
廈門大學



信息学院软件工程系

《JAVA 程序设计》实验报告

实验 10

姓名：黄勛

学号：22920212204392

学院：信息学院

专业：软件工程

完成时间：2023.4.27

一、实验目的及要求

- 熟悉 JavaFX

二、实验题目及实现过程

实验环境：Windows 10 21H2、jdk17、javafx scene builder、utf-8 编码

题目一

(一) 实验题目

◆ 设计界面



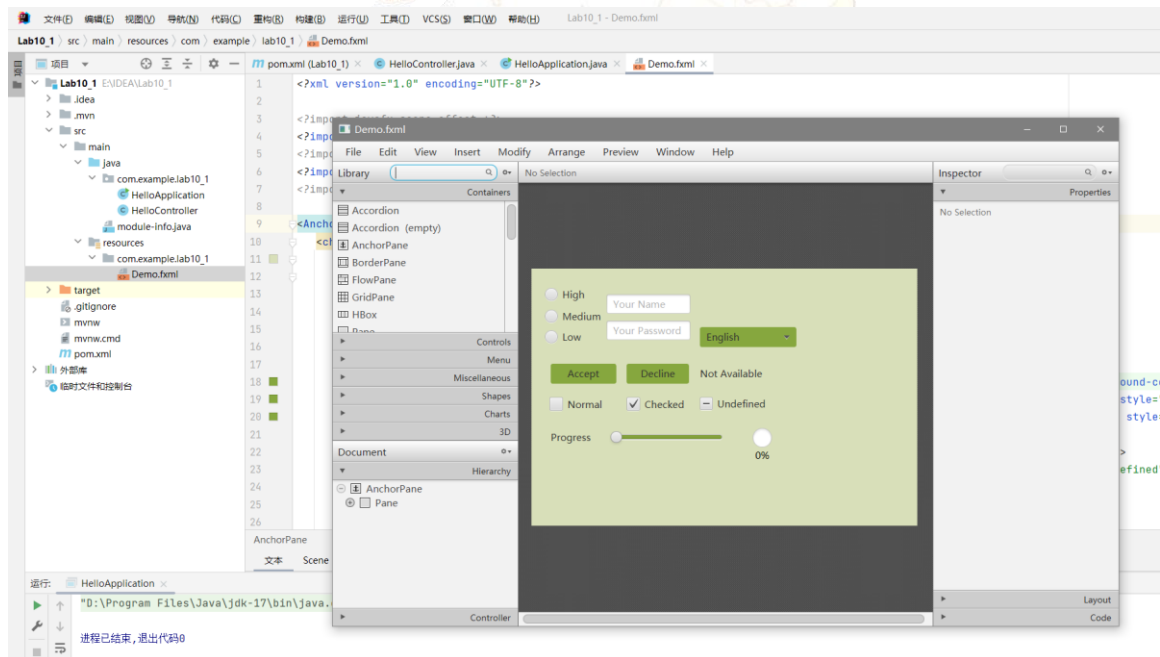
(二) 实现过程

思路：

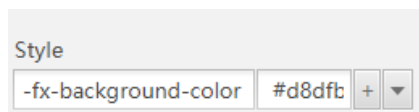
- 1、新建 JavaFx Project;
- 2、新建 fxml 文件，Open with SceneBuilder;



3、拖动组件，按照图片制作界面；



4、在 Properties->Style 中给组件修改颜色；



5、CheckBox->Specific 修改选择框。



5、将 Radio Button 组成组实现单选；

```

25 //单选按钮组
26 ToggleGroup size=new ToggleGroup();
27 RadioButton High=(RadioButton)root.lookup( s: "#High");
28 High.setToggleGroup(size);
29 RadioButton Medium=(RadioButton)root.lookup( s: "#Medium");
30 Medium.setToggleGroup(size);
31 RadioButton Low=(RadioButton)root.lookup( s: "#Low");
32 Low.setToggleGroup(size);

```

6、添加下拉选择项目；

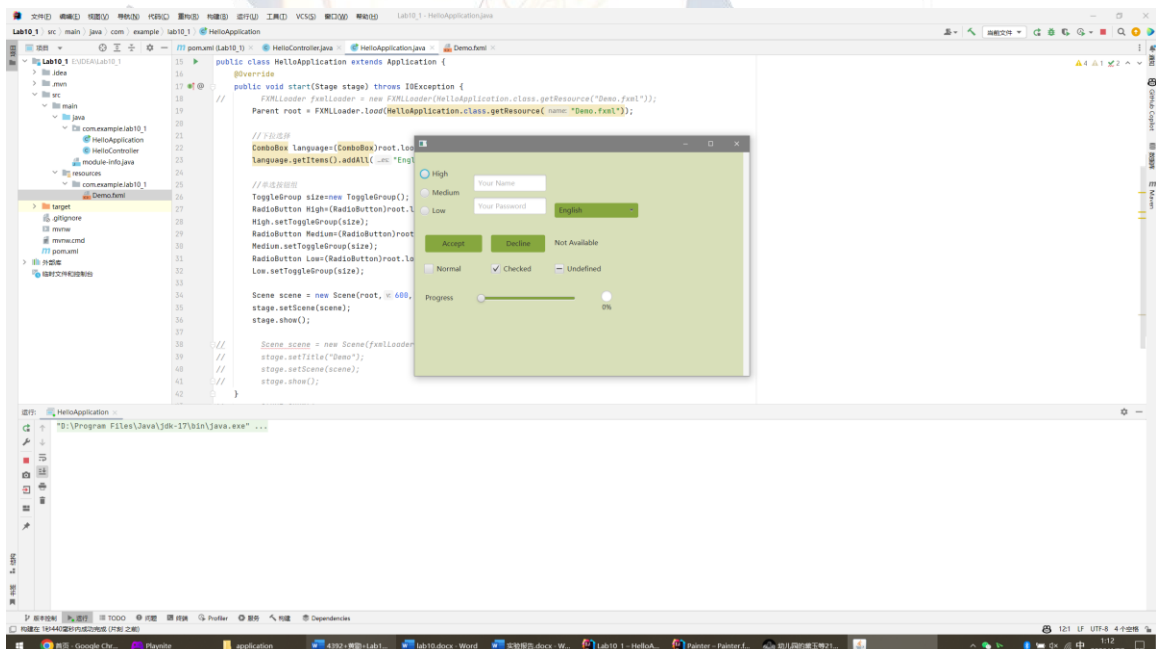
```

21 //下拉选择
22 ComboBox language=(ComboBox)root.lookup( s: "#Language");
23 language.getItems().addAll( ...es: "English", "中文", "Janpanese", "Korean");

```

(三) 过程截图

最终结果 (全屏截图)



题目二

(一) 实验题目

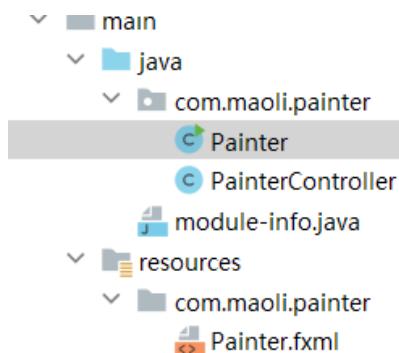
- ◆ 在 painter 例子基础上修改，让程序可以绘制矩形、圆形、直线，并且可以通过对话框修改颜色（可以参考例子 Color Chooser）。

(二) 实现过程

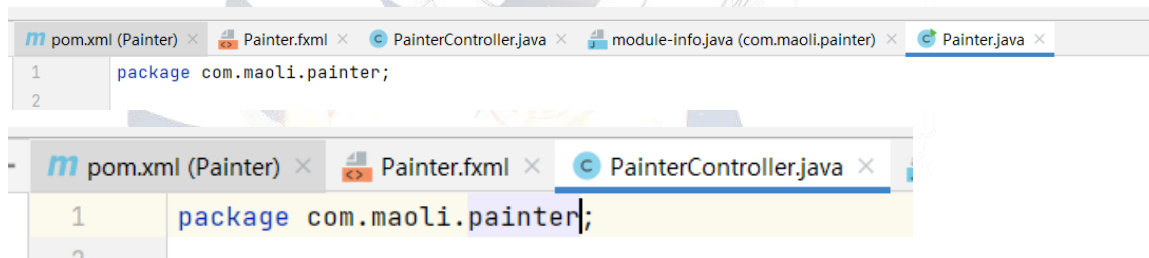
思路：1) 转移 painter 项目

1. 新建 javafx 工程

2. 将 painter 中的文件拷贝到这个工程里面，拷贝完成后目录结构如下：



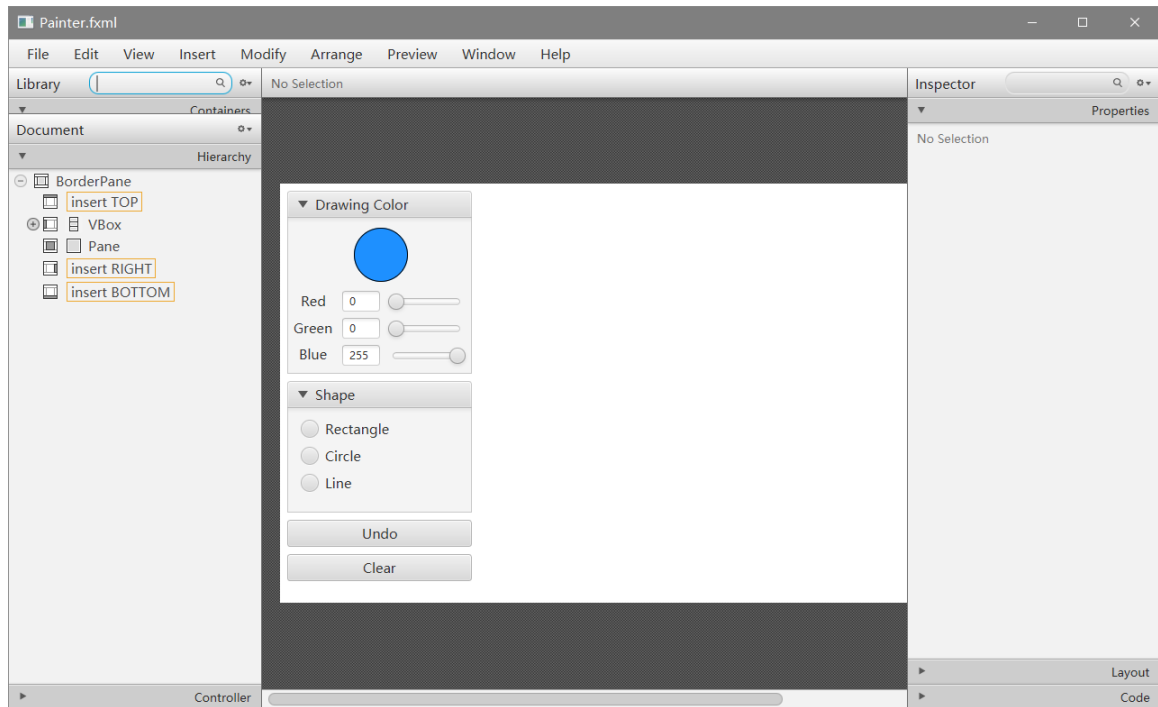
3. 检查两个类前面的 package 信息是否正确，若不正确修改成其所在的包



4. 检查 fxml 文件中：fx:controller="com.example.painter.PainterController"
这里的 controller 类的包名是否正确



2) 新建 fxml 文件，Open with SceneBuilder，拖动组件设计界面；



3) 拖动 Slider 改变颜色 (参考 colorchoose) ;

设置变化范围 RGB 值从 0~255;



TextField 显示 Slider 数据, 即 RGB 数值;

```
public void initialize() {
    // user data on a control can be any Object
    RedT.textProperty().bind(
        RedS.valueProperty().asString(s: "%.0f"));
    GreenT.textProperty().bind(
        GreenS.valueProperty().asString(s: "%.0f"));
    BlueT.textProperty().bind(
        BlueS.valueProperty().asString(s: "%.0f"));
}
```

给 Slider 添加 Listener 记录每次拖动数据, 上面的颜色展示圆圈随颜色选择变化变化而不断变化;

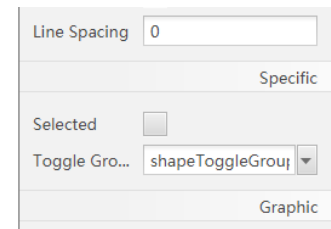
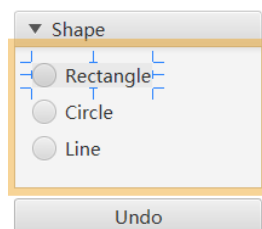
```

52      RedS.valueProperty().addListener(
53          new ChangeListener<Number>() {
54              @Override
55              public void changed(ObservableValue<? extends Number> ov,
56                  Number oldValue, Number newValue) {
57                  red = newValue.intValue();
58                  colorCircle.setFill(Color.rgb(red, green, blue, alpha));
59              }
60          });
61
62      GreenS.valueProperty().addListener(
63          new ChangeListener<Number>() {
64              @Override
65              public void changed(ObservableValue<? extends Number> ov,
66                  Number oldValue, Number newValue) {
67                  green = newValue.intValue();
68                  colorCircle.setFill(Color.rgb(red, green, blue, alpha));
69              }
70          });
71
72      BlueS.valueProperty().addListener(
73          new ChangeListener<Number>() {
74              @Override
75              public void changed(ObservableValue<? extends Number> ov,
76                  Number oldValue, Number newValue) {
77                  blue = newValue.intValue();
78                  colorCircle.setFill(Color.rgb(red, green, blue, alpha));
79              }
80          });
81

```

4) 选择形状在 Pane 上拖动画矩形、直线或圆形;

添加 toggle 使得只能选择一个图形



拖动绘制不同形状:

```

<Pane fx:id="drawingAreaPane" onMousePressed="#drawingAreaMousePress" onMouseReleased="#drawingMouseRelease" p

```

按下鼠标记录下第一个点坐标;

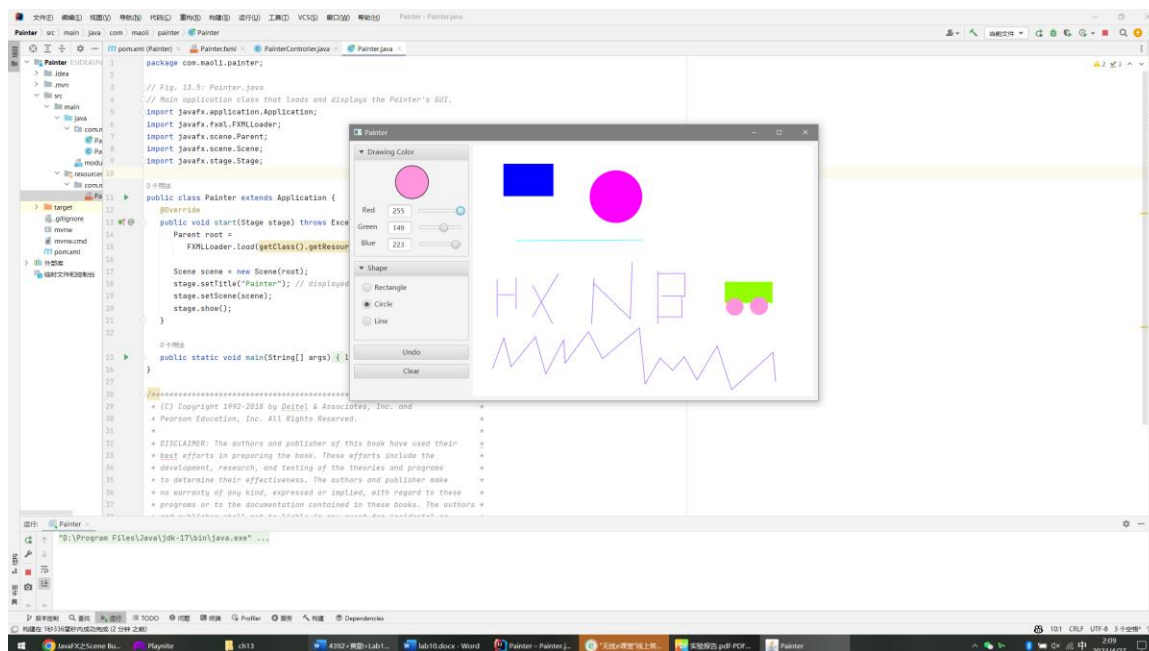
```
89  @ private void drawingAreaMousePress(MouseEvent e) {  
90      x1=e.getX();  
91      y1=e.getY();  
92  }
```

松开鼠标记录下第二个点坐标，通过 isSelected()判断哪个按钮被选择，就用相应的颜色绘制相应的图案；

```
95  private void drawingMouseRelease(MouseEvent e)  
96  {  
97  
98      if(RectRadioButton.isSelected())  
99      {  
100          x2=e.getX()-x1;  
101          y2=e.getY()-y1;  
102          Rectangle rect=new Rectangle(x1,y1,x2,y2);  
103          rect.setFill(Color.rgb(red, green, blue, alpha));  
104          drawingAreaPane.getChildren().add(rect);  
105      }  
106      if(LineRadioButton.isSelected())  
107      {  
108          x2=e.getX();  
109          y2=e.getY();  
110          Line line=new Line(x1,y1,x2,y2);  
111          line.setStroke(Color.rgb(red, green, blue, alpha));  
112          drawingAreaPane.getChildren().add(line);  
113      }  
114      if(CircleRadioButton.isSelected())  
115      {  
116          x2=e.getX()-x1;  
117          y2=e.getY()-y1;  
118          double r=Math.sqrt(x2*x2+y2*y2);  
119          Circle circle=new Circle();  
120          circle.setCenterX(x1);  
121          circle.setCenterY(y1);  
122          circle.setRadius(r);  
123          circle.setFill(Color.rgb(red, green, blue, alpha));  
124          drawingAreaPane.getChildren().add(circle);  
125      }
```

(三) 过程截图

最终结果（全屏截图）



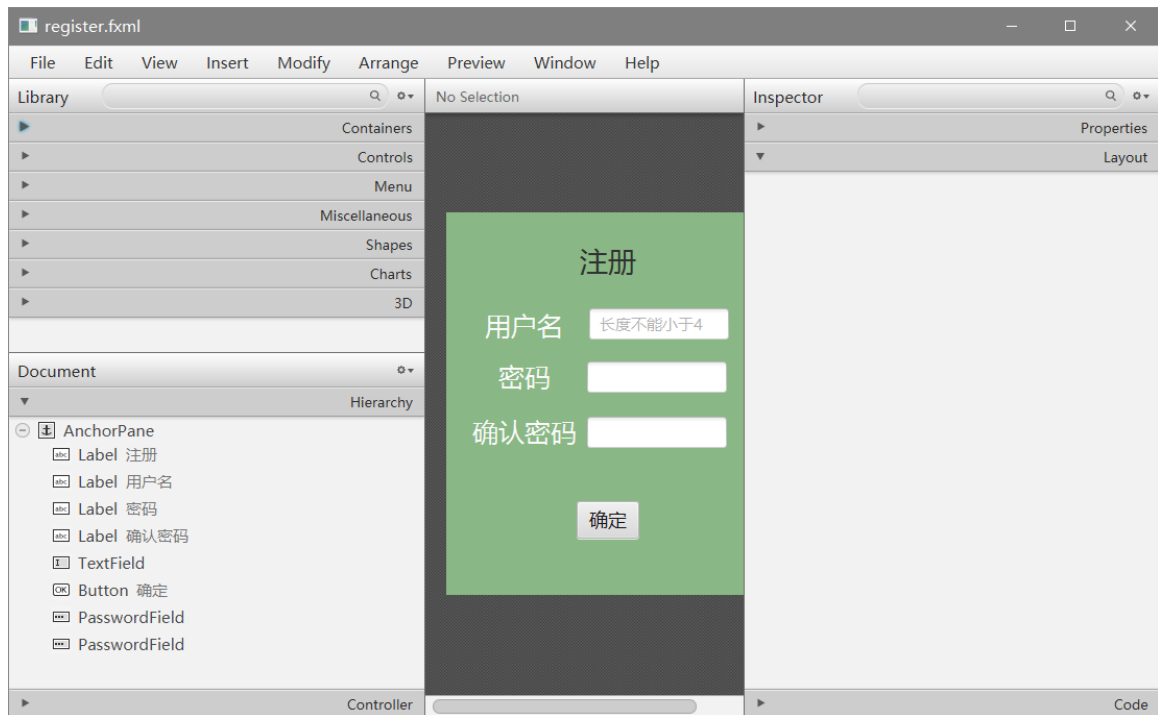
(拓展题目) 题目三

(一) 实验题目

- ◆ 实现一个注册界面，让用户输入用户名、密码、确认密码，其中用户名长度不能小于 4，两次密码必须相同。若输入不满足要求，当用户点击“确定”按钮时，提示相应错误。

(二) 实现过程

思路：1) Scene builder 完成界面设计



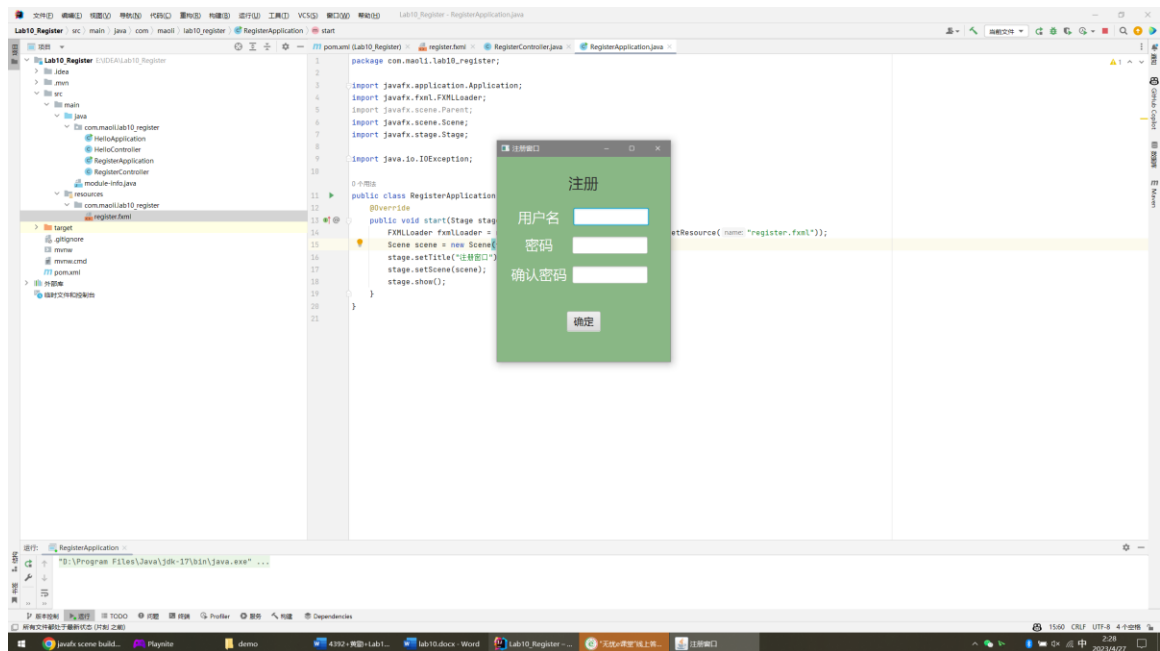
2) 在 controller 里编写对应的脚本。

RegisterController 类里的 click 方法是确认按钮对应的函数，获取用户的输入判断用户名和密码是否符合标准，如果有错误则弹出消息窗口提示错误，如果没有错误则注册窗口关闭，弹出消息窗口提示注册成功。

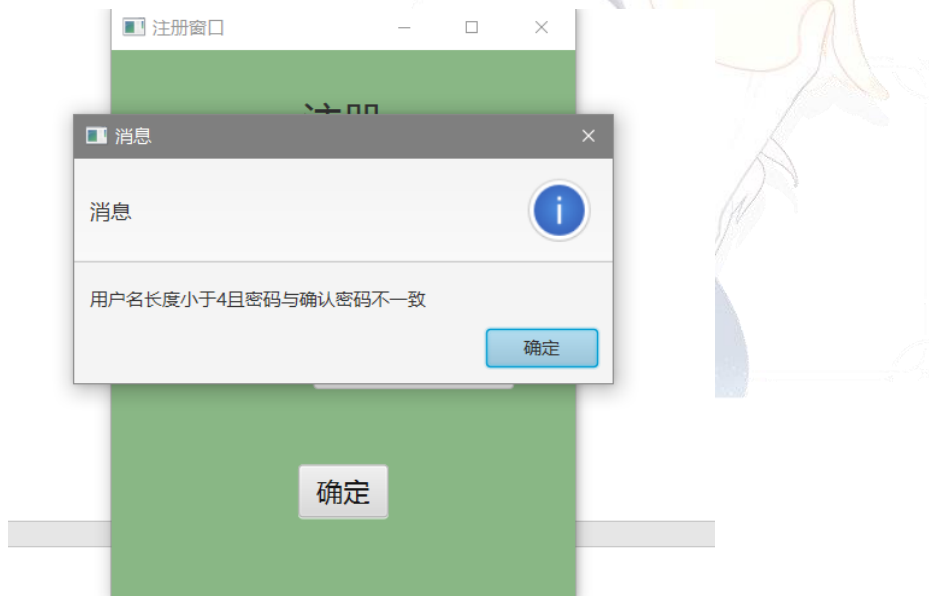
```
<Button fx:id="login" layoutX="110.0" layoutY="278.0" mnemonicParsing="false" onAction="#click" text="确定" textFill="#141414">
    <font>
        <Font size="18.0" />
    </font>
</Button>
```

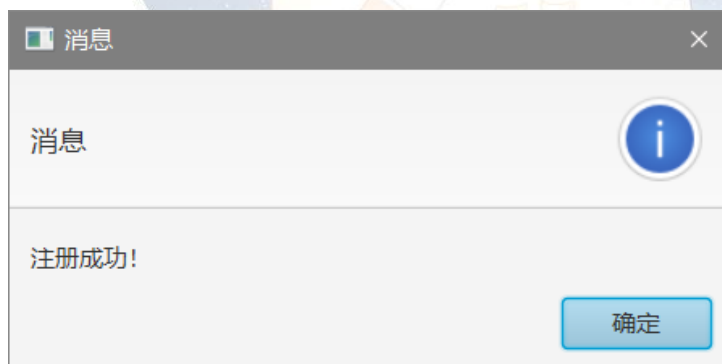
```
18 void click(ActionEvent event) {
19     String name = userName.getText();
20     String pass = password.getText();
21     String identify = passwordIdentify.getText();
22     if (name.length() < 4 && (!pass.equals(identify))) {
23         Alert alert = new Alert(Alert.AlertType.INFORMATION);
24         alert.setContentText("用户名长度小于4且密码与确认密码不一致");
25         alert.show();
26         return;
27     } else if (name.length() < 4) {
28         Alert alert = new Alert(Alert.AlertType.INFORMATION);
29         alert.setContentText("用户名长度小于4");
30         alert.show();
31         return;
32     } else if (!pass.equals(identify)) {
33         Alert alert = new Alert(Alert.AlertType.INFORMATION);
34         alert.setContentText("密码与确认密码不一致");
35         alert.show();
36         return;
37     } else {
38         Alert alert = new Alert(Alert.AlertType.INFORMATION);
39         alert.setContentText("注册成功! ");
40         alert.show();
41         Stage stage = (Stage) login.getScene().getWindow();
42         stage.close();
43         return;
44     }
```

(三) 过程截图



各种情况测试，均符合要求：





三、实验总结与心得记录

通过本次实验，我对 Java 使用 Scene builder 的使用有了更深入的了解。我学会了如何构建一个窗口等操作。在实验的过程中我为有功能的组件设置名字、编写了对应的方法，实现了需要的功能，这对我来说收获颇丰。

