1 import javafx.beans.binding.Bindings;  
 2 import java.util.Random;  
 3 import java.util.Calendar;   
 4 import java.util.GregorianCalendar;  
 5 import javafx.scene.layout.Pane;  
 6 import javafx.scene.paint.Color;  
 7 import javafx.scene.shape.Circle;  
 8 import javafx.scene.shape.Arc;  
 9 import javafx.scene.shape.ArcType;  
 10 import javafx.scene.shape.Line;  
 11 import javafx.scene.text.Text;  
 12 import javafx.application.Application;  
 13 import javafx.geometry.\*;  
 14 import javafx.stage.Stage;  
 15 import javafx.scene.Scene;  
 16 import javafx.scene.control.Label;  
 17 import javafx.scene.layout.\*;  
 18 public class n01185608 extends Application  
 19 {  
 20 @Override  
 21 public void start(Stage primaryStage)  
 22 {  
 23 // Create a clock and a label  
 24 Random rand = new Random();  
 25 int hour = rand.nextInt(23) + 0;  
 26 int minute = rand.nextInt(60) + 0;  
 27 int second = rand.nextInt(60) + 0;  
 28 ClockPane clock = new ClockPane(hour,minute,second);  
 29 String timeString = clock.getHour() + ":" + clock.getMinute()   
 30 + ":" + clock.getSecond();  
 31 Label lblCurrentTime = new Label(timeString);  
 32 FanPane fan = new FanPane();  
 33 // Place clock and label in border pane  
 34 BorderPane pane = new BorderPane();  
 35 pane.setCenter(clock);  
 36 pane.setBottom(lblCurrentTime);  
 37 BorderPane.setAlignment(lblCurrentTime, Pos.TOP\_CENTER);  
 38 pane.setLeft(fan);  
 39 // Create a scene and place it in the stage  
 40 Scene scene = new Scene(pane, 250, 250);  
 41 primaryStage.setTitle("Assignment 6"); // Set the stage title  
 42 primaryStage.setScene(scene); // Place the scene in the stage  
 43 primaryStage.show(); // Display the stage  
 44 }  
 45 public static void main(String[] args) {  
 46 launch(args);  
 47 }  
 48 }  
 49 class ClockPane extends Pane {  
 50 private int hour;  
 51 private int minute;  
 52 private int second;  
 53   
 54 /\*\* Construct a default clock with the current time\*/  
 55 public ClockPane() {  
 56 setCurrentTime();  
 57 }  
 58   
 59 /\*\* Construct a clock with specified hour, minute, and second \*/  
 60 public ClockPane(int hour, int minute, int second) {  
 61 setHour(hour);  
 62 setMinute(minute);  
 63 setSecond(second);  
 64 }  
 65   
 66 /\*\* Return hour \*/  
 67 public int getHour() {  
 68 return hour;  
 69 }  
 70   
 71 /\*\* Set a new hour \*/  
 72 public void setHour(int hour) {  
 73 this.hour = hour;  
 74 paintClock();  
 75 }  
 76   
 77 /\*\* Return minute \*/  
 78 public int getMinute() {  
 79 return minute;  
 80 }  
 81   
 82 /\*\* Set a new minute \*/  
 83 public void setMinute(int minute) {  
 84 this.minute = minute;  
 85 paintClock();  
 86 }  
 87   
 88 /\*\* Return second \*/  
 89 public int getSecond() {  
 90 return second;  
 91 }  
 92   
 93 /\*\* Set a new second \*/  
 94 public void setSecond(int second) {  
 95 this.second = second;  
 96 paintClock();  
 97 }  
 98   
 99 /\* Set the current time for the clock \*/  
100 public void setCurrentTime() {  
101 // Construct a calendar for the current date and time  
102 Calendar calendar = new GregorianCalendar();  
103   
104 // Set current hour, minute and second  
105 this.hour = calendar.get(Calendar.HOUR\_OF\_DAY);  
106 this.minute = calendar.get(Calendar.MINUTE);  
107 this.second = calendar.get(Calendar.SECOND);  
108   
109 paintClock(); // Repaint the clock  
110 }  
111   
112 /\*\* Paint the clock \*/  
113 private void paintClock() {  
114 // Initialize clock parameters  
115 double clockRadius =   
116 Math.min(getWidth(), getHeight()) \* 0.8 \* 0.5;  
117 double centerX = getWidth() / 2;  
118 double centerY = getHeight() / 2;  
119 // Draw circle  
120 Circle circle = new Circle(centerX, centerY, clockRadius);  
121 circle.setFill(Color.WHITE);  
122 circle.setStroke(Color.BLACK);  
123 Text t1 = new Text(centerX - 5, centerY - clockRadius + 12, "12");  
124 Text t2 = new Text(centerX - clockRadius + 3, centerY + 5, "9");  
125 Text t3 = new Text(centerX + clockRadius - 10, centerY + 3, "3");  
126 Text t4 = new Text(centerX - 3, centerY + clockRadius - 3, "6");  
127 Text t5 = new Text(centerX - clockRadius/2-8, centerY + 4, "Shyam Rajendren");  
128 //t5.setX(t5.getX() - t5.getLayoutBounds().getWidth() / 2 );  
129 // Draw second hand  
130 double sLength = clockRadius \* 0.8;  
131 double secondX = centerX + sLength \*   
132 Math.sin(second \* (2 \* Math.PI / 60));  
133 double secondY = centerY - sLength \*   
134 Math.cos(second \* (2 \* Math.PI / 60));  
135 Line sLine = new Line(centerX, centerY, secondX, secondY);  
136 sLine.setStroke(Color.RED);  
137   
138 // Draw minute hand  
139 double mLength = clockRadius \* 0.65;  
140 double xMinute = centerX + mLength \*   
141 Math.sin(minute \* (2 \* Math.PI / 60));  
142 double minuteY = centerY - mLength \*   
143 Math.cos(minute \* (2 \* Math.PI / 60));  
144 Line mLine = new Line(centerX, centerY, xMinute, minuteY);  
145 mLine.setStroke(Color.BLUE);  
146   
147 // Draw hour hand  
148 double hLength = clockRadius \* 0.5;  
149 double hourX = centerX + hLength \*   
150 Math.sin((hour % 12 + minute / 60.0) \* (2 \* Math.PI / 12));  
151 double hourY = centerY - hLength \*  
152 Math.cos((hour % 12 + minute / 60.0) \* (2 \* Math.PI / 12));  
153 Line hLine = new Line(centerX, centerY, hourX, hourY);  
154 hLine.setStroke(Color.GREEN);  
155   
156 getChildren().clear();   
157 getChildren().addAll(circle, t1, t2, t3, t4, t5, sLine, mLine, hLine);  
158 }  
159   
160 @Override  
161 public void setWidth(double width) {  
162 super.setWidth(width);  
163 paintClock();  
164 }  
165   
166 @Override  
167 public void setHeight(double height) {  
168 super.setHeight(height);  
169 paintClock();  
170 }  
171 }  
172 class FanPane extends Pane  
173 {  
174 FanPane()  
175 {  
176 paintFan();  
177 }  
178 public void paintFan()  
179 {  
180 GridPane pane = new GridPane();  
181 for(int i=0;i<2;i++)  
182 {  
183 for(int j=0;j<2;j++)  
184 {  
185 double xCenter = getWidth() / 2;  
186 double yCenter = getHeight() / 2;  
187 double radius = (Math.min(getWidth(), getHeight()) \* 0.4);  
188 Circle circle = new Circle(xCenter, yCenter, radius);  
189 circle.setFill(Color.WHITE);  
190 circle.setStroke(Color.BLACK);  
191 double x = circle.getCenterX();  
192 double y = circle.getCenterY();  
193 Arc arc = new Arc(getWidth() / 2, getHeight() / 2, radius, radius, 0, -30);  
194 arc.setFill(Color.BLACK);  
195 arc.setType(ArcType.ROUND);  
196 pane.add(circle, i ,j);  
197 pane.add(arc, i ,j);  
198 GridPane.setMargin(circle, new Insets(5, 0, 0, 0));  
199 GridPane.setMargin(arc, new Insets(5, 0, 0, 0));  
200 }  
201 }  
202 //pane.setConstraints(new Insets(10,10,10,10));  
203 getChildren().clear();  
204 getChildren().addAll(pane);  
205   
206 }  
207 @Override  
208 public void setWidth(double width) {  
209 super.setWidth(width);  
210 paintFan();  
211 }  
212   
213 @Override  
214 public void setHeight(double height) {  
215 super.setHeight(height);  
216 paintFan();  
217 }  
218 }