

```
1 import components.naturalnumber.NaturalNumber;
2
3
4 /**
5  * Controller class.
6  *
7  * @author Nicholas Cheong
8  */
9 public final class NNCalcController1 implements
   NNCalcController {
10
11     /**
12      * Model object.
13      */
14     private final NNCalcModel model;
15
16     /**
17      * View object.
18      */
19     private final NNCalcView view;
20
21     /**
22      * Useful constants.
23      */
24     private static final NaturalNumber TWO = new
   NaturalNumber2(2),
25         INT_LIMIT = new NaturalNumber2(Integer.MAX_VALUE);
26
27     /**
28      * Updates this.view to display this.model, and to allow
   only operations
29      * that are legal given this.model.
30      *
31      * @param model
32      *         the model
33      * @param view
34      *         the view
35      * @ensures [view has been updated to be consistent with
   model]
36      */
37     private static void updateViewToMatchModel(NNCalcModel
   model,
38         NNCalcView view) {
39
40         NaturalNumber top = model.top();
```

```
41     NaturalNumber bottom = model.bottom();
42     view.updateRootAllowed(model.bottom().compareTo(TWO) >=
0
43         && model.bottom().compareTo(INT_LIMIT) <= 0);
44     view.updateDivideAllowed(model.bottom().compareTo(INT_LIMIT) <=
0);
45     view.updateSubtractAllowed(model.top().compareTo(model.bottom()
) >= 0);
46     view.updatePowerAllowed(model.bottom().compareTo(TWO)
>= 0
47         && model.bottom().compareTo(INT_LIMIT) <= 0);
48
49     view.updateTopDisplay(top);
50     view.updateBottomDisplay(bottom);
51
52 }
53
54 /**
55  * Constructor.
56  *
57  * @param model
58  *         model to connect to
59  * @param view
60  *         view to connect to
61  */
62 public NNCalcController1(NNCalcModel model, NNCalcView
view) {
63     this.model = model;
64     this.view = view;
65     updateViewToMatchModel(model, view);
66 }
67
68 @Override
69 public void processClearEvent() {
70     /*
71      * Get alias to bottom from model
72      */
73     NaturalNumber bottom = this.model.bottom();
74     /*
75      * Update model in response to this event
76      */
77     bottom.clear();
```

```
78      /*
79      * Update view to reflect changes in model
80      */
81      updateViewToMatchModel(this.model, this.view);
82  }
83
84  @Override
85  public void processSwapEvent() {
86      /*
87      * Get aliases to top and bottom from model
88      */
89      NaturalNumber top = this.model.top();
90      NaturalNumber bottom = this.model.bottom();
91      /*
92      * Update model in response to this event
93      */
94      NaturalNumber temp = top.newInstance();
95      temp.transferFrom(top);
96      top.transferFrom(bottom);
97      bottom.transferFrom(temp);
98      /*
99      * Update view to reflect changes in model
100     */
101     updateViewToMatchModel(this.model, this.view);
102 }
103
104 @Override
105 public void processEnterEvent() {
106
107     // get aliases
108     NaturalNumber top = this.model.top();
109     NaturalNumber bottom = this.model.bottom();
110
111     // update model
112     top.copyFrom(bottom);
113
114     // update view
115     updateViewToMatchModel(this.model, this.view);
116
117 }
118
119 @Override
120 public void processAddEvent() {
121
```

```
122         // get aliases
123         NaturalNumber top = this.model.top();
124         NaturalNumber bottom = this.model.bottom();
125
126         // update model
127         top.add(bottom);
128         bottom.transferFrom(top);
129
130         // update view
131         updateViewToMatchModel(this.model, this.view);
132
133     }
134
135     @Override
136     public void processSubtractEvent() {
137
138         // get aliases
139         NaturalNumber top = this.model.top();
140         NaturalNumber bottom = this.model.bottom();
141
142         // update model
143         top.subtract(bottom);
144         bottom.transferFrom(top);
145
146         //update view
147         updateViewToMatchModel(this.model, this.view);
148
149     }
150
151     @Override
152     public void processMultiplyEvent() {
153
154         // get aliases
155         NaturalNumber top = this.model.top();
156         NaturalNumber bottom = this.model.bottom();
157
158         // update model
159         top.multiply(bottom);
160         bottom.transferFrom(top);
161
162         // update view
163         updateViewToMatchModel(this.model, this.view);
164
165     }
```

```
166
167     @Override
168     public void processDivideEvent() {
169
170         // get aliases
171         NaturalNumber top = this.model.top();
172         NaturalNumber bottom = this.model.bottom();
173
174         // update model
175         top.divide(bottom);
176         bottom.transferFrom(top);
177
178         // update view
179         updateViewToMatchModel(this.model, this.view);
180     }
181
182     @Override
183     public void processPowerEvent() {
184
185         // get aliases
186         NaturalNumber top = this.model.top();
187         NaturalNumber bottom = this.model.bottom();
188
189         int bottomInt = bottom.toInt();
190         // update model
191         top.power(bottomInt);
192         bottom.transferFrom(top);
193
194         updateViewToMatchModel(this.model, this.view);
195     }
196
197     @Override
198     public void processRootEvent() {
199
200         // get aliases
201         NaturalNumber top = this.model.top();
202         NaturalNumber bottom = this.model.bottom();
203         int bottomInt = bottom.toInt();
204
205         // updated model
206         top.root(bottomInt);
207         bottom.transferFrom(top);
208
209     }
```

```
210         // update view
211         updateViewToMatchModel(this.model, this.view);
212
213     }
214
215     @Override
216     public void processAddNewDigitEvent(int digit) {
217
218         // get aliases
219         NaturalNumber top = this.model.top();
220         NaturalNumber bottom = this.model.bottom();
221
222         // updated model
223         bottom.multiplyBy10(digit);
224
225         // update view
226         updateViewToMatchModel(this.model, this.view);
227
228     }
229
230 }
231
```