```
 1 package ui.layouts.GridPane;
 2
 3 import javafx.application.Application;
 4 import javafx.event.ActionEvent;
 5 import javafx.event.EventHandler;
 6 import javafx.geometry.HPos;
 7 import javafx.geometry.Pos;
 8 import javafx.geometry.Rectangle2D;
 9 import javafx.scene.Scene;
10 import javafx.scene.control.Button;
11 import javafx.scene.control.Label;
12 import javafx.scene.control.TextField;
13 import javafx.scene.layout.BorderPane;
14 import javafx.scene.layout.GridPane;
15 import javafx.scene.layout.HBox;
16 import javafx.scene.layout.VBox;
17 import javafx.stage.Screen;
18 import javafx.stage.Stage;
19
20 /**
21  * Demo1. Entry point into demonstration application.
22  */
23 public class Demo1 extends Application {
24
25     private BorderPane layout;
26     private Scene scene;
27     private TextField txtFirstName, txtLastName;
28
29     @Override
30     public void start(Stage stage) {
31
32         //Create BorderPane layout manager.
33         layout = new BorderPane();  //This is the "root node".
34
35         //Give Root Node a CSS ID Attribute
36         layout.setId("appContainer");
37
38         //Set Scene Properties.
39         setSceneProperties();
40
41         //Build Demo App Layout
42         buildLeft();
```

```
43          buildTop();
44
45          //Set a few properties of our Application Window
46          stage.setScene(scene);
47          stage.setTitle("Grid Pane Demo");
48          stage.show();
49      }
50
51      /**
52       * main. Application Entry Point.
53       * @param args
54       */
55      public static void main(String[] args) {
56          launch();
57      }
58
59      /**
60       * buildLeft. This method builds the Left Region of BorderPane.
61       * This is VBox containing all buttons.
62       */
63      private void buildLeft() {
64
65          BorderPane leftLayout = new BorderPane();
66
67          // Create a faux border-right effect using a Label.
68          Label divider = new Label();
69          divider.setId("divider1");
70          divider.setPrefWidth(1);
71          divider.setMinHeight(Screen.getPrimary().getBounds().getHeight());
72          leftLayout.setRight(divider);
73
74          //Place all demonstration buttons in a Vercial Box.
75          VBox buttonBox = new VBox();
76
77          //Set Alignment of Buttons in VBox Container.
78          buttonBox.setAlignment(Pos.TOP_CENTER);
79
80          //Give VBox a CSS ID
81          buttonBox.setId("buttonMenuContainer");
82
83          //Create some vertical spacing b/n buttons
84          buttonBox.setSpacing(10);
```

```
85
86          //Add Demonstration Buttons
87          Button btnExample1 = new Button();
88
89          //Set Button Text
90          btnExample1.setText("Example 1");
91
92          //Set All Buttons to the same size.
93          btnExample1.setMaxWidth(Double.MAX_VALUE);
94
95          //Add Click Event.
96          btnExample1.setOnAction(new EventHandler<ActionEvent>() {
97
98              @Override
99              public void handle(ActionEvent event) {
100                 System.out.println("Example 1 Button Clicked.");
101                 layout.setCenter(example1());
102             }
103         });
104
105         //Create Button 2
106         Button btnExample2 = new Button();
107         btnExample2.setText("Useless Button");
108         btnExample2.setMaxWidth(Double.MAX_VALUE);
109         btnExample2.setOnAction(new EventHandler<ActionEvent>() {
110
111             @Override
112             public void handle(ActionEvent event) {
113                 System.out.println("Example 2 Button Clicked.");
114             }
115         });
116
117         //Create Button 3
118         Button btnExample3 = new Button();
119         btnExample3.setText("Useless Button");
120         btnExample3.setMaxWidth(Double.MAX_VALUE);
121         btnExample3.setOnAction(new EventHandler<ActionEvent>() {
122
123             @Override
124             public void handle(ActionEvent event) {
125                 System.out.println("Example 3 Button Clicked.");
126             }
```

```
127         });
128 
129         buttonBox.getChildren().addAll(btnExample1, btnExample2,
btnExample3);
130 
131         //Add VBox to leftLayout.
132         leftLayout.setCenter(buttonBox);
133 
134         //Place into Application.
135         layout.setLeft(leftLayout);
136 
137 
138     }
139 
140     /**
141      * buildTop. Create a Title Bar.
142      */
143     private void buildTop() {
144 
145         BorderPane topLayout = new BorderPane();
146 
147         //Add CSS Style ID.
148         topLayout.setId("topLayoutContainer");
149 
150         // Create a faux border-bottom effect using a Label.
151         Label divider = new Label();
152         divider.setId("divider2");
153         divider.setMaxHeight(1);
154         divider.setMinHeight(1);
155         divider.setMinWidth(Screen.getPrimary().getBounds().getWidth());
156         topLayout.setBottom(divider);
157 
158         //Create an HBox to hold title.
159         //We use the HBox to set text alignment to LEFT, MIDDLE
160         HBox titleBox = new HBox();
161         titleBox.setAlignment(Pos.TOP_LEFT);
162         titleBox.setId("titleBox");
163 
164         //Create title.
165         Label title = new Label();
166         title.setText("GridPane Demo");
167         title.setId("appTitle");
```

```
168
169         //Add Title label to titleBox
170         titleBox.getChildren().add(title);
171
172         //Add Title Box (with label) to topLayout
173         topLayout.setCenter(titleBox);
174
175         //Add topLayout (a BorderPane Manager) to App Layout.
176         layout.setTop(topLayout);
177
178     }
179
180     private void setSceneProperties()
181     {
182         //The percentage values are used as multipliers for screen
width/height.
183         double percentageWidth = 0.98;
184         double percentageHeight = 0.90;
185
186         //Calculate the width / height of screen.
187         Rectangle2D screenSize = Screen.getPrimary().getBounds();
188         percentageWidth *= screenSize.getWidth();
189         percentageHeight *= screenSize.getHeight();
190
191         //Create a scene object. Pass in the layout and set
192         //the dimensions to 98% of screen width & 90% screen height.
193         this.scene = new Scene(layout, percentageWidth, percentageHeight);
194
195         //Add CSS Style Sheet (located in same package as this class).
196         String css =
this.getClass().getResource("Demo1.css").toExternalForm();
197         scene.getStylesheets().add(css);
198
199     }
200
201     /**
202      * example1. This method just creates a simple GridPane with 2
203      * rows and 2 columns. This example demonstrates the use of
204      * showing gridLines.
205      * @return
206      */
207     private VBox example1() {
```

```
208
209         //Create a container to fill 100% space in Center Region of
210         //App BorderPane (layout).
211         VBox exContainer = new VBox();
212         exContainer.setId("exContainer");
213
214         //Create a new GridPane.
215         GridPane gridPane = new GridPane();
216
217         //Turn on GridLines so we can see what is going on.
218         //gridPane.setGridLinesVisible(true);
219
220         //Give the GridPane an ID for CSS Styles.
221         gridPane.setId("gridPane_Example1");
222
223         //Add some spacing between each control.
224         //Comment the next 2 lines out to see what happens when this is
225         //not explicitly set. It will remove the padding you specified.
226         gridPane.setHgap(5);
227         gridPane.setVgap(5);
228
229         //Add a description of what we are doing to GridPane.
230         //This description starts in Row 0, Col 0 and spans
231         //2 columns and one row.
232         Label label = new Label("Turn on the grid lines to see results.");
233         gridPane.add(label, 0,0,2,1);
234
235         //Add A Label. The label starts in Col 0, Row 1 and does not
236         //span any columns or rows.
237         gridPane.add(new Label("First Name"), 0, 1);
238
239         //Add a TextField. The textfield starts in Col 1, Row 1 and
240         //does not span any columns or rows.
241         txtFirstName = new TextField();
242         txtFirstName.setId("txtFirstName");
243         gridPane.add(txtFirstName, 1,1);
244
245         //Add Last Name label in Col 0, Row 2
246         gridPane.add(new Label("Last Name"), 0,2);
247
248         //Add Last Name Text Field in Col 1, Row 2.
249         txtLastName = new TextField();
```

```
250         txtLastName.setId("txtLastName");
251         gridPane.add(txtLastName, 1,2);
252 
253         //Add a Submit Button.
254         Button submitButton = new Button("Submit");
255         submitButton.setOnAction(new EventHandler<ActionEvent>() {
256 
257             @Override
258             public void handle(ActionEvent event) {
259                 System.out.printf("Submit Button Clicked. Hi there %s %s",
260                         txtFirstName.getText(), txtLastName.getText());
261             }
262         });
263         gridPane.add(submitButton, 1,3);
264 
265         //Align the Submit Button to Right.
266         gridPane.setHalignment(submitButton, HPos.RIGHT);
267 
268         //Add GridPane to container.
269         exContainer.getChildren().add(gridPane);
270 
271         //Return Container
272         return exContainer;
273     }
274 }
```