
Objective-C Object-Oriented Programming Techniques

Lewis J. Pinson

Richard S. Wiener

University of Colorado at Colorado Springs

Technische Hochschule Darmstadt
FACHBEREICH INFORMATIK

BIBLIOTHEK

Inventar-Nr.: ...7519.....

Sachgebiete: ...D.1.....

Standort: ...1994.....



ADDISON-WESLEY PUBLISHING COMPANY

Reading, Massachusetts • Menlo Park, California • New York
Don Mills, Ontario • Wokingham, England • Amsterdam • Bonn
Sydney • Singapore • Tokyo • Madrid • San Juan • Milan • Paris

Preface xi

Chapter 1

Basic Principles of Object-Oriented Problem Solving 1

- 1.1 Problem Solving 1
- 1.2 The Object-Oriented Approach 2
 - Objects and Organization 2
 - Definitions 2
 - A Simple Example of Object-Oriented Problem Solving 3
- 1.3 The Object-Oriented Paradigm 6
 - Key Components: Object, Message, Class, Instance, and Method 6
 - Applying the Object-Oriented Paradigm 8
- 1.4 Four Essential Properties 9
 - Abstraction 9
 - Encapsulation 10
 - Inheritance and Multiple Inheritance 10
 - Polymorphism 13
- 1.5 Reusable Software Components 17
 - Why Reinvent the Wheel? 17
 - Better Design of Class Hierarchies 17
- 1.6 Containership, Composition, and Object Dependencies 20
- 1.7 Delegation and Targets 21
 - Delegation: Sharing Responsibility 21
 - Targets: Specific Actions 23
- 1.8 Persistent Objects And Archiving 24

Chapter 2

Methods for Object-Oriented Design 25

- 2.1 Mapping a Problem Specification to an Object-Oriented Solution 26
 - The Steps in an Object-Oriented Design 26
 - Notational Conventions 27
 - Problem Definition and Specification: An Example 28
- 2.2 Identifying Objects and Attributes 35
 - Primary Objects in the Swamp Runner Software System 36
 - Attributes and Supporting Objects 37
- 2.3 Developing an Initial Hierarchy of Classes 41

- 2.4 Identifying Key Messages and Communication Requirements among Objects 42
 - Identification of Key Messages for Primary Objects in Swamp Runner 42
 - Refinement of the Key Messages for Swamp Runner 46
- 2.5 A Prototype Solution: The First Level—Expressions that Will Eventually Work 46
 - The Second Level of a Prototype Solution 47
 - The Third Level of a Prototype Solution 47
 - The Fourth Level of a Prototype Solution 48
 - The Fifth Level of a Prototype Solution 48
 - The Sixth and Other Levels of a Prototype Solution 49
- 2.6 Class Descriptions: What About Reusability, Generality, and Robustness? 50

Chapter 3

The Objective-C Language 57

- 3.1 History and Versions 57
- 3.2 Relationship to the C Language 58
- 3.3 Defining Classes 63
 - Interface Files 63
 - Implementation Files 65
 - Adding Methods to a Class: Categories 65
 - Class Objects 66
 - The Use of self and super 67
 - Naming Conventions for Classes, Objects, and Instance Variables 68
- 3.4 Messages, Dynamic Binding, and Type Checking 68
- 3.5 The Mechanics of Messaging 83
- 3.6 The Object Class 86
 - Inheritance Properties of Class Object 88
 - Method Categories in Class Object 89
 - Comparison of NeXT and Stepstone Implementations of Class Object 89
- 3.7 Some Protocol from Class Object 97
 - Support for Memory Management 98
 - Posing 105
 - Support for Inheritance 105
- 3.8 Support for Streams and Persistence In NeXT Objective-C 109
 - Streams 109
 - Connecting Streams to a Source or Destination 111
 - Archiving Objects to a Typed Stream 112
- 3.9 Support for Persistence Using Stepstone Objective-C 117
 - Writing Objects 118
 - Reading Objects 119
 - An Example 119

- 3.10 Defeating Data Hiding and Other Ways to Weaken
an Object-Oriented Solution 120
 - Public Instance Variables 120
 - The Address of a Method 120
 - Getting the Data Structure of an Object 121
- 3.11 The Major Difference Between Stepstone and NeXT Versions
of Objective-C: Static Objects 121

Chapter 4

Foundation Classes: Using Stepstone's ICpak 101 Library 123

- 4.1 Foundation Classes 123
 - The Array Classes 123
 - The Collection Classes 125
 - The Binary Tree Classes 126
 - The String Class 126
 - The Geometry Support Classes 126
 - The Association Class 127
 - The Sequence Classes 127
- 4.2 Using The Array Classes 127
- 4.3 Using The Collection Classes 129
- 4.4 Using The Binary Tree Classes 130
- 4.5 Using The Geometry Support Classes 131
- 4.6 Summary 132

Chapter 5

More Examples: Using Foundation Classes 133

- 5.1 Use of IdArray Class 134
- 5.2 Use of Classes OrdCltn and Sequence 136
- 5.3 Use of Class Set 137
- 5.4 The Swamp Runner Simulation 140
 - The Hierarchy of Classes 141
 - The Main Simulation Driver Program 142
 - SwampSimulation Class 142
 - Swamp Class 142
 - SwampCell Class 152
 - QuickSand Class 155
 - PathCell Class 155
 - TravelerCell Class 155
 - SwampTraveler Class 156
 - Dimwit Class 160
 - Halfwit Class 160
 - Wit Class 165

Genius Class	165
Conclusions	170

Chapter 6

Examples of Object-Oriented Design Using Objective-C 171

6.1 Solitaire Simulation: An Identification of Important Objects and Classes	171
Specifications	171
Analysis	173
Simulation Output	176
6.2 Generalized Numeric Classes	177
Specifications	177
Design	177
Selected Implementation Details	178

Chapter 7

Software Development Environments: User Interfaces and Visual Programming 181

7.1 Hardware and Software Environments	181
Object-Oriented Programming Environments	181
Visual Programming	182
Supporting Software Development Tools	183
7.2 Building Applications with the NeXT Environment	183
NextStep: Hardware, Applications, Windows, and Window Server	183
Essential Components of an Application	185
Predefined Classes and Software Development Tools	185
7.3 Access to the NeXT Environment Using Objective-C	186
Overview of the Common Classes and Application Kit Classes	186
Program Development from a UNIX Shell	190
Program Development from the Interface Builder	190
7.4 Using the Application Kit Class Library	191
Menus, Scrollable Views, Open Panels, Text, Files, and Printing	192
Matrices, Button Cells, and Scrollable, Selectable Lists	209
Displaying Graphic Images	218

Chapter 8

More Examples of Object-Oriented Problem Solving and Objective-C 231

8.1 A Graphic Interface for Swamp Runner, Using Interface Builder	231
8.2 Needs Statement for MiniBrowser— An Objective-C Class Browser	238
8.3 Specification for MiniBrowser	239
8.4 Design for MiniBrowser: The User Interface	243

8.5	MiniBrowser Application: Objects, Attributes, and Structure	246
	Primary Objects	247
	Attributes for the Primary Objects	247
	Supporting Objects	248
	Class Structure for MiniBrowser	249
8.6	Key Methods for MiniBrowser	250
8.7	Selected Implementation Details for MiniBrowser	252
8.8	Example Operation for MiniBrowser	255

Appendix A

<i>Interface File Listings for Class Object</i>	261
---	-----

Appendix B

<i>An Overview of Stepstone's ICpak 101 Libraries</i>	265
---	-----

B.1	ICpak 101 Specification Sheet	265
B.2	Array Class	267
B.3	IdArray Class	270
B.4	IntArray Class	274
B.5	Cltn Class	276
B.6	OrdCltn Class	280
B.7	Stack Class	283
B.8	Set Class	285
B.9	Assoc Class	288
B.10	Dictionary class	289
B.11	BalNode Class	291
B.12	SortCltn Class	292
B.13	String Class	298
B.14	Point Class	302
B.15	Rectangle Class	303
B.16	IPSequence Class	304
B.17	Sequence Class	305

<i>Index</i>	309
--------------	-----