

Contents

Part 1 Introduction to Object Orientation

| | | |
|----------|---|-----------|
| 1 | Introduction to Object Orientation | 3 |
| 1.1 | Introduction | 3 |
| 1.2 | Programming Paradigms | 3 |
| 1.3 | Revolution Versus Evolution | 4 |
| 1.4 | Why Learn a New Programming Paradigm? | 5 |
| 1.4.1 | Software Industry Blues | 5 |
| 1.4.2 | The Advantages Claimed for Object Orientation | 7 |
| 1.4.3 | What Are the Problems and Pitfalls of Object Orientation? | 7 |
| 1.5 | Pedigree of Object-Oriented Languages | 10 |
| 1.6 | Fundamentals of Object Orientation | 11 |
| 1.7 | The Basic Principles of Object Orientation | 12 |
| 1.8 | Encapsulation | 13 |
| 1.9 | Inheritance | 14 |
| 1.10 | Abstraction | 16 |
| 1.11 | Polymorphism | 17 |
| 1.11.1 | Overloading Operators | 17 |
| 1.11.2 | Overriding Operators | 18 |
| 1.12 | Summary | 18 |
| 1.13 | Further Reading | 19 |
| | | |
| 2 | Elements of Object Orientation | 21 |
| 2.1 | Introduction | 21 |
| 2.2 | Terminology | 21 |
| 2.3 | Types of Hierarchy | 23 |
| 2.4 | The Move to Object Technology | 25 |
| 2.5 | Summary | 25 |
| 2.6 | Exercises | 26 |
| 2.7 | Further Reading | 26 |
| | | |
| 3 | Constructing an Object-Oriented System | 27 |
| 3.1 | Introduction | 27 |
| 3.2 | The Application: Windscreen Wipe Simulation | 27 |
| 3.3 | Where Do We Start? | 28 |
| 3.4 | Identifying the Objects | 29 |
| 3.5 | Identifying the Services Or Methods | 30 |
| 3.6 | Refining the Objects | 31 |
| 3.7 | Bringing it All Together | 31 |
| 3.8 | Where Is the Structure? | 34 |
| 3.9 | Summary | 35 |
| 3.10 | Exercise | 35 |
| 3.11 | Further Reading | 36 |

Part 2 Introduction to the Java Language

| | | |
|----------|---|-----------|
| 4 | A Brief History of Time, the Universe and Java | 39 |
| 4.1 | Introduction | 39 |
| 4.2 | What Is Java? | 39 |
| 4.3 | Objects in Java | 40 |
| 4.4 | History | 40 |
| 4.5 | Commercial Versions of Java | 41 |
| 4.6 | The Java Environment | 41 |
| 4.6.1 | What Is in a Name? | 42 |
| 4.6.2 | What Is JavaScript? | 44 |
| 4.6.3 | Applications and Applets | 44 |
| 4.6.4 | Applications in Java | 44 |
| 4.7 | Further Reading | 45 |
| 4.8 | Where to Get More Information | 45 |
| | | |
| 5 | A Little Java | 47 |
| 5.1 | Introduction | 47 |
| 5.2 | Setting up the Development Environment | 48 |
| 5.3 | Compiling and Executing Java | 48 |
| 5.4 | Using the Java Documentation Tool | 50 |
| 5.5 | Summary | 51 |
| 5.6 | Further Reading | 51 |
| | | |
| 6 | Java Building Blocks | 53 |
| 6.1 | Introduction | 53 |
| 6.2 | The Basics of the Language | 53 |
| 6.2.1 | Some Terminology | 54 |
| 6.2.2 | The Message-Passing Mechanism | 54 |
| 6.2.3 | The Statement Terminator | 54 |
| 6.3 | Classes | 54 |
| 6.3.1 | Class Definitions | 55 |
| 6.3.2 | Classes and Messages | 55 |
| 6.3.3 | Instances and Instance Variables | 56 |
| 6.3.4 | Classes and Inheritance | 56 |
| 6.3.5 | Instance Creation | 58 |
| 6.3.6 | Constructors | 59 |
| 6.3.7 | Static Initialization Blocks | 59 |
| 6.3.8 | Finalize Methods | 60 |
| 6.3.9 | Supplied Classes | 60 |
| 6.4 | Method Definitions | 61 |
| 6.4.1 | The Comments Section | 61 |
| 6.4.2 | The Local Variables Section | 61 |
| 6.4.3 | The Statements Section | 62 |
| 6.4.4 | The Return Operator | 63 |
| 6.4.5 | An Example Method | 63 |
| 6.5 | Interface Definitions | 64 |
| | | |
| 7 | Java Constructs | 67 |
| 7.1 | Introduction | 67 |
| 7.2 | Numbers and Numeric Operators | 67 |
| 7.2.1 | Numeric Values | 67 |
| 7.2.2 | Arithmetic Operators | 68 |

| | | |
|---|---|-----------|
| 7.3 | Characters and Strings | 69 |
| 7.3.1 | Characters | 69 |
| 7.3.2 | Strings | 69 |
| 7.4 | Assignments | 70 |
| 7.5 | Variables | 71 |
| 7.5.1 | Temporary Variables | 71 |
| 7.5.2 | Pseudo Variables | 72 |
| 7.5.3 | Variable Scope | 73 |
| 7.5.4 | Special Values – True, False and Null | 73 |
| 7.6 | Messages and Message Selectors | 73 |
| 7.6.1 | Invoking Methods | 73 |
| 7.6.2 | Precedence | 74 |
| 7.7 | Summary | 74 |
| 7.8 | Further Reading | 74 |
| 8 | An Example Java Class | 75 |
| 8.1 | Introduction | 75 |
| 8.2 | Defining a Class | 75 |
| 8.2.1 | Creating the Class | 75 |
| 8.2.2 | Defining a Class Comment | 76 |
| 8.3 | Defining a Method | 76 |
| 8.3.1 | The main Method | 76 |
| 8.3.2 | The Constructor | 76 |
| 8.3.3 | The Accessor Methods | 77 |
| 8.3.4 | The Updater Method | 78 |
| 8.3.5 | The birthday Method | 78 |
| 8.4 | Creating an Instance | 78 |
| Part 3 Java and Object Orientation | | |
| 9 | Classes, Inheritance and Abstraction | 83 |
| 9.1 | Introduction | 83 |
| 9.2 | Classes Revisited | 83 |
| 9.2.1 | What Are Classes For? | 84 |
| 9.2.2 | Class-Side Methods | 84 |
| 9.2.3 | A Class or an Instance | 85 |
| 9.3 | Inheritance in Classes | 86 |
| 9.3.1 | The Role of a Subclass | 86 |
| 9.3.2 | Capabilities of Classes | 87 |
| 9.3.3 | Restricting a Subclass | 88 |
| 9.4 | Abstract Classes | 89 |
| 9.5 | Constructors and Their Use | 90 |
| 9.6 | The main Method | 91 |
| 10 | Encapsulation and Polymorphism | 93 |
| 10.1 | Introduction | 93 |
| 10.2 | Encapsulation | 93 |
| 10.2.1 | Class Modifiers | 93 |
| 10.2.2 | Variable Modifiers | 93 |
| 10.2.3 | Method Modifiers | 94 |
| 10.3 | Packages | 94 |
| 10.3.1 | Declaring a Package | 94 |
| 10.3.2 | An Example Package | 95 |

| | | |
|-----------|--|------------|
| 10.3.3 | Accessing Package Elements | 96 |
| 10.3.4 | An Example of Using a Package | 96 |
| 10.4 | Polymorphism | 97 |
| 10.4.1 | Dynamic or Late Binding | 97 |
| 10.4.2 | Method Selection | 98 |
| 11 | Inner Classes and Reflection | 101 |
| 11.1 | Introduction | 101 |
| 11.2 | What Are Inner Classes? | 101 |
| 11.3 | Types of Inner Class | 103 |
| 11.3.1 | Nested Top-Level Classes | 103 |
| 11.3.2 | Member Inner Classes | 104 |
| 11.3.3 | Method Level/Local Inner Classes | 104 |
| 11.3.4 | Anonymous Inner Classes | 105 |
| 11.4 | How and When Should I Use Inner Classes? | 105 |
| 11.4.1 | As Helper Classes | 105 |
| 11.4.2 | As Event Handlers | 105 |
| 11.4.3 | As Anonymous Event Handlers | 106 |
| 11.4.4 | Laying Out a Java Class With Inner Classes | 106 |
| 11.4.5 | Inner class guidelines | 107 |
| 11.5 | The Reflection API | 107 |
| 12 | Data Structures | 113 |
| 12.1 | Introduction | 113 |
| 12.2 | Data Structure Classes | 113 |
| 12.3 | The Abstract Class Dictionary | 114 |
| 12.4 | The Hashtable Class | 114 |
| 12.5 | The Vector Class | 115 |
| 12.6 | The Stack Class | 118 |
| 12.7 | A Queue Class | 118 |
| 12.8 | Enumeration | 118 |
| 12.9 | Arrays | 119 |
| 12.9.1 | Arrays of Objects | 120 |
| 12.9.2 | Basic Type Arrays | 121 |
| 12.9.3 | Multi-Dimensional Arrays | 121 |
| 12.9.4 | The main Method | 122 |
| 12.10 | Memory Management | 124 |
| 12.10.1 | Why Have Automatic Memory Management? | 124 |
| 12.10.2 | Memory Management in Java | 125 |
| 12.10.3 | When Is Garbage Collection Performed? | 125 |
| 12.10.4 | Checking the Available Memory | 125 |
| 12.11 | Exercise: Vectors | 126 |
| 12.12 | Summary | 127 |
| 12.13 | Further Reading | 127 |
| 13 | The Collections API | 129 |
| 13.1 | Introduction | 129 |
| 13.2 | What Is in the Collections API? | 129 |
| 13.3 | Collection Interfaces | 131 |
| 13.3.1 | Collection | 131 |
| 13.3.2 | Set | 133 |
| 13.3.3 | List | 133 |
| 13.3.4 | Map | 134 |

| | | |
|--------|-----------------------------|-----|
| 13.3.5 | Comparisons | 135 |
| 13.4 | Abstract Implementations | 135 |
| 13.4.1 | AbstractCollection | 135 |
| 13.4.2 | AbstractSet | 136 |
| 13.4.3 | AbstractList | 136 |
| 13.4.4 | AbstractSequentialList | 137 |
| 13.4.5 | AbstractMap | 137 |
| 13.5 | Concrete Implementations | 137 |
| 13.5.1 | HashSet | 138 |
| 13.5.2 | ArraySet | 139 |
| 13.5.3 | ArrayList | 139 |
| 13.5.4 | LinkedList | 140 |
| 13.5.5 | HashMap | 141 |
| 13.5.6 | ArrayMap | 141 |
| 13.5.7 | TreeMap | 142 |
| 13.6 | The Collections Class | 142 |
| 13.7 | Iteration Over Collections | 143 |
| 13.8 | Array Sorting and Searching | 144 |
| 13.9 | Choosing a Collection class | 144 |
| 13.10 | Summary | 144 |

Part 4 Further Java

| | | |
|--------|--|-----|
| 14 | Control and Iteration | 149 |
| 14.1 | Introduction | 149 |
| 14.2 | Control Structures | 149 |
| 14.2.1 | The If Statement | 149 |
| 14.2.2 | The Conditional Operator | 151 |
| 14.2.3 | The switch Statement | 151 |
| 14.3 | Iteration | 153 |
| 14.3.1 | for Loops | 153 |
| 14.3.2 | while Loops | 154 |
| 14.3.3 | do Loops | 154 |
| 14.3.4 | An Example of Loops | 155 |
| 14.4 | Recursion | 155 |
| 14.5 | Summary | 156 |
| 15 | An Object-Oriented Organizer | 157 |
| 15.1 | Introduction | 157 |
| 15.2 | The Organizer Class | 157 |
| 15.3 | The Class Definition | 158 |
| 15.4 | The Updating Protocol | 158 |
| 15.5 | The Accessing Protocol | 159 |
| 15.6 | The main Method | 160 |
| 15.7 | Exercise – the Financial Manager Project | 161 |
| 16 | Streams and Files | 163 |
| 16.1 | Introduction | 163 |
| 16.2 | Streams | 163 |
| 16.2.1 | What Is a Stream? | 163 |
| 16.2.2 | Readers and Writers: Character Streams | 165 |
| 16.2.3 | Stream Tokenizers | 167 |
| 16.2.4 | Using the IO Classes | 167 |

| | | |
|-----------|--|------------|
| 16.3 | Files | 171 |
| 16.3.1 | Accessing File Information | 171 |
| 16.3.2 | The <code>FileReader</code> Class Constructors | 171 |
| 16.3.3 | The <code>FileWriter</code> Class Constructors | 172 |
| 16.3.4 | The <code>BufferedReader</code> Class Constructors | 172 |
| 16.3.5 | The <code>BufferedWriter</code> Class Constructors | 172 |
| 16.3.6 | The <code>StreamTokenizer</code> Class | 172 |
| 16.3.7 | The <code>PrintWriter</code> Class Constructors | 173 |
| 16.3.8 | Handling File IO Errors | 173 |
| 16.4 | Accessing a File | 173 |
| 16.5 | Creating a File | 175 |
| 16.6 | Input From the Console | 176 |
| 16.7 | Summary | 177 |
| 17 | Serialization | 179 |
| 17.1 | Introduction | 179 |
| 17.1.1 | Saving Objects | 180 |
| 17.1.2 | Reading Objects | 180 |
| 17.2 | The <code>ObjectOutputStream</code> Class | 180 |
| 17.3 | The <code>ObjectInputStream</code> Class | 180 |
| 17.4 | The <code>Serializable</code> Interface | 181 |
| 17.5 | The <code>transient</code> Keyword | 182 |
| 17.6 | The <code>Externalizable</code> Interface | 182 |
| 17.7 | A Simple Serialization Application | 183 |
| 17.7.1 | The <code>Person</code> Class | 183 |
| 17.7.2 | The <code>Family</code> Class | 183 |
| 17.7.3 | The <code>Test</code> Class | 184 |
| 17.8 | Exercise – Using Files With the Financial Manager | 186 |
| 17.9 | Summary | 188 |
| 18 | Observers and Observables | 189 |
| 18.1 | Introduction | 189 |
| 18.2 | The Dependency Mechanism | 189 |
| 18.2.1 | Why Do We Want Dependency? | 189 |
| 18.2.2 | How Does Dependency Work? | 190 |
| 18.2.3 | Constructing Dependencies | 190 |
| 18.2.4 | A Simple Dependency Example | 191 |
| 18.2.5 | Making Dependency Work for You | 192 |
| 18.2.6 | The “Changed” Methods | 193 |
| 18.3 | The <code>Observer</code> Interface | 193 |
| 18.4 | Extending the Dependency Example | 194 |
| 18.5 | Exercise – Dependency and the Financial Manager | 196 |
| 18.6 | Summary | 196 |

Part 5 Graphical Interfaces and Applets

| | | |
|-----------|--|------------|
| 19 | Graphic Programming Using the Abstract Window Toolkit | 199 |
| 19.1 | Introduction | 199 |
| 19.2 | Windows as Objects | 199 |
| 19.3 | Windows in Java | 200 |
| 19.4 | The Abstract Window Toolkit | 200 |
| 19.5 | The <code>Component</code> Class | 201 |
| 19.6 | The <code>Container</code> Class | 203 |

| | | |
|-----------|---------------------------------------|------------|
| 19.7 | The Panel Class | 204 |
| 19.8 | The Frame Class | 204 |
| 19.9 | The Graphics Class | 205 |
| 19.10 | A Worked Graphical Application | 207 |
| 19.10.1 | The ExampleFrame Class | 207 |
| 19.10.2 | The ExamplePanel Class | 209 |
| 19.11 | Further Reading | 210 |
| 20 | User Interface Programming | 211 |
| 20.1 | Introduction | 211 |
| 20.2 | The Event Delegation Model | 211 |
| 20.2.1 | The Philosophy Behind the Event Model | 211 |
| 20.2.2 | A Partial Example | 213 |
| 20.2.3 | The Listener Interfaces | 214 |
| 20.3 | GUI Component Classes | 215 |
| 20.3.1 | The Button Class | 215 |
| 20.3.2 | The Checkbox Class | 216 |
| 20.3.3 | The CheckboxGroup Class | 217 |
| 20.3.4 | The Label Class | 218 |
| 20.3.5 | The TextComponent Class | 218 |
| 20.3.6 | The TextField Class | 219 |
| 20.3.7 | The TextArea Class | 219 |
| 20.4 | Additional AWT Classes | 220 |
| 21 | Managing Component Layout | 221 |
| 21.1 | Introduction | 221 |
| 21.2 | The FlowLayout Manager | 221 |
| 21.3 | The BorderLayout Manager | 222 |
| 21.4 | The GridLayout Manager | 223 |
| 21.5 | The GridBagLayout Manager | 224 |
| 21.6 | The CardLayout Manager | 227 |
| 21.7 | A Simple GUI Example | 227 |
| 22 | Putting the Swing into Java | 231 |
| 22.1 | Introduction | 231 |
| 22.2 | Swing, the JFC and the JDK | 231 |
| 22.3 | What Is the MVC? | 232 |
| 22.4 | Swinging the MVC Into Action | 233 |
| 22.5 | Transitioning to Swing | 234 |
| 22.5.1 | The SimpleGallery Application | 234 |
| 22.6 | A Swinging Gallery | 237 |
| 22.7 | Things to Remember | 242 |
| 22.8 | Online References | 243 |
| 23 | A GUI Case Study | 245 |
| 23.1 | Introduction | 245 |
| 23.1.1 | The Model–View–Handler Architecture | 246 |
| 23.1.2 | Event Listeners | 247 |
| 23.1.3 | Observers and Observables | 247 |
| 23.1.4 | Frames, Panels and Layout Managers | 247 |
| 23.2 | The Class Structure | 248 |
| 23.2.1 | The AccountInterface Class | 248 |

| | | |
|-----------|---|------------|
| 23.2.2 | The WindowHandler class | 250 |
| 23.2.3 | The JPanel Classes | 250 |
| 23.2.4 | The Handler Classes | 251 |
| 23.2.5 | The Account Class | 254 |
| 23.3 | The Instance Structure | 255 |
| 23.4 | Exercise – A GUI for the Financial Manager | 255 |
| 23.5 | Summary | 256 |
| 23.6 | Further Reading | 256 |
| 24 | The Lowdown on Layouts, Borders and Containers | 257 |
| 24.1 | Introduction | 257 |
| 24.2 | Containers in Swing | 257 |
| 24.3 | Layouts for Containers | 258 |
| 24.3.1 | The FlowLayout Manager | 258 |
| 24.3.2 | The BorderLayout Manager | 259 |
| 24.3.3 | The GridLayout Manager | 259 |
| 24.3.4 | The GridBagLayout Manager | 260 |
| 24.3.5 | BoxLayout | 260 |
| 24.3.6 | OverlayLayout | 262 |
| 24.4 | Borders in Swing | 262 |
| 24.4.1 | The BorderFactory | 264 |
| 24.4.2 | Empty Borders | 265 |
| 24.4.3 | LineBorder class | 266 |
| 24.4.4 | BevelBorder and SoftBevelBorder | 266 |
| 24.4.5 | EtchedBorder | 267 |
| 24.4.6 | The MatteBorder class | 267 |
| 24.4.7 | The TitledBorder | 268 |
| 24.4.8 | CompoundBorder class | 269 |
| 24.5 | Using Panels, Layouts and Borders | 270 |
| 24.6 | Planning a Display | 270 |
| 24.7 | Online References | 271 |
| 25 | Combining Graphics and GUI Components | 273 |
| 25.1 | Introduction | 273 |
| 25.2 | The SwingDraw Application | 273 |
| 25.3 | The Structure of the Application | 273 |
| 25.4 | The Interactions Between Objects | 277 |
| 25.4.1 | The Draw Constructor | 278 |
| 25.4.2 | Changing the Type of Graphic Object | 278 |
| 25.4.3 | Adding a Graphic Object | 279 |
| 25.4.4 | Moving a Graphic Object | 280 |
| 25.5 | The Classes | 281 |
| 25.5.1 | The SwingDraw Class | 281 |
| 25.5.2 | The DrawMenuBar class | 283 |
| 25.5.3 | The DrawToolBar and ButtonPanel Classes | 284 |
| 25.5.4 | The ToolHandler Class | 285 |
| 25.5.5 | The Drawing Class | 286 |
| 25.5.6 | The DrawingPanel Class | 287 |
| 25.5.7 | The DrawingController Class | 289 |
| 25.5.8 | The LookAndFeelPanel Class | 291 |
| 25.5.9 | The DrawingException Class | 292 |
| 25.5.10 | The Figure Class | 293 |
| 25.5.11 | The Square Class | 295 |

| | | |
|------------------------------------|--------------------------------------|------------|
| 25.5.12 | The Circle Class | 295 |
| 25.5.13 | The Line Class | 295 |
| 25.5.14 | The TextFigure Class | 297 |
| 25.5.15 | The TextDialog Class | 298 |
| 25.6 | Exercises | 298 |
| 25.7 | Summary | 299 |
| 26 | Swing Data Model Case Study | 301 |
| 26.1 | Introduction | 301 |
| 26.2 | The JTree Swing Component | 301 |
| 26.2.1 | The JTree Constructors | 302 |
| 26.2.2 | The JTree API | 303 |
| 26.2.3 | Changing the Model | 304 |
| 26.3 | The JTree Package | 305 |
| 26.3.1 | Interfaces | 305 |
| 26.3.2 | Classes | 305 |
| 26.3.3 | The DefaultMutableTreeNode class | 306 |
| 26.4 | Building the Data Model | 307 |
| 26.5 | Building the GUI Application | 308 |
| 26.6 | Online Resources | 314 |
| 27 | Java: Speaking in Tongues | 315 |
| 27.1 | Introduction | 315 |
| 27.2 | Locale | 316 |
| 27.3 | Properties Objects | 317 |
| 27.4 | ResourceBundle Introduction | 318 |
| 27.4.1 | Naming Properties Files | 319 |
| 27.4.2 | Property Inheritance | 320 |
| 27.4.3 | Accessing Property Values | 320 |
| 27.5 | Formatting Output | 321 |
| 27.6 | Summary | 322 |
| 27.7 | Online References | 322 |
| 28 | The CUTting Edge | 323 |
| 28.1 | Introduction | 323 |
| 28.2 | The Simple Editor | 323 |
| 28.3 | Cutting and Copying Data | 324 |
| 28.4 | Pasting Data | 325 |
| 28.5 | Drag and Drop | 327 |
| 28.5.1 | DragLabel | 327 |
| 28.5.2 | DropTextArea | 329 |
| 28.6 | Summary | 331 |
| Part 6 Internet Working | | |
| 29 | Sockets in Java | 335 |
| 29.1 | Introduction | 335 |
| 29.2 | Socket to socket communication | 335 |
| 29.3 | Setting Up a Connection | 335 |
| 29.4 | An Example Client-Server Application | 336 |
| 29.4.1 | The System Structure | 336 |
| 29.4.2 | Implementing the Server Application | 336 |
| 29.4.3 | Implementing the Client Application | 338 |

| | | |
|-----------|---|-----|
| 30 | Applets and the Internet | 341 |
| 30.1 | Introduction | 341 |
| 30.2 | Applet Security | 342 |
| 30.3 | The Applet Class | 342 |
| 30.4 | Working With Applets | 344 |
| 30.5 | The Account Applet | 344 |
| 30.5.1 | Change the Root Class Definition | 345 |
| 30.5.2 | Define the <code>init</code> Method | 345 |
| 30.6 | A Brief Introduction to HTML | 346 |
| 30.7 | The <code><applet></code> HTML Tag | 347 |
| 30.8 | Accessing HTML Files | 348 |
| 30.9 | Swing and Applets | 349 |
| 30.10 | Exercise: Tic-Tac-Toe Applet | 349 |
| 30.10.1 | What to Do | 349 |
| 30.10.2 | Notes | 350 |
| 30.11 | Summary | 350 |
| 30.12 | Further Reading | 350 |
| | | |
| 31 | Servlets: Serving Java up on the Web | 353 |
| 31.1 | Introduction | 353 |
| 31.2 | How Servlets Work | 354 |
| 31.3 | The Structure of the Servlet API | 355 |
| 31.4 | An Example Servlet | 356 |
| 31.5 | Why Use Servlets? | 359 |
| 31.6 | Summary | 360 |
| 31.7 | Further Reading | 360 |
| | | |
| 32 | Java Server Pages | 361 |
| 32.1 | Introduction | 361 |
| 32.2 | What Is a JSP? | 361 |
| 32.3 | A Very Simple JSP | 363 |
| 32.4 | The Components of a JSP | 364 |
| 32.4.1 | Directives | 365 |
| 32.4.2 | Actions | 365 |
| 32.4.3 | Implicit Objects | 366 |
| 32.4.4 | JSP Scripting | 366 |
| 32.5 | Making JSPs Interactive | 367 |
| 32.6 | Why Use JSPs? | 368 |
| 32.7 | Problems With JSPs | 368 |
| | | |
| 33 | Java Database Connectivity | 371 |
| 33.1 | Introduction | 371 |
| 33.2 | What Is JDBC? | 371 |
| 33.3 | What the Driver Provides | 373 |
| 33.4 | Registering Drivers | 373 |
| 33.5 | Opening a Connection | 374 |
| 33.6 | Obtaining Data From a Database | 375 |
| 33.7 | Creating a Table | 377 |
| 33.8 | Applets and Databases | 378 |
| 33.9 | Mini SQL | 379 |
| 33.10 | Further Reading | 379 |
| 33.11 | Online References | 379 |

Part 7 Java Development

| | | |
|-----------|-------------------------------------|-----|
| 34 | Java Style Guidelines | 383 |
| 34.1 | Introduction | 383 |
| 34.2 | Code Layout | 383 |
| 34.3 | Variables | 383 |
| 34.3.1 | Naming Variables | 383 |
| 34.3.2 | Using Variables | 385 |
| 34.4 | Classes | 386 |
| 34.4.1 | Naming Classes | 386 |
| 34.4.2 | The Role of a Class | 387 |
| 34.4.3 | Creating New Data Structure Classes | 387 |
| 34.4.4 | Class Comments | 387 |
| 34.4.5 | Using a Class or an Instance | 388 |
| 34.5 | Interfaces | 388 |
| 34.6 | Enumerated Types | 388 |
| 34.7 | Methods | 389 |
| 34.7.1 | Naming Methods | 389 |
| 34.7.2 | Using Methods | 390 |
| 34.7.3 | Class Methods and Instance Methods | 391 |
| 34.7.4 | Static Initialization Blocks | 391 |
| 34.7.5 | Constructors | 391 |
| 34.7.6 | The finalize Method | 391 |
| 34.7.7 | Programming in Terms of Objects | 391 |
| 34.7.8 | Positioning of Methods | 391 |
| 34.8 | Scoping | 392 |
| 34.9 | Statement Labels | 393 |
| 35 | Exception Handling | 395 |
| 35.1 | Introduction | 395 |
| 35.2 | What Is an Exception? | 395 |
| 35.3 | What Is Exception Handling? | 396 |
| 35.4 | Throwing an Exception | 397 |
| 35.5 | Catching an Exception | 398 |
| 35.6 | Defining an Exception | 400 |
| 36 | Concurrency | 403 |
| 36.1 | Introduction | 403 |
| 36.2 | Concurrent Processes | 403 |
| 36.3 | Threads | 404 |
| 36.3.1 | Thread States | 404 |
| 36.3.2 | Creating a Thread | 405 |
| 36.3.3 | Thread Groups | 405 |
| 36.4 | The Thread Class | 405 |
| 36.4.1 | Implementing a Thread | 406 |
| 36.4.2 | Synchronization | 407 |
| 36.5 | A Time Slicing Example | 408 |
| 36.5.1 | Using Schedulers | 408 |
| 36.5.2 | The Time Slicing Source Code | 408 |

Part 8 Object-Oriented Design

| | | |
|-----------|--|------------|
| 37 | Object-Oriented Analysis and Design | 415 |
| 37.1 | Introduction | 415 |
| 37.2 | Object-Oriented Design Methods | 415 |
| 37.3 | Object-Oriented Analysis | 415 |
| 37.4 | The Booch Method | 416 |
| 37.4.1 | The Steps in the Booch Method | 416 |
| 37.4.2 | Strengths and Weaknesses | 417 |
| 37.5 | The Object Modeling Technique | 417 |
| 37.5.1 | The Analysis Phase | 417 |
| 37.5.2 | The Design Phase | 418 |
| 37.5.3 | The Implementation Phase | 418 |
| 37.5.4 | Strengths and Weaknesses | 419 |
| 37.6 | The Objectory Method | 419 |
| 37.6.1 | The Requirements Phase | 419 |
| 37.6.2 | The Analysis Phase | 419 |
| 37.6.3 | The Construction Phase | 419 |
| 37.6.4 | Strengths and Weaknesses | 420 |
| 37.7 | The Fusion Method | 420 |
| 37.8 | The Unified Modeling Language | 421 |
| 37.9 | Summary | 421 |
| | | |
| 38 | The Unified Modeling Language | 423 |
| 38.1 | Introduction | 423 |
| 38.2 | The Meta-Model | 424 |
| 38.3 | The Models | 424 |
| 38.4 | Use Case Diagrams | 425 |
| 38.5 | The Object Model | 426 |
| 38.5.1 | Representing Classes | 426 |
| 38.5.2 | Representing Objects | 428 |
| 38.5.3 | Representing Relationships | 428 |
| 38.6 | Packages | 432 |
| 38.7 | Sequence Diagrams | 434 |
| 38.8 | Collaboration Diagrams | 435 |
| 38.9 | State Machine Diagrams | 437 |
| 38.9.1 | Start Points | 437 |
| 38.9.2 | Events | 438 |
| 38.9.3 | A Set of Transitions | 438 |
| 38.9.4 | A Set of State Variables | 439 |
| 38.9.5 | A Set of States | 439 |
| 38.9.6 | A Set of Exit Points | 440 |
| 38.10 | Deployment Diagrams | 440 |
| 38.11 | Summary | 440 |
| | | |
| 39 | The Unified Process | 443 |
| 39.1 | Introduction | 443 |
| 39.2 | The Unified Process | 443 |
| 39.2.1 | Overview of the Unified Process | 444 |
| 39.2.2 | Life Cycle Phases | 445 |
| 39.2.3 | Phases, Iterations and Workflows | 446 |
| 39.2.4 | Workflows and Activities | 448 |
| 39.3 | Requirements Workflow | 448 |

| | | | |
|------|---------|--|-----|
| | 39.3.1 | Interface Descriptions | 449 |
| 39.4 | | Analysis Workflow | 449 |
| | 39.4.1 | Analysis Model Classes | 449 |
| | 39.4.2 | Constructing the Analysis Model | 451 |
| 39.5 | | Design Workflow | 454 |
| | 39.5.1 | Identifying Classes | 454 |
| | 39.5.2 | Refining the Set of Classes | 455 |
| | 39.5.3 | Identifying and Refining Attributes | 455 |
| | 39.5.4 | Identifying and Refining Operations | 455 |
| | 39.5.5 | Design Use Case Realizations | 456 |
| | 39.5.6 | Generating a Sequence Diagram | 456 |
| | 39.5.7 | Building a Statechart Diagram | 456 |
| | 39.5.8 | Identifying and Refining Associations | 457 |
| | 39.5.9 | Identifying Interfaces and Inheritance | 457 |
| | 39.5.10 | Remaining Steps | 458 |
| 39.6 | | Implementation Workflow | 458 |
| 39.7 | | Testing Workflow | 458 |
| 39.8 | | Summary | 458 |

Part 9 The Future

| | | | |
|-----------|--------|--|------------|
| 40 | | Areas of Java and Object Technology not Covered | 461 |
| 40.1 | | Introduction | 461 |
| 40.2 | | Language Areas | 461 |
| | 40.2.1 | Native Code | 461 |
| 40.3 | | Java Virtual Machine | 462 |
| 40.4 | | JavaBeans | 462 |
| 40.5 | | Enterprise JavaBeans | 463 |
| 40.6 | | Remote Method Invocation | 464 |
| 40.7 | | Java and C++ | 465 |
| 40.8 | | CORBA | 466 |
| 40.9 | | Java Naming and Directory Interface | 467 |
| 40.10 | | Java and XML | 468 |
| 40.11 | | Java 2D and Java 3D | 468 |
| 40.12 | | Object-Oriented Databases | 469 |

Appendices

| | | |
|---|-----------------------|-----|
| A | The Java API Packages | 471 |
| B | Java Keywords | 472 |

| | |
|-------------------|------------|
| References | 473 |
|-------------------|------------|

| | |
|--------------|------------|
| Index | 477 |
|--------------|------------|



<http://www.springer.com/978-1-85233-569-4>

Java and Object Orientation: An Introduction

Hunt, J.

2002, XXVIII, 484 p. 76 illus., Softcover

ISBN: 978-1-85233-569-4