For: Melissa Bakke

Assignment: Control A Clock Exercise 15.32

|  |
| --- |
| **Screenshot(s)** |
|  |

|  |
| --- |
| **Code** |
| *package control.a.clock.exercise.pkg15.pkg32;*  *import javafx.animation.KeyFrame;*  *import javafx.animation.Timeline;*  *import javafx.application.Application;*  *import javafx.geometry.Pos;*  *import javafx.stage.Stage;*  *import javafx.scene.Scene;*  *import javafx.scene.control.Button;*  *import javafx.scene.control.Label;*  *import javafx.scene.layout.BorderPane;*  *import javafx.scene.layout.HBox;*  *import javafx.util.Duration;*  */\*\**  *\* Class: ControlAClockExercise1532*  *\* Developer: Melissa Bakke*  *\* Date: 02/09/2017*  *\* Purpose: Clock animation with functioning start and stop buttons*  *\*/*  *public class ControlAClockExercise1532 extends Application {*  *@Override*  *public void start (Stage primaryStage) {*  *// Create a clock and a label*  *ClockPane clock = new ClockPane();*  *String timeString = clock.getHour() + ":" + clock.getMinute()*  *+ ":" + clock.getSecond();*  *Label lblCurrentTime = new Label(timeString);*    *// Place clock and label in border pane*  *BorderPane pane = new BorderPane();*  *pane.setCenter(clock);*  *pane.setBottom(lblCurrentTime);*  *BorderPane.setAlignment(lblCurrentTime, Pos.TOP\_CENTER);*    *// Create start and stop buttons*  *HBox paneForButtons = new HBox(20);*  *Button btStop = new Button("Stop");*  *Button btStart = new Button("Start");*  *paneForButtons.getChildren().addAll(btStop, btStart);*  *paneForButtons.setAlignment(Pos.CENTER);*  *pane.setBottom(paneForButtons);*    *// Create a timeline and set duration and cycle count*  *Timeline animation = new Timeline(new KeyFrame(Duration.millis(1000), e -> initalStart(clock)));*  *animation.setCycleCount(Timeline.INDEFINITE);*    *// Clicking on buttons calls appropriate method and passes animation to them*  *btStop.setOnAction(e -> clockStop(animation));*  *btStart.setOnAction(e -> clockStart(clock, animation));*    *// Create a scene and place it in the stage*  *Scene scene = new Scene(pane, 250, 250);*  *primaryStage.setTitle("DisplayClock");*  *primaryStage.setScene(scene);*  *primaryStage.show();*  *}*    *// Stops the clock*  *public void clockStop(Timeline animation){*  *animation.stop();*  *}*    *// Starts the clock initally*  *public void initalStart(ClockPane clock){*  *clock.setCurrentTime();*  *}*    *// Starts the clock after being stopped*  *public void clockStart(ClockPane clock, Timeline animation){*  *animation.play();*  *}*    */\*\**  *\* @param args the command line arguments*  *\*/*  *public static void main(String[] args) {*  *Application.launch(args);*  *}*  *}*  *package control.a.clock.exercise.pkg15.pkg32;*  *import java.util.Calendar;*  *import java.util.GregorianCalendar;*  *import javafx.scene.layout.Pane;*  *import javafx.scene.paint.Color;*  *import javafx.scene.shape.Circle;*  *import javafx.scene.shape.Line;*  *import javafx.scene.text.Text;*  */\*\**  *\* Class: ClockPane*  *\* Developer: Melissa Bakke*  *\* Date: 02/14/2017*  *\* Purpose: Class that create a clock*  *\*/*  *public class ClockPane extends Pane {*  *private int hour;*  *private int minute;*  *private int second;*    */\*\* Construct a default clock with the current time\*/*  *public ClockPane() {*  *setCurrentTime();*  *}*  */\*\* Construct a clock with specified hour, minute, and second \*/*  *public ClockPane(int hour, int minute, int second) {*  *this.hour = hour;*  *this.minute = minute;*  *this.second = second;*  *}*  */\*\* Return hour \*/*  *public int getHour() {*  *return hour;*  *}*  */\*\* Set a new hour \*/*  *public void setHour(int hour) {*  *this.hour = hour;*  *paintClock();*  *}*  */\*\* Return minute \*/*  *public int getMinute() {*  *return minute;*  *}*  */\*\* Set a new minute \*/*  *public void setMinute(int minute) {*  *this.minute = minute;*  *paintClock();*  *}*  */\*\* Return second \*/*  *public int getSecond() {*  *return second;*  *}*  */\*\* Set a new second \*/*  *public void setSecond(int second) {*  *this.second = second;*  *paintClock();*  *}*    */\* Set the current time for the clock \*/*  *public void setCurrentTime() {*  *// Construct a calendar for the current date and time*  *Calendar calendar = new GregorianCalendar();*  *// Set current hour, minute and second*  *this.hour = calendar.get(Calendar.HOUR\_OF\_DAY);*  *this.minute = calendar.get(Calendar.MINUTE);*  *this.second = calendar.get(Calendar.SECOND);*  *paintClock(); // Repaint the clock*  *}*    */\*\* Paint the clock \*/*  *private void paintClock() {*  *// Initialize clock parameters*  *double clockRadius = Math.min(getWidth(), getHeight()) \* 0.8 \* 0.5;*  *double centerX = getWidth() / 2;*  *double centerY = getHeight() / 2;*  *// Draw circle*  *Circle circle = new Circle(centerX, centerY, clockRadius);*  *circle.setFill(Color.WHITE);*  *circle.setStroke(Color.BLACK);*  *Text t1 = new Text(centerX - 5, centerY - clockRadius + 12, "12");*  *Text t2 = new Text(centerX - clockRadius + 3, centerY + 5, "9");*  *Text t3 = new Text(centerX + clockRadius - 10, centerY + 3, "3");*  *Text t4 = new Text(centerX - 3, centerY + clockRadius - 3, "6");*  *// Draw second hand*  *double sLength = clockRadius \* 0.8;*  *double secondX = centerX + sLength \* Math.sin(second \* (2 \* Math.PI / 60));*  *double secondY = centerY - sLength \* Math.cos(second \* (2 \* Math.PI / 60));*  *Line sLine = new Line(centerX, centerY, secondX, secondY);*  *sLine.setStroke(Color.RED);*  *// Draw minute hand*  *double mLength = clockRadius \* 0.65;*  *double xMinute = centerX + mLength \* Math.sin(minute \* (2 \* Math.PI / 60));*  *double minuteY = centerY - mLength \* Math.cos(minute \* (2 \* Math.PI / 60));*  *Line mLine = new Line(centerX, centerY, xMinute, minuteY);*  *mLine.setStroke(Color.BLUE);*  *// Draw hour hand*  *double hLength = clockRadius \* 0.5;*  *double hourX = centerX + hLength \* Math.sin((hour % 12 + minute / 60.0) \* (2 \* Math.PI / 12));*  *double hourY = centerY - hLength \* Math.cos((hour % 12 + minute / 60.0) \* (2 \* Math.PI / 12));*  *Line hLine = new Line(centerX, centerY, hourX, hourY);*  *hLine.setStroke(Color.GREEN);*  *getChildren().clear();*  *getChildren().addAll(circle, t1, t2, t3, t4, sLine, mLine, hLine);*  *}*    *@Override*  *public void setWidth(double width) {*  *super.setWidth(width);*  *paintClock();*  *}*  *@Override*  *public void setHeight(double height) {*  *super.setHeight(height);*  *paintClock();*  *}*  *}* |
|  |