

```
1 // Class TestEmployeeDM.java
2 // Copyright 2001 ASR Strategic Resources
3 // AITP NCC 2001 Java Competition Event
4 // Java Contest Solution
5 // Design: Don Baldwin & Craig Slinkman PhD
6 // Code: Eugene Wasserman & Don Baldwin
7
8 package test;
9 import dm.*;
10 import hr.Boss;
11 import hr.Employee;
12 import hr.HourlyWorker;
13 import javax.swing.*;
14 import java.awt.BorderLayout;
15 import java.awt.event.*;
16
17 /**
18  * Test harness (runs as a main application) used to test Data Management classes.
19  */
20 public class TestEmployeeDM extends JFrame implements ActionListener {
21
22     JButton btRun = new JButton("Run Tests");
23     JTextArea jta = new JTextArea();
24
25     public TestEmployeeDM() {
26         super("TestEmployeeDM");
27         addWindowListener(new MainFrameWindowListener());
28         setBounds(100, 100, 600, 400);
29         getContentPane().setLayout(new BorderLayout());
30         getContentPane().add(new JScrollPane(jta), BorderLayout.CENTER);
31         btRun.addActionListener(this);
32         getContentPane().add(btRun, BorderLayout.SOUTH);
33         setVisible(true);
34     }
35
36     public void actionPerformed(ActionEvent ae) {
37         if(ae.getSource() == btRun)
38             runTest();
39     }
40
41     public static void main(String argv[]) {
42         TestEmployeeDM te = new TestEmployeeDM();
43     }
44
45     private void runTest() {
46
```

```
47         int bossId = 25, hourlyWorkerId = 27;
48
49         jta.append("\n" + "Running tests on dm package...");
50
51         jta.append("\n" + "\nTesting BossDM class");
52         try {
53             BossDM bossDM = new BossDM();
54
55             Boss boss = new Boss(bossId, "Yankee", "Doodle", 12.34);
56             jta.append("\n" + "Saving Boss: " + boss);
57             bossDM.save(boss);
58
59
60             boss = bossDM.read(bossId);
61             jta.append("\n" + "Read Boss(" + bossId + "): " + boss);
62
63             boss.setWeeklySalary(56.78);
64             jta.append("\n" + "Updating Boss: " + boss);
65             bossDM.update(boss);
66
67             boss = bossDM.read(bossId);
68             jta.append("\n" + "Read Boss(" + bossId + "): " + boss);
69
70             jta.append("\n" + "Deleting Boss(" + bossId + "): " + boss);
71             bossDM.delete(bossId);
72
73             jta.append("\n" + "Listing Bosses:");
74             Boss list[] = bossDM.list();
75             for(int i=0; i<list.length; i++)
76                 jta.append("\n" + list[i]);
77
78         } catch(Exception e) {
79             e.printStackTrace();
80         }
81
82         jta.append("\n" + "\nTesting HourlyWorkerDM class");
83         try {
84             HourlyWorkerDM hourlyWorkerDM = new HourlyWorkerDM();
85
86             HourlyWorker hourlyWorker = new HourlyWorker(hourlyWorkerId, "Yankee", "Doodle", 12.34,
87                 25);
88             jta.append("\n" + "Saving HourlyWorker: " + hourlyWorker);
89             hourlyWorkerDM.save(hourlyWorker);
90
91             hourlyWorker = hourlyWorkerDM.read(hourlyWorkerId);
```

```
92         jta.append("\n" + "Read HourlyWorker(" + hourlyWorkerId+ "): " + hourlyWorker);
93
94         hourlyWorker.setWage(56.78);
95         hourlyWorker.setHours(30);
96         jta.append("\n" + "Updating HourlyWorker: " + hourlyWorker);
97         hourlyWorkerDM.update(hourlyWorker);
98
99         hourlyWorker = hourlyWorkerDM.read(hourlyWorkerId);
100        jta.append("\n" + "Read HourlyWorker(" + hourlyWorkerId+ "): " + hourlyWorker);
101
102        jta.append("\n" + "Deleting HourlyWorker(" + hourlyWorkerId+ "): " + hourlyWorker);
103        hourlyWorkerDM.delete(hourlyWorkerId);
104
105        jta.append("\n" + "Listing HourlyWorkeres:");
106        HourlyWorker list[] = hourlyWorkerDM.list();
107        for(int i=0; i<list.length; i++)
108            jta.append("\n" +list[i]);
109
110    } catch(Exception e) {
111        e.printStackTrace();
112    }
113
114        jta.append("\n" + "Finished running tests on dm package.");
115    }
116
117
118    /*
119     * The inner listener class for the window's actions.
120     */
121    private class MainFrameWindowListener implements WindowListener {
122        public void windowOpened(WindowEvent e){}
123        public void windowClosed(WindowEvent e){}
124        public void windowIconified(WindowEvent e){}
125        public void windowDeiconified(WindowEvent e){}
126        public void windowActivated(WindowEvent e){}
127        public void windowDeactivated(WindowEvent e){}
128
129        /**
130         * Called if the user clicks on the close window button or
131         * closes the window in another way.
132         */
133        public void windowClosing(WindowEvent e) {
134            dispose();
135            System.exit(0);
136        }
137    }
```

TestEmployeeDM.java

12/10/2001

```
138 } // END Class TestEmployeeDM.java
```