

## SampleApp.java

```
1 package msoe.se2030.sequence;
2
3 /**
4  * This app creates a UI and processes data
5  * @author hornick
6  */
7 public class SampleApp {
8     private UserInterface ui; // the UI for this program
9     /**
10    * @param args not used
11    */
12    public static void main(String[] args) {
13        new SampleApp();
14    }
15
16    public SampleApp() {
17        ui = new UserInterface(this, "My Sample App"); // create the UI
18    }
19
20    /**
21    * This method does some processing of data collected from the UI
22    * @param input the string entered
23    * @return nothing
24    */
25    public void notify( String input ) {
26        CalculationHelper helper = new CalculationHelper();
27        int n = helper.countChars(input);
28        // int n = helper.countDigits(input);
29
30        if( n == -1 ) { // an error occurred
31            ui.displayError("Input is empty or null!");
32        } else {
33            ui.displayNumberOfChars(n); // tell the UI to display the result
34            // ui.displayNumberOfDigits(n); // tell the UI to display the result
35        }
36    }
37 }
38
```

## UserInterface.java

```
1 /**
2  *
3  */
4 package msoe.se2030.sequence;
5
6 import java.awt.Container;
7 import java.awt.Dimension;
8 import java.awt.FlowLayout;
9 import java.awt.event.ActionEvent;
10 import java.awt.event.ActionListener;
11
12 import javax.swing.JButton;
13 import javax.swing.JFrame;
14 import javax.swing.JLabel;
15 import javax.swing.JTextField;
16
17
18 /**
19  * This class creates a UI and processes events
20  * @author hornick
21  */
22 public class UserInterface {
23     private JFrame ui; // the encapsulated JFrame that displays this UI
24     private SampleApp controller; // the app that creates this UI
25     private JButton button1; // a button on the UI
26     private JLabel output; // a field for displaying results
27     private JTextField input; // a input field for input
28
29     private class ButtonHandler implements ActionListener {
30
31         @Override
32         // respond to button presses by notifying the controller
33         public void actionPerformed(ActionEvent arg0) {
34
35             String s = input.getText(); // retrieve the input from the JTextField
36
37             if( arg0.getSource() == button1) { // the button was pressed...
38                 controller.notify( s ); // just tell the app/controller about it; this UI is too dumb to know what
39                 // to do about it...
40             }
41         }
42     }
43
44     /**
45     * Constructor for the UI
46     * @param c the parent, or controller class that creates this UI
47     * @param t the title of the window
48     */
49     public UserInterface(SampleApp c, String t) {
50         controller = c;
51         ui = new JFrame(t);
52     }
53 }
```

## UserInterface.java

```
51     ui.setSize(300, 500);
52     Container pane = ui.getContentPane();
53     pane.setLayout(new FlowLayout());
54
55     JLabel instructions = new JLabel("Enter some characters into the input field and press the button.");
56     pane.add(instructions);
57
58     input = new JTextField(""); // initial value in input field
59     input.setPreferredSize(new Dimension(200, 25)); // need this to set a fixed width; otherwise, the input
field will be too small.
60     pane.add(input);
61
62     output = new JLabel(""); // initial output in output field
63     output.setPreferredSize(new Dimension(200, 25));
64     pane.add(output);
65
66     ButtonHandler handler = new ButtonHandler();
67
68     button1 = new JButton("Compute!");
69     pane.add(button1);
70
71     button1.addActionListener(handler); // listen for button presses
72
73
74     // other code not shown
75     ui.pack();
76     ui.setVisible(true);
77 }
78
79 /**
80  * This method displays the number of characters entered in the input field
81  * @param v the value to display
82  */
83 public void displayNumberOfChars( int v ) {
84     output.setText("#chars = " + v);
85 }
86
87 /**
88  * This method displays the number of digits entered in the input field
89  * @param v the value to display
90  */
91 public void displayNumberOfDigits( int v ) {
92     output.setText("#digits = " + v);
93 }
94
95 /**
96  * This method displays an error message in the output
97  * @param msg the message to display
98  */
```

UserInterface.java

```
99  public void displayError( String msg ) {  
100     output.setText( msg );  
101 }  
102  
103 }
```

## Cal cul ati onHel per. j ava

```
1 package msoe.se2030.sequence;
2
3 /**
4  * This class implements methods that analyze a given String
5  * @author hornick
6  */
7 public class CalculationHelper {
8     public CalculationHelper() {
9         // nothing to initialize
10    }
11
12
13    /**
14     * This method counts chars in a String
15     * @param s a String
16     * @return number of chars, -1 if the String is empty or null
17     */
18    public int countChars( String s ) {
19        if( s == null || s.isEmpty() ) // input error check
20            return -1;
21
22        return s.length();
23    }
24
25
26    /**
27     * This method counts digits in a String
28     * @param s a String
29     * @return number of digits, -1 if the String is empty or null
30     */
31    public int countDigits( String s ) {
32        int nDigits = 0;
33
34        if( s == null || s.isEmpty() ) // input error check
35            return -1;
36
37        for( int index=0; index<s.length(); index++ ) {
38            Character ch = s.charAt(index);
39            if( Character.isDigit(ch) ) {
40                nDigits++;
41            }
42        }
43        return nDigits;
44    }
45 }
46 }
47
```