

```

1 package poo_unidade_11_ex06_randomcharacters;
2
3 // Fig. 23.17: RunnableObject.java
4 // Runnable that writes a random character to a JLabel
5 import java.awt.Color;
6 import java.util.Random;
7 import java.util.concurrent.locks.Condition;
8 import java.util.concurrent.locks.Lock;
9 import javax.swing.JLabel;
10 import javax.swing.SwingUtilities;
11
12 public class RunnableObject implements Runnable {
13
14     private static Random generator = new Random(); // for random letters
15     private Lock lockObject; // application lock; passed in to constructor
16     private Condition suspend; // used to suspend and resume thread
17     private boolean suspended = false; // true if thread suspended
18     private JLabel output; // JLabel for output
19
20     public RunnableObject(Lock theLock, JLabel label) {
21         lockObject = theLock; // store the Lock for the application
22         suspend = lockObject.newCondition(); // create new Condition
23         output = label; // store JLabel for outputting character
24     } // end RunnableObject constructor
25
26     // place random characters in GUI
27     @Override
28     public void run() {
29         // get name of executing thread
30         final String threadName = Thread.currentThread().getName();
31
32         while (true){ // infinite loop; will be terminated from outside
33             try {
34                 // sleep for up to 1 second
35                 Thread.sleep(generator.nextInt(1000));
36
37                 lockObject.lock(); // obtain the lock
38                 try {
39                     while (suspended){ // loop until not suspended
40                         suspend.await(); // suspend thread execution
41                     } // end while
42                 } // end try
43                 finally {
44                     lockObject.unlock(); // unlock the lock
45                 } // end finally
46             } // end try
47             // if thread interrupted during wait/sleep
48             catch (InterruptedException exception) {
49                 exception.printStackTrace(); // print stack trace
50             } // end catch
51
52             // display character on corresponding JLabel
53             SwingUtilities.invokeLater(
54                 new Runnable() {
55                     // pick random character and display it
56
57                     @Override

```

```

60         public void run() {
61             // select random uppercase letter
62             char displayChar =
63                 (char) (generator.nextInt(26) + 65);
64
65             // output character in JLabel
66             output.setText(threadName + ": " + displayChar);
67         } // end method run
68     } // end inner class
69 } // end call to SwingUtilities.invokeLater
70 } // end while
71 } // end method run
72
73 // change the suspended/running state
74 public void toggle() {
75     suspended = !suspended; // toggle boolean controlling state
76
77     // change label color on suspend/resume
78     output.setBackground(suspended ? Color.RED : Color.GREEN);
79
80     lockObject.lock(); // obtain lock
81     try {
82         if (!suspended) // if thread resumed
83         {
84             suspend.signal(); // resume thread
85         } // end if
86     } // end try
87     finally {
88         lockObject.unlock(); // release lock
89     } // end finally
90 } // end method toggle
91 } // end class RunnableObject
92
93 /*****
94 * (C) Copyright 1992-2005 by Deitel & Associates, Inc. and
95 * Pearson Education, Inc. All Rights Reserved.
96 *
97 * DISCLAIMER: The authors and publisher of this book have used their
98 * best efforts in preparing the book. These efforts include the
99 * development, research, and testing of the theories and programs
100 * to determine their effectiveness. The authors and publisher make
101 * no warranty of any kind, expressed or implied, with regard to these
102 * programs or to the documentation contained in these books. The authors
103 * and publisher shall not be liable in any event for incidental or
104 * consequential damages in connection with, or arising out of, the
105 * furnishing, performance, or use of these programs.
106 *****/

```