**实 验 报 告**

**课程名称** 面向对象技术（Java）

**实验项目**  Java GUI综合编程

**实验仪器**  个人计算机 \_\_

**学 院** 计算机学院

**专 业** 计算机大类

**班级/学号** 计类1801/2018011140

**学生姓名**  张京宇

**实验日期** 2019/5/8

**指导教师 徐 莉**

## 实验项目四 Java GUI综合编程

**一、实验目的**

掌握开发Java GUI应用程序的方法，培养学生针对特定软件需求，完成可重用软件模块的分析、设计、测试的能力，提高学生将相关知识和对象描述方法用于分析复杂工程问题、并提出解决方案的能力。

**二、实验要求**

1. 上机实验前，按照每一个实验题目的具体要求提前完成准备工作，如：查找资料、设计算法、写出全部源程序，做好预习笔记，做到有准备地上机。
2. 上机实验课内，积极调试及修改程序。得到正确运行结果后，上传程序到过程化考核平台/http://211.68.40.252:8008/
3. 上机实验课后，学生应根据实验数据及结果，完成实验报告。

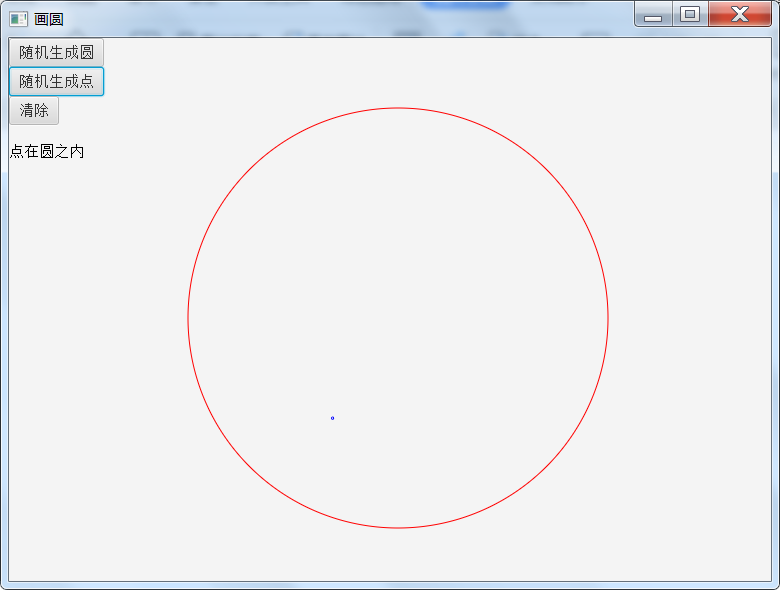
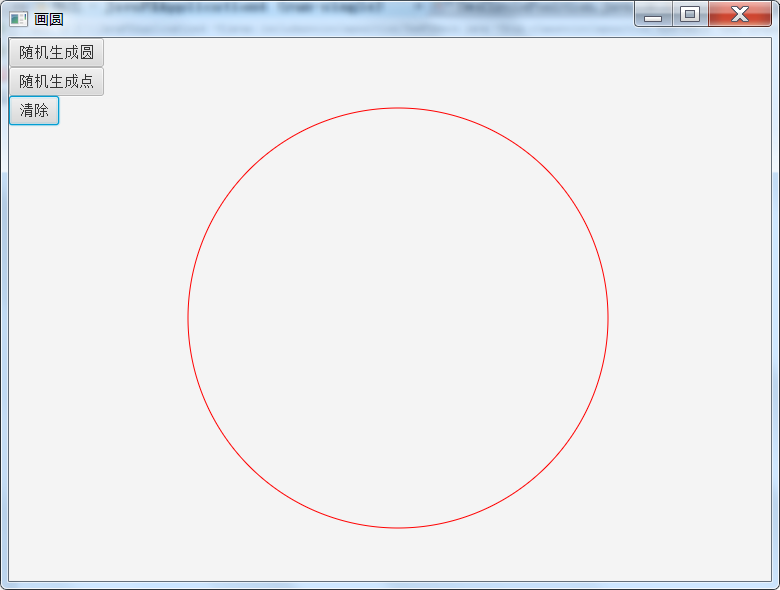
**三、实验设备**

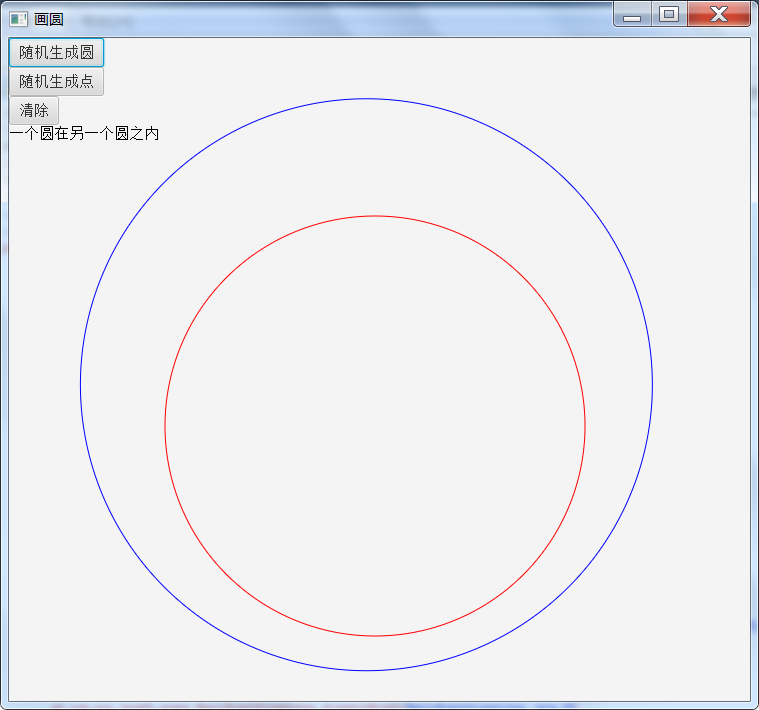
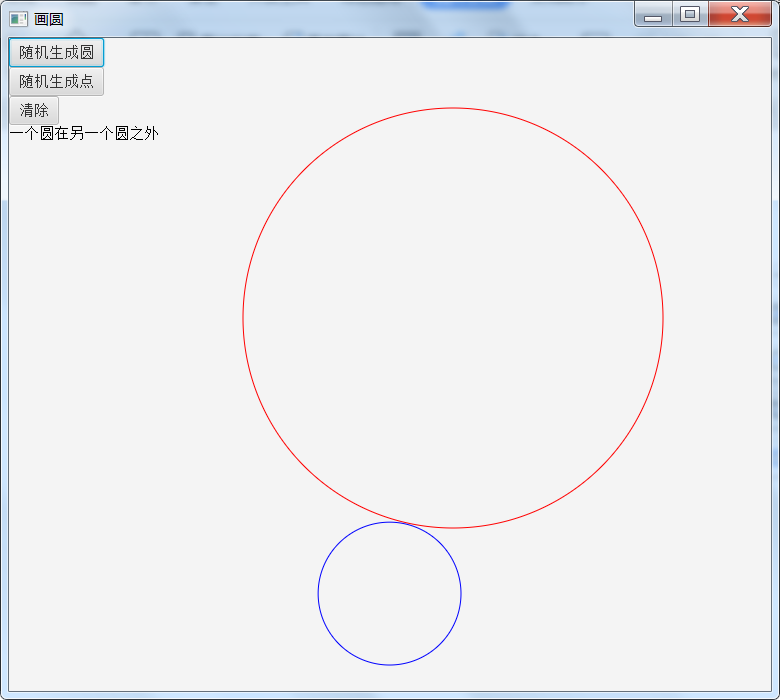
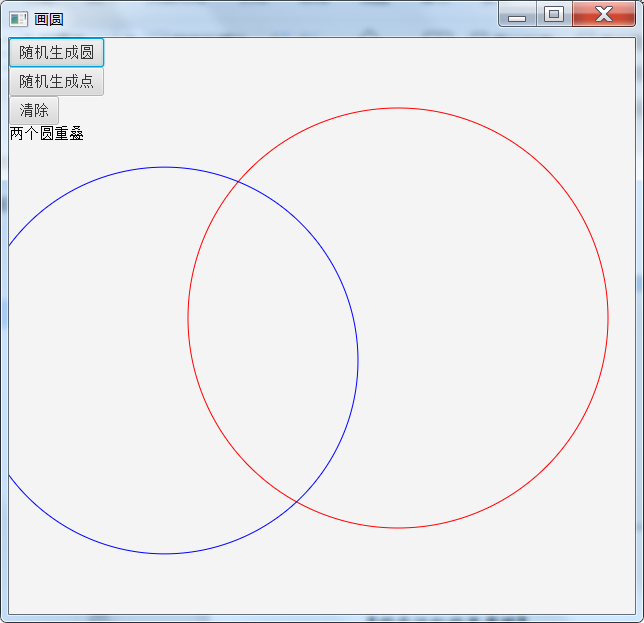
硬件环境：个人电脑一台

软件环境：JDK1.8及以上版本，NetBeans8.0及以上版本

**四、实验内容**

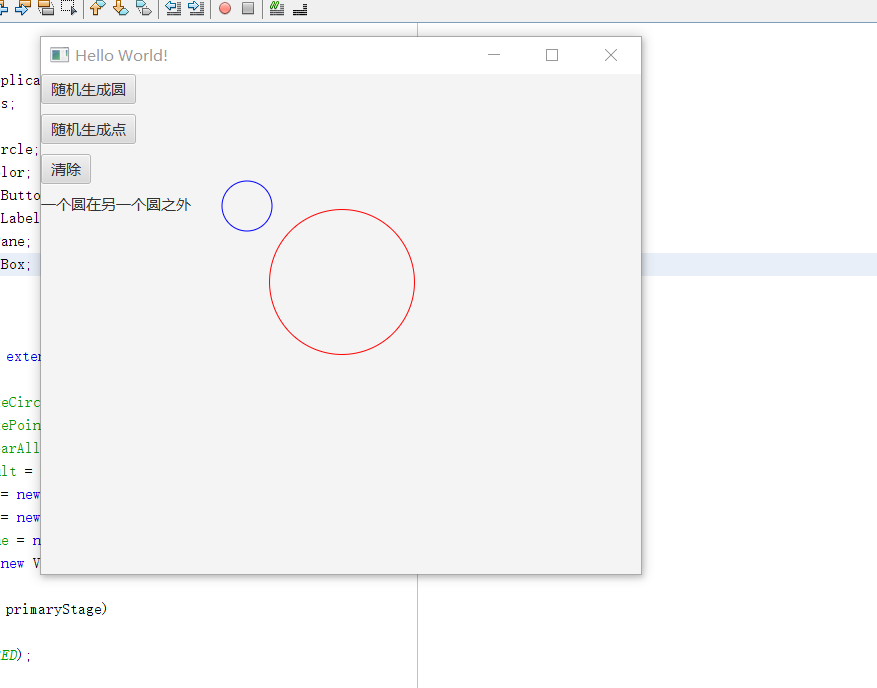
**任务1** 、（本题50分）设计一个GUI应用程序，先在GUI界面上通过鼠标拖动画出一个圆，然后再通过点击“随机生成圆”按钮，随机生成另一个圆，并在GUI界面上输出两个圆的位置关系（重叠，一个圆在另一个圆之内，一个圆在另一个圆之外）的信息，通过点击“随机生成点”按钮，随机生成一个点，并在GUI界面上输出点和是否在圆内的信息。参考界面如下所示。（提示设计的界面可以和参考界面不同）。





**【UML图】**

**【程序运行结果截图】**

****

****

**【程序结构与清单】**

package experiment4\_2

import javafx.application.Application;

import javafx.geometry.Insets;

import javafx.scene.Scene;

import javafx.scene.shape.Circle;

import javafx.scene.paint.Color;

import javafx.scene.control.Button;

import javafx.scene.control.Label;

import javafx.scene.layout.Pane;

import javafx.scene.layout.VBox;

import javafx.stage.Stage;

import java.util.Random;

public class Experiment4\_2 extends Application

{

private final Button makeCircle = new Button("随机生成圆");

private final Button makePoint = new Button("随机生成点");

private final Button clearAll = new Button("清除");

private final Label result = new Label("");

private final Pane pane = new Pane();

private final Circle c1 = new Circle();

private final Scene scene = new Scene(pane, 600, 500);

private final VBox v1 = new VBox(10);

@Override

public void start(Stage primaryStage)

{

c1.setStroke(Color.RED);

c1.setFill(null);

v1.getChildren().add(makeCircle);

v1.getChildren().add(makePoint);

v1.getChildren().add(clearAll);

v1.getChildren().add(result);

pane.setPadding(new Insets(10));

pane.getChildren().add(v1);

pane.getChildren().add(c1);

pane.setOnMousePressed(e -> {

c1.setCenterX(e.getSceneX());

c1.setCenterY(e.getSceneY());

c1.setRadius(0);

});

pane.setOnMouseDragged(e -> {

c1.setRadius(Math.pow((c1.getCenterX() - e.getSceneX())\*(c1.getCenterX() - e.getSceneX()) + (c1.getCenterY() - e.getSceneY())\*(c1.getCenterY() - e.getSceneY()), 0.5));

});

makeCircle.setOnAction(e -> drawCircle());

makePoint.setOnAction(e -> drawPoint());

clearAll.setOnAction(e -> clear());

primaryStage.setTitle("Hello World!");

primaryStage.setScene(scene);

primaryStage.show();

}

private void drawCircle()

{

clear();

double height = scene.getHeight();

double width = scene.getWidth();

int randomRadius = (int)(Math.random() \* ((height + width) / 4));

int randomX = (int)(Math.random() \* width);

int randomY = (int)(Math.random() \* height);

Circle c2 = new Circle(randomX, randomY, randomRadius);

c2.setStroke(Color.BLUE);

c2.setFill(null);

pane.getChildren().add(c2);

double distance;

distance = Math.sqrt((c2.getCenterX() - c1.getCenterX()) \* (c2.getCenterX() - c1.getCenterX()) + (c2.getCenterY() - c1.getCenterY()) \* (c2.getCenterY() - c1.getCenterY()));

if(c2.getRadius() + distance < c1.getRadius())

{

result.setText("一个圆在另一个圆之内");

}

else if(c1.getRadius() + c2.getRadius() <= distance)

{

result.setText("一个圆在另一个圆之外");

}

else

{

result.setText("两个圆重叠");

}

}

private void drawPoint()

{

clear();

double height = scene.getHeight();

double width = scene.getWidth();

int randomX = (int)(Math.random() \* width);

int randomY = (int)(Math.random() \* height);

Circle c2 = new Circle(randomX, randomY, 0.1);

c2.setStroke(Color.BLACK);

c2.setFill(null);

pane.getChildren().add(c2);

double distance;

distance = Math.sqrt((c2.getCenterX() - c1.getCenterX()) \* (c2.getCenterX() - c1.getCenterX()) + (c2.getCenterY() - c1.getCenterY()) \* (c2.getCenterY() - c1.getCenterY()));

if(c2.getRadius() + distance < c1.getRadius())

{

result.setText("点在圆之内");

}

else

{

result.setText("点在圆之外");

}

}

private void clear()

{

pane.getChildren().clear();

pane.getChildren().add(v1);

pane.getChildren().add(c1);

}

public static void main(String[] args)

{

launch(args);

}

}

**【任务1实验小结】**

主要谈谈你在该实验中用到了哪些具体技术或知识点？该技术或知识点解决了什麽问题？你的设计的优缺点？你在实验中遇到的问题，你是如何解决的？

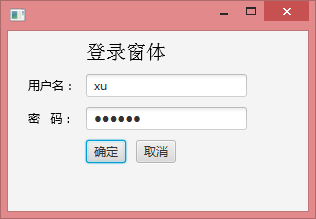
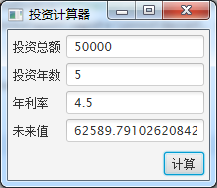
我在实验中采用了pane布局方法将三个按钮放置于左上角并利用三个按钮的处理方法实现生成随机圆、生成随机点、以及清除界面内容功能，然后通过判断方法判断点和圆、圆和圆的位置关系，以实现题目所预期的功能。在题目中我参考书上内容得到利用画笔画圆的方法，利用数学方法实现判断点和圆、圆和圆的位置关系，利用pane布局实现结点固定位置。

**任务2** 、（本题50分）设计一个GUI应用程序，启动程序，打开登录窗体，输入用户名与密码后按“确定”按钮，如果用户名与密码正确，打开投资计算器，否则提示“用户或密码错误”。

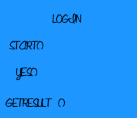
计算投资值在给定利率以及给定年数下的未来值。计算公式如下：

未来值=投资总额\*（1+年利率）年数

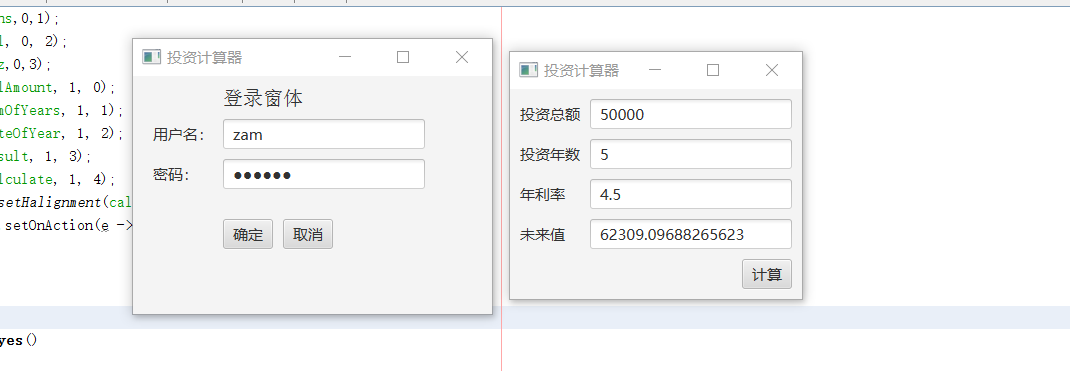
使用TextField显示利率、投资总额、年数，当用户点击“计算”按钮时在文本域显示未来值。

**【UML图】**



**【程序运行结果截图】**

****

**【程序结构与清单】**

package login;

import javafx.scene.control.PasswordField;

import javafx.scene.control.Label;

import javafx.scene.layout.HBox;

import javafx.scene.layout.GridPane;

import javafx.application.Application;

import javafx.geometry.HPos;

import javafx.geometry.Insets;

import javafx.scene.Scene;

import javafx.scene.control.Button;

import javafx.scene.control.TextField;

import javafx.scene.text.Font;

import javafx.scene.text.FontWeight;

import javafx.stage.Stage;

public class Login extends Application

{

private final GridPane w1 = new GridPane();

private final Scene scene1 = new Scene(w1);

private final Label title = new Label("登录窗体");

private final Label yhm = new Label("用户名：");

private final TextField userName = new TextField();

private final Label mm = new Label("密码：");

private final PasswordField password = new PasswordField();

private final Button confirm = new Button("确定");

private final Button cancel = new Button("取消");

private final GridPane w2 = new GridPane();

private final Scene scene2 = new Scene(w2);

private final Label tzze = new Label("投资总额");

private final TextField allAmount = new TextField();

private final Label tzns = new Label("投资年数");

private final TextField numOfYears = new TextField();

private final Label nll = new Label("年利率");

private final TextField rateOfYear = new TextField();

private final Label wlz = new Label("未来值");

private final TextField result = new TextField();

private final Button calculate = new Button("计算");

@Override

public void start(Stage primaryStage)

{

title.setFont(Font.font("Arial",FontWeight.BOLD,20));

HBox h1 = new HBox(10);

primaryStage.setScene(scene1);

primaryStage.show();

w1.setHgap(10);

w1.setVgap(10);

w1.add(title,2,1,3,1);

w1.setPadding(new Insets(0,10,0,10));

w1.add(yhm, 1, 2);

w1.add(userName,2,2);

w1.add(mm,1,3);

w1.add(password,2,3);

w1.add(h1,2,6,3,6);

h1.getChildren().add(confirm);

h1.getChildren().add(cancel);

confirm.setOnAction(e -> yes());

cancel.setOnAction(e -> primaryStage.close());

primaryStage.setTitle("投资计算器");

w2.setHgap(10);

w2.setVgap(10);

w2.setPadding(new Insets(10));

w2.add(tzze, 0, 0);

w2.add(tzns,0,1);

w2.add(nll, 0, 2);

w2.add(wlz,0,3);

w2.add(allAmount, 1, 0);

w2.add(numOfYears, 1, 1);

w2.add(rateOfYear, 1, 2);

w2.add(result, 1, 3);

w2.add(calculate, 1, 4);

GridPane.setHalignment(calculate,HPos.RIGHT);

calculate.setOnAction(e -> getResult());

}

private void yes()

{

if(userName.getText().equals("zam") && password.getText().equals("123456"))

{

Stage stage = new Stage();

stage.setTitle("投资计算器");

stage.setScene(scene2);

stage.show();

}

}

private void getResult()

{

double a = Double.parseDouble(allAmount.getText());

double n = Double.parseDouble(numOfYears.getText());

double r =Double.parseDouble(rateOfYear.getText());

double f;

f = a \* Math.pow((1 + (r \* 0.01)), n);

result.setText(String.valueOf(f));

}

public static void main(String[] args)

{

launch(args);

}

}

**【任务2实验小结】**

主要谈谈你在该实验中用到了哪些具体技术或知识点？该技术或知识点解决了什麽问题？你的设计的优缺点？你在实验中遇到的问题，你是如何解决的？

我在实验中创建了两个场景，利用gridpane布局方法分别将相应结点固定在对应位置，然后通过结点对应的实现方法实现判断、登录、计算等功能，以实现题目所预期的功能。在题目中我自定义了用户名和密码“zam”“123456”，输入正确后将弹出计算窗口，然后通过数学公式对未来值进行计算。

**五、实验总结**

通过本次实验，我基本掌握了开发Java GUI应用程序的方法，提升了我针对特定软件需求，完成可重用软件模块的分析、设计、测试的能力，提高了我将相关知识和对象描述方法用于分析复杂工程问题、并提出解决方案的能力，我体会到了面向对象技术的深层次意义，使我受益匪浅。