```
 1  // joi/1/lights/TrafficLight.java
 2  //
 3  //
 4  // Copyright 2003 Bill Campbell and Ethan Bolker
 5
 6  import java.awt.*;
 7  import java.awt.event.*;
 8
 9  /**
10   * A TrafficLight has three lenses: red, yellow and green.
11   * It can be set to signal Go, Caution, Stop or Walk.
12   *
13   * @version 1
14   */
15
16  public class TrafficLight extends Panel
17  {
18      // Three Lenses and a Button
19
20      private Lens red          = new Lens( Color.red );
21      private Lens yellow       = new Lens( Color.yellow );
22      private Lens green        = new Lens( Color.green );
23      private Button nextButton = new Button("Next");
24
25      /**
26       * Construct a traffic light.
27       */
28
29      public TrafficLight()
30      {
31          this.setLayout(new BorderLayout());
32
33          // create a Panel for the Lenses
34          Panel lensPanel = new Panel();
35          lensPanel.setLayout( new GridLayout( 3, 1 ) );
36          lensPanel.add( red );
37          lensPanel.add( yellow );
38          lensPanel.add( green );
39          this.add( BorderLayout.NORTH, lensPanel );
40
41          // configure the "Next" button
42          Sequencer sequencer = new Sequencer( this );
43          NextButtonListener payAttention =
44              new NextButtonListener( sequencer );
45          nextButton.addActionListener( payAttention );
46          this.add( BorderLayout.CENTER, nextButton);
47      }
48
49      // Methods that change the light
50
51      /**
52       * Set the light to stop (red).
53       */
54
55      public void setStop()
56      {
```



```
57          red.turnOn();
58          yellow.turnOff();
59          green.turnOff();
60      }
61
62      /**
63       * Set the light to caution (yellow).
64       */
65
66      public void setCaution()
67      {
68          red.turnOff();
69          yellow.turnOn();
70          green.turnOff();
71      }
72
73      /**
74       * Set the light to go (green).
75       */
76
77      public void setGo()
78      {
79          red.turnOff();
80          yellow.turnOff();
81          green.turnOn();
82      }
83
84      /**
85       * Set the light to walk.
86       *
87       * (In Boston, red and yellow signal walk.)
88       */
89
90      public void setWalk()
91      {
92          red.turnOn();
93          yellow.turnOn();
94          green.turnOff();
95      }
96
97      /**
98       * The traffic light simulation starts at main.
99       *
100      * @param args ignored.
101      */
102
103     public static void main( String[] args )
104     {
105         Frame frame         = new Frame();
106         TrafficLight light  = new TrafficLight();
107         frame.add( light );
108         frame.addWindowListener( new ShutDownLight() );
109         frame.pack();
110         frame.show();
111     }
112
```

```
   113       // A ShutDownLight instance handles close events generated
   114       // by the underlying window system with its windowClosing
   115       // method.
   116       //
   117       // This is an inner class, declared inside the
   118       // TrafficLight class since it's used only here.
   119
   120      private static class ShutDownLight extends WindowAdapter
   121      {
   122           // Close the window by shutting down the light.
   123
   124          public void windowClosing (WindowEvent e)
   125          {
   126              System.exit(0);
   127          }
   128      }
   129  }
   130
   131
```

```
 1  // joi/1/lights/NextButtonListener.java
 2  //
 3  //
 4  // Copyright 2003 Bill Campbell and Ethan Bolker
 5
 6  import java.awt.event.*;
 7
 8  /**
 9   * A NextButtonListener sends a "next" message to its
10   * Sequencer each time a button to which it is listening
11   * is pressed.
12   *
13   * @version 1
14   */
15
16  public class NextButtonListener implements ActionListener
17  {
18      private Sequencer sequencer;
19
20      /**
21       * Construct a listener that "listens for" a user's
22       * pressing the "Next" button.
23       *
24       * @param sequencer the Sequencer for the TrafficLight.
25       */
26
27      public NextButtonListener( Sequencer sequencer )
28      {
29          this.sequencer = sequencer;
30      }
31
32      /**
33       * The action performed when a push of the button is detected:
34       * send a next message to the Sequencer to advance it to
35       * its next state.
36       *
37       * @param event the event detected at the button.
38       */
39
40      public void actionPerformed( ActionEvent event )
41      {
42          this.sequencer.next();
43      }
44  }
```

```
 1  // joi/1/lights/Sequencer.java
 2  //
 3  //
 4  // Copyright 2003 Bill Campbell and Ethan Bolker
 5
 6  /**
 7   * A Sequencer controls a TrafficLight. It maintains fields
 8   * for the light itself and the current state of the light.
 9   *
10   * Each time it receives a "next" message, it advances to the
11   * next state and sends the light an appropriate message.
12   *
13   * @version 1
14   */
15
16  public class Sequencer
17  {
18      // the TrafficLight this Sequencer controls
19      private TrafficLight light;
20
21      // represent the states by ints
22      private final static int GO       = 0;
23      private final static int CAUTION  = 1;
24      private final static int STOP     = 2;
25
26      private int currentState;
27
28      /**
29       * Construct a sequencer to control a TrafficLight.
30       *
31       * @param light the TrafficLight we wish to control.
32       */
33
34      public Sequencer( TrafficLight light )
35      {
36          this.light = light;
37          this.currentState = GO;
38          this.light.setGo();
39      }
40
41      /**
42       * How the light changes when a next Button is pressed
43       * depends on the current state. The sequence is
44       * GO -> CAUTION -> STOP -> GO.
45       */
46
47      public void next()
48      {
49          switch ( currentState ) {
50
51          case GO:
52              this.currentState = CAUTION;
53              this.light.setCaution();
54              break;
55
56          case CAUTION:
```



```
57              this.currentState = STOP;
58              this.light.setStop();
59              break;
60
61          case STOP:
62              this.currentState = GO;
63              this.light.setGo();
64              break;
65
66          default: // This will never happen
67              System.err.println("What color is the light?!");
68          }
69      }
70  }
```

```
 1  // joi/1/lights/Lens.java
 2  //
 3  //
 4  // Copyright 2003 Bill Campbell and Ethan Bolker
 5
 6  import java.awt.*;
 7
 8  /**
 9   * A Lens has a certain color and can either be turned on
10   * (the color) or turned off (black).
11   *
12   * @version 1
13   */
14
15  public class Lens extends Canvas
16  {
17      private Color onColor;               // color on
18      private Color offColor = Color.black; // color off
19      private Color currentColor;          // color the lens is now
20
21      private final static int SIZE = 100;  // how big is this Lens?
22      private final static int OFFSET = 20; // offset of Lens in Canvas
23
24      /**
25       * Construct a Lens to display a given color.
26       *
27       * The lens is black when it's turned off.
28       *
29       * @param color the color of the lens when it is turned on.
30       */
31
32      public Lens( Color color )
33      {
34          this.setBackground( Color.black );
35          this.onColor = color;
36          this.setSize( SIZE , SIZE );
37          this.turnOff();
38      }
39
40      /**
41       * How this Lens paints itself.
42       *
43       * @param g a Graphics object to manage brush and color information.
44       */
45
46      public void paint( Graphics g )
47      {
48          g.setColor( this.currentColor );
49          g.fillOval( OFFSET, OFFSET,
50                      SIZE - OFFSET*2, SIZE - OFFSET*2 );
51      }
52
53      /**
54       * Have this Lens display its color.
55       */
56
```



```
57      public void turnOn()
58      {
59          currentColor = onColor;
60          this.repaint();
61      }
62
63      /**
64       * Darken this lens.
65       */
66
67      public void turnOff()
68      {
69          currentColor = offColor;
70          this.repaint();
71      }
72  }
```