CONTENTS IN DETAIL

ACKNOWLEDGMENTS XIX

INTRODUCTION XXI

Who Is This Book For? xxii
Python Version(s) and Installation xxii
How Will I Explain OOP? xxiii
What's in the Book xxiv
Development Environments xxvi
Widgets and Example Games xxvi

PART I: INTRODUCING OBJECT-ORIENTED PROGRAMMING

1
PROCEDURAL PYTHON EXAMPLES 3

Higher or Lower Card Game 4
Representing the Data 4
Implementation 4
Reusable Code 7
Bank Account Simulations 7
Analysis of Required Operations and Data 7
Implementation 1—Single Account Without Functions 8
Implementation 2—Single Account with Functions 10
Implementation 3—Two Accounts 12
Implementation 4—Multiple Accounts Using Lists 13
Implementation 5—List of Account Dictionaries 16
Common Problems with Procedural Implementation 18
Object-Oriented Solution—First Look at a Class 19
Summary 20

2
MODELING PHYSICAL OBJECTS WITH OBJECT-ORIENTED PROGRAMMING 21

Building Software Models of Physical Objects 22
State and Behavior: Light Switch Example 22
Introduction to Classes and Objects 23
Classes, Objects, and Instantiation 25
Writing a Class in Python 26
Scope and Instance Variables 27

Differences Between Functions and Methods 28
Creating an Object from a Class 28
Calling Methods of an Object 30
Creating Multiple Instances from the Same Class 31
Python Data Types Are Implemented as Classes 32
Definition of an Object 33
Building a Slightly More Complicated Class 33
Representing a More Complicated Physical Object as a Class 35
Passing Arguments to a Method 40
Multiple Instances 41
Initialization Parameters 43
Classes in Use 45
OOP as a Solution 45
Summary 46

3
MENTAL MODELS OF OBJECTS AND THE MEANING OF "SELF" 47

Revisiting the DimmerSwitch Class 48
High-Level Mental Model #1 49
A Deeper Mental Model #2 49
What Is the Meaning of "self"? 52
Summary 55

4
MANAGING MULTIPLE OBJECTS 57

Bank Account Class 58
Importing Class Code 60
Creating Some Test Code 61
Creating Multiple Accounts 62
Multiple Account Objects in a List 64
Multiple Objects with Unique Identifiers 66
Building an Interactive Menu 68
Creating an Object Manager Object 70
Building the Object Manager Object 72
Main Code That Creates an Object Manager Object 74
Better Error Handling with Exceptions 76
try and except 76
The raise Statement and Custom Exceptions 77
Using Exceptions in Our Bank Program 78
Account Class with Exceptions 78
Optimized Bank Class 79
Main Code That Handles Exceptions 81
Calling the Same Method on a List of Objects 83
Interface vs. Implementation 84
Summary 85

PART II: GRAPHICAL USER INTERFACES WITH PYGAME

5
INTRODUCTION TO PYGAME 89

Installing Pygame 90
Window Details 91
The Window Coordinate System 91
Pixel Colors 94
Event-Driven Programs 95
Using Pygame 96
Bringing Up a Blank Window 97
Drawing an Image 100
Detecting a Mouse Click 102
Handling the Keyboard 105
Creating a Location-Based Animation 109
Using Pygame rects 111
Playing Sounds 114
Playing Sound Effects 114
Playing Background Music 115
Drawing Shapes 116
Reference for Primitive Shapes 118
Summary 120

6
OBJECT-ORIENTED PYGAME 121

Building the Screensaver Ball with OOP Pygame 121
Creating a Ball Class 122
Using the Ball Class 124
Creating Many Ball Objects 125
Creating Many, Many Ball Objects 126
Building a Reusable Object-Oriented Button 127
Building a Button Class 128
Main Code Using a SimpleButton 130
Creating a Program with Multiple Buttons 131
Building a Reusable Object-Oriented Text Display 133
Steps to Display Text 133
Creating a SimpleText Class 133
Demo Ball with SimpleText and SimpleButton 135
Interface vs. Implementation 137
Callbacks 137
Creating a Callback 138
Using a Callback with SimpleButton 139
Summary 141

7
PYGAME GUI WIDGETS 143

Passing Arguments into a Function or Method . . . 144
Positional and Keyword Parameters. . . . 145
Additional Notes on Keyword Parameters . . . 146
Using None as a Default Value . . . 146
Choosing Keywords and Default Values . . . 148
Default Values in GUI Widgets . . . 148
The pygwidgets Package . . . 148
Setting Up . . . 149
Overall Design Approach . . . 150
Adding an Image . . . 151
Adding Buttons, Checkboxes, and Radio Buttons. . . . 152
Text Output and Input . . . 154
Other pygwidgets Classes . . . 157
pygwidgets Example Program . . . 157
The Importance of a Consistent API . . . 158
Summary . . . 158

PART III: ENCAPSULATION, POLYMORPHISM, AND INHERITANCE

8
ENCAPSULATION 163

Encapsulation with Functions . . . 164
Encapsulation with Objects . . . 164
Objects Own Their Data . . . 165
Interpretations of Encapsulation . . . 165
Direct Access and Why You Should Avoid It . . . 166
Strict Interpretation with Getters and Setters . . . 170
Safe Direct Access. . . . 172
Making Instance Variables More Private . . . 172
Implicitly Private . . . 172
More Explicitly Private . . . 173
Decorators and @property . . . 174
Encapsulation in pygwidgets Classes . . . 177
A Story from the Real World . . . 178
Abstraction . . . 179
Summary . . . 182

9
POLYMORPHISM 183

Sending Messages to Real-World Objects . . . 184
A Classic Example of Polymorphism in Programming . . . 184
Example Using Pygame Shapes . . . 185
The Square Shape Class . . . 186
The Circle and Triangle Shape Classes . . . 187

The Main Program Creating Shapes . . . 190
Extending a Pattern . . . 192
pygwidgets Exhibits Polymorphism . . . 192
Polymorphism for Operators . . . 193
Magic Methods . . . 194
Comparison Operator Magic Methods . . . 195
A Rectangle Class with Magic Methods . . . 196
Main Program Using Magic Methods . . . 198
Math Operator Magic Methods . . . 200
Vector Example . . . 201
Creating a String Representation of Values in an Object . . . 203
A Fraction Class with Magic Methods . . . 205
Summary . . . 208

10
INHERITANCE 211

Inheritance in Object-Oriented Programming . . . 212
Implementing Inheritance . . . 213
Employee and Manager Example . . . 214
Base Class: Employee . . . 214
Subclass: Manager . . . 215
Test Code . . . 217
The Client's View of a Subclass . . . 218
Real-World Examples of Inheritance . . . 219
InputNumber . . . 219
DisplayMoney . . . 222
Example Usage . . . 224
Multiple Classes Inheriting from the Same Base Class . . . 227
Abstract Classes and Methods . . . 231
How pygwidgets Uses Inheritance . . . 234
Class Hierarchy . . . 236
The Difficulty of Programming with Inheritance . . . 238
Summary . . . 239

11
MANAGING MEMORY USED BY OBJECTS 241

Object Lifetime . . . 242
Reference Count . . . 242
Garbage Collection . . . 248
Class Variables . . . 248
Class Variable Constants . . . 249
Class Variables for Counting . . . 250
Putting It All Together: Balloon Sample Program . . . 251
Module of Constants . . . 253
Main Program Code . . . 254
Balloon Manager . . . 256
Balloon Class and Objects . . . 258
Managing Memory: Slots . . . 261
Summary . . . 263

PART IV: USING OOP IN GAME DEVELOPMENT

12
CARD GAMES 267

The Card Class 268
The Deck Class 270
The Higher or Lower Game 272
Main Program 272
Game Object 274
Testing with __name__ 276
Other Card Games 278
Blackjack Deck 278
Games with Unusual Card Decks 279
Summary 279

13
TIMERS 281

Timer Demonstration Program 282
Three Approaches for Implementing Timers 283
Counting Frames 283
Timer Event. 284
Building a Timer by Calculating Elapsed Time. 285
Installing pyghelpers 287
The Timer Class 287
Displaying Time 290
CountUpTimer. 291
CountDownTimer 293
Summary 294

14
ANIMATION 295

Building Animation Classes 296
SimpleAnimation Class 296
SimpleSpriteSheetAnimation Class 300
Merging Two Classes 304
Animation Classes in pygwidgets. 304
Animation Class 305
SpriteSheetAnimation Class 306
Common Base Class: PygAnimation 307
Example Animation Program 308
Summary 310

15
SCENES 311

The State Machine Approach 312
A pygame Example with a State Machine. 314
A Scene Manager for Managing Many Scenes 319

A Demo Program Using a Scene Manager 320
The Main Program 322
Building the Scenes 323
A Typical Scene 326
Rock, Paper, Scissors Using Scenes 328
Communication Between Scenes 332
Requesting Information from a Target Scene 333
Sending Information to a Target Scene. 333
Sending Information to All Scenes 334
Testing Communications Among Scenes. 334
Implementation of the Scene Manager 334
run() Method 336
Main Methods 337
Communication Between Scenes. 338
Summary 340

16
FULL GAME: DODGER 341

Modal Dialogs 342
Yes/No and Alert Dialogs 342
Answer Dialogs 345
Building a Full Game: Dodger 347
Game Overview 347
Implementation 348
Extensions to the Game 366
Summary 366

17
DESIGN PATTERNS AND WRAP-UP 367

Model View Controller 367
File Display Example 368
Statistical Display Example. 368
Advantages of the MVC Pattern 373
Wrap-Up 374

INDEX 377