

# 廈門大學



## 信息学院软件工程系

### 《JAVA 程序设计》实验报告

#### 实验十

姓名：陈澄

学号：32420212202930

学院：信息学院

专业：软件工程专业

完成时间：2023.04.25

## 一、实验目的及要求

### 实验目的：

- 熟悉 JavaFX

### 实验要求：

- 按照题目要求写代码和实验报告，并上传到 FTP

## 二、实验题目及实现过程

题目 1：设计界面：



（一）实验环境（集成开发环境、jdk 版本、字符编码等）

集成开发环境：IntelliJ IDEA

Jdk 版本：17.0.5

字符编码：ASCII

（二）实现过程（**本部分为主要评分依据**，请描述解题思路，比如总共设计几个类，各个类的用途、成员、主要方法等及其之间调用关系等）

1. 依次创建各个类型的变量，并根据题目描述添加相应描述。

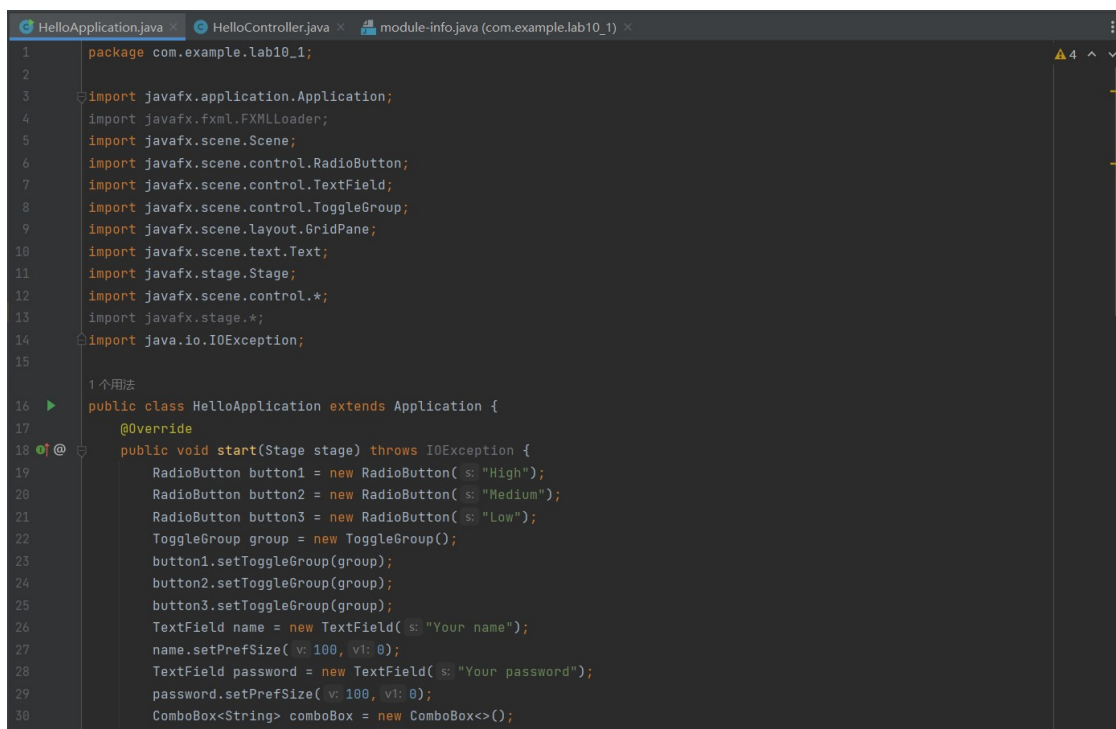
其中：三选按钮通过新建三个 `RadioButton` 后将其添加到同一个 `ToggleGroup` 来实现，即调用 `RadioButton` 中的 `setToggleGroup` 方法。输入框 `TextField`，勾选框 `CheckBox`，可以通过 `setAllowIndeterminate` 方法设置其是否可以为未定义状态，即图中 `Undefined`。Slider 滑动条，Text 文本。

2. 创建一个布局管理器 `GridPane`，并将按钮添加到其中。

3. 创建一个场景 `Scene`，并将布局管理器添加到其中。

4. 将场景设置到主舞台中，并显示舞台。

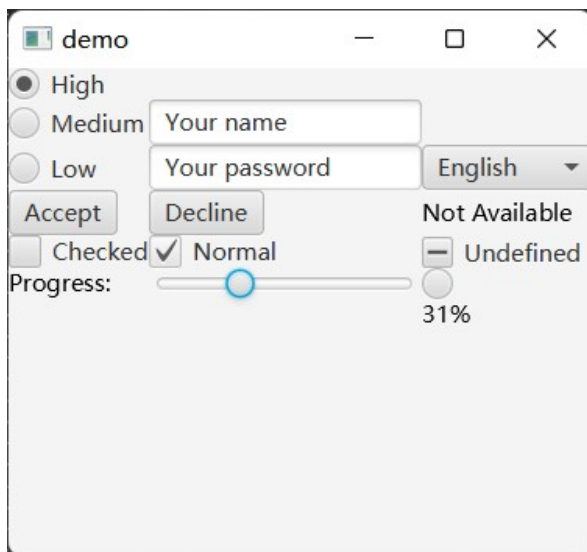
（三）过程截图（本部分为主要评分依据，一张全屏截图（必须）、若干运行结果展示图（可选），主要代码（可选））



```
1 package com.example.lab10_1;
2
3 import javafx.application.Application;
4 import javafx.fxml.FXMLLoader;
5 import javafx.scene.Scene;
6 import javafx.scene.control.RadioButton;
7 import javafx.scene.control.TextField;
8 import javafx.scene.control.ToggleGroup;
9 import javafx.scene.layout.GridPane;
10 import javafx.scene.text.Text;
11 import javafx.stage.Stage;
12 import javafx.scene.control.*;
13 import javafx.stage.*;
14 import java.io.IOException;
15
16 1 个用法
17 public class HelloApplication extends Application {
18     @Override
19     public void start(Stage stage) throws IOException {
20         RadioButton button1 = new RadioButton( s: "High");
21         RadioButton button2 = new RadioButton( s: "Medium");
22         RadioButton button3 = new RadioButton( s: "Low");
23         ToggleGroup group = new ToggleGroup();
24         button1.setToggleGroup(group);
25         button2.setToggleGroup(group);
26         button3.setToggleGroup(group);
27         TextField name = new TextField( s: "Your name");
28         name.setPrefSize( w: 100, v: 0);
29         TextField password = new TextField( s: "Your password");
30         password.setPrefSize( w: 100, v: 0);
31         ComboBox<String> comboBox = new ComboBox<>();
```

```
HelloApplication.java x HelloController.java x module-info.java (com.example.lab10_1) x
30 ComboBox<String> comboBox = new ComboBox<>();
31 comboBox.getItems().addAll(...es: "English", "Chinese");
32 Button accept = new Button(s: "Accept");
33 Button decline = new Button(s: "Decline");
34 Text text1 = new Text(s: "Not Available");
35 CheckBox checked = new CheckBox(s: "Checked");
36 CheckBox normal = new CheckBox(s: "Normal");
37 CheckBox undefined = new CheckBox(s: "Undefined");
38 undefined.setAllowIndeterminate(true);
39 Slider slider = new Slider();
40 Text text2 = new Text(s: "Progress:");
41 RadioButton button = new RadioButton();
42 slider.setMin(0);
43 slider.setMax(100);
44 slider.setValue(50);
45 Text text3 = new Text();
46 slider.valueProperty().addListener((observable, oldValue, newValue) -> {
47     text3.setText((int)(newValue.doubleValue())+"%");
48 });
49 // 创建一个布局管理器，并将按钮添加到其中
50 GridPane grid = new GridPane();
51 grid.add(button1, i: 0, i1: 0);
52 grid.add(button2, i: 0, i1: 1);
53 grid.add(button3, i: 0, i1: 2);
54 grid.add(name, i: 1, i1: 1);
55 grid.add(password, i: 1, i1: 2);
56 grid.add(comboBox, i: 2, i1: 2);
57 grid.add(accept, i: 0, i1: 3);
58 grid.add(decline, i: 1, i1: 3);
59 grid.add(text1, i: 2, i1: 3);
60 grid.add(checked, i: 0, i1: 4);
```

```
HelloApplication.java x HelloController.java x module-info.java (com.example.lab10_1) x
48     });
49     // 创建一个布局管理器，并将按钮添加到其中
50     GridPane grid = new GridPane();
51     grid.add(button1, i: 0, i1: 0);
52     grid.add(button2, i: 0, i1: 1);
53     grid.add(button3, i: 0, i1: 2);
54     grid.add(name, i: 1, i1: 1);
55     grid.add(password, i: 1, i1: 2);
56     grid.add(comboBox, i: 2, i1: 2);
57     grid.add(accept, i: 0, i1: 3);
58     grid.add(decline, i: 1, i1: 3);
59     grid.add(text1, i: 2, i1: 3);
60     grid.add(checked, i: 0, i1: 4);
61     grid.add(normal, i: 1, i1: 4);
62     grid.add(undefined, i: 2, i1: 4);
63     grid.add(slider, i: 1, i1: 5);
64     grid.add(text2, i: 0, i1: 5);
65     grid.add(button, i: 2, i1: 5);
66     grid.add(text3, i: 2, i1: 6);
67     // 创建一个场景，并将布局管理器添加到其中
68     Scene scene = new Scene(grid, w: 300, h: 250);
69     // 将场景设置到主舞台中，并显示舞台
70     stage.setTitle("demo");
71     stage.setScene(scene);
72     stage.show();
73 }
74
75 0 个用法
76 public static void main(String[] args) { launch(); }
77 }
```



题目 2: 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)。

(一) 实验环境 (集成开发环境、jdk 版本、字符编码等)

集成开发环境: IntelliJ IDEA

Jdk 版本: 17.0.5

字符编码: ASCII

(二) 实现过程 (本部分为主要评分依据, 请描述解题思路, 比如总共设计几个类, 各个类的用途、成员、主要方法等及其之间调用关系等)

1. 打开 fxml 文件, 在原有文件基础上, 添加一个 TitlePane, 在其下添加一个 VBox, 再在其下添加四个 RadioButton(Line,Rectangle,Cicle,Point), 创建一个名为 typrGroup 的 ToggleGroup, 将四个按钮全添加到这个 ToggleGroup 中, 以实现只选中一个的功能。在 AnchorPane 组件中, 添加 On Mouse Pressed 的相应函数 drawingAreaMousePressed, 和 On Mouse Released 的相应函数 drawingAreaMouseReleased。
2. 打开 PainterController, 依次添加上述变量以及函数。新建四个 double 型变量 x1,x2,y1,y2 用于获取鼠标按下和松开时的坐标 (后续会以此坐标来绘制各种图形)。
3. 在原有的 initialize 方法中添加一个 typeGroup 的初始化, 通过 selectToggle 将其默认按钮选择为 pointButton, 即不绘制任何图形。
4. 在原有的 drawingAreaMouseDragged 方法下添加一个 Tooggle 型的变量 selectedType, 该变量通过 TooggleGroup 下的方法 getSelectedToggle 来获取选中的按钮并进行判断。在此处添加一条选中按钮为 pointButton 的判断。
5. 创建新方法 drawingAreaMousePressed, 该方法会在鼠标点击时获取点击事件, 该方法下根据点击位置获取点击坐标 x1,y1 即可。
6. 创建新方法 drawingAreaMouseReleased, 该方法会在鼠标松开时获取松开事件, 在该方法下实现其他三个选项的绘制, 即鼠标松开时绘制图形。

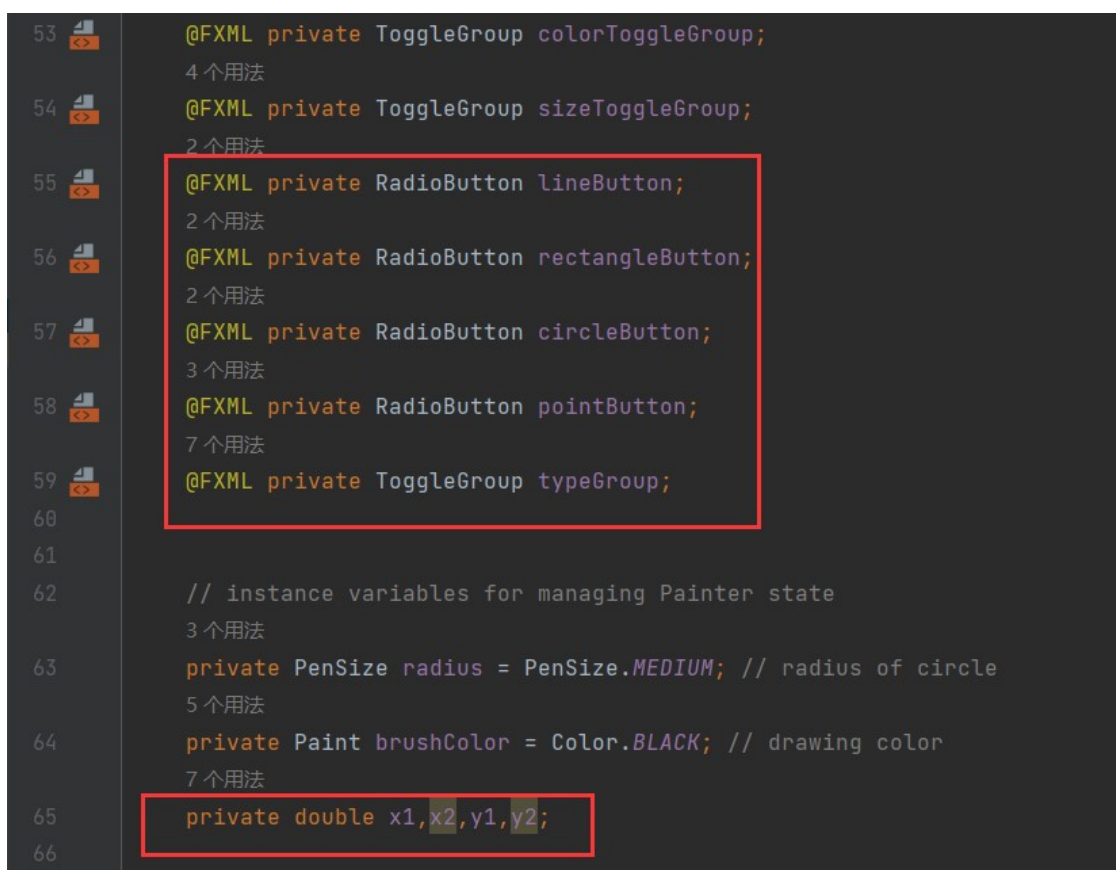
同样先获取坐标  $x_2, y_2$ ，再创建 `selectedType`（同 4）进行判断按钮。

圆：将鼠标按下的坐标和松开坐标的连线作为直径绘制。将其添加到 `drawingAreaAnchorPane` 中。

矩形：将鼠标按下位置作为矩形左上角，松开位置作为矩形右上角。

直线：按下位置作为起点，松开位置作为终点绘制。

（三）过程截图（**本部分为主要评分依据**，一张全屏截图（必须）、若干运行结果展示图（可选），主要代码（可选））



```
53  @FXML private ToggleGroup colorToggleGroup;
    4 个用法
54  @FXML private ToggleGroup sizeToggleGroup;
    2 个用法
55  @FXML private RadioButton lineButton;
    2 个用法
56  @FXML private RadioButton rectangleButton;
    2 个用法
57  @FXML private RadioButton circleButton;
    3 个用法
58  @FXML private RadioButton pointButton;
    7 个用法
59  @FXML private ToggleGroup typeGroup;
60
61
62  // instance variables for managing Painter state
    3 个用法
63  private PenSize radius = PenSize.MEDIUM; // radius of circle
    5 个用法
64  private Paint brushColor = Color.BLACK; // drawing color
    7 个用法
65  private double x1, x2, y1, y2;
66
```



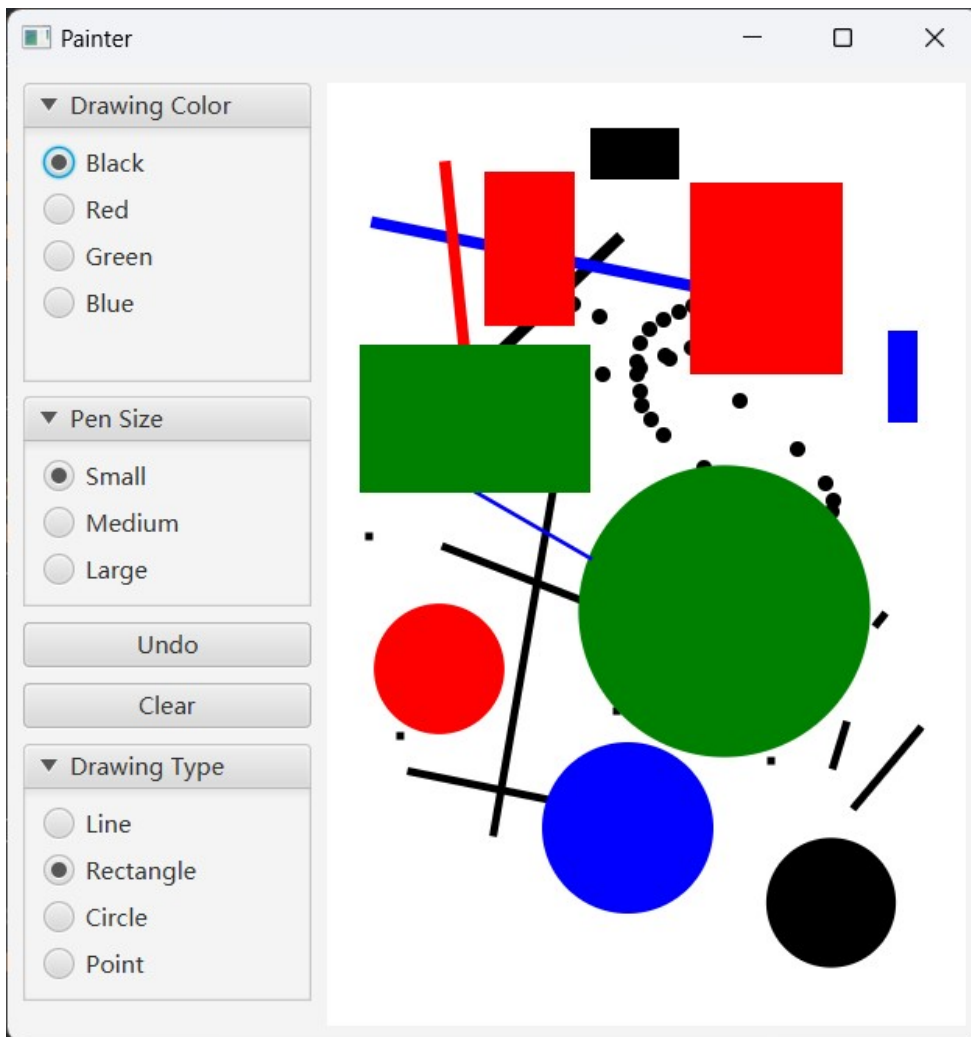
```
68     public void initialize()
69     {
70         // user data on a control can be any Object
71         blackRadioButton.setUserData(Color.BLACK);
72         redRadioButton.setUserData(Color.RED);
73         greenRadioButton.setUserData(Color.GREEN);
74         blueRadioButton.setUserData(Color.BLUE);
75         smallRadioButton.setUserData(PenSize.SMALL);
76         mediumRadioButton.setUserData(PenSize.MEDIUM);
77         largeRadioButton.setUserData(PenSize.LARGE);
78         typeGroup.selectToggle(pointButton);
79     }
80
```

```
82     @FXML
83     private void drawingAreaMouseDragged(MouseEvent e)
84     {
85         Toggle selectedType = typeGroup.getSelectedToggle();
86         if(selectedType==pointButton){
87             Circle newCircle = new Circle(e.getX(), e.getY(), radius.getRadius(), brushColor);
88             drawingAreaAnchorPane.getChildren().add(newCircle);
89         }
90     }
91
```

```
1 个用法
92     @FXML
93     @
94     public void drawingAreaMousePressed(MouseEvent e){
95         x1=e.getX();
96         y1=e.getY();
97     }
98     1 个用法
```



```
97 @FXML
98 public void drawingAreaMouseReleased(MouseEvent e){
99     Toggle selectedType = typeGroup.getSelectedToggle();
100     x2=e.getX();
101     y2=e.getY();
102     if(selectedType==circleButton){
103         Circle newCircle = new Circle( v: (x1+x2)/2, v1: (y1+y2)/2, v2: Math.sqrt((x2-x1)*(x2-x1)+(y2-y1)*(y2-y1))/2, brushColor);
104         drawingAreaAnchorPane.getChildren().add(newCircle);
105     }
106     if(selectedType==rectangleButton){
107         Rectangle newRectangle = new Rectangle( v: 0, v1: 0, brushColor);
108         newRectangle.setX(x1);
109         newRectangle.setY(y1);
110         newRectangle.setHeight(y2-y1);
111         newRectangle.setWidth(x2-x1);
112         drawingAreaAnchorPane.getChildren().add(newRectangle);
113     }
114     if(selectedType==lineButton){
115         Line newLine = new Line(x1,y1,x2,y2);
116         newLine.setStroke(brushColor);
117         newLine.setStrokeWidth(radius.getRadius());
118         drawingAreaAnchorPane.getChildren().add(newLine);
119     }
120 }
```



题目 3：为博饼程序设计一个图形界面。

（一）实验环境（集成开发环境、jdk 版本、字符编码等）

集成开发环境：IntelliJ IDEA

Jdk 版本：17.0.5

字符编码：ASCII

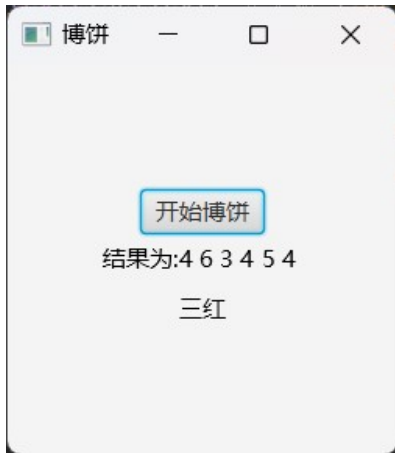
（二）实现过程（**本部分为主要评分依据**，请描述解题思路，比如总共设计几个类，各个类的用途、成员、主要方法等及其之间调用关系等）

1. 新建一个 Button 名为 start 用于开始博饼，建立两个 Text 组件用于输出博饼结果。
2. 通过 Button 下的 setOnAction 方法，获取按钮的点击事件，在该方法中创建 int 数组用于博饼结果的储存，循环产生随机数并输出为第一个文本，再根据数组中 4 的个数判断奖项输出为第二个文本。
3. 增加一个组件管理器 StackPane 将上述组件添加进去，创建一个 Scene 将 Stackpane 添加进去，展示主舞台。

（三）过程截图（**本部分为主要评分依据**，一张全屏截图（必须）、若干运行结果展示图（可选），主要代码（可选））

```
m pom.xml (Lab10_3_1) x hello-view.fxml x Bobing.java x BobingController.java x
1 package com.example.lab10_3_1;
2 import javafx.application.Application;
3 import javafx.scene.Scene;
4 import javafx.scene.control.Button;
5 import javafx.scene.image.Image;
6 import javafx.scene.image.ImageView;
7 import javafx.scene.layout.StackPane;
8 import javafx.scene.text.Text;
9 import javafx.stage.Stage;
10 import javafx.geometry.Insets;
11 import java.io.IOException;
12 import java.util.Random;
13
14 1 个用法
15 public class Bobing extends Application {
16     @Override
17     public void start(Stage stage) throws IOException {
18         Button start = new Button("开始博饼");
19         Text text1 = new Text();
20         Text text2 = new Text();
21         start.setOnAction(event -> {
22             String result="结果为:";
23             Random random = new Random();
24             int a[]=new int[6];
25             int count=0;
26             for(int n:a){
27                 n=random.nextInt( bound: 6)+1;
28                 result+=n+" ";
29                 if(n==4)count++;
30             }
31             text1.setText(result);
32         });
33     }
34 }
```

```
m pom.xml (Lab10_3_1) x hello-view.fxml x Bobing.java x BobingController.java x
28         if(n==4)count++;
29     }
30     text1.setText(result);
31     switch(count){
32         case 0:text2.setText("无奖项");break;
33         case 1:text2.setText("一秀");break;
34         case 2:text2.setText("二举");break;
35         case 3:text2.setText("三红");break;
36         case 4:text2.setText("状元");break;
37     }
38 });
39 StackPane pane = new StackPane();
40 pane.getChildren().add(start);
41 pane.getChildren().add(text1);
42 pane.getChildren().add(text2);
43 StackPane.setMargin(start, new Insets( w: -50, v1: 0, v2: 0, v3: 0));
44 StackPane.setMargin(text2, new Insets( w: 50, v1: 0, v2: 0, v3: 0));
45 Scene scene = new Scene(pane, w: 200, v1: 200 );
46 stage.setTitle("博饼");
47 stage.setScene(scene);
48 stage.show();
49 }
50
51 0 个用法
52 public static void main(String[] args) { launch(); }
53 }
```



题目 4：写一个程序，点击界面上一个按钮后弹出多种对话框。

（一）实验环境（集成开发环境、jdk 版本、字符编码等）

集成开发环境：IntelliJ IDEA

Jdk 版本：17.0.5

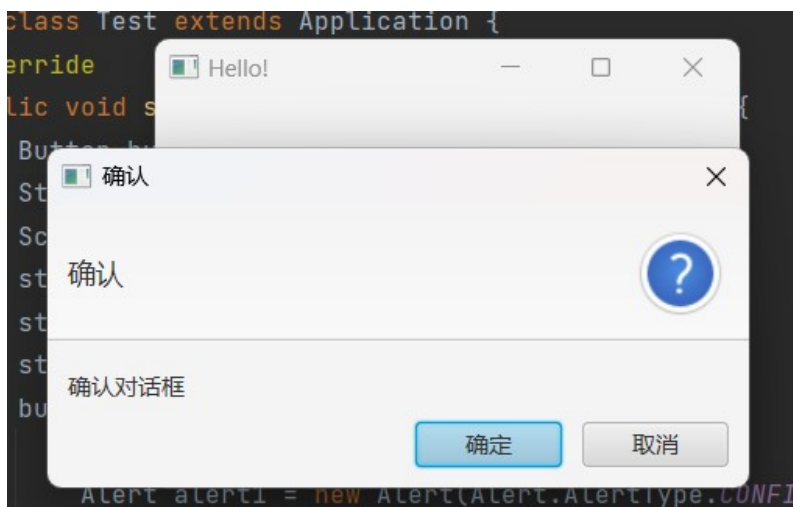
字符编码：ASCII

（二）实现过程（**本部分为主要评分依据**，请描述解题思路，比如总共设计几个类，各个类的用途、成员、主要方法等及其之间调用关系等）

1. 创建按钮，添加相应事件等等与上题相同不再赘述。
2. 对话框即为 Alert 组件，通过其构造方法以及 AlertType 方法创建不同类型的对话框。再通过 showAndWait 方法弹出。

（三）过程截图（**本部分为主要评分依据**，一张全屏截图（必须）、若干运行结果展示图（可选），主要代码（可选））

```
pom.xml (Lab10_3_2) × hello-view.fxml × module-info.java (com.example.lab10_3_2) × Test.java ×
7 import java.io.IOException;
8 import javafx.scene.layout.StackPane;
9
10 1 个用法
11 public class Test extends Application {
12     @Override
13     public void start(Stage stage) throws IOException {
14         Button button = new Button( s: "显示对话框");
15         StackPane pane = new StackPane(button);
16         Scene scene = new Scene(pane, w: 300, h: 200);
17         stage.setTitle("Hello!");
18         stage.setScene(scene);
19         stage.show();
20         button.setOnAction(event -> {
21             // 创建多个对话框，并显示出来
22             Alert alert1 = new Alert(Alert.AlertType.CONFIRMATION, s: "确认对话框");
23             alert1.showAndWait();
24             Alert alert2 = new Alert(Alert.AlertType.INFORMATION, s: "信息对话框");
25             alert2.showAndWait();
26             Alert alert3 = new Alert(Alert.AlertType.ERROR, s: "错误对话框");
27             alert3.showAndWait();
28             Alert alert4 = new Alert(Alert.AlertType.WARNING, s: "警告对话框");
29             alert4.showAndWait();
30             Alert alert5 = new Alert(Alert.AlertType.NONE, s: "空对话框");
31             alert5.showAndWait();
32         });
33     }
34
35     0 个用法
36     public static void main(String[] args) {
37         launch();
38     }
39 }
```



题目 5：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。

（一）实验环境（集成开发环境、jdk 版本、字符编码等）

集成开发环境：IntelliJ IDEA

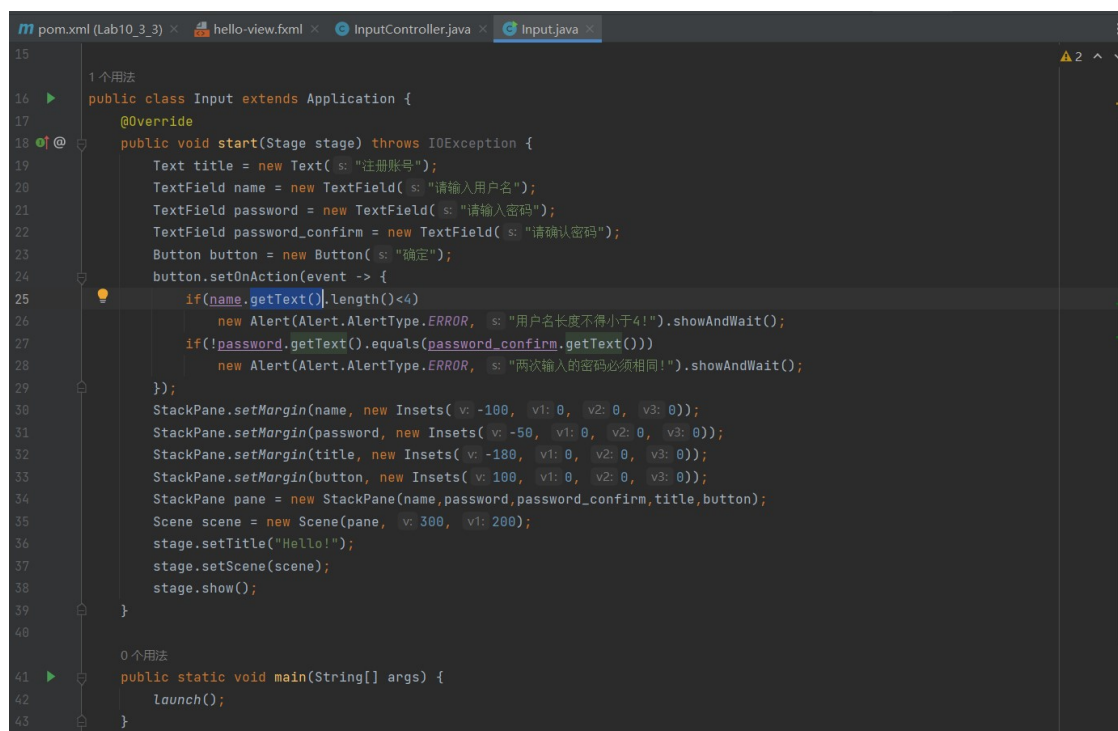
Jdk 版本: 17.0.5

字符编码: ASCII

(二) 实现过程 (本部分为主要评分依据, 请描述解题思路, 比如总共设计几个类, 各个类的用途、成员、主要方法等及其之间调用关系等)

1. 新建组件 TextField, 即输入框。新建一个按钮并添加响应事件。
2. 在按钮响应事件中通过 `getText()` 方法对输入框中的字符串进行判断, 若不符合要求则 `new` 创建一个弹窗提示错误。

(三) 过程截图 (本部分为主要评分依据, 一张全屏截图 (必须)、若干运行结果展示图 (可选), 主要代码 (可选))



```
15  
16 1 个用法  
17 public class Input extends Application {  
18     @Override  
19     public void start(Stage stage) throws IOException {  
20         Text title = new Text(s: "注册账号");  
21         TextField name = new TextField(s: "请输入用户名");  
22         TextField password = new TextField(s: "请输入密码");  
23         TextField password_confirm = new TextField(s: "请确认密码");  
24         Button button = new Button(s: "确定");  
25         button.setOnAction(event -> {  
26             if(name.getText().length()<4)  
27                 new Alert(Alert.AlertType.ERROR, s: "用户名长度不得小于4!").showAndWait();  
28             if(!password.getText().equals(password_confirm.getText()))  
29                 new Alert(Alert.AlertType.ERROR, s: "两次输入的密码必须相同!").showAndWait();  
30         });  
31         StackPane.setMargin(name, new Insets(v: -100, v1: 0, v2: 0, v3: 0));  
32         StackPane.setMargin(password, new Insets(v: -50, v1: 0, v2: 0, v3: 0));  
33         StackPane.setMargin(title, new Insets(v: -100, v1: 0, v2: 0, v3: 0));  
34         StackPane.setMargin(button, new Insets(v: 100, v1: 0, v2: 0, v3: 0));  
35         StackPane pane = new StackPane(name, password, password_confirm, title, button);  
36         Scene scene = new Scene(pane, v: 300, v1: 200);  
37         stage.setTitle("Hello!");  
38         stage.setScene(scene);  
39         stage.show();  
40     }  
41 0 个用法  
42 public static void main(String[] args) {  
43     launch();  
44 }
```



### 三、实验总结与心得记录

本部分根据实验过程的所得所想描述，记录可供以后复习回看 {可以记录调试过程遇到的问题，自己哪些知识点掌握不够，设计是否有缺陷（比如耗时？耗内存？）是否有亮点，是否有精妙的算法，或者设计模式的应用，可吐槽，也可与其他语言作适当对比。}（本部分不作为平时评分依据）

**备注：**

建议附带代码提交的方式：导出工程压缩包。

平时实验成绩以考查参与度为主，所有实验要求自己完成，一旦发现抄袭或者其他投机取巧，取消所有平时成绩