List of Figures

Figure 1-1: Isomorphic Mapping Between Problem Domain and Design Domain 8
Figure 1-2: Results of Running Example 1.2 16
Figure 1-3: javadoc Tool Being Used to Generate TestClass API Documentation 20
Figure 1-4: Example HTML Documentation Page Created With javadoc 20
Figure 2-1: Java Project Creation Process 28
Figure 2-2: Sun's Java Download Page 32
Figure 2-3: Custom Setup Dialog 32
Figure 2-4: Window Showing Subfolders and Files in the j2sdk1.4.2_01 Folder 33
Figure 2-5: Contents Listing of C:\j2sdk1.4.2_01\bin 33
Figure 2-6: Setting the PATH Environment Variable in Microsoft Windows 2000/XP 34
Figure 2-7: Testing PATH Environment Variable by Typing javac 35
Figure 2-8: Setting Some Important Folder Options 36
Figure 2-9: SampleClass Project Source-Code Directory Structure 37
Figure 2-10: Final SampleClass Subdirectory Structure 37
Figure 2-11: Compiling the Java Source Files 38
Figure 2-12: SampleClass Sub-directory Structure After Compilation 38
Figure 2-13: The .class Files are Located in Their Proper Package Structure 38
Figure 2-14: Results of Running ApplicationClass Program 39
Figure 2-15: JBuilder with No Open Projects 40
Figure 2-16: New Project Menu 40
Figure 2-17: Project Wizard Step 1 of 3 41
Figure 2-18: Project Wizard Step 2 of 3 41
Figure 2-19: Project Wizard Step 3 of 3 42
Figure 2-20: Creating ApplicationClass.java Source File 42
Figure 2-21: TestProject After Creating SampleClass.java and ApplicationClass.java 43
Figure 2-22: Project Properties Dialog 44
Figure 2-23: Runtime Configuration Properties Dialog 44
Figure 2-24: JBuilder Project Menu Showing the Make Project Item 45
Figure 2-25: Running TestProject 46
Figure 2-26: Downloading a Linux Java Self-Extracting Bin File 47
Figure 2-27: Executing the Self-Extracting Bin File 47
Figure 2-28: New Subdirectory Resulting from SDK Extraction 47
Figure 2-29: Contents of the j2sdk1.4.2_02 Directory 48
Figure 2-30: Checking Environment Variables with the env Command 48
Figure 2-31: Using the tree Command to Show Directory Structure 50
Figure 2-32: Running ApplicationClass in the Linux Environment 50
Figure 2-33: Xcode New Project Assistant 51
Figure 2-34: New Java Tool Window 52
Figure 2-35: Xcode ApplicationClass Project Window 52
Figure 2-36: Editing the Manifest File to Reflect the Correct Package Location of the Main Application Class 53
Figure 2-37: Running ApplicationClass Project from Xcode IDE 53
Figure 2-38: Executing the MyApp.jar File Using the java Command 54
Figure 3-1: Tight Spiral Development Cycle Deployment 61
Figure 3-2: Robot Rat Viewed As Attributes 65
Figure 3-3: Robot Rat Floor Sketch 65
Figure 3-4: Complete Robot Rat Attributes 66
Figure 3-5: RobotRat UML Class Diagram 68
Figure 3-6: Compiling & Testing RobotRat Class - First Iteration 70
Figure 3-7: Compiling & Testing RobotRat.class - Second Iteration 72
Figure 3-8: Testing Menu Commands 75

© 2006 Rick Miller — All Rights Reserved

Figure 3-9: pen_position State Transition Diagram 76
Figure 3-10: State Transition Diagram for the direction Variable 77
Figure 3-11: printFloor() Method Test 79
Figure 3-12: Testing the getSpaces() and move() Methods 82
Figure 3-13: Two Floor Patterns Printed To The Console 82
Figure 3-14: Partial RobotRat Javadoc Documentation 83
Figure 4-1: Typical Power Mac G4 System 92
Figure 4-2: System Unit 93
Figure 4-3: Main Logic Board Block Diagram 93
Figure 4-4: PowerPC G4 Processor 94
Figure 4-5: Motorola PowerPC 7400 Block Diagram 94
Figure 4-6: Memory Hierarchy 96
Figure 4-7: Simplified Memory Subsystem Diagram 96
Figure 4-8: Simplified Main Memory Diagram 97
Figure 4-9: Processing Cycle 99
Figure 4-10: Dumb Sort Results 1 101
Figure 4-11: Dumb Sort Results 2 101
Figure 4-12: Dumb Sort Results 3 101
Figure 4-13: Algorithmic Growth Rates 101
Figure 4-14: Java HotSpot™ Virtual Machine Targets Specific Hardware Platforms 102
Figure 4-15: Java HotSpot™ Virtual Machine Architecture 103
Figure 5-1: Java Platform Version 1.4.2 Package Architectural Overview 110
Figure 5-2: Java 2 Platform API Version 1.4.2 Specification Page 111
Figure 5-3: Partial Listing For java.lang Package 111
Figure 5-4: Detailed Information For java.lang.String class 112
Figure 5-5: String Inheritance Hierarchy UML Diagram 113
Figure 5-6: JButton Inheritance Hierarchy Information 113
Figure 5-7: JButton Inheritance Hierarchy 114
Figure 5-8: JButton Inherited Methods Grouped by Base Class (partial listing) 115
Figure 5-9: setVisible() Function Description 115
Figure 6-1: Compiling and Executing SimpleApplication 125
Figure 6-2: Results of Running IdentifierTest Program 126
Figure 6-3: TestClassOne Mod 1 Output 131
Figure 6-4: TestClassOne Mod 3 Output 132
Figure 6-5: TestClassOne Mod 4 Output 132
Figure 6-6: TestClassOne Mod 5 Output 133
Figure 6-7: Compiler Error Message Resulting From Attempt To Change A Constant's Value 133
Figure 6-8: Results of Running TestClassOne with the Input 1 2 3 134
Figure 6-9: Running TestClassOne Again with Different Input Values 134
Figure 6-10: Results of Running Example 6.12 135
Figure 6-11: Creating an Object with the new Operator 136
Figure 6-12: Creating Another Object with the new Operator 136
Figure 6-13: Reusing the object_reference Variable 137
Figure 6-14: Results of Running Example 6.14 138
Figure 6-15: Results of running Example 6.15 139
Figure 6-16: Results of Running Example 6.16 142
Figure 6-17: Results of Running Example 6.17 142
Figure 6-18: Results of Running Example 6.18 143
Figure 6-19: Results of Running Example 6.19 143
Figure 6-20: Results of Running Example 6.20 144
Figure 6-21: Results of Running Example 6.21 145
Figure 6-22: Results of Running Example 6.22 145
Figure 6-23: Results of Running Example 6.23 146
Figure 6-24: Results of Running Example 6.24 146
Figure 6-25: Results of Running Example 6.25 147
Figure 6-26: Results of Running Example 6.26 148
Figure 6-27: Bitwise Operator Truth Tables 148
Figure 6-28: Results of Running Example 6.27 148
Figure 6-29: Results of Running Example 6.28 149

© 2006 Rick Miller — All Rights Reserved

Figure 7-1: if Statement Execution Diagram 156
Figure 7-2: Results of Running Example 7.1 157
Figure 7-3: Results of Running Example 7.2 157
Figure 7-4: Results of Running Example 7.3 158
Figure 7-5: if/else Statement Execution Diagram 158
Figure 7-6: Results of Running Example 7.4 159
Figure 7-7: Results of Running Example 7.5 159
Figure 7-8: switch Statement Execution Diagram 160
Figure 7-9: Results of Running Example 7.6 160
Figure 7-10: Results of Running Example 7.7 161
Figure 7-11: Results of Running Example 7.8 162
Figure 7-12: while Statement Execution Diagram 163
Figure 7-13: Results of Running Example 7.9 163
Figure 7-14: do/while Statement Execution Diagram 164
Figure 7-15: Results of Running Example 7-10 164
Figure 7-16: for Statement Execution Diagram 165
Figure 7-17: Results of Running Example 7.11 166
Figure 7-18: Results of Running Example 7.12 166
Figure 7-19: Results of Running CheckBookBalancer 168
Figure 7-20: Results of Running Example 7.14 169
Figure 7-21: Results of Running Example 7.15 169
Figure 7-22: Results of Running Example 7.16 170
Figure 7-23: Results of Running Example 7.17 with Different Loop Limits 171
Figure 8-1: Array Elements are Contiguous and Homogeneous 180
Figure 8-2: Specifying Array Component Type 181
Figure 8-3: Array-Type Inheritance Hierarchy 182
Figure 8-4: Results of Running Example 8.1 184
Figure 8-5: Memory Representation of Primitive Type Array int_array Showing Default Initialization 184
Figure 8-6: Results of Running Example 8.2 185
Figure 8-7: Element Values of int_array After Initialization Performed by Second for Loop 185
Figure 8-8: Results of Running Example 8.3 186
Figure 8-9: Results of Running Example 8.4 186
Figure 8-10: Results of Running Example 8.5 188
Figure 8-11: State of Affairs After Line 3 of Example 8.5 Executes 188
Figure 8-12: State of Affairs After Line 5 of Example 8.5 Executes. 189
Figure 8-13: State of Affairs After Line 10 of Example 8.5 Executes 189
Figure 8-14: Final State of Affairs: All object_array Elements Point to an Object object 190
Figure 8-15: Results of Running Example 8.6 190
Figure 8-16: Results of Running Example 8.7 191
Figure 8-17: Results of Running Example 8.8 193
Figure 8-18: Array Declaration Syntax for a Two-Dimensional Array 194
Figure 8-19: A Two Dimensional Array with Dimensions 10 by 10 195
Figure 8-20: Results of Running Example 8.9 195
Figure 8-21: Memory Representation of int_2d_array with 2 Rows and 10 Columns 196
Figure 8-22: Results of Running Example 8.10 197
Figure 8-23: Results of Running Example 8.11 198
Figure 8-24: Results of Running Example 8.12 200
Figure 8-25: Results of Running Example 8.13 201
Figure 8-26: Results of Running Example 8.14 201
Figure 9-1: People Management Program Project Specification 211
Figure 9-2: Class Diagram for People Manager Classes 213
Figure 9-3: Static and Non-Static Fields 215
Figure 9-4: Horizontal Access Controlled via Access Modifiers public and private 216
Figure 9-5: Method Definition Structure 218
Figure 9-6: Results of Running Example 9.5 224
Figure 9-7: Results of Running Example 9.7 225
Figure 9-8: Results of Running Example 9.9 226
Figure 9-9: Results of Running Example 9.11 227
Figure 9-10: Results of Running Example 9.16 230

© 2006 Rick Miller — All Rights Reserved

Figure 9-11: Results of Running Example 9.18 232
Figure 9-12: Primitive and Reference Argument Values are Copied to Method Parameters 233
Figure 9-13: Results of Running Example 9.19 235
Figure 9-14: Linked List with Three Nodes 239
Figure 10-1: UML Diagram Showing Simple Aggregation 247
Figure 10-2: Part Class Shared Between Simple Aggregate Classes 247
Figure 10-3: UML Diagram Showing Composite Aggregation 247
Figure 10-4: Simple Aggregation Example 248
Figure 10-5: Results of Running Example 10.3 249
Figure 10-6: Composite Aggregation Example 249
Figure 10-7: Results of Running Example 10.6 250
Figure 10-8: Sequence Diagram – Simple Aggregation 250
Figure 10-9: Sequence Diagram – Composite Aggregation 251
Figure 10-10: Aircraft Engine Project Specification 252
Figure 10-11: Engine Simulation Class Diagram 253
Figure 10-12: Engine Class 253
Figure 10-13: Aircraft Engine Create Engine Object Sequence 255
Figure 10-14: Result of Running Example 10.8 256
Figure 10-15: Simple Aggregation Class Diagram 261
Figure 10-16: Composite Aggregation Class Diagram 261
Figure 11-1: Inheritance Hierarchy Illustrating Generalized & Specialized Behavior 266
Figure 11-2: UML Class Diagram Showing DerivedClass Inheriting From BaseClass 268
Figure 11-3: UML Diagram of BaseClass & DerivedClass Showing Fields and Methods 269
Figure 11-4: Results of Running Example 11.3 271
Figure 11-5: UML Diagram Showing Student Class Inheritance Hierarchy 272
Figure 11-6: Results of Running Example 11.6 274
Figure 11-7: Results of Running Example 11.7 274
Figure 11-8: UML Class Diagram For BaseClass & DerivedClass 275
Figure 11-9: Results of Running Example 11.3 with Modified Version of DerivedClass 276
Figure 11-10: Expressing an Abstract Class in the UML 277
Figure 11-11: UML Class Diagram Showing the AbstractClass & DerivedClass Inheritance Hierarchy 277
Figure 11-12: Results of Running Example 11.11 279
Figure 11-13: Two Types of UML Interface Diagrams 280
Figure 11-14: UML Diagram Showing the Simple Form of Realization 281
Figure 11-15: UML Diagram Showing the Expanded Form of Realization 281
Figure 11-16: UML Diagram Showing the MessagePrinterClass Realizing the MessagePrinter Interface 281
Figure 11-17: Results of Running Example 11.14 282
Figure 11-18: Horizontal And Vertical Access In Multi-Package Environment 283
Figure 11-19: Employee Class Inheritance Hierarchy 286
Figure 11-20: Results of Running Example 11.24 288
Figure 11-21: UML Class Diagram For Aircraft Engine Simulator 289
Figure 11-22: Results of Running Example 11.38 297
Figure 12-1: Standard Algebraic Coordinate System 307
Figure 12-2: Standard Computer-Screen Coordinate System 307
Figure 12-3: Components and Bounds 308
Figure 12-4: Top-Level Container Hierarchy 309
Figure 12-5: Screen Shot of an Empty JWindow 309
Figure 12-6: Structure of a JWindow 309
Figure 12-7: ScreenShot of an Empty JFrame 310
Figure 12-8: Structure of an Empty JFrame 310
Figure 12-9: JFrame with Menubar 310
Figure 12-10: Structure of JFrame with Menubar 310
Figure 12-11: A JDialog with a Label and Three Buttons 310
Figure 12-12: TestFrame GUI 311
Figure 12-13: TestFrameWithContents GUI 317
Figure 12-14: TestFrameWithContents Console Output 317
Figure 12-15: TestFrameWithContents Resized Larger 318
Figure 12-16: TestFrameWithContents Resized Smaller 318
Figure 12-17: TestFrameWithFlowLayout GUI 319

 © 2006 Rick Miller — All Rights Reserved

Figure 12-18: TestFrameWithFlowLayout Console Output 319
Figure 12-19: TestFrameWithFlowLayout Resized Wider 319
Figure 12-20: TestFrameWithFlowLayout Resized Taller 319
Figure 12-21: Coordinates for a Sample GridLayout with 4 Rows and 2 Columns 320
Figure 12-22: TestFrameWithGridLayout GUI 321
Figure 12-23: TestFrameWithGridLayout Console Output 321
Figure 12-24: TestFrameWithGridLayout Resized Wider 321
Figure 12-25: TestFrameWithGridLayout Resized Taller 321
Figure 12-26: BorderLayout Positions 322
Figure 12-27: TestFrameWithBorderLayout GUI 322
Figure 12-28: TestFrameWithBorderLayout Console Output 323
Figure 12-29: TestFrameWithBorderLayout Resized Wider 323
Figure 12-30: TestFrameWithBorderLayout Resized Taller 323
Figure 12-31: GridBagLayoutExample GUI 324
Figure 12-32: GridBagLayoutExample Console Output 324
Figure 12-33: GridBagLayoutExample GUI Variation 1 325
Figure 12-34: GridBagLayoutExample GUI Variation 2 325
Figure 12-35: GridBagLayoutExample GUI Variation 3 325
Figure 12-36: GridBagLayoutExample GUI Variation 4 326
Figure 12-37: GridBagLayoutExample GUI Variation 5 326
Figure 12-38: CombinedLayoutsExample GUI 328
Figure 12-39: CombinedLayoutExample Console Output 329
Figure 12-40: JComponent Inheritance Hierarchy 329
Figure 12-41: MainFrame GUI 342
Figure 12-42: Visual Guide to the Components in MainFrame 343
Figure 12-43: MainFrame Layout 344
Figure 12-44: Exercise3: Default Size 345
Figure 12-45: Exercise3: Stretched Horizontally 346
Figure 12-46: Alternate Border Layout 346
Figure 13-1: EventObject Inheritance Hierarchy 350
Figure 13-2: ACME Product Services Confirmation 353
Figure 13-3: Event-Handling Division of Labor 353
Figure 13-4: Inheritance Hierarchy for the Examples Used in this Chapter 354
Figure 13-5: ActionEvent Inheritance Hierarchy 356
Figure 13-6: ActionListener Inheritance Hierarchy 357
Figure 13-7: MouseEvent Inheritance Hierarchy 359
Figure 13-8: MouseListener, MouseMotionListener, and MouseWheelListener Inheritance Hierarchy 361
Figure 13-9: KeyEvent Inheritance Hierarchy 364
Figure 13-10: KeyListener Inheritance Hierarchy 365
Figure 13-11: ChangeEvent Inheritance Hierarchy 371
Figure 13-12: ChangeListener Inheritance Hierarchy 372
Figure 13-13: ListSelectionEvent Inheritance Hierarchy 373
Figure 13-14: ListSelectionListener Inheritance Hierarchy 373
Figure 14-1: This Chapter's Completed Application 381
Figure 14-2: Graphics Drawing Operations and Property-Related Methods 383
Figure 14-3: A JFrame Containing a JList with a Custom ListCellRenderer 394
Figure 14-4: A JFrame Containing a JTable 394
Figure 14-5: A JFrame Containing a Highly Customized JTree 395
Figure 14-6: Sequence Diagram for a JList Using CheckboxListCell – First Version 399
Figure 14-7: Maneuvering Through the Swing API 400
Figure 14-8: Sequence Diagram for a JList Using CheckboxListCell – Second Version 408
Figure 14-9: TreeCellEditor and TableCellEditor Inheritance Hierarchy 409
Figure 14-10: DefaultListModel Inheritance Hierarchy 414
Figure 15-1: Throwable Class Hierarchy 428
Figure 15-2: NumberFormatException Class Inheritance Hierarchy 429
Figure 15-3: Results of Running Example 15.1 with Good and Bad Input Strings 430
Figure 15-4: Results of Running Example 15.2 431
Figure 15-5: Results of Running Example 15.5 434
Figure 15-6: Results of Running Example 15.6 435

 © 2006 Rick Miller — All Rights Reserved

Figure 15-7: Results of Running Example 15.7 436
Figure 15-8: Results of Running Example 15.9 438
Figure 15-9: Results of Running Example 15.11 439
Figure 16-1: Results of Running Example 16.1 445
Figure 16-2: Results of Running Example 16.2 446
Figure 16-3: Results of Running Example 16.11 458
Figure 16-4: Results of Running Example 16.13 459
Figure 16-5: Breaker.java Thread Interaction 460
Figure 16-6: Acquiring and Releasing Locks 461
Figure 16-7: Breaker(2) and Breaker(3) Both Succeed 463
Figure 16-8: Breaker(2) Fails Because Not All Threads Synchronized 464
Figure 16-9: Breaker(3) Fails Because Not All Threads Synchronized 464
Figure 16-10: Results of Running Example 16.15 465
Figure 16-11: Breaker(2) Fails Because Threads Synchronized on Different Locks 465
Figure 16-12: Results of Running Example 16.16 466
Figure 16-13: Consumer Thread Waits 472
Figure 16-14: Producer Thread Waits 473
Figure 16-15: Deadlocked Threads 474
Figure 16-16: Results of Running Example 16.25 476
Figure 16-17: Deadlock Due to Nested Synchronization 476
Figure 17-1: Results of Testing DynamicArray 483
Figure 17-2: Results of Running Example 17.3 485
Figure 17-3: Results of Running Example 17.4 485
Figure 17-4: Java 1.4.2 Collections Framework Core Interface Hierarchy 487
Figure 17-5: Array of Object References Before Insertion 488
Figure 17-6: New Reference to be Inserted at Array Element 3 (index 2) 488
Figure 17-7: Array After New Reference Insertion 489
Figure 17-8: Linked List Node Organization 489
Figure 17-9: Linked List Before New Element Insertion 489
Figure 17-10: New Reference Being Inserted Into Second Element Position 490
Figure 17-11: References of Previous, New, and Next List Elements Must Be Manipulated 490
Figure 17-12: Linked List Insertion Complete 490
Figure 17-13: A Hash Function Transforms a Key Value into an Array Index 491
Figure 17-14: Hash Table Collisions are Resolved by Linking Nodes Together 491
Figure 17-15: Red-Black Tree Node Data Elements 492
Figure 17-16: Red-Black Tree After Inserting Integer Values 9, 3, 5, 6, 7, 8, 4, 1 492
Figure 17-17: Results of Running Example 17.5 494
Figure 17-18: Results of Running Example 17.8 496
Figure 17-19: Results of Running Example 17.9 497
Figure 17-20: Results of Running Example 17.12 499
Figure 17-21: Java 5 Collections Framework Core Interface Hierarchy 500
Figure 17-22: Results of Running Example 17.13 501
Figure 17-23: Results of Running Example 17.14 502
Figure 18-1: Partial java.io Package Hierarchy 509
Figure 18-2: Results of Running Example 18.1 513
Figure 18-3: Results of Running Example 18.2 514
Figure 18-4: Results of Running Example 18.3 516
Figure 18-5: Contents of test.txt After Executing Example 18.3 Six Times 516
Figure 18-6: Results of Running Example 18.4 517
Figure 18-7: Contents of test.txt File After Executing Example 18.4 517
Figure 18-8: Results of Running Example 18.5 518
Figure 18-9: Contents of test.txt After Running Example 18.5 518
Figure 18-10: Results of Running Example 18.7 519
Figure 18-11: Contents of People.dat File Viewed with Text Editor 520
Figure 18-12: Contents of Output.txt File After Example 18.8 Executes 520
Figure 18-13: Results of Running Example 18.9 521
Figure 18-14: Results of Running Example 18.10. 522
Figure 18-15: Results of Running Example 18.11 523
Figure 18-16: Warning Produced When Compiling Example 18.12 524

 © 2006 Rick Miller — All Rights Reserved

Figure 18-17: Results of Running Example 18.12 . . . 524
Figure 18-18: Contents of test.txt File After Example 18.13 Executes . . . 525
Figure 18-19: Contents of test.txt After Example 18.14 Executes . . . 526
Figure 18-20: Contents of test.txt File After Example 18.15 Executes . . . 526
Figure 18-21: Contents of test.txt File After Example 18.16 Executes . . . 527
Figure 18-22: Results of Running Example 18.17 . . . 528
Figure 18-23: Results of Running Example 18.18 . . . 528
Figure 18-24: Results of Running Example 18.19 . . . 529
Figure 18-25: Initial Execution of PropertiesTesterApp (Example 18.21) . . . 530
Figure 18-26: Contents of app_prop.xml After Example 18.21 Executes . . . 531
Figure 18-27: Legacy Data File Adapter Project Specification . . . 532
Figure 18-28: Contents of books.dat Example Legacy Datafile Viewed with Text Editor . . . 533
Figure 18-29: Header and Record Length Analysis . . . 533
Figure 18-30: Results of Running Example 18.29. . . . 544
Figure 19-1: A Simple Computer Network . . . 554
Figure 19-2: Local Area Network Connected to the Internet . . . 555
Figure 19-3: The Internet – A Network of Networks Communicating via Internet Protocols . . . 556
Figure 19-4: Client and Server Hardware and Applications . . . 557
Figure 19-5: Client and Server Applications Physically Deployed to Same Computer . . . 558
Figure 19-6: Client and Server Applications Require Separate Java Virtual Machines . . . 559
Figure 19-7: Starting Multiple Terminal Windows Using start Command . . . 559
Figure 19-8: Multiple JVMs Launched As Separate Processes In Mac OSX . . . 560
Figure 19-9: Killing Unix Processes with the kill Command . . . 560
Figure 19-10: Running Multiple Client JVMs On Same Hardware . . . 561
Figure 19-11: Client and Server Applications Deployed On Different Computers . . . 561
Figure 19-12: Physically Distributed Client and Server Applications Need A JVM . . . 562
Figure 19-13: A Multi-tiered Application . . . 562
Figure 19-14: Physically Deploying Logical Application Tiers on Same Computer . . . 563
Figure 19-15: Logical Application Tiers Physically Deployed to Different Computers . . . 563
Figure 19-16: TCP/IP Protocol Stack . . . 564
Figure 19-17: Internet Protocol Stack Operations . . . 566
Figure 19-18: Java Server Application Utilizing a ServerSocket Object . . . 567
Figure 19-19: Incoming Client Connection . . . 568
Figure 19-20: Connection Between Client & Server Established . . . 568
Figure 19-21: Retrieve IOStream Objects from Server and Client Socket Objects . . . 568
Figure 19-22: Results of Running Example 19.1 . . . 569
Figure 19-23: The Remote Method Invocation (RMI) Concept . . . 570
Figure 19-24: Class Diagram for RemoteSystemMonitorInterface & RemoteSystemMonitorImplementation . . . 572
Figure 19-25: SystemMonitorServer Running on Host Machine . . . 574
Figure 19-26: Results of Running the SystemMonitorClient Application Connecting to the Locally Served Server Application . . . 574
Figure 19-27: Results of the SystemMonitorClient Application after Running on a remote PC. . . . 574
Figure 19-28: SystemMonitorClient Invoking the Remote Method on a PC Running the SystemMonitorServer Application . . . 575
Figure 20-1: Client and Server Applications . . . 582
Figure 20-2: Incoming Client Connection . . . 583
Figure 20-3: The Connection is Established – There are Sockets at Both Ends of the Connection . . . 583
Figure 20-4: The Socket Objects are Used to Retrieve the IOStream Objects . . . 583
Figure 20-5: SimpleServer Running & Waiting for Incoming Client Connections . . . 586
Figure 20-6: SimpleClient Console Output and GUI . . . 587
Figure 20-7: SimpleServer Console After Detecting Incoming Client Connection . . . 587
Figure 20-8: Several Messages Exchanged with the Server From SimpleClient . . . 587
Figure 20-9: rat.gif . . . 589
Figure 20-10: First Draft Class Diagram for the NetRatServer Application . . . 591
Figure 20-11: NetRatServer Application Upon Start-up . . . 596
Figure 20-12: NetRatServer Application After Approximately 10 move Button Clicks . . . 596
Figure 20-13: RMI-Enabled NetRatServer Application at Start-up . . . 599
Figure 20-14: RMI-Enabled NetRatServer Application After Approximately 10 move Button Clicks . . . 599
Figure 20-15: RMI_NetRatClient Application . . . 601
Figure 20-16: The Floor After Testing RMI_NetRatClient. . . . 601
Figure 20-17: Updated Robot Rat Server Application Class Diagram . . . 603

 © 2006 Rick Miller — All Rights Reserved

Figure 20-18: The Floor After Approximately 15 Clicks of the Server-Side move Button 606
Figure 20-19: Server Floor After RMI-Client-Controlled Robot Rat Moves South Several Clicks 607
Figure 20-20: Server Floor After Second RMI-Client-Controlled Robot Rat Appears 607
Figure 20-21: Final NetRatServer Application Design Class Diagram 614
Figure 20-22: Console Output on NetRatServer Application Startup 619
Figure 20-23: Empty Floor Displayed as a Result of Explicitly Loading the RobotRat Class 619
Figure 21-1: BasicApplet Inheritance Hierarchy 627
Figure 21-2: BasicApplet Running In Web Browser 628
Figure 21-3: Console Log Showing BasicApplet Life Cycle Messages 629
Figure 21-4: Console Log Showing BasicApplet Life Cycle Messages After Browser Shuts Down 629
Figure 21-5: Applet Life Cycle Stages 629
Figure 21-6: The <applet> Tag and its Attributes 630
Figure 21-7: An Applet Can Only Connect To The Server From Which It Was Served 631
Figure 21-8: AppletServer Applet Running In A Browser And Being Accessed By The SimpleClient Application 633
Figure 21-9: Results of Attempting to Connect to AppletServer from a Computer Other Than its Server 634
Figure 21-10: Results of Running ParameterApplet 636
Figure 21-11: Poetry Applet In Action 639
Figure 21-12: Employee Training Management System Architecture Diagram 642
Figure 21-13: Server-Side Component Class Diagram 642
Figure 21-14: Client-Side Component Class Diagram 642
Figure 21-15: MySQL Monitor Program on Startup 645
Figure 21-16: Results of Entering "show databases;" at the Monitor Prompt 646
Figure 21-17: Results of Changing to the mysql Database with "use mysql;" and Entering "show tables;" 646
Figure 21-18: Structure of the user Table Located in the mysql Database 647
Figure 21-19: Structure of the db Table 648
Figure 21-20: Contents of the db Table 648
Figure 21-21: Structure of the tables_priv Table 649
Figure 21-22: Structure of the columns_priv Table 649
Figure 21-23: Structure of the host Table 649
Figure 21-24: Entity Diagram for Employee and Employee_Training Tables 650
Figure 21-25: Newly Created chapter_21 Database Tables 651
Figure 21-26: Results of Executing the Select Statement Against the employees Table 652
Figure 21-27: Results of Selecting Only the First_Name and Last_Name Columns from the employees Table 652
Figure 21-28: Results of the Update Statement – Note the Middle_Name is Changed to 'W' 653
Figure 21-29: Employees Table with Additional Data Added 653
Figure 21-30: Employee_Training Table Populated with Data 653
Figure 21-31: Results of Joining the employees Table with the employee_training Table 653
Figure 21-32: Results of Executing the Nested Select Statement Shown in Example 21.16 654
Figure 21-33: Results of Running Example 21.17 655
Figure 21-34: Results of Running Example 21.18 with employee Table Metadata Displayed 656
Figure 21-35: Employee Training Management System Architecture 656
Figure 21-36: Employee Training Management System Source Code Package Structure 657
Figure 21-37: Terminal Output Showing DBServerApp Startup Sequence 668
Figure 21-38: EmployeeTrainingApplet Appearance on First Access 669
Figure 21-39: Complete List of Employees 669
Figure 21-40: Training Records for Homer Simpson 669
Figure 21-41: Add New Employee Dialog 670
Figure 21-42: Add New Employee Dialog with Text Fields Filled In 670
Figure 21-43: New Employee Added to the Database 670
Figure 22-1: Meyer's Inheritance Taxonomy 683
Figure 22-2: Person-Employee Inheritance Diagram 685
Figure 22-3: Revised Person - Employee Example 689
Figure 22-4: Results of Running Example 22.9 693
Figure 23-1: Results of Running Example 23.4 705
Figure 23-2: Results of Running Example 23.6 707
Figure 23-3: Concept of a Shallow Copy 708
Figure 23-4: Concept of a Deep Copy 709
Figure 23-5: Results of Running Example 23.8 710
Figure 23-6: Results of Running Example 23.10 712

© 2006 Rick Miller — All Rights Reserved

Figure 23-7: Results of Running Example 23.11 714
Figure 23-8: Results of Running Example 23.13 715
Figure 24-1: Results of Running Example 24.2 726
Figure 24-2: Results of Running Example 24.4 728
Figure 24-3: Results of Running Example 24.6 730
Figure 24-4: Results of Running Example 24.8 732
Figure 24-5: Strong vs. Weak Types 733
Figure 24-6: Results of Running Example 24.12 734
Figure 24-7: Naval Fleet Class Inheritance Hierarchy 737
Figure 24-8: Results of Running Example 24.22 740
Figure 24-9: Traditional Top-Down Functional Dependencies 740
Figure 25-1: Results of Running Example 25.8 751
Figure 25-2: Model-View-Controller Pattern 752
Figure 25-3: Results of Running Example 25.11 and Clicking the "Next Message" Button Several Times 753
Figure 25-4: Employee Management Application UML Class Diagram 757
Figure 25-5: Interacting with the Employee Management Application 776

© 2006 Rick Miller — All Rights Reserved

© 2006 Rick Miller — All Rights Reserved