

View Javadoc

```

1  package view;
2
3  import java.awt.Graphics;
4  import java.awt.Graphics2D;
5  import java.awt.GraphicsConfiguration;
6  import java.awt.event.KeyEvent;
7  import java.awt.event.KeyListener;
8  import javax.swing.JFrame;
9  import javax.swing.JOptionPane;
10
11 import contract.ControllerOrder;
12 import contract.IController;
13 import contract.IModel;
14
15 /**
16  * The Class ViewFrame.
17  *
18  * @author Group 5
19  */
20 class ViewFrame extends JFrame implements KeyListener {
21
22     /** The panel. */
23     public static ViewPanel      pan;
24
25     /** The model. */
26     private IModel                model;
27
28     /** The controller. */
29     private IController           controller;
30
31     /** The Constant serialVersionUID. */
32     private static final long     serialVersionUID = -697358409737458175L;
33
34     /**
35      * Instantiates a new view frame.
36      *
37      * @param model
38      *         the model
39      */
40     public ViewFrame(final IModel model) {
41         this.buildViewFrame(model);
42     }
43
44     /**
45      * Instantiates a new view frame.
46      *
47      * @param model
48      *         the model
49      * @param gc
50      *         the gc
51      */
52     public ViewFrame(final IModel model, final GraphicsConfiguration gc) {
53         super(gc);
54         this.buildViewFrame(model);
55     }
56
57     /**
58      * Instantiates a new view frame.
59      *
60      * @param model
61      *         the model
62      * @param title
63      *         the title
64      */
65     public ViewFrame(final IModel model, final String title) {
66         super(title);

```

```
67         this.buildViewFrame(model);
68     }
69
70     /**
71     * Instantiates a new view frame.
72     *
73     * @param model
74     *         the model
75     * @param title
76     *         the title
77     * @param gc
78     *         the gc
79     */
80     public ViewFrame(final IModel model, final String title, final GraphicsConfiguration gc) {
81         super(title, gc);
82         this.buildViewFrame(model);
83     }
84
85     /**
86     * Gets the controller.
87     *
88     * @return the controller
89     */
90     private IController getController() {
91         return this.controller;
92     }
93
94     /**
95     * Sets the controller.
96     *
97     * @param controller
98     *         the new controller
99     */
100    protected void setController(final IController controller) {
101        this.controller = controller;
102    }
103
104    /**
105    * Gets the model.
106    *
107    * @return the model
108    */
109    protected IModel getModel() {
110        return this.model;
111    }
112
113    /**
114    * Sets the model.
115    *
116    * @param model
117    *         the new model
118    */
119    private void setModel(final IModel model) {
120        this.model = model;
121    }
122
123    /**
124    * Builds the view frame.
125    *
126    * @param model
127    *         the model
128    */
129    private void buildViewFrame(final IModel model) {
130        this.setModel(model);
131        this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
132        this.setResizable(false);
133        this.addKeyListener(this);
134        this.setContentPane(new ViewPanel(this));
135        this.setSize(635, 385);
```

```
136         this.setLocationRelativeTo(null);
137         this.setVisible(true);
138     }
139
140     /**
141      * Prints the message.
142      *
143      * @param message
144      *     the message
145      */
146     public void printMessage(final String message) {
147         JOptionPane.showMessageDialog(null, message);
148     }
149
150     /**
151      * (non-Javadoc)
152      *
153      * @see java.awt.event.KeyListener#keyTyped(java.awt.event.KeyEvent)
154      */
155     public void keyTyped(final KeyEvent e) {
156     }
157
158     /**
159      * (non-Javadoc)
160      *
161      * @see java.awt.event.KeyListener#keyPressed(java.awt.event.KeyEvent)
162      */
163     public void keyPressed(final KeyEvent e) {
164         this.getController().orderPerform(View.keyCodeToControllerOrder(e.getKeyCode()));
165     }
166
167     /**
168      * (non-Javadoc)
169      *
170      * @see java.awt.event.KeyListener#keyReleased(java.awt.event.KeyEvent)
171      */
172     public void keyReleased(final KeyEvent e) {
173     }
174
175 }
176
177 }
```

Copyright © 2019. All rights reserved.