



STUDENTS-HUB.com

Uploaded By: Malak Obalid

>> How to Display a Stage



STUDENTS-HUB.com

>> How to Display a Stage and but button on it

```
import javafx.application.Application;
    import javafx.scene.Scene;
    import javafx.scene.layout.Pane;
    import javafx.scene.control.Button;
    import javafx.stage.Stage;
    public class MyJavaFX extends Application {
           @Override
          public void start(Stage primaryStage) {
                 Button bt = new Button("OK");
                 Pane pane = new Pane();
                 pane.getChildren().add(bt);
                 Scene scene = new Scene(pane, 200, 250);
                 primaryStage.setTitle("MyJavaFX");
                 primaryStage.setScene(scene);
                 primaryStage.show();
           }
    }
عرفنا Button وضفناه على البرنامج السابق بنقدر نظيف ال button مباشرة على ال Scene بس رح
يغطى الواجهة كاملة عشان هيك بنلجا لحل نحط ال button في داخل pane ورح يطلع معنا البرنامج
  بالشكّل الاتي، احنا لهسا بس بنتعلم نبني اشكال وبعدين بنتعلم تُنفيذ الاوامر عند الضغط على الزر
                     MyJavaFX
                                         Х
                                    OK
```

ملخص سريع لهسا احنا عرفنا انه بنحتاج عشان نبني ال Application تاعنا لازم اول اشي يكون عنا Stage والي هو عبارة عن الواجهة الي بتظهر وما في مانع لو كان في اكثر من Stage في نفس البرنامج، وبنعرف كمان انه لازم يكون في عنا Scene في داخل ال Stage ومعلومة مهمة كثير انه اذا عرفنا Scene لا يمكن وضعه الا في Stage واحد فقط، وشفنا كمان انه بقدر احط pane جوا ال Scene، ايش ال pane هو وسيلة عشان نقدر نرتب الاشياء جوا ال Scene وفي اله انواع كثيرة بنتعرف عليها لقدام.

>> import in JavaFX

import javafx.packageName.packageName.ClassName(*);

```
import javafx.packageName.className(*);
```

same package we need to import :

import	javafx.application.Application;
import	javafx.stage.Stage;
import	javafx.scene.Scene;
import	javafx.scene.control.*;
import	javafx.scene.layout.*;
import	javafx.scene.paint.*;
import	javafx.scene.shape.*;
import	javafx.geometry.*;

>> How to Display a Shape

import javafx.scene.shape.*;



نستطيع اعطاء لون لهذه الاشكال او اعطاء لون للحواف الخاصة بها باستخدام Class مخصص موجود في javafx.scene.paint اسمه Color مثال:

Rectangle rectangle = new Rectangle(30,30);

rectangle.setFill(Color.BLUE);

rectangle.setStroke(Color.BLACK);

يعنى كلاس Color فيه الوان معرفة Static عشان نقدر نستخدمها .

وال Object الي عرفناه من نوع Rectangle فيه ميثود اسمها setFill بتساعدنا عشان نلون الشكل من جوا والميثود الثانية هي setStroke عشان نلون الحواف.

ملااحظة في ميثود اسمها setStyle قصتها قصة موضحة تحت.

في السطور السابقة قمنا بتعريف object من Class موجودة في javafx.scene.shape والان سوف نقوم باظهار هذه الاشكال في البرنامج عن طريق وضعها في ال Scene مباشرة او عن طريق وضعها في احد انواع ال Pane واظهار هذه ال Pane في ال Scene وبالتالي يؤدي الى ظهور الاشكال الهندسية في البرامج.

Example:

```
public void start(Stage stage) {
    Circle circle= new Circle(50);
    circle.setFill(Color.BLUE);
    circle.setStroke(Color.BLACK);
    StackPane pane = new StackPane();
    pane.getChildren().addAll(circle);
    Scene scene = new Scene(pane,250, 250);
    stage.setTitle("A7A");
    stage.setScene(scene);
    stage.show();
```

```
STUDENTS-HUB.com
```

}

The result:



طبعا استخدمنا نوع خاص من ال Pane اسمه StackPane وهو برتب الاشياء الي بنحطها جواته فوق بعض يعنى شكل فوق شكل وهكزا. كله جاينه في التفصيل

>>Binding Properties

ال Stage الي بظهر ممكن تكبره وتصغره زي ما بدك بس بتلاحظ لما تكبره او تصغره انه حجم الاشكال الي جواه بضل ثابت وهون بنيجي لاشي اسمه ال Binding عشان نقدر نربط حجم الاشكال مع حجم ال Stage او حجم ال Scene او حجم ال Pane الي بريحك بس <mark>غالبا بنريطه مع ال Pane</mark>.

عشان نعمل Binding بنستخدم اشي اسمه Property وهاذ بكون معرف جاهز جوا الكلاس (جزء من المعنان أي المعنان أي المعن ال data field تاعت الكلاس) وهي مثال عليه :





- : بنحط اسم ال object الي بدنا اياه يكبر او يصغر لما نحرك ال Stage : بنحط اسم البعد الي بملكه ال object وهو البعد الي رح يكبر او يصغر : بنحط اسم ال Pane او ال Stage الى بدنا نربط ال object فيها
- : بنحط اسم البعد الى بملكه ال Pane او ال Stage وهو الى رح نربط قيمته بقيمة بعد الشكل
 - : بنحط العملية الى بدها تربط بين البعدين
 - : بنحط قيمة ال Scale بين البعدين

مثلا بدنا نعدل ع اخر مثال عملناه تاع الدائرة ونعمله Binding يعني لما نكبر او نصغر ال Stage رح نلاقي انه الدائرة بتكبر او بتصغر مع ال Stage . (بس بنظيف هذا السطر لل Code السابق)

circle.radiusProperty().bind(pane.heightProperty().divide(2));

مثلا لو بدنا نربط مستطيل بنربط طول المستطيل بطول ال pane وبنربط عرض المستطيل بعرض ال Pane وهكزا.

>> Image View

Image image = new Image("C:\\Users\\user\\Documents\\PSPice\\1.png"); ImageView imageView = new ImageView(image); يعني هون في عنا event (حدث) هاذ الحدث الي هوا حدا يكبس على ال button او انه حدا يختار RadioButton او حدا يختار comboBox وهكذا فالفكرة انه لازم حدا يروح يعمل handle فهاذ ال handle تطور في الجافا على اربع مراحل الي هي كالتالي

- 1- اول اشي انه عنا class منفصل لازم هاذ ال class هو الي يعمل handle طبعا هاذ الكلاس بكون class الله عنا دامي class منفصل لازم هاذ ال class هو الي يعمل override ل
 Implements EventHandler<ActionEvent وجواه بنروح بنعمل override ل</p>
 method اسمها handle طبعا هاي الطريقة فيها مشكلة لانه جوا handle ما بنقدر نوصل لل
 object الموجودين جوا ال class الي بحتوي على ال botact action action
- 2- مرحلة ال Inner Class في هاي المرحلة بندخل ال class الي بحتوي على handle جوا ال class الي بحتوي على bart method جوا ال class وكمان بنحط جميع ال object الي بدنا نستخدمهم dataField في ال class الخارجي وطبعا لسا الموضوع تطور اكثر لانه الحكي هاذ غلبة لوصلنا للمرحلة الثالثة
- 3- مرحلة Anonymous Inner Classes الي هوا بنعمل object مباشرة بدون ما نعمل كلاس بس برضو ضل الموضوع يتطور لوصلنا ل
 - 4- (<-) وهاي هي المرحلة النهائية طبعا كلهم توضحوا تحت</p>

رح نعمل البرنامج الاتي :



STUDENTS-HUB.com

هاذ البرنامج بسيط نتيجة ال run اله :
Run
بدنا نظيف على هذا البرنامج حدث يعني لما نضغط على ال button يصير اشي معين :
و هاذ الموضوع بتم عن طريق
button.setOnAction();
عن طريق هاي ال method طبعا رح نطبق عليها الاربع حالا الي فوق
ما ننسی هاي :
<pre>import javafx.event.*;</pre>

1- الحالة الاولى

هون بنعمل object من ال class الي مسوؤل عن ال handle وبنعطي ال method هاذ ال





3- الحالة الثالثة

في هاي المرحلة بنرجع ندخل ال object الي طلعناهم برا لانه رح نعمل Anonymous Inner الي طلعناهم برا لانه رح نعمل Classes

كيف نعمل الصيغة هاي حفظ

```
public void start(Stage primaryStage) {
    VBox vBox = new VBox();
    Button button = new Button("Run");
    Circle circle = new Circle(30);
```

button.setOnAction(new EventHandler<ActionEvent> (){

```
@Override
public void handle (ActionEvent e) {
    circle.setRadius()+10);
}
```

```
vBox.getChildren().addAll(circle,button);
vBox.setAlignment(Pos.CENTER);
vBox.setSpacing(10);
Scene scene = new Scene(vBox,400,400);
primaryStage.setScene(scene);
primaryStage.show();
```

وجوا ال handle هاي بنكتب شو ما بدنا

طبعا المرحلة الرابعة اختصار لهاي المرحلة

});

4- المرحلة الرابعة



طبعا رمز ال (<-) هو عبارة عن بديل للخطة رقم 3 يعني مسموح نستخدمه لما يكون عنا one abstract method لانه هو عبارة عن بديل لهاي ال method عشان هيك بنحكي انه لازم نستخدمه مع SAM class وهذول همي ال classes الي بحتوا على one abstract method بس

هسا بضل علينا نطبق ونحل اسئلة

هيك كل اشي جاهز باذن الله