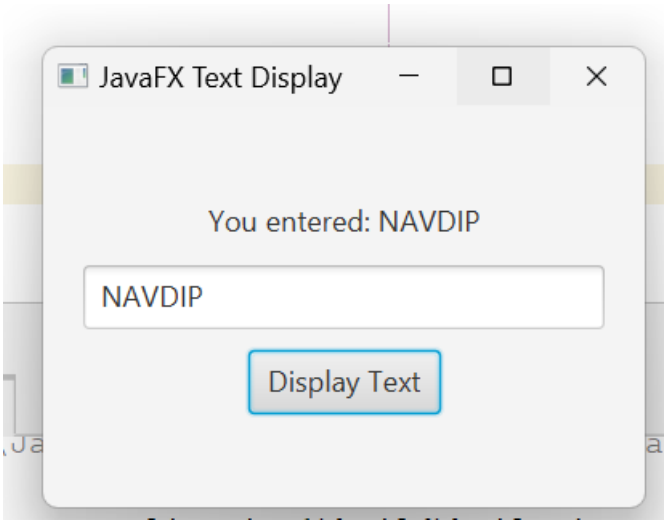
LAB Assignment – 6

Subject : Programming with Java

.

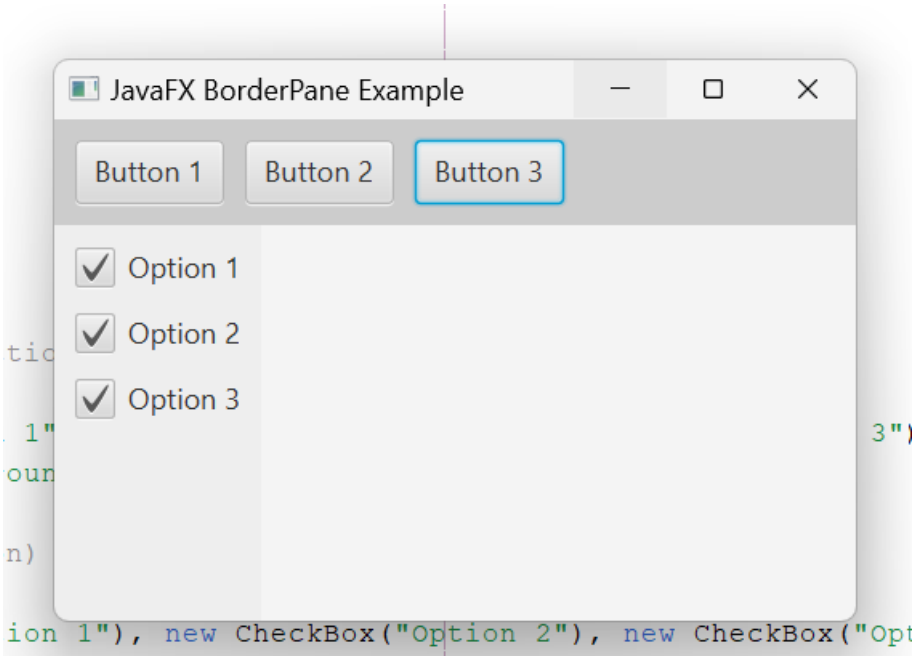
1. Create a JavaFX application with a Label, a Button, and a TextField. When the button is clicked, get the text from the TextField and display it in the Label.

```
import javafx.application.Application;
import javafx.scene.Scene;
import javafx.scene.control.*;
import javafx.scene.layout.VBox;
import javafx.stage.Stage;

public class JavaFXTextDisplay extends Application {
    @Override
    public void start(Stage primaryStage) {
        Label label = new Label("Enter text below:");
        TextField textField = new TextField();
        Button button = new Button("Display Text");

        button.setOnAction(e -> {
            String text = textField.getText();
            label.setText("You entered: " + text);
        });

        VBox layout = new VBox(10, label, textField, button);
        layout.setStyle("-fx-padding: 20px; -fx-alignment: center;");

        Scene scene = new Scene(layout, 300, 200);
        primaryStage.setTitle("JavaFX Text Display");
        primaryStage.setScene(scene);
        primaryStage.show();
    }

    public static void main(String[] args) {
        launch(args);
    }
}
```



2. Create a layout using a BorderPane. Place an HBox at the top with three buttons. Place a VBox on the left with a list of CheckBoxes.

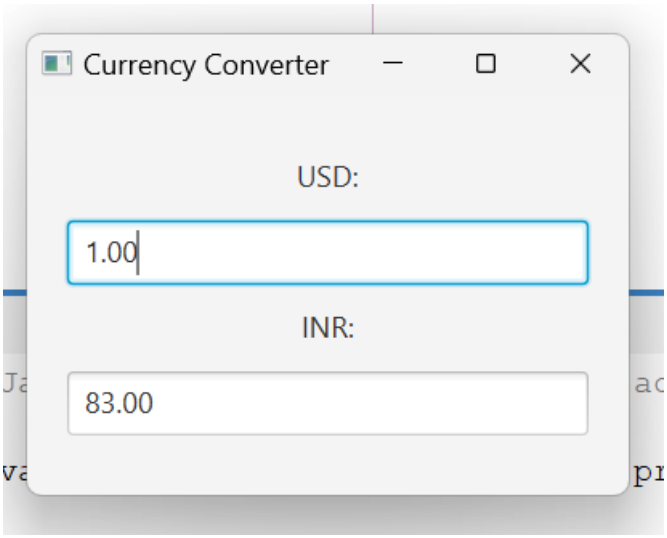
```
import javafx.application.Application;
import javafx.scene.Scene;
import javafx.scene.control.*;
import javafx.scene.layout.*;
import javafx.stage.Stage;

public class SimpleBorderPane extends Application {
    @Override
    public void start(Stage stage) {
        HBox top = new HBox(10, new Button("Button1"), new Button("Button2"),
new Button("Button3"));
        VBox left = new VBox(10, new CheckBox("Option1"), new
CheckBox("Option2"), new CheckBox("Option3"));

        BorderPane layout = new BorderPane();
        layout.setTop(top);
        layout.setLeft(left);

        stage.setScene(new Scene(layout, 300, 200));
        stage.setTitle("BorderPane Layout");
        stage.show();
    }

    public static void main(String[] args) {
        launch();
    }
}
```



3. Create two TextFields for entering amounts in two different currencies. Use property binding to automatically update the other TextField when one is changed.

```
import javafx.application.Application;
import javafx.scene.Scene;
import javafx.scene.control.Label;
import javafx.scene.control.TextField;
import javafx.scene.layout.VBox;
import javafx.stage.Stage;

public class SimpleCurrencyConverter extends Application {
    @Override
    public void start(Stage stage) {
        TextField usdField = new TextField();
        TextField inrField = new TextField();
        double rate = 83.0; // 1 USD = 83 INR

        usdField.textProperty().addListener((o, oldVal, newVal) -> {
            try { inrField.setText(String.format("%.2f",
Double.parseDouble(newVal) * rate)); }
            catch (NumberFormatException e) {}
        });

        inrField.textProperty().addListener((o, oldVal, newVal) -> {
            try { usdField.setText(String.format("%.2f",
Double.parseDouble(newVal) / rate)); }
            catch (NumberFormatException e) {}
        });

        VBox root = new VBox(10, new Label("USD:"), usdField, new
Label("INR:"), inrField);
```

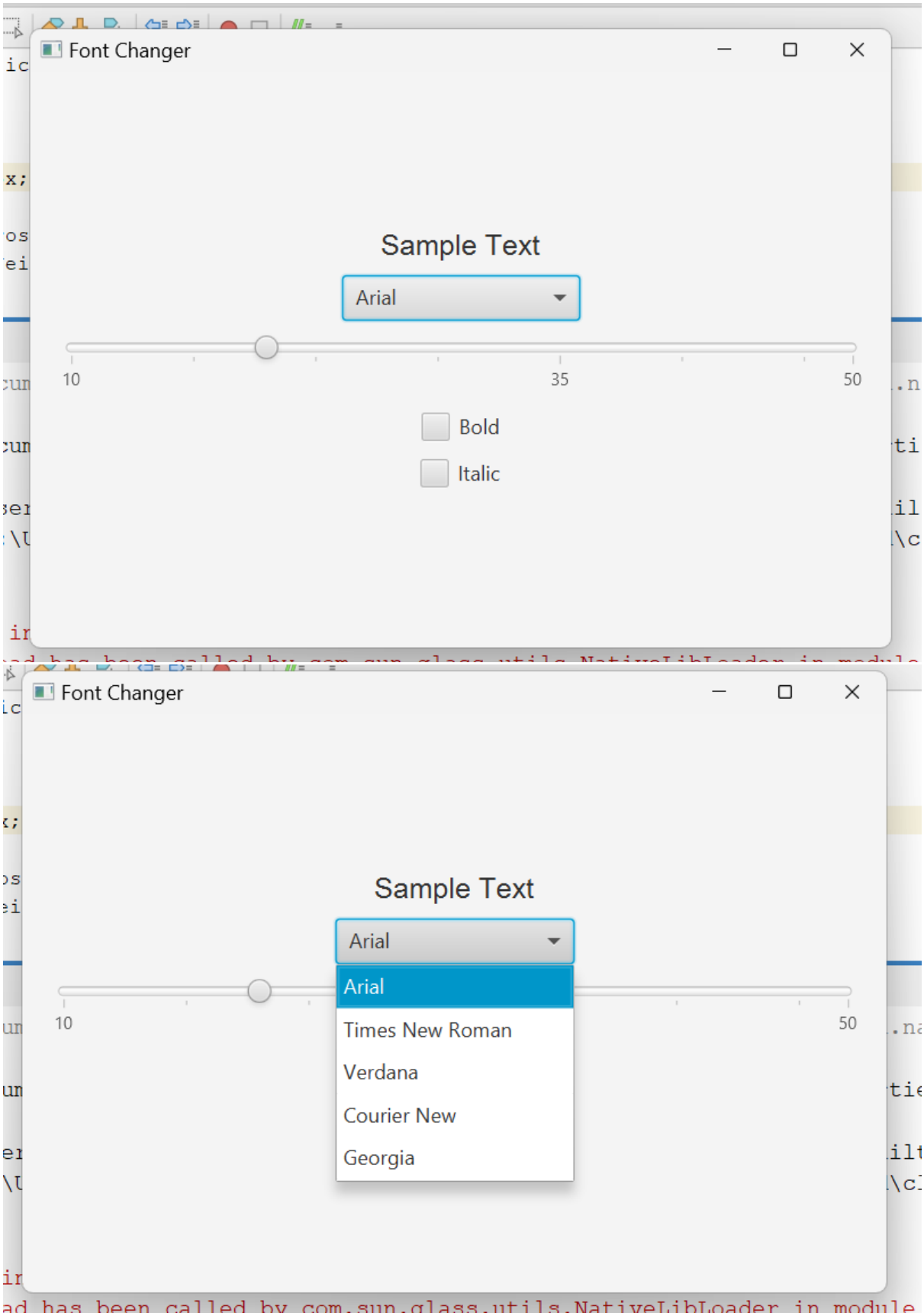
```
        root.setStyle("-fx-padding: 20px; -fx-alignment: center;");

        stage.setScene(new Scene(root, 250, 150));
        stage.setTitle("Converter");
        stage.show();
    }

    public static void main(String[] args) {
        launch();
    }
}
```



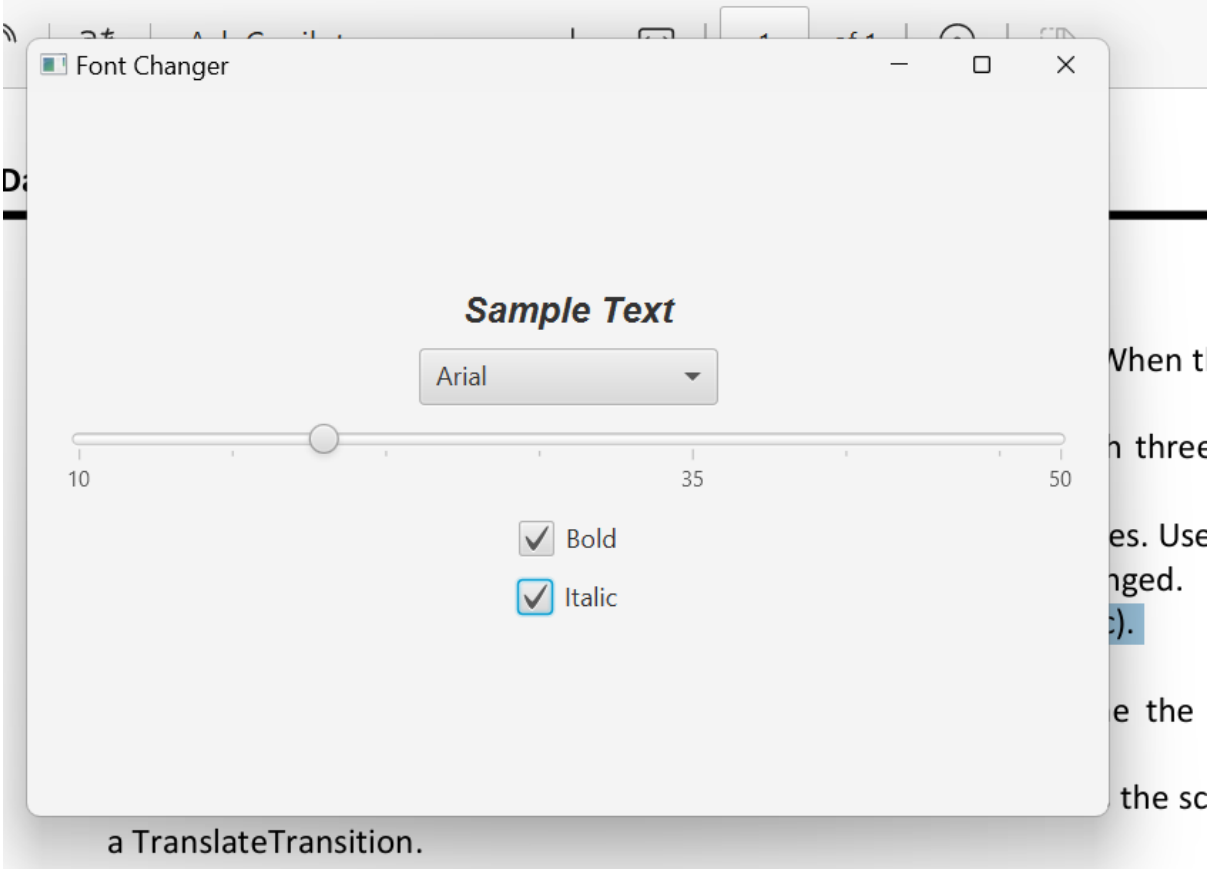
4. Create a Label and controls to change its font (family, size, bold, italic).

```
import javafx.application.Application;
import javafx.geometry.Pos;
import javafx.scene.Scene;
import javafx.scene.control.*;
import javafx.scene.layout.VBox;
import javafx.scene.text.Font;
import javafx.scene.text.FontPosture;
import javafx.scene.text.FontWeight;
import javafx.stage.Stage;

public class FontChanger extends Application {
    @Override
    public void start(Stage primaryStage) {
        Label textLabel = new Label("Sample Text");
        textLabel.setFont(Font.font("Arial", 20));

        // Font Family Selection
        ComboBox<String> fontBox = new ComboBox<>();
```

```
        fontBox.getItems().addAll("Arial", "Times New Roman", "Verdana",
"Courier New", "Georgia");
        fontBox.setValue("Arial");

        // Font Size Selection
        Slider sizeSlider = new Slider(10, 50, 20);
        sizeSlider.setShowTickLabels(true);
        sizeSlider.setShowTickMarks(true);

        // Bold & Italic CheckBoxes
        CheckBox boldCheck = new CheckBox("Bold");
        CheckBox italicCheck = new CheckBox("Italic");

        // Listener to update font
        Runnable updateFont = () -> {
            String family = fontBox.getValue();
            double size = sizeSlider.getValue();
            FontWeight weight = boldCheck.isSelected() ? FontWeight.BOLD :
FontWeight.NORMAL;
            FontPosture posture = italicCheck.isSelected() ?
FontPosture.ITALIC : FontPosture.REGULAR;
            textLabel.setFont(Font.font(family, weight, posture, size));
        };

        // Event Listeners
        fontBox.setOnAction(e -> updateFont.run());
        sizeSlider.valueProperty().addListener((obs, oldVal, newVal) ->
updateFont.run());
        boldCheck.setOnAction(e -> updateFont.run());
        italicCheck.setOnAction(e -> updateFont.run());

        // Layout
        VBox layout = new VBox(10, textLabel, fontBox, sizeSlider, boldCheck,
italicCheck);
        layout.setAlignment(Pos.CENTER);
        layout.setStyle("-fx-padding: 20px;");

        Scene scene = new Scene(layout, 350, 250);
        primaryStage.setTitle("Font Changer");
        primaryStage.setScene(scene);
        primaryStage.show();
    }

    public static void main(String[] args) {
        launch(args);
    }
}
```


Font Changer
Sample Text
Arial
10
35
50
Bold
Italic

Font Changer
Sample Text
Arial
Arial
Times New Roman
Verdana
Courier New
Georgia
10
50



5. Create a simple image gallery using ImageView and layout panes.

```
import javafx.application.Application;
import javafx.geometry.Pos;
import javafx.scene.Scene;
import javafx.scene.control.Label;
import javafx.scene.image.Image;
import javafx.scene.image.ImageView;
import javafx.scene.layout.*;
import javafx.stage.Stage;

import java.io.File;

public class ImageGallery extends Application {
    @Override
    public void start(Stage primaryStage) {
        // Corrected file paths
        String[] imagePaths = {
            "C:/Users/navdip/Pictures/685679.jpg",
            "C:/Users/navdip/Pictures/685679.jpg",
            "C:/Users/navdip/Pictures/685679.jpg",
            "C:/Users/navdip/Pictures/685679.jpg",
            "C:/Users/navdip/Pictures/685679.jpg"
        };

        // Label to display the selected image
```

```
        Label previewLabel = new Label("Click an image to preview");
        ImageView previewImage = new ImageView();
        previewImage.setFitWidth(300);
        previewImage.setFitHeight(300);
        previewImage.setPreserveRatio(true);

        // TilePane for thumbnail images
        TilePane tilePane = new TilePane(10, 10);
        tilePane.setAlignment(Pos.CENTER);

        for (String path : imagePaths) {
            File file = new File(path);
            if (file.exists()) {  // Check if the image file exists
                ImageView thumbnail = new ImageView(new
Image(file.toURI().toString()));
                thumbnail.setFitWidth(80);
                thumbnail.setFitHeight(80);
                thumbnail.setPreserveRatio(true);

                thumbnail.setOnMouseClicked(e -> {
                    previewImage.setImage(new Image(file.toURI().toString()));
                    previewLabel.setText("");  // Remove text when an image is
selected
                });

                tilePane.getChildren().add(thumbnail);
            } else {
                System.out.println("Image not found: " + path);
            }
        }

        // Layout organization
        VBox previewBox = new VBox(10, previewLabel, previewImage);
        previewBox.setAlignment(Pos.CENTER);

        VBox root = new VBox(20, previewBox, tilePane);
        root.setAlignment(Pos.CENTER);
        root.setStyle("-fx-padding: 20px;");

        Scene scene = new Scene(root, 500, 600);
        primaryStage.setTitle("Image Gallery");
        primaryStage.setScene(scene);
        primaryStage.show();
    }

    public static void main(String[] args) {
        launch(args);
    }
```

```
}
```

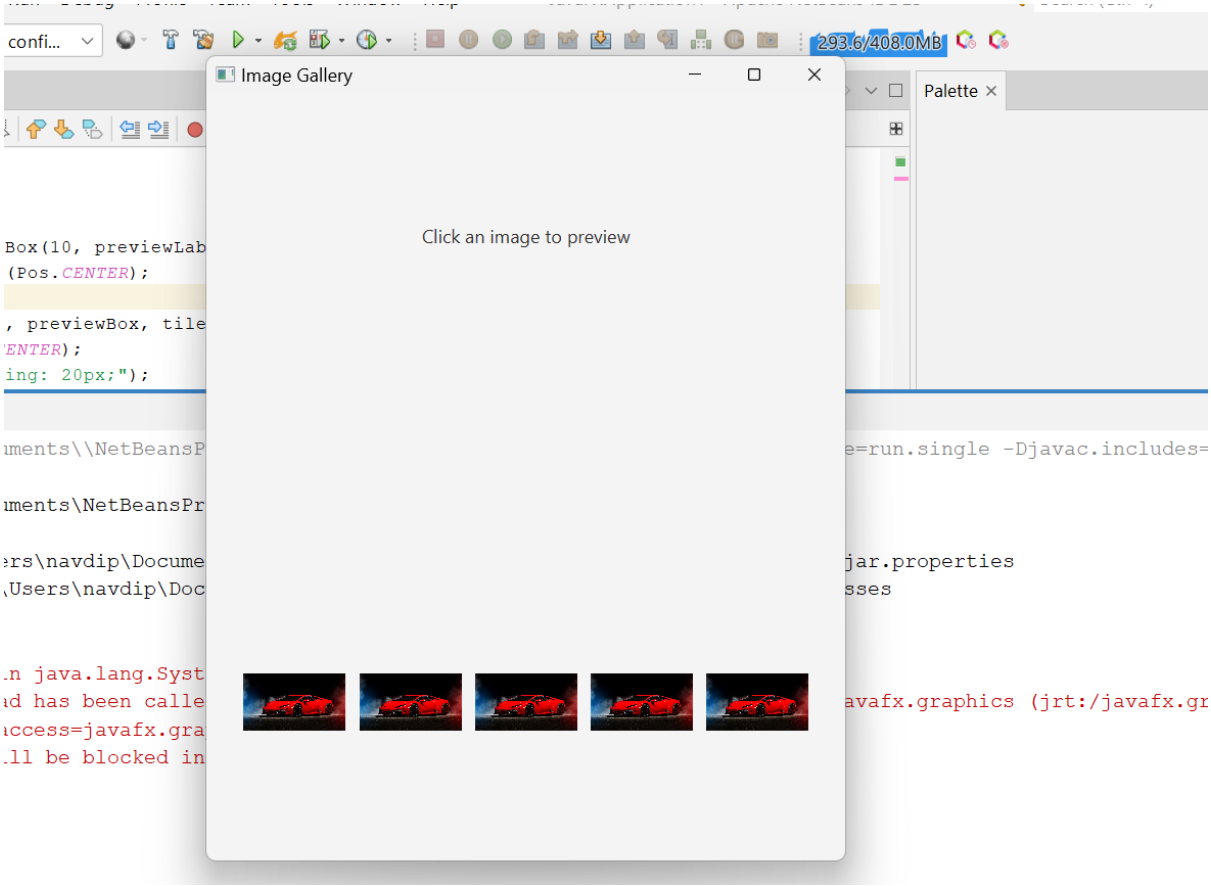
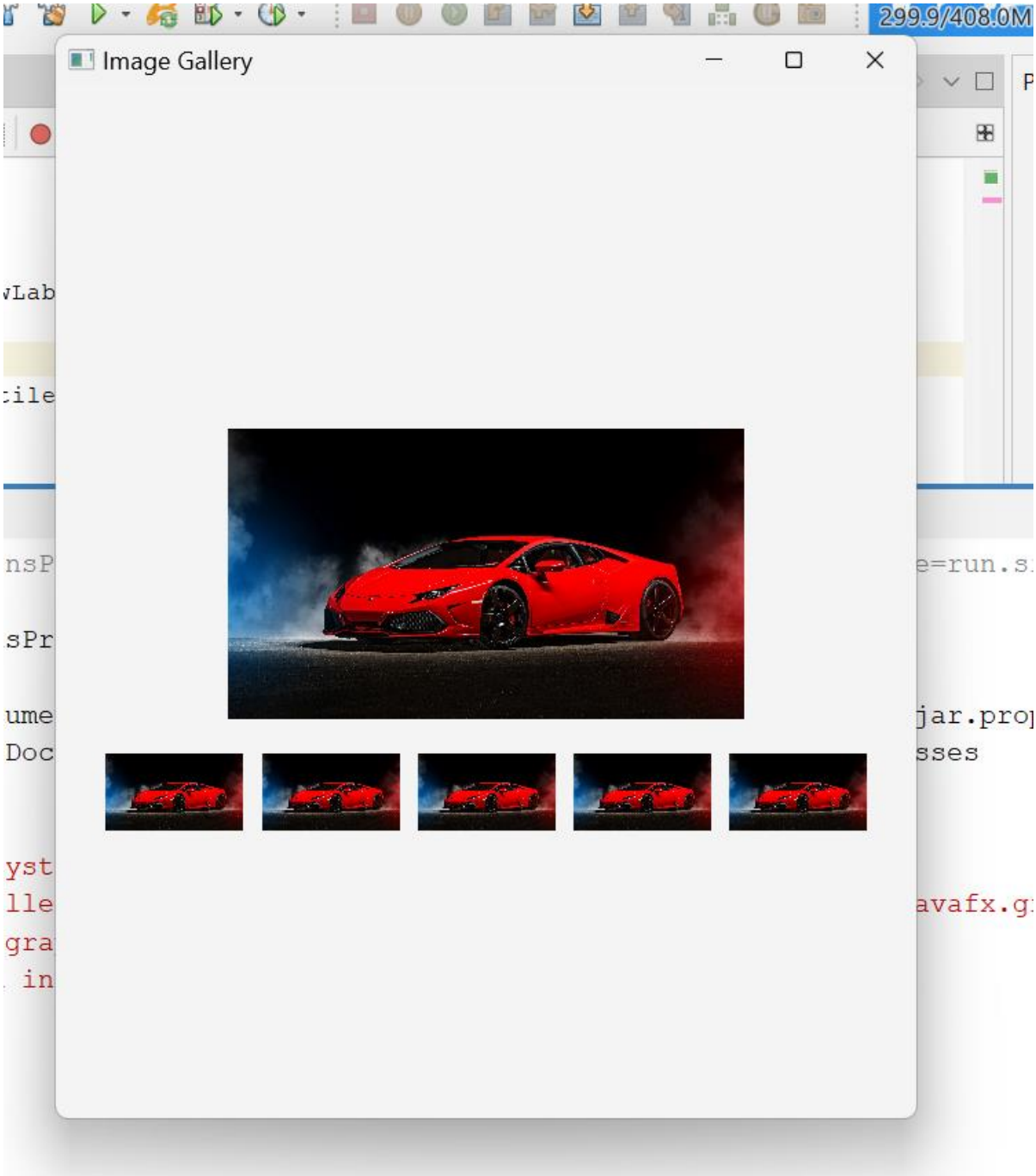


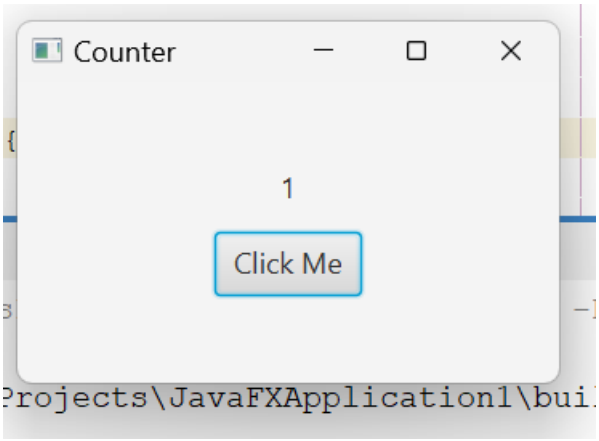
Image Gallery
Click an image to preview
Palette
293.6/408.0MB



6. Create a button and Display a counter that increments each time the button is clicked.

```
import javafx.application.Application;
import javafx.scene.Scene;
import javafx.scene.control.Button;
import javafx.scene.control.Label;
import javafx.scene.layout.VBox;
import javafx.stage.Stage;

public class SimpleCounter extends Application {
    private int count = 0;

    @Override
```

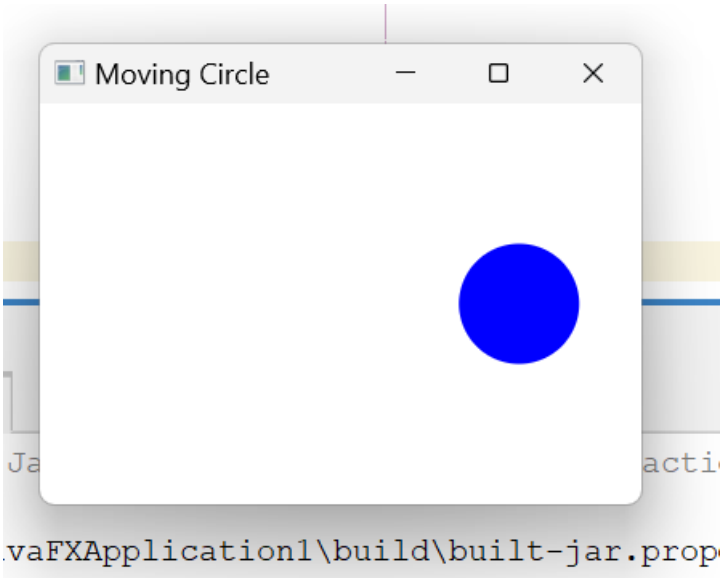
```
    public void start(Stage stage) {
        Label label = new Label("0");
        Button button = new Button("Click Me");

        button.setOnAction(e -> label.setText(String.valueOf(++count)));

        VBox root = new VBox(10, label, button);
        root.setStyle("-fx-padding: 20; -fx-alignment: center;");

        stage.setScene(new Scene(root, 200, 150));
        stage.setTitle("Counter");
        stage.show();
    }

    public static void main(String[] args) {
        launch();
    }
}
```



7. Create a shape (e.g., a Rectangle or Circle) and make it move across the scene using a TranslateTransition.

```
import javafx.animation.TranslateTransition;
import javafx.application.Application;
import javafx.scene.Scene;
import javafx.scene.paint.Color;
import javafx.scene.shape.Circle;
import javafx.scene.layout.Pane;
import javafx.stage.Stage;
import javafx.util.Duration;

public class SimpleMovingCircle extends Application {
    @Override
    public void start(Stage stage) {
        Circle circle = new Circle(30, Color.BLUE); // Create a circle
        circle.setCenterX(50);
```

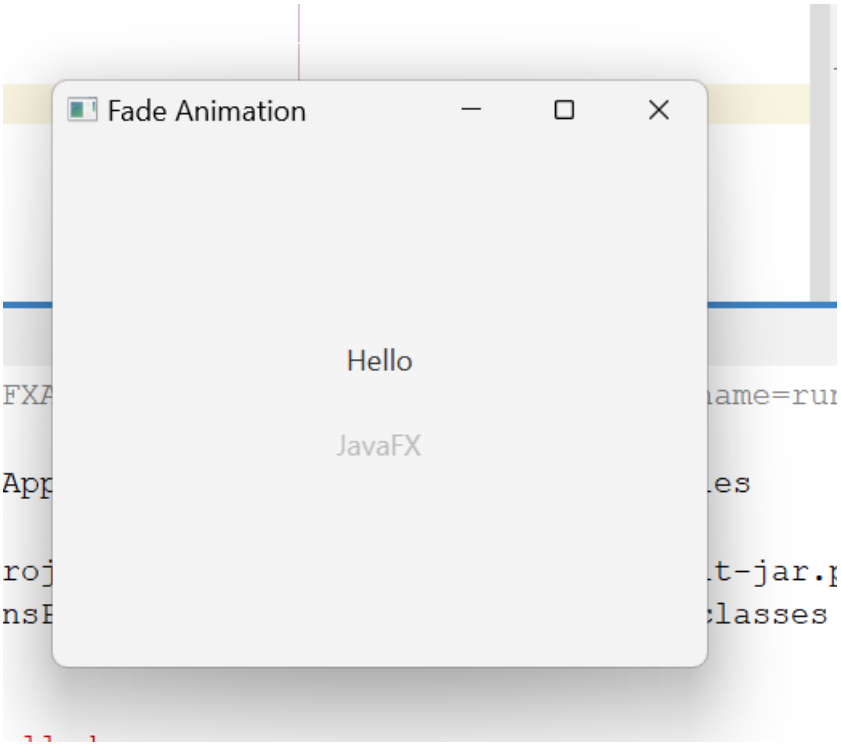
```
        circle.setCenterY(100);

        TranslateTransition transition = new
TranslateTransition(Duration.seconds(2), circle);
        transition.setByX(200); // Move 200 pixels right
        transition.setAutoReverse(true);
        transition.setCycleCount(TranslateTransition.INDEFINITE);
        transition.play();

        Pane root = new Pane(circle);
        stage.setScene(new Scene(root, 300, 200));
        stage.setTitle("Moving Circle");
        stage.show();
    }

    public static void main(String[] args) {
        launch();
    }
}
```



8. Create a sequence where several UI elements (Labels, ImageViews) fade in and out one after another using FadeTransition and SequentialTransition.

```
import javafx.animation.FadeTransition;
import javafx.animation.SequentialTransition;
import javafx.application.Application;
import javafx.scene.Scene;
import javafx.scene.control.Label;
import javafx.scene.layout.VBox;
import javafx.stage.Stage;
import javafx.util.Duration;
```

```
public class SimpleFadeSequence extends Application {
    @Override
    public void start(Stage stage) {
        Label label1 = new Label("Hello");
        Label label2 = new Label("JavaFX");

        FadeTransition fade1 = new FadeTransition(Duration.seconds(1),
label1);
        fade1.setFromValue(0);
        fade1.setToValue(1);

        FadeTransition fade2 = new FadeTransition(Duration.seconds(1),
label2);
        fade2.setFromValue(0);
        fade2.setToValue(1);

        SequentialTransition sequence = new SequentialTransition(fade1,
fade2);
        sequence.setCycleCount(SequentialTransition.INDEFINITE);
        sequence.play();

        VBox root = new VBox(20, label1, label2);
        root.setStyle("-fx-padding: 30px; -fx-alignment: center;");

        stage.setScene(new Scene(root, 200, 150));
        stage.setTitle("Fade Animation");
        stage.show();
    }

    public static void main(String[] args) {
        launch();
    }
}
```


Fade Animation
Hello
JavaFX