

```
/*
 * CS5405
 * Homework 05
 * Grant Broadwater (grbcp5)
 * October 16, 2018
 *
 * CircleDataPane.java
 */
```

```
package code;
```

```
import javafx.scene.layout.VBox;
import javafx.scene.control.Label;
import javafx.scene.control.TextField;
import javafx.scene.control.Button;
import javafx.scene.control.ToggleGroup;
import javafx.scene.control.RadioButton;
import javafx.geometry.Insets;
import javafx.event.EventHandler;
import javafx.event.ActionEvent;
import javafx.scene.paint.Color;
import javafx.beans.value.ChangeListener;
import javafx.beans.value.ObservableValue;
import javafx.scene.control.Toggle;
```

```
public class CircleDataPane extends VBox implements EventHandler<ActionEvent>,
    ChangeListener<Toggle> {
```

```
    public static final double DEFAULT_SPACING = 20;
```

```
    private CircleDataModel circleDataModel;
```

```
    private EventHandler<ActionEvent> parent;
```

```
    private Label lbl_circlesData;
```

```
    private TextField tf_circlesData;
```

```
    private Button btn_submitChanges;
```

```
    private ToggleGroup group;
```

```
    private Label lbl_circle1;
```

```
    private Label lbl_circle2;
```

```
    public CircleDataPane(EventHandler<ActionEvent> parent) {
        super(CircleDataPane.DEFAULT_SPACING);
```

```
        this.parent = parent;
```

```
        this.circleDataModel = new CircleDataModel();
```

```

this.setPrefWidth(300);
this.setPadding(new Insets(10, 30, 50, 20));

this.lbl_circlesData = new Label("Circles Data:");
this.tf_circlesData = new TextField();
this.tf_circlesData.setOnAction(this);
this.btn_submitChanges = new Button("Submit Changes");
this.btn_submitChanges.setOnAction(this);
this.getChildren().addAll(this.lbl_circlesData, this.tf_circlesData,
    this.btn_submitChanges);

this.group = new ToggleGroup();
for(int i = 0; i < this.circleDataModel.getNumTestCases(); i++) {
    RadioButton rb = new RadioButton("Test Case " + (i + 1));
    rb.setToggleGroup(group);
    rb.setUserData(i);
    this.getChildren().add(rb);
}
group.selectedToggleProperty().addListener(this);

this.lbl_circle1 = new Label("Circle 1:\n");
this.lbl_circle1.setTextFill(Color.RED);
this.lbl_circle2 = new Label("Circle 2:\n");
this.lbl_circle2.setTextFill(Color.BLUE);
this.getChildren().addAll(this.lbl_circle1, this.lbl_circle2);

this.updateUI();
}

@Override
public void handle(ActionEvent actionEvent) {
    circleDataModel.setCircleData(this.tf_circlesData.getText());
    this.updateUI();

    parent.handle(actionEvent);
}

@Override
public void changed(
    ObservableValue<? extends Toggle> ov,
    Toggle old_toggle,
    Toggle new_toggle
) {
    if (group.getSelectedToggle() != null) {
        int idx = (Integer)group.getSelectedToggle().getUserData();

        this.circleDataModel.setCircleData(this.circleDataModel.getTestCase(idx)
        );
        this.updateUI();
    }
}

```

```
        parent.handle(null);
    }
}

public void updateUI() {

    this.tf_circlesData.setText(this.circleDataModel.toString());

    this.lbl_circle1.setText("Circle 1:\n" +
        this.circleDataModel.getCircle1().toString());
    this.lbl_circle2.setText("Circle 2:\n" +
        this.circleDataModel.getCircle2().toString());

    this.lbl_circlesData.setText(this.circleDataModel.getCircleRelationship());
}

public CircleDataModel getModel() {
    return this.circleDataModel;
}
}
```