1 import javafx.application.Application;

2 import javafx.event.ActionEvent;

3 import javafx.geometry.Insets;

4 import javafx.scene.Scene;

5 import javafx.scene.control.\*;

5 import javafx.scene.layout.VBox;

6 import javafx.stage.Stage;

8

9 public class ColorPickerDemo extends Application

10 {

11 ColorPicker aColorPicker= new ColorPicker(Color.ORANGE);

12 Button pickColor, likeColor;

13 VBox root;

14

15 @Override

16 public void start(Stage primaryStage)

17 {

18 root = new VBox();

19 pickColor = new Button("Pick Color");

20 likeColor = new Button("Like Color");

21

22 aColorPicker.setOnAction(e-> handle(e));

23 pickColor.setOnAction(e-> handle(e));

24 likeColor.setOnAction(e-> handle(e));

25

26 root.setAlignment(Pos.CENTER);

27 root.setSpacing(15.0);

28 root.getChildren().addAll(pickColor, likeColor);

29

30 Scene scene = new Scene(root, 400, 300);

31

32 primaryStage.setTitle("Color Picker Demo");

33 primaryStage.setScene(scene);

34 primaryStage.show();

35 }

36

37 public void handle(ActionEvent e)

38 {

39 if(e.getSource() == pickColor &&

40 !root.getChildren().contains(aColorPicker))

41 {

42 root.getChildren().add(aColorPicker);

43 }

44 else if(e.getSource() == aColorPicker)

45 {

46 String c = aColorPicker.getValue().toString();

47 c = “#” + c.substring(2,8);

48 root.setStyle("-fx-background-color: " + c);

49 root.getChildren().remove(aColorPicker);

50 }

51 else if(e.getSource() == likeColor &&

52 !root.getChildren().contains(aColorPicker))

53 {

54 root.getChildren().removeAll(pickColor, likeColor);

55 }

56 }

57

58 public static void main(String[] args)

59 {

60 launch(args);

61 }

62 }

**Figure 12.19**

The application ColorPickerDemo.