

```
1 /**
2  * Controller class.
3  *
4  * @author Bruce W. Weide
5  * @author Paolo Bucci
6  */
7 public final class AppendUndoController1 implements
AppendUndoController {
8
9     /**
10      * Model object.
11      */
12     private final AppendUndoModel model;
13
14     /**
15      * View object.
16      */
17     private final AppendUndoView view;
18
19     /**
20      * Updates view to display model.
21      *
22      * @param model
23      *         the model
24      * @param view
25      *         the view
26      */
27     private static void updateViewToMatchModel(AppendUndoModel
model,
28         AppendUndoView view) {
29         /**
30          * Get model info
31          */
32         String input = model.input();
33         /**
34          * Update view to reflect changes in model
35          */
36         view.updateInputDisplay(input);
37     }
38
39
40     /**
41      * Constructor; connects {@code this} to the model and view it
coordinates.
```

```
42     *
43     * @param model
44     *         model to connect to
45     * @param view
46     *         view to connect to
47     */
48     public AppendUndoController1(AppendUndoModel model,
AppendUndoView view) {
49         this.model = model;
50         this.view = view;
51         /*
52          * Update view to reflect initial value of model
53          */
54         updateViewToMatchModel(this.model, this.view);
55     }
56
57     /**
58      * Processes append event.
59      */
60     @Override
61     public void processAppendEvent(String input) {
62         /*
63          * Update model in response to this event
64          */
65         this.model.setInput(this.model.input() + input);
66
67         /*
68          * Update view to reflect changes in model
69          */
70         updateViewToMatchModel(this.model, this.view);
71     }
72
73     /**
74      * Processes Undo event.
75      *
76      * @param input
77      *         value of input text (provided by view)
78      */
79     @Override
80     public void processUndoEvent() {
81         /*
82          * Update model in response to this event
83          */
84     }
```

```
85      /*
86      * Update view to reflect changes in model
87      */
88      updateViewToMatchModel(this.model, this.view);
89  }
90
91 }
92
```