```java
/**
 * Controller class.
 *
 * @author Bruce W. Weide
 * @author Paolo Bucci
 */
public final class AppendUndoController1 implements AppendUndoController {

    /**
     * Model object.
     */
    private final AppendUndoModel model;

    /**
     * View object.
     */
    private final AppendUndoView view;

    /**
     * Updates view to display model.
     *
     * @param model
     *            the model
     * @param view
     *            the view
     */
    private static void updateViewToMatchModel(AppendUndoModel model,
            AppendUndoView view) {
        /*
         * Get model info
         */
        String input = model.input();
        String output = model.output().top();
        /*
         * Update view to reflect changes in model
         */
        if (model.output().length() > 1) {
            view.updateUndoAllowed(true);
        } else {
            view.updateUndoAllowed(false);
        }

        view.updateInputDisplay(input);
        view.updateOutputDisplay(output);
    }

    /**
     * Constructor; connects {@code this} to the model and view it coordinates.
     *
     * @param model
     *            model to connect to
```

```
 * @param view
 *            view to connect to
 */
public AppendUndoController1(AppendUndoModel model, AppendUndoView view) {
    this.model = model;
    this.view = view;
    /*
     * Update view to reflect initial value of model
     */
    updateViewToMatchModel(this.model, this.view);
}

/**
 * Processes reset event.
 */
@Override
public void processResetEvent() {
    /*
     * Update model in response to this event
     */
    this.model.setInput("");
    this.model.output().clear();
    this.model.output().push("");

    /*
     * Update view to reflect changes in model
     */
    updateViewToMatchModel(this.model, this.view);
}

/**
 * Processes append event.
 *
 * @param input
 *            value of input text (provided by view)
 */
@Override
public void processAppendEvent(String input) {
    /*
     * Update model in response to this event
     */
    this.model.setInput("");

    String top = this.model.output().top();
    this.model.output().push(top + input);

    /*
     * Update view to reflect changes in model
     */
    updateViewToMatchModel(this.model, this.view);
}
```

```
    /**
     * Processes undo event.
     *
     */
    @Override
    public void processUndoEvent() {
        /*
         * Update model in response to this event
         */
        this.model.output().pop();

        /*
         * Update view to reflect changes in model
         */
        updateViewToMatchModel(this.model, this.view);
    }

}
```