

Java2 Lab 10

(JavaFx2)

[Experimental Objective]

深入理解JavaFx的图表，2d和3d图像的建立

[JavaFx中的charts]

JavaFx 中有许多统计图：

如饼图(pieChart)、折线图(lineChart)、条形图(barChart)、散点图(scatterChart)等

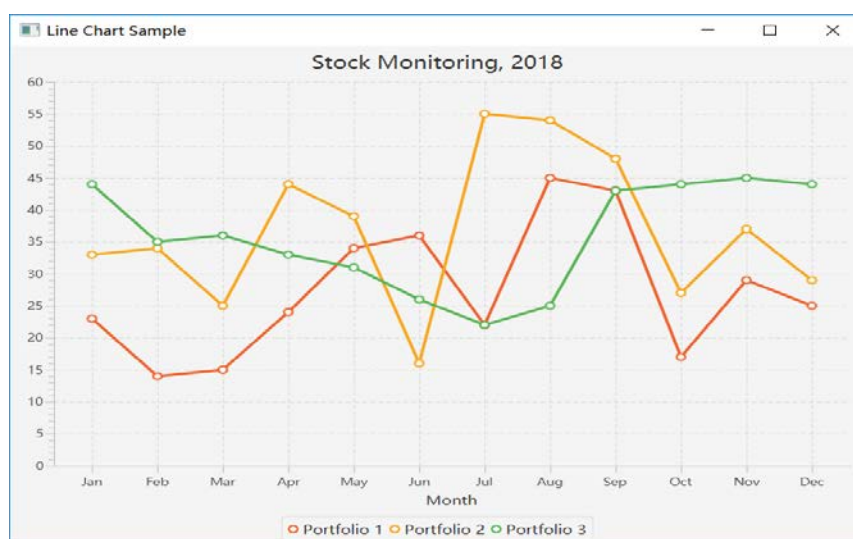
[案例1-折线图]

我们可以用JAVAFX绘制一个折线图，描绘股票价格的波动情况。先建好x轴和y轴，分别加入到折线图中，然后填写图表数据并与折线图绑定，即可。

```
public void start(Stage primaryStage) {  
    primaryStage.setTitle("line Chart");  
    NumberAxis xAxis = new NumberAxis();  
    NumberAxis yAxis = new NumberAxis();  
    xAxis.setLabel("number of month");  
    LineChart<Number, Number> lineChart = new LineChart<Number, Number>(xAxis, yAxis);  
    lineChart.setTitle("stock monitor, 2018");  
    XYChart.Series series = new XYChart.Series();  
    series.setName("my portfolio");  
    series.getData().add(new XYChart.Data(1, 23));  
    series.getData().add(new XYChart.Data(2, 14));  
    series.getData().add(new XYChart.Data(3, 15));  
    series.getData().add(new XYChart.Data(4, 24));  
    series.getData().add(new XYChart.Data(5, 34));  
    series.getData().add(new XYChart.Data(6, 36));  
    series.getData().add(new XYChart.Data(7, 22));  
  
    lineChart.getData().add(series);  
    Scene scene = new Scene(lineChart, 800, 600);  
    primaryStage.setScene(scene);  
    primaryStage.show();  
}
```



在案例1的基础上，绘制三条折线，分别表示3个不同投资组合的走势情况，实现的样图如下，数据自己拟定即可。



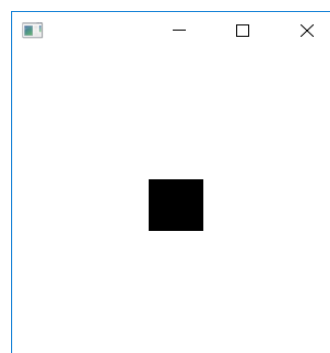
[案例2-2d图像]

我们可以用JAVAFX绘制一个2D图画。一些常用的类库在javafx.scene.shape包中。这里我们可以简单绘制一个2D的矩形，然后将矩形加入到pane，再加入到stage中即可：

```
public class twod1 extends Application {
    @Override
    public void start(Stage stage) {

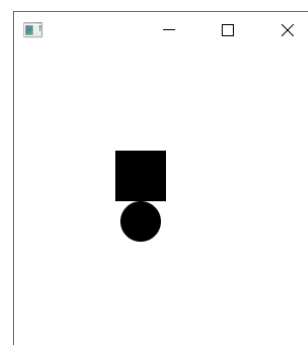
        Rectangle rect = new Rectangle(100,100,50,50);

        StackPane pane = new StackPane();
        pane.getChildren().add(rect);
        Scene scene = new Scene(pane, 300, 300);
        stage.setScene(scene);
        stage.show();
    }
    public static void main(String[] args) {
        launch(args);
    }
}
```



[练习2-2d图像]

在案例2 的基础上，我们可以再画一个圆形，与之相接。当然也可以自己创造一些由其他几个图形相组合，形成一个复杂的图案。



[案例3-3d图像]

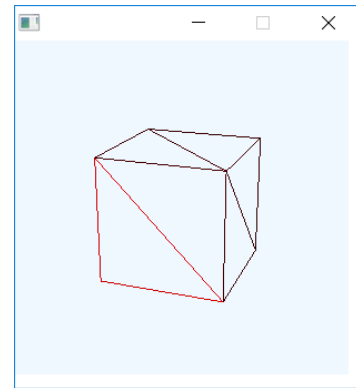
我们可以用JAVAFX绘制一个3D图画。在这里我们简单绘制一个3D的盒子：设置好Box的大小，线型，再设置相机视角，旋转一个角度才可以看全整个盒子。再将相机和Box分组绑定，再加到场景和舞台中，即可。

```
public Parent createContent() throws Exception {
    // Box
    Box testBox = new Box(3, 3, 3);
    testBox.setMaterial(new PhongMaterial(Color.RED));
    testBox.setDrawMode(DrawMode.LINE);
    // Create and position camera
    PerspectiveCamera camera = new PerspectiveCamera(true);
    camera.getTransforms().addAll (
        new Rotate(-20, Rotate.Y_AXIS),
        new Rotate(-20, Rotate.X_AXIS),
        new Translate(0, 0, -15));

    // Build the Scene Graph
    Group root = new Group();
    root.getChildren().add(camera);
    root.getChildren().add(testBox);

    // Use a SubScene
    SubScene subScene = new SubScene(root, 300, 300);
    subScene.setFill(Color.ALICEBLUE);
    subScene.setCamera(camera);
    Group group = new Group();
    group.getChildren().add(subScene);
    return group;
}

@Override
public void start(Stage primaryStage) throws Exception {
    Scene scene = new Scene(createContent());
    primaryStage.setScene(scene);
    primaryStage.show();
}
```



【练习3-3d图形】

在案例3的基础上，再画一个小盒子，使之放置在大盒子上方，效果如图：

