

```
1 import javafx.application.*;
2 import javafx.scene.*;
3 import javafx.scene.layout.*;
4 import javafx.scene.control.*;
5 import javafx.scene.paint.*;
6 import javafx.stage.*;
7
8 ///////////////////////////////////////////////////
9 // 問1 下のウィンドウを表示するコードを書きなさい。
10 ///////////////////////////////////////////////////
11
12 public class Assignment3_1 extends Application
13 {
14     public void start(Stage stage) throws Exception
15     {
16         // レイアウトを生成します
17         FlowPane flowpane = new FlowPane();
18
19         // シーンを生成します
20         Scene scene = new Scene(flowpane);
21         scene.setFill(Color.GREEN);
22
23         // ステージを生成します
24         stage.setScene(scene);
25         stage.setTitle("oooシステム バージョン 2.3.1");
26         stage.setWidth(450);
27         stage.setHeight(250);
28
29         // ステージ (ウィンドウ) を表示します
30         stage.show();
31     }
32
33     public static void main(String[] args)
34     {
35         launch(args);
36     }
37 }
38
```

```

1 import javafx.application.*;
2 import javafx.scene.*;
3 import javafx.scene.layout.*;
4 import javafx.scene.control.*;
5 import javafx.scene.paint.*;
6 import javafx.stage.*;
7
8 ///////////////////////////////////////////////////
9 // 問2 下のように4つのウィンドウをタイル状に表示するコード...
10 ///////////////////////////////////////////////////
11
12 public class Assignment3_2 extends Application
13 {
14     public void start(Stage stage) throws Exception
15     {
16         // レイアウトを4つ生成します
17         FlowPane flowpane1 = new FlowPane();
18         FlowPane flowpane2 = new FlowPane();
19         FlowPane flowpane3 = new FlowPane();
20         FlowPane flowpane4 = new FlowPane();
21
22         // シーンを4つ生成します
23         Scene scenel = new Scene(flowpane1);
24         Scene scene2 = new Scene(flowpane2);
25         Scene scene3 = new Scene(flowpane3);
26         Scene scene4 = new Scene(flowpane4);
27         scenel.setFill(Color.RED);
28         scene2.setFill(Color.BLUE);
29         scene3.setFill(Color.YELLOW);
30         scene4.setFill(Color.BLACK);
31
32         // ステージを4つ生成します
33         Stage stagel = new Stage();
34         Stage stage2 = new Stage();
35         Stage stage3 = new Stage();
36         Stage stage4 = new Stage();
37         stagel.setScene(scenel);
38         stage2.setScene(scene2);
39         stage3.setScene(scene3);
40         stage4.setScene(scene4);
41         stagel.setTitle("赤色");
42         stage2.setTitle("青色");
43         stage3.setTitle("黄色");
44         stage4.setTitle("黒色");
45         stagel.setWidth(300); stagel.setHeight(300);
46         stage2.setWidth(300); stage2.setHeight(300);
47         stage3.setWidth(300); stage3.setHeight(300);
48         stage4.setWidth(300); stage4.setHeight(300);
49         stagel.setX(10); stagel.setY(10);
50         stage2.setX(310); stage2.setY(10);
51         stage3.setX(10); stage3.setY(310);
52         stage4.setX(310); stage4.setY(310);
53
54         // ステージ (ウィンドウ) を表示します
55         stagel.show();
56         stage2.show();
57         stage3.show();
58         stage4.show();
59     }
60
61     public static void main(String[] args)
62     {
63         launch(args);
64     }
65 }
66

```

```
1 import javafx.application.*;
2 import javafx.scene.*;
3 import javafx.scene.layout.*;
4 import javafx.scene.control.*;
5 import javafx.scene.paint.*;
6 import javafx.stage.*;
7
8 import javafx.scene.image.*;
9 import javafx.geometry.*;
10 import javafx.animation.*;
11 import javafx.util.*;
12
13 ///////////////////////////////////////////////////
14 // 問3 下のようなロゴ画面を表示するコードを書きなさい...
15 ///////////////////////////////////////////////////
16
17 public class Assignment3_3 extends Application
18 {
19     public void start(Stage stage) throws Exception
20     {
21         // ロゴの準備
22         Image logo = new Image("halloween.jpg");
23         ImagePattern imgptn = new ImagePattern(logo);
24
25         // レイアウトを生成します
26         FlowPane logofp = new FlowPane();
27
28         // シーンを生成します
29         Scene logosc = new Scene(logofp, 400, 300, imgptn);
30
31         // ステージにシーンを設定します
32         Stage logost = new Stage();
33         logost.initStyle(StageStyle.UNDECORATED);
34         logost.setScene(logosc);
35
36         // 画面サイズを取得します
37         Rectangle2D screensize = Screen.getPrimary().getBounds();
38
39         // 画面中央に表示します
40         logost.setX(screensize.getWidth()/2 - 200);
41         logost.setY(screensize.getHeight()/2 - 150);
42
43         // ステージを表示します
44         logost.show();
45     }
46
47     public static void main(String[] args)
48     {
49         launch(args);
50     }
51 }
52
```

```

1 import javafx.application.*;
2 import javafx.scene.*;
3 import javafx.scene.layout.*;
4 import javafx.scene.control.*;
5 import javafx.scene.paint.*;
6 import javafx.stage.*;
7
8 import javafx.scene.image.*;
9 import javafx.geometry.*;
10 import javafx.animation.*;
11 import javafx.util.*;
12
13 // ※これは参考としてログ表示からアプリ起動までを行うコードです
14 ///////////////////////////////////////////////////////////////////
15 // 問3 下のようなログ画面を表示するコードを書きなさい...
16 ///////////////////////////////////////////////////////////////////
17
18 public class Assignment3_3s extends Application
19 {
20     // ログ用ステージ
21     private Stage logost;
22     // アプリ用ステージ
23     private Stage appst;
24     // ※ネストクラスから参照するためメンバーとして宣言
25
26     public void start(Stage stage) throws Exception
27     {
28         // ログの準備
29         Image logo = new Image("halloween.jpg");
30         ImagePattern imgptn = new ImagePattern(logo);
31
32         // レイアウトを生成します
33         FlowPane logofp = new FlowPane();
34         FlowPane appfp = new FlowPane();
35
36         // シーンを生成します
37         Scene logosc = new Scene(logofp, 400, 300, imgptn);
38         Scene appsc = new Scene(appfp, 700, 500);
39
40         // ステージにシーンを設定します
41         logost = new Stage();
42         appst = new Stage();
43         logost.initStyle(StageStyle.UNDECORATED);
44         logost.setScene(logosc);
45         appst.setTitle("ハロウィンディナ予約");
46
47         // 画面サイズを取得します
48         Rectangle2D screensize = Screen.getPrimary().getBounds();
49
50         // ログを画面中央に表示します
51         logost.setX(screensize.getWidth()/2 - 200);
52         logost.setY(screensize.getHeight()/2 - 150);
53
54         // ログを表示します
55         logost.show();
56
57         // 5秒後にログを非表示にします
58         Thread th=new Thread(new Runnable() {
59             public void run() {
60                 try{
61                     Thread.sleep(5000);
62                     Platform.runLater(new Runnable() {
63                         public void run() {
64                             // ログを消してアプリを表示します
65                             logost.close();
66                             appst.show();
67                         }
68                     });
69                 }catch(InterruptedException e){}
70             }
71         });
72         th.start();
73     }
74
75     public static void main(String[] args)
76     {
77         launch(args);
78     }
79 }
80

```