



جامعة طرابلس  
كلية تقنية المعلومات  
قسم هندسة البرمجيات



## البرمجة المرئية Visual Programming خريف 2024

المحاضرة الثانية



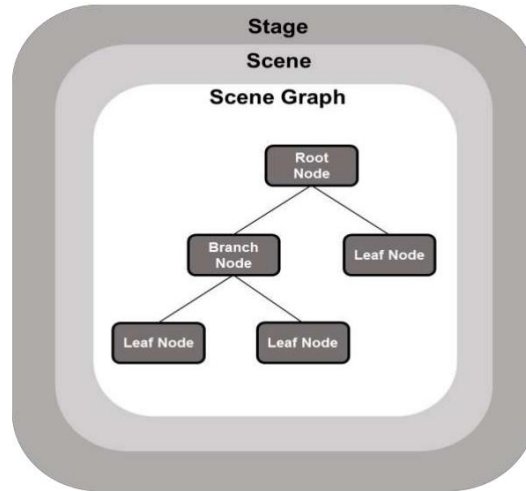
### مواضيع المحاضرة

- ▶ مكونات تطبيق JavaFX
- ▶ ما هو Stage؟
- ▶ ما هو المشهد Scene؟
- ▶ ما هي العقدة Node؟
- ▶ نظام الإحداثيات للعقد Nodes
- ▶ انشاء JavaFX Application
- ▶ مثال تطبيق JavaFX
- ▶ مثال عرض Stage الرئيسية
- ▶ مثال تحديد حدود Stage
- ▶ عرض Stage في وضع ملء الشاشة



## مكونات تطبيق JavaFX

يتكون تطبيق JavaFX من ثلاث مكونات أساسية هي: Stage و Scene و Nodes ▶



▶ 3



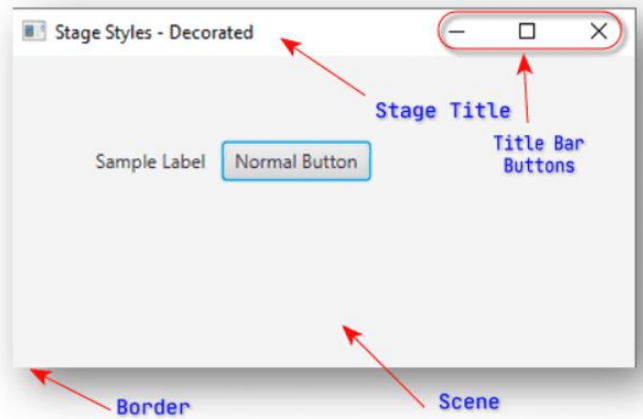
## ما هو Stage؟

- ▶ نافذة (window) تحوي كل عناصر التطبيق.
- ▶ تمثل بالحزمة `javafx.stage`.
- ▶ Stage الأولى تنشأ من المنصة نفسها وتمرر كعامل (parameter) الى دالة `start()`.
- ▶ اعتمادا على الجهاز المستخدم قد يوجد stage واحدة فقط التي تحوي scene واحدة او عدة.
- ▶ يمكن إنشاء Stages إضافية حسب الحاجة.
- ▶ لعرض محتوى stage لابد من استدعاء الدالة `show()`.
- ▶ لها اثنان من العوامل لتحديد موقعها `Width` و `Height`.

▶ 4



## ما هو Stage؟

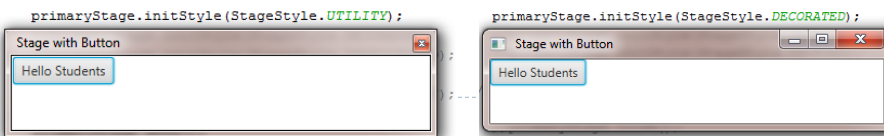


▶ 5



## ما هو Stage؟

- ▶ تقسم الى جزئين: مساحة المحتوى (content area) والزخارف (decorations) التي تشمل Title Bar و Borders.
- ▶ يمكن ان تحدد شكل stage باستخدام دالة `initStyle()` بالقيم الخمسة الآتية:
- ▶ **Decorated**: هي النافذة المعتادة التي لها title bar و ازرار minimize و maximize و close وخلفية بيضاء.
- ▶ **Undcorated**: هي النافذة المعتادة لكن بدون الازرار.
- ▶ **Transparent**: هي مثل `undecorated window` لكن مع خلفية شفافة.
- ▶ **Unified**: هي مثل `decorated window` لكن بدون اطار بين `content area` و `decoration area`.
- ▶ **Utility**: هي مثل `decorated window` لكن مع الحد الأدنى من الزخارف.





## ما هو المشهد Scene؟

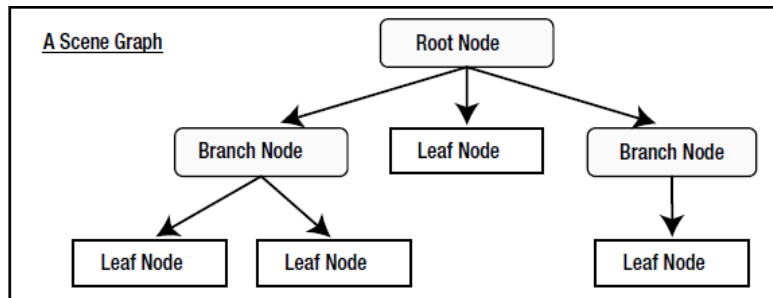
- ▶ يتكون Scene من scene graph والتي تحتوي على عقد مرئية nodes.
- ▶ يعمل Scene كحاوية لكل العقد scene graph.
- ▶ يكون scene graph كشجرة تحتوي على عقد، يمكن أن تكون العقدة إما عقدة فرعية branch node أو عقدة طرفية leaf node.
- ▶ يمكن أن تحتوي العقدة الفرعية على عقد للأبناء children nodes، بينما لا يمكن للعقدة الطرفية ذلك.
- ▶ تسمى العقدة الأولى في scene بعقدة الجذر root node.
- ▶ العقدة node في scene graph عبارة عن كائن من javafx.scene.

▶ 7



## ما هو المشهد Scene؟

- ▶ يمكن أن تحتوي root node على عقد أبناء children nodes؛ ومع ذلك، فإنها لا تحتوي على عقدة أب parent node أعلى منها.



▶ 8



## ما هي العقدة Node؟

- ▶ يمكن ان تمثل Node الاتي:
  - ▶ كائنات هندسية (2D و 3D) مثل: الدائرة Circle والمستطيل Rectangle والمضلع Polygon.... الخ
  - ▶ عناصر التحكم (UI Controls) مثل: الزر Button ومربع الاختيار Check Box و Choice Box ومربع النص Text Area... الخ
  - ▶ الحاويات (Layout Panes) مثل: Border Pane و Grid Pane و Flow Pane.... الخ
  - ▶ عناصر الوسائط (Media) مثل: كائنات الصوت Audio و الفيديو Video والصورة Image.

▶ 9



## ما هي العقدة Node؟

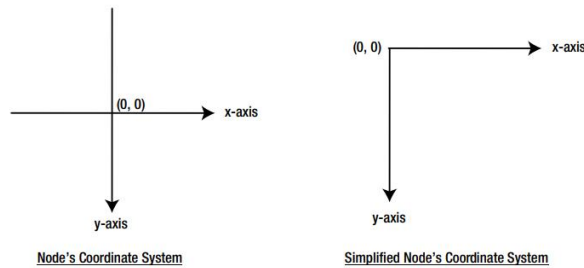
- ▶ توجد ثلاث أنواع من Node:
  - ▶ Root Node: اول Scene Graph تسمى Root node
  - ▶ Branch Node/ Parent Node: هي Node التي لها أبناء. يمكن ان تكون احدي الأنواع التالية:
    - Group: هي عقدة مجمعة التي تحوي مجموعة من العقد الأبناء. بحيث كل التغيرات التي تحدث group ستنطبق على العقد الأبناء.
    - Region: هي الفئة الأساس لكل UI Controls مثل Chart و Pane و Control.
    - WebView: تتحكم هذه العقدة بمحرك الويب وتعرض محتوياته.
  - ▶ العقدة الطرفية (Leaf Node): هي العقدة التي بدون أبناء مثل Rectangle و Ellipse و Box و ImageView و MediaView.

▶ 10



## نظام الإحداثيات للعقد Nodes

- ▶ كل عقدة Node في scene graph لها إحداثيات خاص بها. تستخدم العقدة نظام إحداثيات يتكون من محور X ومحور Y. في أنظمة الكمبيوتر ، تزيد القيم الموجودة على المحور x جهة اليمين وتزداد القيم على المحور y إلى أسفل .

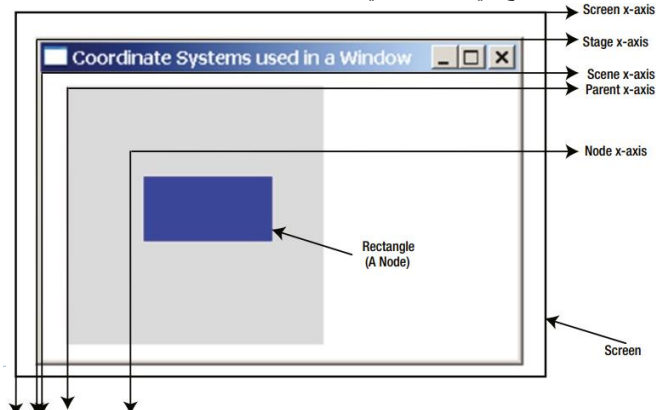


▶ 11



## نظام الإحداثيات للعقد Nodes

- ▶ في تطبيق واجهة المستخدم الرسومية GUI، يتم وضع العقد داخل أبائهم. العقدة الجذرية root node هي الأصل لجميع العقد ويتم وضعها داخل المشهد Scene. يتم وضع Scene داخل Stage ويتم وضع Stage داخل الشاشة Screen. كل عنصر يتكون من نافذة ، له نظام إحداثيات خاص به، كما هو موضح في الشكل التالي:



▶ 12



## انشاء JavaFX Application

- ▶ نقطة البداية هي الفئة Application للحزمة javafx.application.
- ▶ لانشاء تطبيق JavaFx عليك بوراثة الفئة application وتنفيذ دالته start() التي ستحمل كامل الاسطر البرمجية.
- ▶ في دالة main() عليك ببدأ التطبيق باستخدام الدالة launch().
- ▶ داخل دالة start() عليك بتجهيز scene graph بالعقد المطلوبة. ثم تجهيز Scene بالأبعاد المحددة وازافة scene graph له. ثم تجهيز stage بإضافة scene له وعرض محتويات stage.

▶ 13



## انشاء JavaFX Application

▶ تجهيز Scene Graph

```
Group root = new Group();
ObservableList list = root.getChildren();
list.add(NodeObject);
Group root = new Group(NodeObject);
```

▶ تجهيز Scene

```
Scene scene = new Scene(root)
Scene scene = new Scene(root, 600, 300);
```

▶ تجهيز Stage

```
primaryStage.setTitle("Sample application");
primaryStage.setScene(scene);
primaryStage.show();
```

▶ 14



## مثال تطبيق JavaFX

```
import javafx.application.Application;
import javafx.stage.Stage;

public class Lect2Ex1 extends Application {
    public void start(Stage primaryStage) {
        primaryStage.show();
    }

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

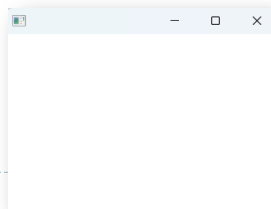
▶ 15



## مثال تطبيق JavaFX

▶ لاحظ ثلاثة أشياء رئيسية:

- ▶ أن الفئة تستخدم "extends Application"، وهذا يعلم Java أنك ستستخدم inheritance لجعل هذا التطبيق يستخدم JavaFX. فالفئة Application توفر وظائف دورة حياة التطبيق من Initializing و launching و starting و stoping خلال وقت التنفيذ مما يوفر الية لتطبيقات الجافا من تشغيل مكونات JavaFX GUI بشكل منفصل عن main thread.
- ▶ يحتوي على الطريقة المطلوبة وهي "start (Stage primaryStage)", حيث أن primaryStage هو الذي سيظهر.
- ▶ أخيرا سنطلب من stage أن تظهر بواسطة "primaryStage.show()" وهذا ما يجعل التطبيق مرئي.



▶ 16





## مثال عرض Stage الرئيسية

```

package runningapp;

import javafx.application.Application;
import javafx.application.Platform;
import javafx.stage.Stage;

/**
 *
 * @author Hassan
 */
public class RunningApp extends Application {

    @Override
    public void start(Stage primaryStage) {
        Platform.exit();
    }

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        launch(args);
    }
}

```

إيقاف تطبيق JavaFX باستخدام Platform.exit()

▶ 17

```

package runningapp;

import javafx.application.Application;
import javafx.stage.Stage;

/**
 *
 * @author Hassan
 */
public class RunningApp extends Application {

    @Override
    public void start(Stage primaryStage) {
        //Do not write any code
    }

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        launch(args);
    }
}

```

تطبيق JavaFX بدون اسطر برمجية



## مثال عرض Stage الرئيسية

- ▶ في المثال الموضح في الشريحة السابقة، لا تحتوي start() على أي اسطر برمجية. وعليه، عند تنفيذ التطبيق، لا تظهر أي نافذة. وبالتالي، يبقى البرنامج يشتغل دون توقف ever-running program.
- ▶ وعليه، يتم إيقاف البرنامج باستخدام Platform.exit() أو system.exit() داخل start() لمعالجة مشكلة ever-running program.

▶ 18



## مثال عرض Stage الرئيسية

```
public class RunningApp extends Application {
    @Override
    public void start(Stage primaryStage) {
        //Platform.exit();
        primaryStage.show();
        primaryStage.close();
    }

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        launch(args);
    }
}
```

► يمكن أيضا إنهاء البرنامج السابق باستخدام الدالة `close()` مع stage ولكن يجب أن يتم عرض stage أولاً.

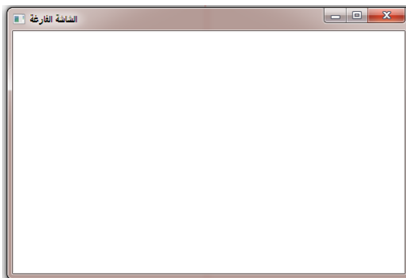
► لأن الدالة `close()` لا تقوم بإنهاء stage طالما أن stage غير نشطة.

► ملاحظة: في المثال الموضح سيتم إنهاء البرنامج مباشرة بعد تنفيذه.

19



## مثال تحديد حدود Stage



```
import javafx.application.Application;
import javafx.stage.Stage;

/**
 *
 * @author Hassan
 */
public class RunningApp extends Application {

    @Override
    public void start(Stage primaryStage) {
        //Platform.exit();
        primaryStage.setTitle("النافذة الفارغة");
        primaryStage.show();
        //primaryStage.close();
    }

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        launch(args);
    }
}
```

20



## مثال تحديد حدود Stage

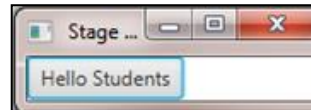
▶ سنقوم بالتعديل على البرنامج السابق بإضافة زر Button داخل Group ثم يتم اضافتها إلى Scene.

▶ تكون start() المعدلة كما يلي:

```
import javafx.application.Application;
import javafx.scene.Group;
import javafx.scene.Scene;
import javafx.scene.control.Button;
import javafx.stage.Stage;

/**
 *
 * @author Hassan
 */
public class RunningApp extends Application {

    @Override
    public void start(Stage primaryStage) {
        primaryStage.setTitle(" Stage with Button");
        Group root = new Group(new Button("Hello Students"));
        Scene scene = new Scene(root);
        primaryStage.setScene(scene);
        primaryStage.show();
        //primaryStage.close();
    }
}
```



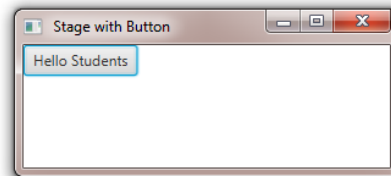
## مثال تحديد حدود Stage

▶ لضبط عرض وارتفاع Scene إلى 300 و 100، يكون على النحو التالي:

```
import javafx.application.Application;
import javafx.scene.Group;
import javafx.scene.Scene;
import javafx.scene.control.Button;
import javafx.stage.Stage;

/**
 *
 * @author Hassan
 */
public class RunningApp extends Application {

    @Override
    public void start(Stage primaryStage) {
        primaryStage.setTitle(" Stage with Button");
        Group root = new Group(new Button("Hello Students"));
        Scene scene = new Scene(root, 300, 100);
        primaryStage.setScene(scene);
        primaryStage.show();
        //primaryStage.close();
    }
}
```





## مثال تحديد حدود Stage

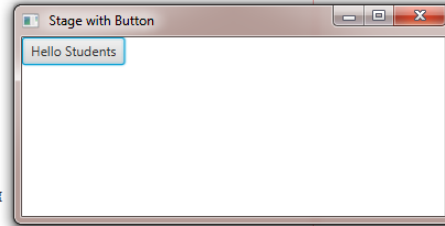
يمكن التعديل في حجم stage و Scene باستخدام الكود التالي:

```
import javafx.application.Application;
import javafx.scene.Group;
import javafx.scene.Scene;
import javafx.scene.control.Button;

import javafx.stage.Stage;

/**
 *
 * @author Hassan
 */
public class RunningApp extends Application {

    @Override
    public void start(Stage primaryStage) {
        primaryStage.setTitle(" Stage with Button");
        Group root = new Group(new Button("Hello Students"));
        Scene scene = new Scene(root, 300, 100);
        primaryStage.setScene(scene);
        primaryStage.setWidth(400);
        primaryStage.setHeight(200);
        primaryStage.show();
        //primaryStage.close();
    }
}
```



23



## عرض Stage في وضع ملء الشاشة

يمكن وضع Stage في وضع ملء الشاشة full-screen عن طريق استدعاء `.setFullScreen(true)`

عندما يتم عرض Stage في وضع ملء الشاشة، يتم عرض رسالة مختصرة حول كيفية الخروج من وضع ملء الشاشة، يتم الضغط على مفتاح ESC للخروج من وضع ملء الشاشة.

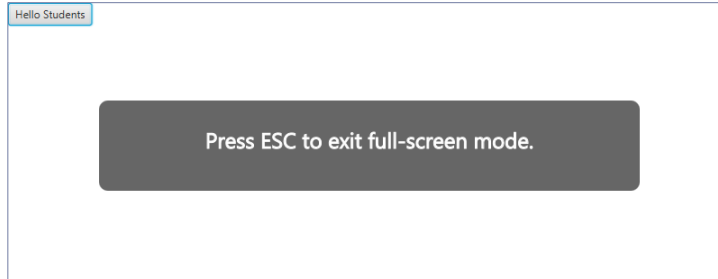
يمكنك الخروج من وضع ملء الشاشة برمجياً عن طريق استدعاء `.setFullScreen(false)`

كما يمكن استخدام `isFullScreen()` للتحقق مما إذا كانت stage في وضع ملء الشاشة او لا.

24



## عرض Stage في وضع ملء الشاشة



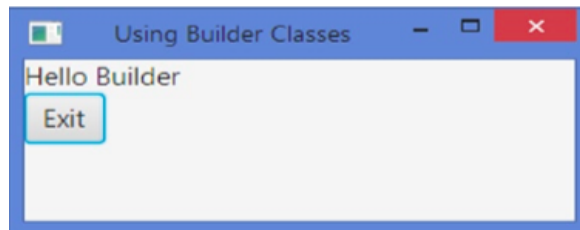
```
public class RunningApp extends Application {
    @Override
    public void start(Stage primaryStage) {
        primaryStage.setTitle(" Stage with Button");
        Group root = new Group(new Button("Hello Students"));
        Scene scene = new Scene(root, 300, 100);
        primaryStage.setScene(scene);
        // primaryStage.setWidth(400);
        // primaryStage.setHeight(200);
        primaryStage.setFullScreen(true);
        primaryStage.show();
        //primaryStage.close();
    }
}
```

▶ 25



## تمرين

▶ أكتب برنامج لعرض واجهة المستخدم التالية:



▶ 26

نهاية المحاضرة

