Concept	193
Basic Class Construct	193
Parameterized Class and Bounded Class Construct	194
Interface Construct	194
Association	194
Class Aggregation	196
Generalization/Specialization of Class	196
Generalization/Specialization of Interface	197
Polymorphism	197
Instantiation	197
Refining the Model	197
<b>Subsystems</b>	<b>198</b>
Domain	198

Bridge	199
<b>Organizing Subsystems</b>	<b>200</b>
Horizontal Layers	200
Vertical Partitions	201
Combination	201
<b>Identifying Subsystems</b>	<b>201</b>
Documenting Subsystem	201
<b>Recommended Approach</b>	<b>202</b>
Subsystems	205
<b>11 Design</b>	<b>210</b>
<b>Introduction</b>	<b>211</b>
<b>System Design</b>	<b>211</b>
Subsystems	211
Architectural Frameworks	212
Software Control Within a Framework	214
Documenting System Design	215
<b>Detailed Design</b>	<b>217</b>
Class Design	217
Association Design	219
Generalization and Inheritance	220
Delegation	221
Orlando Treaty	223
Multiple Inheritance	223
Documenting Detailed Design	225
<b>12 Java Fundamentals</b>	<b>227</b>
<b>Introduction to the Java Language</b>	<b>227</b>
Example Statements	227
Unicode Character	228
<b>Programming Elements</b>	<b>229</b>
Comments	230
Reserved Words (Keywords)	231
Identifiers	232
Literals	232
Operators	233
Arithmetic Operators	234
Boolean Operators	235
Conditional and Relational Operators	235
Bitwise Operators	237
Class-Defined Operators	238
Punctuators	238
<b>Primitive Data Types</b>	<b>239</b>

Primitive Data Types	239
Constant Values	240
Symbolic Variables	241
Reference Variables	242
Constants	242
Class “Data Type”	242
<b>What Is a Statement?</b>	<b>243</b>
Expressions	244
Null Statement	245
Assignment Statement	245
Service Request Statement	246
Block (Compound Statement)	246
<b>Statement Flow Control</b>	<b>247</b>
If Statement	247
If-Else Statement	247
For Statement	248
Switch Statement	248
While Statement	250
Do-While Statement	250
<b>Branching Statements</b>	<b>251</b>
The Break Statement	251
The Continue Statement	253
The Return Statement	254
<b>Exception Handling</b>	<b>255</b>
Throws and Try-Catch Block	255
Throwable Class	257
<b>NameSpace</b>	<b>259</b>
No Global Variables	259
Package and Imports	260
Packages of the Java API	261
Filenames and Directories	262
Access to Members of a Class	262
Public	262
Private	263
Protected	263
Package	264
Summary	264
<b>Type Conversion</b>	<b>265</b>
Casting Between Primitive Data Types	265
Casting Objects	266
Converting Primitive Data Types to Objects	267
<b>Recommended Approach</b>	<b>267</b>

<b>13</b>	<b>Implementing Class and Interface</b>	<b>273</b>
	<b>Components of a Class</b>	<b>273</b>
	<b>Class Definition</b>	<b>274</b>
	<b>Class Body</b>	<b>275</b>
	Visibility	275
	Data Variables	276
	Instance Data Variables	276
	Class Data Variables	277
	Constants	278
	Initialization of Member Variables	279
	Other Attributes of a Class Member	280
	Member Methods	280
	Class Method	283
	Details of Member Method Prototype	284
	Abstract Class	285
	<b>Nested, Inner, and Anonymous Inner Classes</b>	<b>286</b>
	Nested Classes	286
	Inner Class	287
	Anonymous Inner Classes	288
	<b>Predefined Java Classes</b>	<b>289</b>
	String Data Type	289
	String Concatenation	289
	Conversion of an Object to a String Instance	289
	Conversion from String to Integer and Vice Versa	290
	Substrings	290
	Length of String	290
	Individual Characters in a String	290
	Immutable Property	291
	StringBuffer Data Type	292
	Number Data Types	293
	Array Data Type	294
	Declaring a Variable to Hold an Array	295
	Initialization of an Array	295
	Access and Assignment by Position	297
	Multidimensional Array	298
	Interface	298
	Interface Declaration	299
	Interface Body	300
	Using an Interface as a Data Type	300
	Implementing the Interface	300
	Why Can't I Just Use an Abstract Class?	303
	Do Interfaces Provide for Multiple Inheritance?	303
	What Can I Use Interfaces For?	303
	<b>Recommended Approach</b>	<b>304</b>

<b>14</b>	<b>Implementing Static Behavior</b>	<b>310</b>
	<b>What Is a Service?</b>	<b>310</b>
	<b>Method Definition</b>	<b>312</b>
	Method (Service) Prototype	312
	Method Implementation	313
	Return Type	315
	Return Statement	315
	<b>Body</b>	<b>316</b>
	this	316
	super	317
	Local Variables	318
	Service Argument List	318
	<b>Passing Arguments</b>	<b>319</b>
	Pass-By-Value	319
	Pass-By-Reference (or Pointer)	320
	<b>Identifier's Scope</b>	<b>321</b>
	<b>Polymorphism</b>	<b>321</b>
	Overriding Methods	322
	Calling the Original Method	323
	<b>Creating Objects and Destroying Objects</b>	<b>323</b>
	Introduction	324
	Constructor Methods	324
	Calling Another Constructor	327
	Overloading Constructors	327
	Calling Sequence of Constructors	328
	Visibility of Constructor	329
	Finalizer Method	329
	<b>Constructor and Finalizer Coding Guidelines</b>	<b>329</b>
	<b>Recommended Approach</b>	<b>330</b>
<b>15</b>	<b>Implementing Dynamic Behavior</b>	<b>333</b>
	<b>Elements of Dynamic Behavior</b>	<b>333</b>
	<b>Simple State Diagrams</b>	<b>334</b>
	Implement the Actions Performed by the Object	334
	Adding State	336
	Introduce Helper Methods for Managing State Entry and Exit Actions	337
	Implement the Event Handlers (Methods) for Responding to Events	339
	<b>Nested State Diagrams</b>	<b>341</b>
	Implement the Actions Performed by the Object	341
	Adding State	342
	Introduce Helper Methods for Managing State Entry and Exit Actions	343
	Implement the Event Handlers (Methods) for Responding to Events	346

<b>Concurrent State Diagrams</b>	<b>347</b>
Introduction	348
Changes to the Microwave Class	349
Creating a Parallel State Machine	351
Extending Thread Class	351
Implement the Actions Performed by the Object	352
Creating States	352
Introduce Helper Methods for Managing State Entry and Exit Actions	352
Implement the Event Handlers (Methods) for Responding to Events	353
<b>16 Implementing Generalization/Specialization</b>	<b>356</b>
<b>Introduction</b>	<b>356</b>
Java Inheritance Hierarchy	357
Generalization/Specialization Mapping to Inheritance	357
<b>Inheritance</b>	<b>357</b>
Derived Class and Base Class	357
Specifying a Derived Class	358
Inheriting from a Derived Class and Implementing Association	359
Adding Polymorphism	361
Abstract Class	363
Abstract Service (Method)	364
<b>Implementing Generalization/Specialization</b>	<b>366</b>
Interfaces	366
Aggregation with Delegation	369
<b>The Object Class</b>	<b>372</b>
Clone Method	372
Equals and hashCode Methods	373
Finalize Method	374
ToString Method	375
getClass Method	375
Notify, NotifyAll, and Wait Methods	376
<b>17 Implementing More Relationships</b>	<b>380</b>
<b>Introduction</b>	<b>380</b>
Implementing Association	381
Implementing Attributes of an Association	382
Implementing Aggregation	382
<b>References</b>	<b>382</b>
<b>Static Members</b>	<b>384</b>
<b>Implementing Association</b>	<b>385</b>
Binary Association	385
Many-to-One Association	387
Many-to-Many Association	389

<b>Implementing Aggregation</b>	<b>390</b>
Buried Reference	390
Inner Classes	391
<b>Implementing Static Members</b>	<b>391</b>
<b>Recommended Approach</b>	<b>393</b>
<b>A Unified Modeling Language</b>	<b>395</b>
<b>Introduction</b>	<b>395</b>
What Is the Unified Modeling Language?	396
What UML Is Not	396
What Are the Goals of UML?	396
Why Use UML?	396
What Are the Diagrams of UML?	397
What Are the Most Important UML Diagrams?	397
<b>UML Diagrams</b>	<b>398</b>
Use-Case Diagram	398
Class Diagram	400
Sequence Diagram	410
Collaboration Diagram	412
Statechart Diagram	414
Activity Diagram	416
Component Diagram	418
Deployment Diagram	420
<b>UML Glossary</b>	<b>421</b>
<b>B Java Language Summary</b>	<b>425</b>
<b>Reserved Words</b>	<b>426</b>
<b>Comments</b>	<b>426</b>
<b>Literals</b>	<b>427</b>
<b>Variable Declaration</b>	<b>427</b>
<b>Variable Assignment</b>	<b>428</b>
<b>Operators</b>	<b>429</b>
<b>Loops and Conditions</b>	<b>430</b>
<b>Class and Interface Definitions</b>	<b>431</b>
<b>Objects</b>	<b>431</b>
<b>Arrays</b>	<b>432</b>
<b>Method and Constructor Definitions</b>	<b>432</b>
<b>Packages and Importing</b>	<b>434</b>
<b>Exceptions and Synchronization</b>	<b>434</b>

<b>C</b>	<b>How the Java Language Differs from C and C++</b>	<b>435</b>
	<b>C Data Types Not Supported</b>	<b>436</b>
	Pointers	436
	Struct and Union	436
	Enumerated Data Types	438
	<b>Primitive Data Types</b>	<b>438</b>
	<b>Operators</b>	<b>439</b>
	<b>Arguments</b>	<b>439</b>
	Command Line	439
	Method Arguments	440
	<b>Arrays</b>	<b>440</b>
	<b>Java Strings</b>	<b>441</b>
	<b>Memory Management</b>	<b>442</b>
	<b>Miscellaneous</b>	<b>443</b>
	<b>Bibliography</b>	<b>444</b>
	<b>Index</b>	<b>447</b>