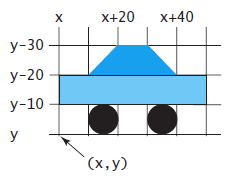
**Java实验报告**

本文是Java实验的实验报告。

**习题1 赛车**

**题目描述**：绘制一辆赛车（颜色任选，款式如图），使用上下左右箭头控制赛车的移动。注意不能让赛车的任何部位超出界面的边界。

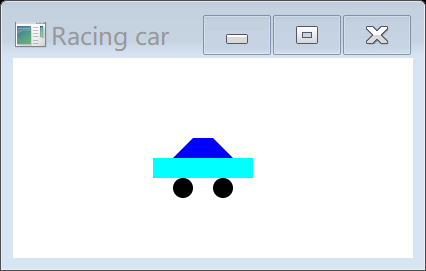
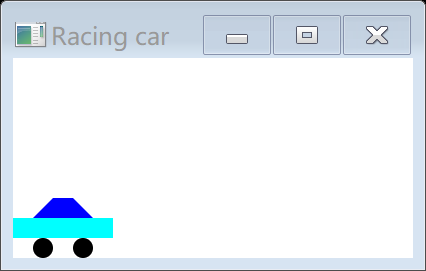


**解题思路**：本题将小车打包成一个结点，然后放置在一个Pane面板上。并编写四个函数分别处理上下左右四个事件的响应。

**源代码**：

import javafx.application.Application;  
import javafx.collections.ObservableList;  
import javafx.scene.Scene;  
import javafx.scene.layout.Pane;  
import static javafx.scene.paint.Color.\*;  
import javafx.scene.shape.Circle;  
import javafx.scene.shape.Polygon;  
import javafx.scene.shape.Rectangle;  
import javafx.stage.Stage;  
  
public class RacingCar extends Application {  
 @Override  
 public void start(Stage primaryStage) {  
 Pane pane = new Pane();  
 Car carPane = new Car();  
  
 carPane.setOnKeyPressed(e -> {  
 switch(e.getCode()) {  
 case *UP*: {  
 carPane.up();  
 }break;  
 case *DOWN*: {  
 carPane.down();  
 }break;  
 case *LEFT*: {  
 carPane.left();  
 }break;  
 case *RIGHT*: {  
 carPane.right();  
 }break;  
 }  
 });  
  
 pane.getChildren().add(carPane);  
  
 Scene scene = new Scene(pane, 200, 100);  
 primaryStage.setScene(scene);  
 primaryStage.setTitle("Racing car");  
 primaryStage.show();  
 carPane.requestFocus();  
 }  
}  
  
class Car extends Pane {  
 Circle leftWheel = new Circle(5, *BLACK*);  
 Circle rightWheel = new Circle(5, *BLACK*);  
 Rectangle body = new Rectangle(50, 10, *AQUA*);  
 Polygon hat = new Polygon();  
  
 Car() {  
 leftWheel.setCenterX(15);  
 leftWheel.setCenterY(95);  
 rightWheel.setCenterX(35);  
 rightWheel.setCenterY(95);  
 body.setX(0);  
 body.setY(80);  
 ObservableList<Double> list = hat.getPoints();  
 list.addAll(20.0, 70.0,  
 30.0, 70.0,  
 40.0, 80.0,  
 10.0, 80.0);  
 hat.setFill(*BLUE*);  
 getChildren().addAll(leftWheel, rightWheel, body, hat);  
 }  
  
 public void up() {  
 if(leftWheel.getCenterY() - 10 >= 25) {  
 leftWheel.setCenterY(leftWheel.getCenterY() - 10);  
 rightWheel.setCenterY(rightWheel.getCenterY() - 10);  
 body.setY(body.getY() - 10);  
 ObservableList<Double> list = hat.getPoints();  
 for(int i = 1; i < 8; i = i + 2)  
 list.set(i, list.get(i) - 10);  
 }  
 }  
  
 public void down() {  
 if(leftWheel.getCenterY() + 10 <= 95) {  
 leftWheel.setCenterY(leftWheel.getCenterY() + 10);  
 rightWheel.setCenterY(rightWheel.getCenterY() + 10);  
 body.setY(body.getY() + 10);  
 ObservableList<Double> list = hat.getPoints();  
 for(int i = 1; i < 8; i = i + 2)  
 list.set(i, list.get(i) + 10);  
 }  
 }  
  
 public void left() {  
 if(leftWheel.getCenterX() - 10 >= 15) {  
 leftWheel.setCenterX(leftWheel.getCenterX() - 10);  
 rightWheel.setCenterX(rightWheel.getCenterX() - 10);  
 body.setX(body.getX() - 10);  
 ObservableList<Double> list = hat.getPoints();  
 for(int i = 0; i < 8; i = i + 2)  
 list.set(i, list.get(i) - 10);  
 }  
 }  
  
 public void right() {  
 if(leftWheel.getCenterX() + 10 <= 165) {  
 leftWheel.setCenterX(leftWheel.getCenterX() + 10);  
 rightWheel.setCenterX(rightWheel.getCenterX() + 10);  
 body.setX(body.getX() + 10);  
 ObservableList<Double> list = hat.getPoints();  
 for(int i = 0; i < 8; i = i + 2)  
 list.set(i, list.get(i) + 10);  
 }  
 }  
}

**测试样例**：

**习题2 幻灯片放映**

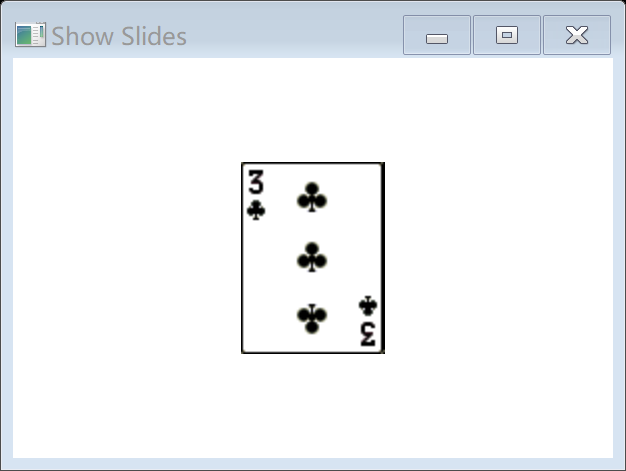
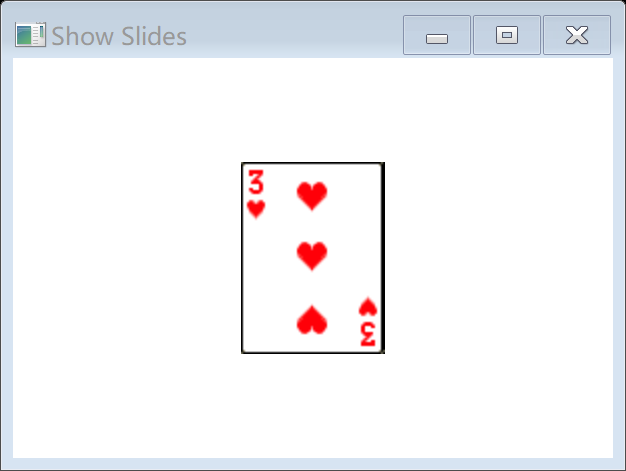
**题目描述**：写一个程序，将54张扑克牌图片，按照每次1张，每张1秒的放映间隔，按顺序循环显示图片。并且要求：（1）向上箭头加快放映速度；向下箭头减缓放映速度；（2）双击鼠标左键暂停放映，再次双击则继续。程序界面自定。

**解题思路**：设置一个时间轴，将每一张图片的查看放置在时间轴上，并设置相关的事件响应。

**源代码**：

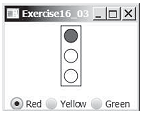
import java.io.IOException;  
import java.io.InputStream;  
import java.util.jar.JarEntry;  
import java.util.jar.JarFile;  
import javafx.animation.KeyFrame;  
import javafx.animation.Timeline;  
import javafx.application.Application;  
import javafx.event.ActionEvent;  
import javafx.event.EventHandler;  
import javafx.scene.Scene;  
import javafx.scene.image.Image;  
import javafx.scene.image.ImageView;  
import javafx.scene.input.KeyCode;  
import javafx.scene.input.MouseButton;  
import javafx.scene.layout.BorderPane;  
import javafx.stage.Stage;  
import javafx.util.Duration;  
  
public class ShowSlides extends Application {  
 private int imgNumber = 1;  
 private boolean isPlay = true;  
  
 @Override  
 public void start(Stage primaryStage) {  
 BorderPane pane = new BorderPane();  
  
 String imgAddress = "cards/" + imgNumber + ".png";  
 ImageView imgView = new ImageView();  
 imgView.setImage(*readImageFromJar*(".\\cards.jar", imgAddress));  
  
 pane.setCenter(imgView);  
  
 EventHandler<ActionEvent> eventHandler = e -> {  
 imgNumber = imgNumber % 54 + 1;  
 String imgAddressNew = "cards/" + imgNumber + ".png";  
 imgView.setImage(*readImageFromJar*(".\\cards.jar", imgAddressNew));  
 };  
  
 Timeline tl = new Timeline(new KeyFrame(Duration.*millis*(1000), eventHandler));  
 tl.setCycleCount(Timeline.*INDEFINITE*);  
 tl.play();  
  
 imgView.setOnKeyPressed(e -> {  
 if(e.getCode().compareTo(KeyCode.*UP*) == 0) {  
 if(tl.getRate() < 1000)  
 tl.setRate(tl.getRate() \* 2);  
 }  
 else if(e.getCode().compareTo(KeyCode.*DOWN*) == 0) {  
 if(tl.getRate() >= 0.5)  
 tl.setRate(tl.getRate() / 2);  
 }  
 });  
 imgView.setOnMouseClicked(e -> {  
 if(e.getButton() == MouseButton.*PRIMARY* && e.getClickCount() == 2) {  
 if(isPlay == true) {  
 tl.pause();  
 isPlay = false;  
 }  
 else {  
 tl.play();  
 isPlay = true;  
 }  
 }  
 });  
  
 Scene scene = new Scene(pane, 300, 200);  
 primaryStage.setTitle("Show Slides");  
 primaryStage.setScene(scene);  
 primaryStage.show();  
 imgView.requestFocus();  
 }  
  
 public static Image readImageFromJar(String jarname, String picname) {  
 Image image = null;  
 try {  
 JarFile jarFile = new JarFile(jarname);  
 JarEntry entry = jarFile.getJarEntry(picname);  
 InputStream in = jarFile.getInputStream(entry);  
 image = new Image(in);  
 in.close();  
 jarFile.close();  
 }  
 catch (IOException e) {  
 System.*err*.println("read file error.");  
 }  
 return image;  
 }  
}

**测试样例**：



**习题3 交通灯**

**题目描述**：如图，当用户选择交通灯下方的颜色时，自动填充（相当于点亮）上面对应的颜色灯，并注意清空（相当于熄灭）原先点亮的灯。程序开始自动点亮红灯。

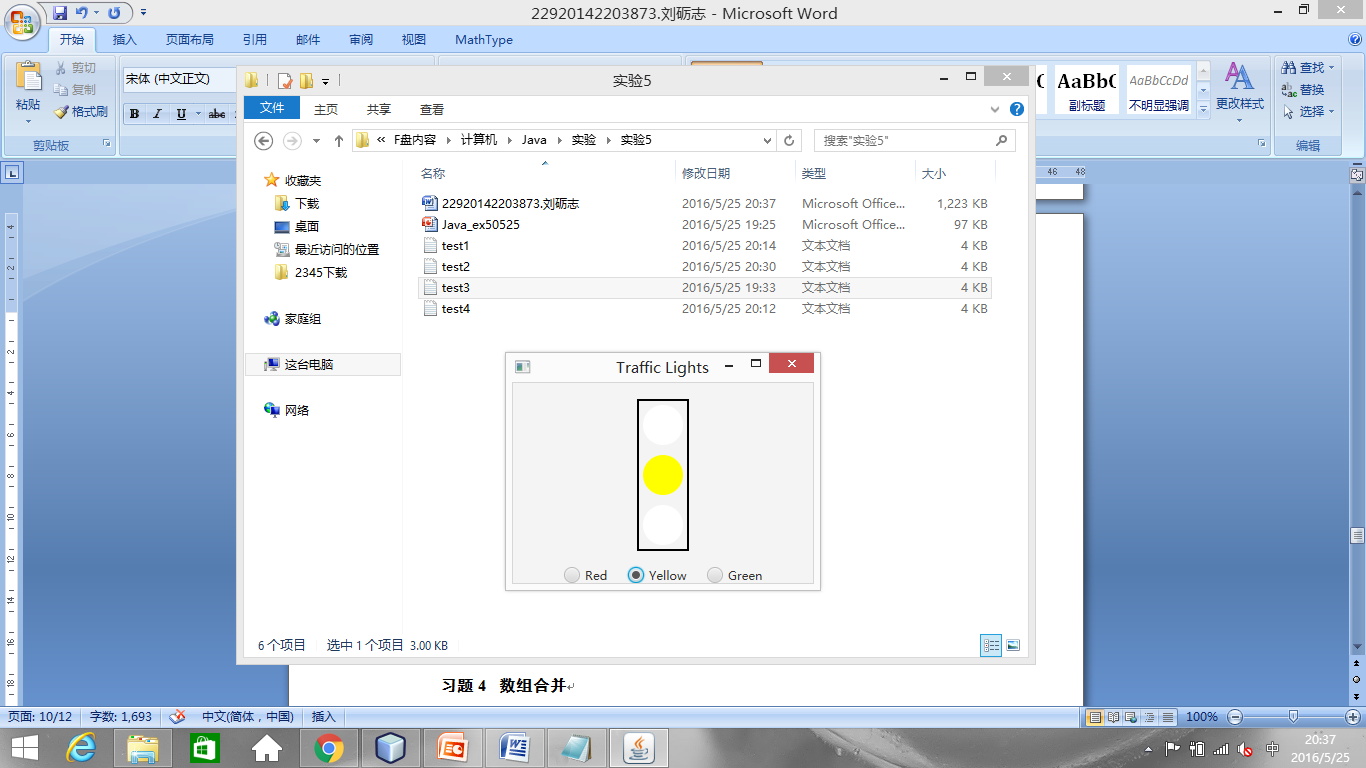
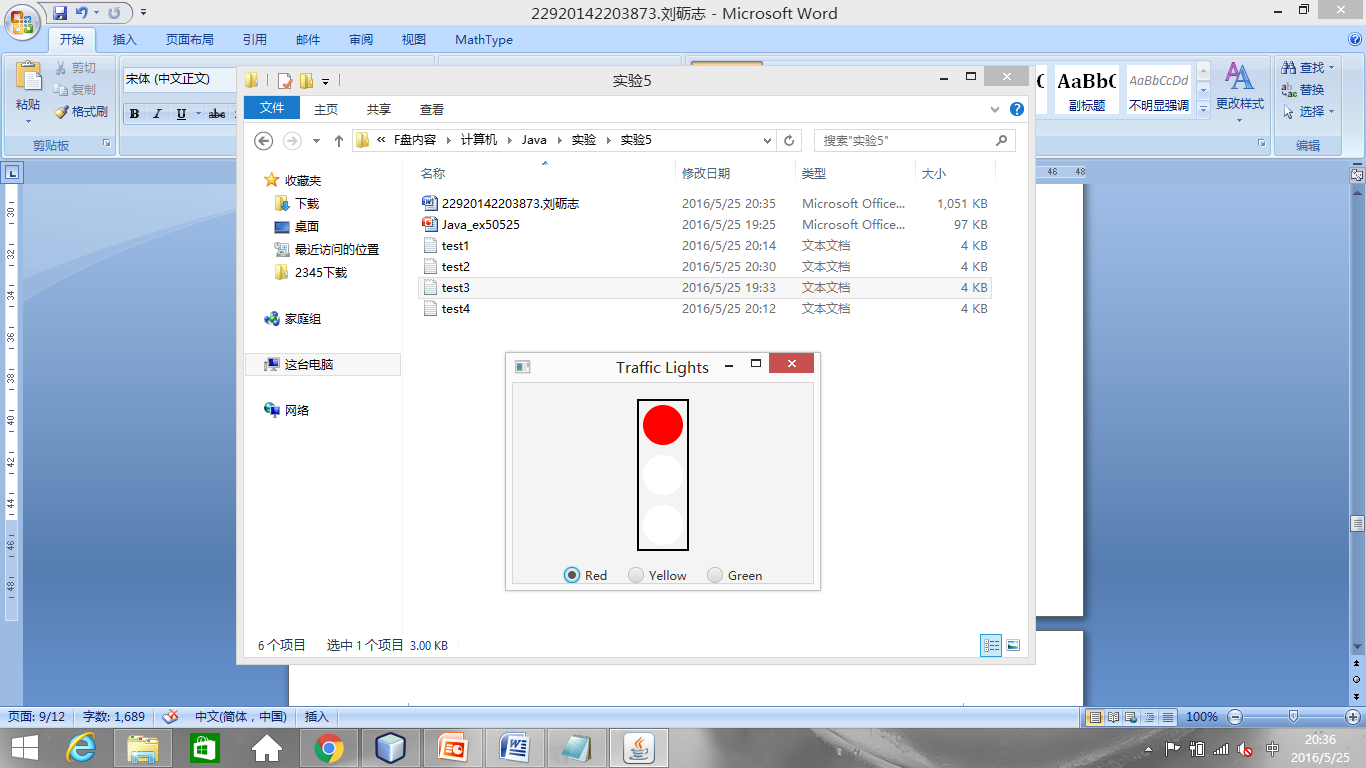
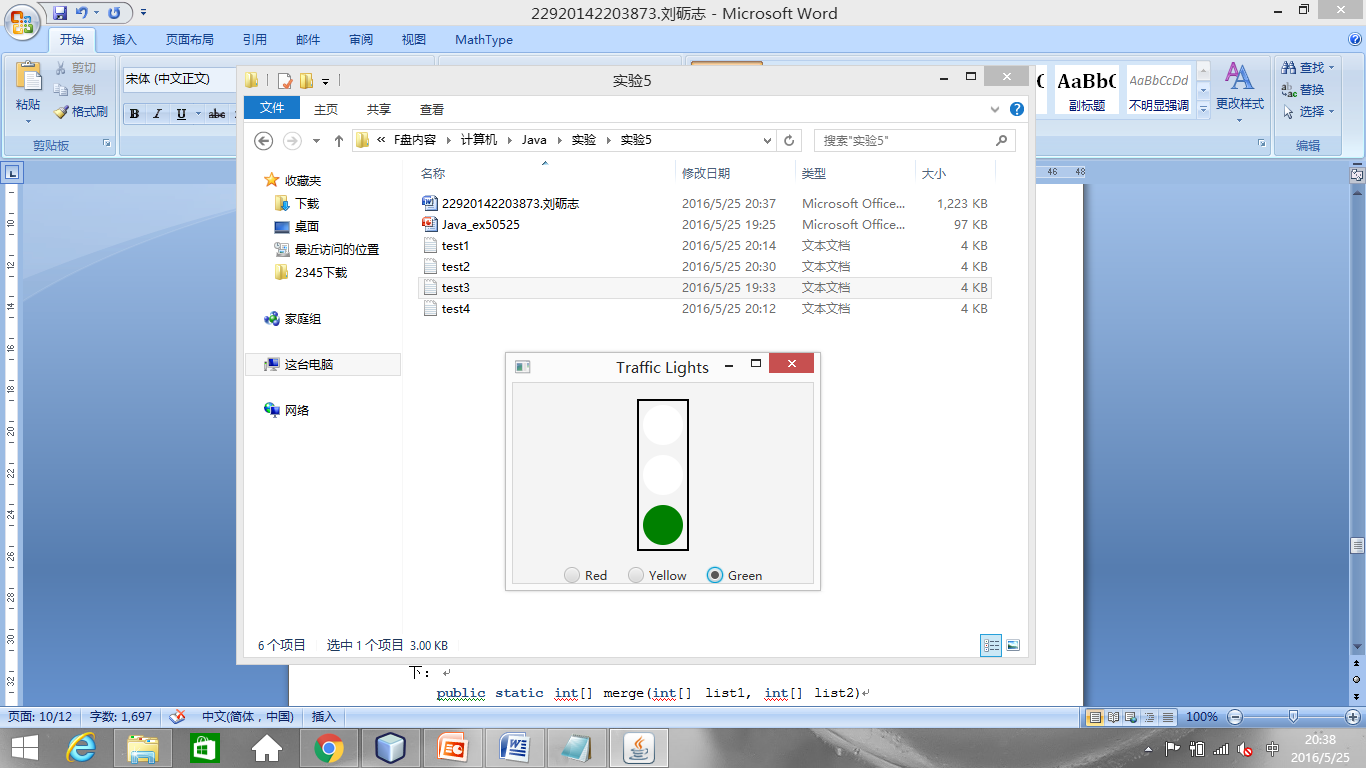


**解题思路**：在主面板上放置一个由矩形和三个圆组成的栈面板，以及一排单选框。并且设置单选框与三个圆颜色的响应。

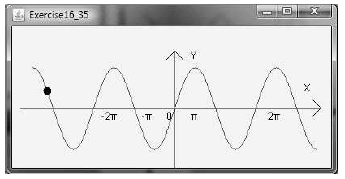
**源代码**：

import java.awt.Container;  
import java.awt.GridLayout;  
import java.awt.event.WindowAdapter;  
import java.awt.event.ItemListener;  
import java.awt.event.ItemEvent;  
import java.awt.event.WindowEvent;  
import javax.swing.JFrame;  
import javax.swing.JPanel;  
import javax.swing.JRadioButton;  
import javax.swing.ButtonGroup;  
import javax.swing.BorderFactory;  
import javax.swing.ImageIcon;  
  
import javafx.application.Application;  
import javafx.geometry.Pos;  
import javafx.scene.Scene;  
import javafx.scene.control.RadioButton;  
import javafx.scene.control.ToggleGroup;  
import javafx.scene.layout.BorderPane;  
import javafx.scene.layout.HBox;  
import javafx.scene.layout.StackPane;  
import javafx.scene.layout.VBox;  
import javafx.scene.paint.Color;  
import javafx.scene.shape.Circle;  
import javafx.scene.shape.Rectangle;  
import javafx.stage.Stage;  
  
//完成单选的实现  
class MyRadio {  
 private String imRed ="C:\\Users\\Apple\\Desktop\\codes\\java\\190625\\src\\Red.jpg";  
 private String imYellow = "C:\\Users\\Apple\\Desktop\\codes\\java\\190625\\src\\Yellow.jpg";  
 private String imGreen ="C:\\Users\\Apple\\Desktop\\codes\\java\\190625\\src\\Green.jpg";  
 //设置一窗体  
 private JFrame frame = new JFrame("单选按钮");  
 //获取窗体容器  
 private Container cont = frame.getContentPane();  
 //定义三个窗体按钮  
 private JRadioButton jradio1 = new JRadioButton("红灯");  
 private JRadioButton jradio2 = new JRadioButton("黄灯");  
 private JRadioButton jradio3 = new JRadioButton("绿灯");  
 //定义一个面板  
 private JPanel pan = new JPanel();  
 public MyRadio() {  
 //设置边框显示条  
 pan.setBorder(BorderFactory.*createTitledBorder*("红绿灯："));  
 //定义排版样式  
 pan.setLayout(new GridLayout(1, 3));  
 //面板中加入两个单选按钮  
 pan.add(this.jradio1);  
 pan.add(this.jradio2);  
 pan.add(this.jradio3);  
 //定义按钮组  
 ButtonGroup group=new ButtonGroup();  
 //把单选按钮添加到按钮组中，这样只能选组中的一个按钮，真正实现单选  
 group.add(jradio1);  
 group.add(jradio2);  
 group.add(jradio3);  
 //单选按钮设置监听器  
 jradio1.addItemListener(new MyItemListener());  
 jradio2.addItemListener(new MyItemListener());  
 jradio3.addItemListener(new MyItemListener());  
  
 cont.add(pan);  
 this.frame.setSize(400, 200);  
 this.frame.setVisible(true);  
 //监听窗体关闭事件  
 this.frame.addWindowListener(new WindowAdapter() {  
 public void windowClosing(WindowEvent obj) {  
 System.*exit*(1);  
 }  
 });  
 }  
  
 class MyItemListener implements ItemListener {  
 public void itemStateChanged(ItemEvent e) {  
 JRadioButton btn = (JRadioButton) e.getSource();//发生Event的源头  
 int state = e.getStateChange();  
 switch (state) {  
 case ItemEvent.*SELECTED*:  
 //System.out.println(btn.getText() + "被选中");  
 break;  
 case ItemEvent.*DESELECTED*:  
 //System.out.println(btn.getText() + "取消");  
 break;  
 }  
 }  
 }  
  
 public void itemStateChanged(ItemEvent e) {  
 if (e.getSource() == jradio1) {  
 jradio1.setIcon(new ImageIcon(imRed));  
 jradio2.setIcon(null);  
 jradio3.setIcon(null);  
 } else if (e.getSource() == jradio2){  
 jradio2.setIcon(new ImageIcon(imYellow));  
 jradio1.setIcon(null);  
 jradio3.setIcon(null);  
 }  
 else {  
 jradio3.setIcon(new ImageIcon(imGreen));  
 jradio1.setIcon(null);  
 jradio2.setIcon(null);  
 }  
  
 }  
}  
  
public class JRadioButtonDemo01 {  
 public static void main(String args[]) {  
 new MyRadio();  
 }  
}

**测试样例**：

**习题4 沿正弦曲线运动的小球**

**题目描述**：如图，编程实现让小球沿着正弦曲线从左向右运动，如果到达曲线右边界，则回到最左边重新开始运动。用户按一次空格键可以暂停小球运动，再按一次空格键可以让小球继续运动。

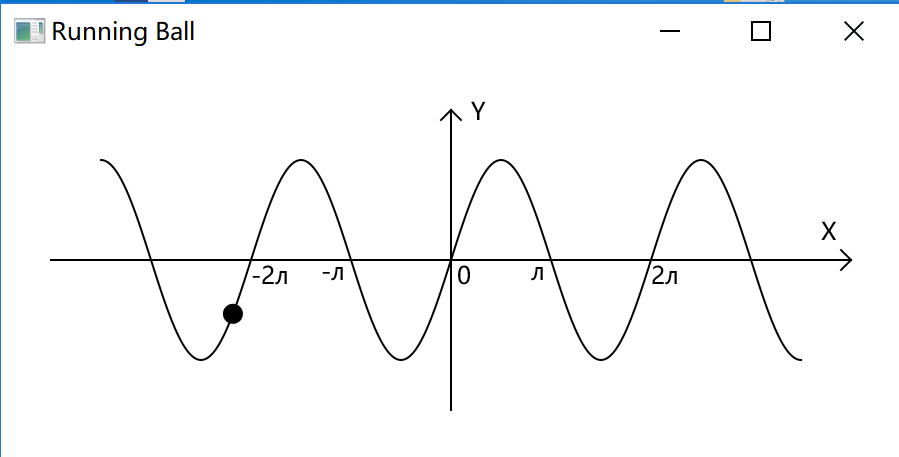


**解题思路**：利用折线绘制正弦波图像，然后将小球放到折线运动，并设置相关的事件响应。

**源代码**：

import javafx.animation.PathTransition;  
import javafx.application.Application;  
import javafx.collections.ObservableList;  
import javafx.scene.Scene;  
import javafx.scene.input.KeyCode;  
import javafx.scene.layout.Pane;  
import javafx.scene.paint.Color;  
import javafx.scene.shape.Circle;  
import javafx.scene.shape.Line;  
import javafx.scene.shape.Polyline;  
import javafx.scene.text.Text;  
import javafx.stage.Stage;  
import javafx.util.Duration;  
  
public class RunningBall extends Application {  
 private boolean isPlay;  
  
 @Override  
 public void start(Stage primaryStage) {  
 Pane pane = new Pane();  
  
 Polyline polyline = new Polyline();  
 ObservableList<Double> list = polyline.getPoints();  
 for(int i = 50; i <= 400; ++i) {  
 list.add((double)i);  
 list.add(50 \* Math.*cos*(((double)i) / 50.0 \* Math.*PI*) + 100);  
 }  
 pane.getChildren().addAll(polyline);  
  
 Line axisX = new Line(25.0, 100.0, 425.0, 100.0);  
 Line arrowX1 = new Line(420.0, 95.0, 425.0, 100.0);  
 Line arrowX2 = new Line(420.0, 105.0, 425.0, 100.0);  
 Line axisY = new Line(225.0, 175.0, 225.0, 25.0);  
 Line arrowY1 = new Line(220.0, 30.0, 225.0, 25.0);  
 Line arrowY2 = new Line(230.0, 30.0, 225.0, 25.0);  
 pane.getChildren().addAll(axisX, arrowX1, arrowX2, axisY, arrowY1, arrowY2);  
  
 Text X = new Text(410.0, 90.0, "X");  
 Text Y = new Text(235.0, 30.0, "Y");  
 Text Zero = new Text(228.0, 112.0, "0");  
 Text Pi = new Text(265.0, 110.0, "л");  
 Text minusPi = new Text(160.0, 110.0, "-л");  
 Text doublePi = new Text(325.0, 112.0, "2л");  
 Text minusDoublePi = new Text(125.0, 112.0, "-2л");  
 pane.getChildren().addAll(X, Y, Zero, Pi, minusPi, doublePi, minusDoublePi);  
  
 Circle ball = new Circle(5);  
 ball.setCenterX(50);  
 ball.setCenterY(50);  
 ball.setFill(Color.*BLACK*);  
 pane.getChildren().add(ball);  
  
 PathTransition pt = new PathTransition();  
 pt.setNode(ball);  
 pt.setPath(polyline);  
 pt.setDuration(Duration.*millis*(6000));  
 pt.play();  
  
 pt.currentTimeProperty().addListener((ov, old\_val, new\_val) -> {  
 if(pt.getCurrentTime().compareTo(pt.getDuration()) >= 0)  
 pt.playFromStart();  
 });  
  
 isPlay = true;  
 ball.setOnKeyPressed(e -> {  
 if(e.getCode() == KeyCode.*SPACE*) {  
 if(isPlay == true) {  
 pt.pause();  
 isPlay = false;  
 }  
 else {  
 pt.play();  
 isPlay = true;  
 }  
 }  
 });  
  
 Scene scene = new Scene(pane, 450, 200);  
 primaryStage.setTitle("Running Ball");  
 primaryStage.setScene(scene);  
 primaryStage.show();  
 ball.requestFocus();  
 }  
}

**测试样例**：



总结

通过本次实验，我进一步熟悉了JavaFX的使用方法和注意事项。通过练习，我对Java已经有了一些了解，但是显然这样的程度是非常浅的。在第三题中，由于路径的原因，RadioButton无法使用，所以改用JRadioButton 在以后的时间里，我还要刻苦努力，加深对Java的了解和熟悉。