# 实验9 ： JavaFX基础

## 实验目的

## 实验题目

### P498 14.1

原题：（显示图像）请写一个程序，在一个网格面板里面显示4个图像

源程序：

**import** javafx.application.Application;

**import** javafx.geometry.Pos;

**import** javafx.scene.Scene;

**import** javafx.scene.image.Image;

**import** javafx.scene.image.ImageView;

**import** javafx.scene.layout.GridPane;

**import** javafx.scene.layout.Pane;

**import** javafx.stage.Stage;

**public** **class** Test0 **extends** Application {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

Application.*launch*(args);

}

@Override

**public** **void** start(Stage primaryStage) **throws** Exception {

// **TODO** Auto-generated method stub

GridPane pane = **new** GridPane();

Image image = **new** Image("image/us.gif");

ImageView imageView1 = **new** ImageView(image);

imageView1.setFitHeight(200);

imageView1.setFitWidth(200);

ImageView imageView2 = **new** ImageView(image);

imageView2.setFitHeight(200);

imageView2.setFitWidth(200);

ImageView imageView3 = **new** ImageView(image);

imageView3.setFitHeight(200);

imageView3.setFitWidth(200);

ImageView imageView4 = **new** ImageView(image);

imageView4.setFitHeight(200);

imageView4.setFitWidth(200);

pane.add(imageView1, 0, 0);

pane.add(imageView2, 0, 1);

pane.add(imageView3, 1, 0);

pane.add(imageView4, 1, 1);

Scene scene = **new** Scene(pane);

primaryStage.setTitle("ShowImage");

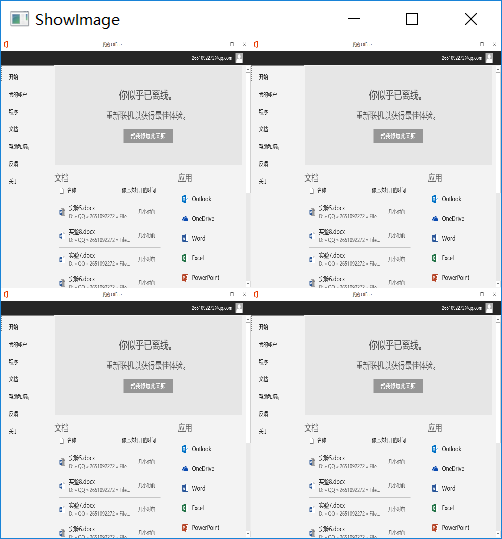
primaryStage.setScene(scene);

primaryStage.show();

}

}

结果及截图：



### P500 14.11

原题：（绘制一个笑脸）请写一个绘制笑脸的程序。

源程序：

**import** javafx.application.Application;

**import** javafx.scene.Scene;

**import** javafx.scene.layout.Pane;

**import** javafx.scene.paint.Color;

**import** javafx.scene.shape.Arc;

**import** javafx.scene.shape.ArcType;

**import** javafx.scene.shape.Circle;

**import** javafx.scene.shape.Ellipse;

**import** javafx.scene.shape.Polygon;

**import** javafx.stage.Stage;

**public** **class** Test1 **extends** Application {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

Application.*launch*(args);

}

@Override

**public** **void** start(Stage primaryStage) **throws** Exception {

// **TODO** Auto-generated method stub

Pane pane = **new** Pane();

Circle circle1 = **new** Circle(150,150,150,Color.***YELLOW***);

Circle circle2 = **new** Circle(75,90,20,Color.***BLACK***);

Circle circle3 = **new** Circle(225,90,20,Color.***BLACK***);

Polygon polygon = **new** Polygon();

polygon.getPoints().addAll(**new** Double[] {

150.0,120.0,

130.0,170.0,

170.0,170.0

});

polygon.setFill(Color.***RED***);

Ellipse ellipse1 = **new** Ellipse();

ellipse1.setCenterX(75.0f);

ellipse1.setCenterY(90.0f);

ellipse1.setRadiusX(40.0f);

ellipse1.setRadiusY(30.0f);

ellipse1.setStroke(Color.***BLACK***);

ellipse1.setFill(**null**);

Ellipse ellipse2 = **new** Ellipse();

ellipse2.setCenterX(225.0f);

ellipse2.setCenterY(90.0f);

ellipse2.setRadiusX(40.0f);

ellipse2.setRadiusY(30.0f);

ellipse2.setStroke(Color.***BLACK***);

ellipse2.setFill(**null**);

Arc arc = **new** Arc();

arc.setCenterX(150.0f);

arc.setCenterY(200.0f);

arc.setRadiusX(60.0f);

arc.setRadiusY(30.0f);

arc.setStartAngle(200.0f);

arc.setLength(140.0f);

arc.setType(ArcType.***OPEN***);

arc.setFill(Color.***BLUE***);

pane.getChildren().add(circle1);

pane.getChildren().add(circle2);

pane.getChildren().add(circle3);

pane.getChildren().add(polygon);

pane.getChildren().add(ellipse1);

pane.getChildren().add(ellipse2);

pane.getChildren().add(arc);

Scene scene = **new** Scene(pane);

primaryStage.setScene(scene);

primaryStage.setTitle("ShowSmileFace");

primaryStage.show();

}

}

结果及截图：



### P500 14.18

原题：（绘制平方函数）请写一个程序，绘制表示函数f(x)=x\*x的图

源程序：

**import** javafx.application.Application;

**import** javafx.collections.ObservableList;

**import** javafx.scene.Scene;

**import** javafx.scene.layout.Pane;

**import** javafx.scene.shape.Line;

**import** javafx.scene.shape.Polyline;

**import** javafx.stage.Stage;

**public** **class** Test2 **extends** Application{

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

Application.*launch*(args);

}

@Override

**public** **void** start(Stage primaryStage) **throws** Exception {

// **TODO** Auto-generated method stub

Pane pane = **new** Pane();

Polyline polyline = **new** Polyline();

ObservableList<Double> list = polyline.getPoints();

**double** scaleFactor = 0.0125;

**for**(**int** x=-100;x<=100;x++)

{

list.add(x+120.0);

list.add(-scaleFactor\*x\*x+120);

}

Line line1 = **new** Line();

line1.setStartX(0);

line1.setStartY(120);

line1.setEndX(240);

line1.setEndY(120);

Line line2 = **new** Line();

line2.setStartX(120);

line2.setStartY(0);

line2.setEndX(120);

line2.setEndY(120);

pane.getChildren().add(polyline);

pane.getChildren().add(line1);

pane.getChildren().add(line2);

Scene scene = **new** Scene(pane);

primaryStage.setScene(scene);

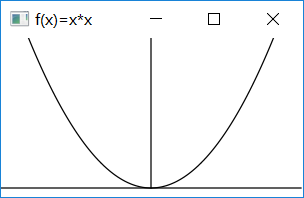
primaryStage.setTitle("f(x)=x\*x");

primaryStage.show();

}

}

结果及截图：



### P502 14.27

原题：（绘制一个详细的时钟）修改14.12节的ClockPane类，绘制一个包含小时和分钟的更加详细信息的时钟。

源程序：

**package** project;

**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;

**public** **class** ClockPane **extends** Pane {

**private** **int** hour;

**private** **int** minute;

**private** **int** second;

**public** ClockPane() {

setCurrentTime();

}

**public** ClockPane(**int** hour, **int** minute, **int** second) {

**this**.hour = hour;

**this**.minute = minute;

**this**.second = second;

}

**public** **int** getHour() {

**return** hour;

}

**public** **void** setHour(**int** hour) {

**this**.hour = hour;

paintClock();

}

**public** **int** getMinute() {

**return** minute;

}

**public** **void** setMinute(**int** minute) {

**this**.minute = minute;

paintClock();

}

**public** **int** getSecond() {

**return** second;

}

**public** **void** setSecond(**int** second) {

**this**.second = second;

paintClock();

}

**public** **void** setCurrentTime() {

Calendar calendar = **new** GregorianCalendar();

**this**.hour = calendar.get(Calendar.***HOUR\_OF\_DAY***);

**this**.minute = calendar.get(Calendar.***MINUTE***);

**this**.second = calendar.get(Calendar.***SECOND***);

paintClock();

}

**private** **void** paintClock() {

**double** clockRadius =

Math.*min*(getWidth(), getHeight()) \* 0.8 \* 0.5;

**double** centerX = getWidth() / 2;

**double** centerY = getHeight() / 2;

Circle circle = **new** Circle(centerX, centerY, clockRadius);

circle.setFill(Color.***WHITE***);

circle.setStroke(Color.***BLACK***);

Text t12 = **new** Text(centerX-3 , centerY - clockRadius +12,"12");

Text t9 = **new** Text(centerX - clockRadius + 3, centerY , "9");

Text t3 = **new** Text(centerX + clockRadius - 12, centerY , "3");

Text t6 = **new** Text(centerX - 3, centerY + clockRadius - 3, "6");

Text t1 = **new** Text(centerX +clockRadius\*Math.*sin*(Math.***PI***/6)-5, centerY-clockRadius\*Math.*cos*(Math.***PI***/6)+12 , "1");

Text t2 = **new** Text(centerX +clockRadius\*Math.*sin*(Math.***PI***/3)-12, centerY-clockRadius\*Math.*cos*(Math.***PI***/3)+5 , "2");

Text t4 = **new** Text(centerX +clockRadius\*Math.*sin*(Math.***PI***/3)-9, centerY+clockRadius\*Math.*cos*(Math.***PI***/3)+1 , "4");

Text t5 = **new** Text(centerX +clockRadius\*Math.*sin*(Math.***PI***/6)-5, centerY+clockRadius\*Math.*cos*(Math.***PI***/6)-3 , "5");

Text t7 = **new** Text(centerX -clockRadius\*Math.*sin*(Math.***PI***/6)+1, centerY+clockRadius\*Math.*cos*(Math.***PI***/6)-4 , "7");

Text t8 = **new** Text(centerX -clockRadius\*Math.*sin*(Math.***PI***/3)+1, centerY+clockRadius\*Math.*cos*(Math.***PI***/3)+0 , "8");

Text t10 = **new** Text(centerX -clockRadius\*Math.*sin*(Math.***PI***/3)+2, centerY-clockRadius\*Math.*cos*(Math.***PI***/3)+6 , "10");

Text t11 = **new** Text(centerX -clockRadius\*Math.*sin*(Math.***PI***/6)-3, centerY-clockRadius\*Math.*cos*(Math.***PI***/6)+12 , "11");

**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***);

**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***);

**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,t5,t6,t7,t8,t9,t10,t11,t12, sLine, mLine, hLine);

}

@Override

**public** **void** setWidth(**double** width) {

**super**.setWidth(width);

paintClock();

}

@Override

**public** **void** setHeight(**double** height) {

**super**.setHeight(height);

paintClock();

}

}

**package** project;

**import** javafx.application.Application;

**import** javafx.geometry.Pos;

**import** javafx.stage.Stage;

**import** javafx.scene.Scene;

**import** javafx.scene.control.Label;

**import** javafx.scene.layout.BorderPane;

**public** **class** Test3 **extends** Application {

@Override // Override the start method in the Application class

**public** **void** start(Stage primaryStage) {

ClockPane clock = **new** ClockPane();

String timeString = clock.getHour() + ":" + clock.getMinute()

+ ":" + clock.getSecond();

Label lblCurrentTime = **new** Label(timeString);

BorderPane pane = **new** BorderPane();

pane.setCenter(clock);

pane.setBottom(lblCurrentTime);

BorderPane.*setAlignment*(lblCurrentTime, Pos.***TOP\_CENTER***);

Scene scene = **new** Scene(pane, 250, 250);

primaryStage.setTitle("DisplayClock");

primaryStage.setScene(scene);

primaryStage.show();

}

**public** **static** **void** main(String[] args) {

Application.*launch*(args);

}

}

结果及截图：

