



جامعة طرابلس
كلية تقنية المعلومات
قسم هندسة البرمجيات



البرمجة المرئية Visual Programming خريف 2024

المحاضرة الرابعة
Layout Panes – الجزء الأول



مواضيع المحاضرة

- ما هو Layout Pane
- الفئات Class داخل Layout Pane
- Group
 - بناء الفئة المجموعة Group Class
- الفئة Hbox Class
 - بناء الفئة Hbox Class
 - خصائص الفئة Hbox Class
- الفئة VBox Class
 - بناء الفئة VBox Class
 - خصائص الفئة VBox Class
- الفئة BorderPane
- الفئة GridPane Class
 - إنشاء الفئة GridPane Class
- تمارين



Layout Panes - مقدمة

- Container layout هو ترتيب الأبناء داخل الاب بعد تعريفهم.
- JavaFX توفر العديد من layouts مثل: Hbox و VBox و Border Pane و Stack Pane و Text Flow و Anchor Pane و Title Pane و Grid Pane و Flow Pane الخ
- جميع Layouts تمثل بفئات تنتمي للحزمة javafx.scene.layout

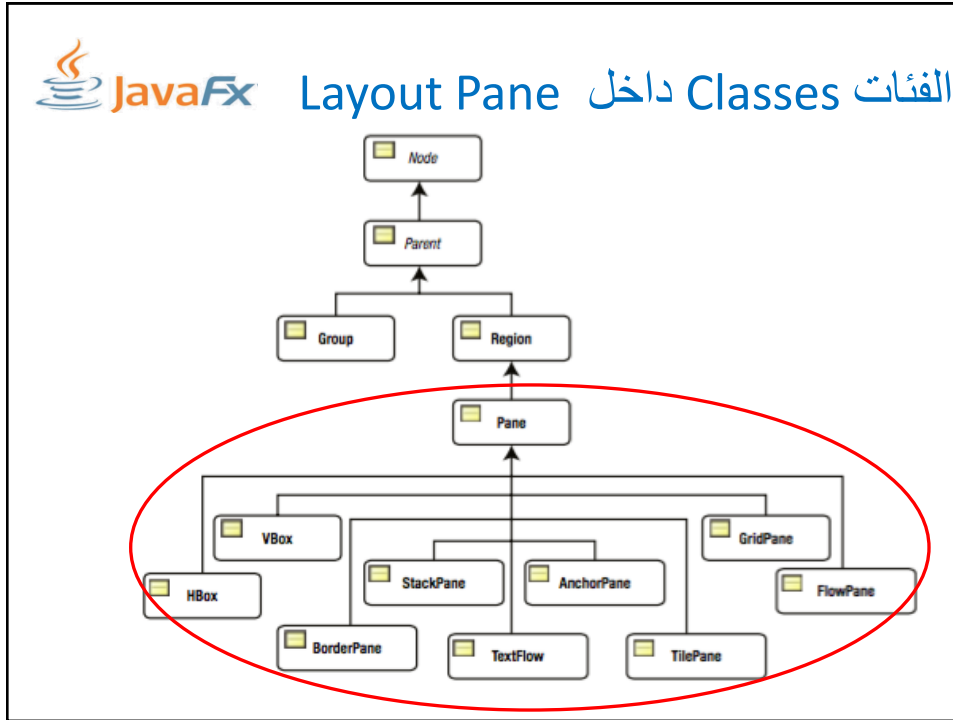
3



Layout Panes - مقدمة

- لإنشاء Layout نحتاج للخطوات التالية:
 - تكوين الأبناء nodes
 - مثلا `Button stopButton = new Button("stop");`
 - تعريف الفئة الصحيحة التي تناسب layout المستخدم
 - مثلا `HBox hbox = new HBox();`
 - إضافة خصائص layout
 - مثلا `hbox.setSpacing(10);`
 - إضافة جميع الأبناء المنشئين سابقا الى layout
 - مثلا `hbox.getChildren().add(stopButton)`

4



JavaFx Group

- Group لها ميزات Layout لكن ليست Layout. فهي فئة فرعية من فئة الأصل Parent.
- هي عقدة جماعية تحتوي على قائمة بالعقد الفرعية.
- يعرض العقد nodes بالترتيب الذي تمت إضافتهم به.
- يتم وضع جميع العقد عند الاحداثيات (0، 0) بشكل افتراضي.
- يستخدم خصائص layoutX و layoutY لتحديد موقع كل ابن.
- بشكل افتراضي، يقوم بتغيير حجم جميع الابناء إلى الحجم المفضل لديهم.

6



Group

- يتم استخدام دالة الانشاء constructor ليتم انشاء كائن من group باستخدام عدة طرق منها:
– انشاء Group بدون تحديد الأبناء:

```
Group emptyGroup = new Group();
```

- إنشاء Group بعد انشاء الابناء و اضافتهم إليه:

```
Button smallBtn = new Button("Small Button");
```

```
Button bigBtn = new Button("This is a big button");
```

```
Group group1 = new Group(smallBtn, bigBtn);
```

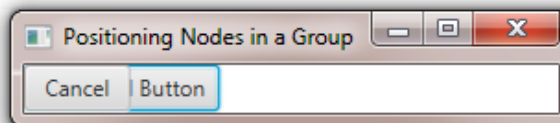
- تذكر اضافة المكتبة: `import javafx.scene.Group;`

7



مثال

```
public void start(Stage stage) {
    // Create two buttons
    Button okBtn = new Button(" OK Small Button");
    Button cancelBtn = new Button(" Cancel ");
    Group root = new Group();
    root.getChildren().addAll(okBtn, cancelBtn);
    Scene scene = new Scene(root);
    stage.setScene(scene);
    stage.setTitle("Positioning Nodes in a Group");
    stage.show();
}
}
```

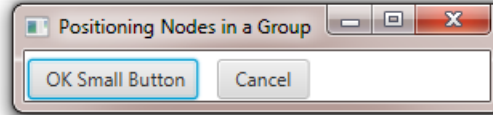


8



مثال

```
public void start(Stage stage) {
    // Create two buttons
    Button okBtn = new Button(" OK Small Button");
    Button cancelBtn = new Button(" Cancel ");
    // Set the location of the OK button
    okBtn.setLayoutX(2);
    okBtn.setLayoutY(5);
    cancelBtn.setLayoutX(120);
    cancelBtn.setLayoutY(5);
    Group root = new Group();
    root.getChildren().addAll(okBtn, cancelBtn);
    Scene scene = new Scene(root);
    stage.setScene(scene);
    stage.setTitle("Positioning Nodes in a Group");
    stage.show();
}
}
```



9



HBox Layout

- تضع الفئة HBox أبنائها في صف أفقي واحد بدءاً من الجانب الأيسر الأعلى للواجهة.
- يتيح إمكانية ضبط التباعد الأفقي بين الأبناء المتجاورة، وكذلك الهوامش.
- يستخدم القيمة صفر كتباعد افتراضي بين الأبناء المتجاورة.
- العرض Width الافتراضي للمنطقة التي بداخله كافيه لعرض جميع الأبناء الخاصة به في العرض المفضل لديهم.
- الارتفاع Height الافتراضي هو أكبر من ارتفاع جميع الأبناء.
- لا يمكنك تعيين المواقع للأبناء في HBox، حيث يتم حسابها تلقائياً بواسطة HBox.

10



HBox Layout

- يتم استخدام دالة الانشاء Constructor لإنشاء الكائن HBox object باستخدام عدة طرق منها:
 - انشاء Hbox بدون تحديد المسافة بين الابناء.
`HBox hbox1 = new HBox();`
 - انشاء Hbox مع تحديد المسافة 10 بين الابناء.
`HBox hbox2 = new HBox(10);`
 - انشاء Hbox مع اضافة الابناء و تحديد المسافة 10 بينهم.
`Button okBtn = new Button("OK");`
`Button cancelBtn = new Button("Cancel");`
`HBox hbox3 = new HBox(10, okBtn, cancelBtn);`

11



خصائص HBox Layout

- **alignment**: محاذاة الأبناء في حدود Hbox Layout ويمكن تخصيص قيمة لها باستخدام `setAlignment()` حيث أن القيمة الافتراضية هي `Pos.TOP_LEFT`
- **fillHeight**: هذه الخاصية تحدد ما اذا كان على الأبناء ان يتم تغيير حجمهم لملائمة حجم layout ام يبقون على احجامهم المحددة. يمكن ضبط قيمة هذه الخاصية باستخدام `setFillHeight()` حيث أن القيمة الافتراضية هي `Boolean.True`
- **spacing**: تمثل المسافة بين الأبناء ويمكن تعيين قيمة لهذه الخاصية باستخدام `setSpacing()` والقيمة الافتراضية لها هو صفر
- **setHgrow()**: ضبط أولوية الزيادة الأفقية للأبناء بحيث تقبل الدالة اسم الابن وقيمة الأولوية.
- **setMargin()**: لتعيين هوامش HBox Layout.

12

مثال

```

import javafx.application.Application;
import javafx.geometry.Insets;
import javafx.scene.Scene;
import javafx.scene.layout.HBox;
import javafx.scene.shape.Rectangle;
import javafx.stage.Stage;

public class Lect4Ex1 extends Application {

    @Override
    public void start(Stage primaryStage) {
        HBox hbox = new HBox(50); // space between child nodes only
        hbox.setPadding(new Insets(10)); // padding between hbox border and row.
        Rectangle r1 = new Rectangle(100, 100); // Square
        Rectangle r2 = new Rectangle(200, 200); // Big Square
        Rectangle r3 = new Rectangle(50, 200); // Vertical rectangle
        Rectangle r4 = new Rectangle(200, 50); // Horizontal rectangle
        HBox.setMargin(r1, new Insets(20,20,20,20)); // space between border & child node
        hbox.getChildren().addAll(r1, r2, r3, r4);

        Scene scene = new Scene(hbox);

        primaryStage.setTitle("HBox Example");
        primaryStage.setScene(scene);
        primaryStage.show();
    }

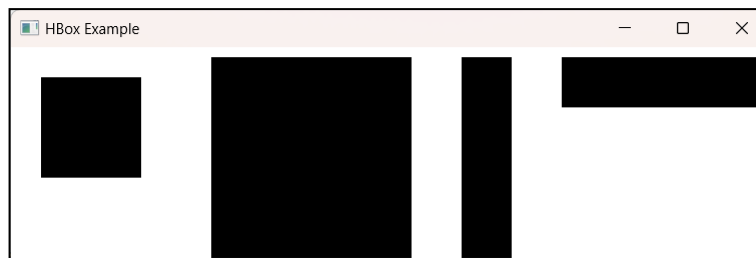
    public static void main(String[] args) {
        launch(args);
    }
}

```

13



مثال



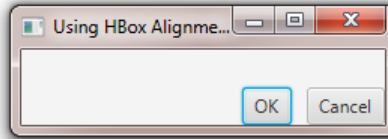
Width = 760 ??
Height = 240 ??

14



مثال

```
public void start(Stage stage) {
    Button okBtn = new Button("OK");
    Button cancelBtn = new Button("Cancel");
    //Hbox with spacing 10 between button
    HBox hbox = new HBox(10);
    hbox.setPrefSize(100, 50);
    hbox.getChildren().addAll(okBtn, cancelBtn);
    // Set the alignment to bottom right
    hbox.setAlignment(Pos.BOTTOM_RIGHT);
    Scene scene = new Scene(hbox);
    stage.setScene(scene);
    stage.setTitle("Using HBox Alignment Property");
    stage.show();
}
```

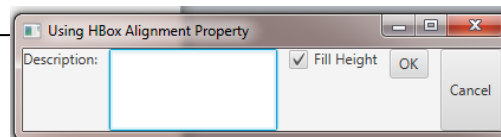


15



مثال

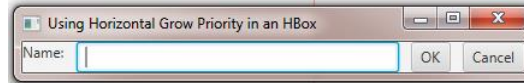
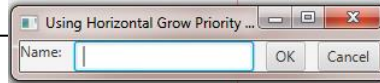
```
public void start(Stage stage) {
    HBox root = new HBox(10); // 10px spacing
    Label descLbl = new Label("Description:");
    TextArea desc = new TextArea();
    desc.setPrefColumnCount(10); //number cols
    desc.setPrefRowCount(3); // number rows
    Button okBtn = new Button("OK");
    Button cancelBtn = new Button("Cancel");
    // Let the Cancel button expand vertically
    cancelBtn.setMaxHeight(Double.MAX_VALUE);
    CheckBox fillHeightCb = new CheckBox("Fill Height");
    fillHeightCb.setSelected(true);
    root.getChildren().addAll(descLbl, desc, fillHeightCb, okBtn, cancelBtn);
    Scene scene = new Scene(root);
    stage.setScene(scene);
    stage.setTitle("Using HBox Alignment Property");
    stage.show();
}
```





مثال

```
public void start(Stage stage) {
    Label nameLbl = new Label("Name:");
    TextField nameFld = new TextField();
    Button okBtn = new Button("OK");
    Button cancelBtn = new Button("Cancel");
    HBox root = new HBox(10);
    root.getChildren().addAll(nameLbl, nameFld, okBtn, cancelBtn);
    // Let the TextField always grow horizontally
    HBox.setHgrow(nameFld, Priority.ALWAYS);
    Scene scene = new Scene(root);
    stage.setScene(scene);
    stage.setTitle("Using Horizontal Grow Priority in an HBox");
    stage.show();
}
```



17



VBox Layout

- هذه الفئة مشابهة للفئة السابقة HBOX مع وجود اختلاف بسيط.
- تضع الفئة VBox ابنائها في عمود رأسي واحد، كما تتيح ضبط التباعد الرأسي بين الابناء المتجاورين، والهوامش.
- تستخدم القيمة صفر كمسافة افتراضية بين ابناءها.
- الارتفاع الافتراضي لمساحة المحتوى في VBox مرتفع بما يكفي لعرض جميع الابناء بارتفاعاتهم المفضلة.
- العرض الافتراضي هو أكبر من عرض لجميع الابناء.
- لا يمكنك تعيين مواقع الابناء في VBox بل يتم حسابها تلقائيًا بواسطة VBox.

18



VBox Layout

- يتم استخدام الدالة Constructor لإنشاء الكائن VBox Object باستخدام عدة طرق منها:
 - إنشاء VBox بدون تحديد المسافة بين الأبناء.
`VBox vbox1 = new VBox();`
 - إنشاء VBox مع تحديد المسافة 10 بين الأبناء.
`VBox vbox2 = new VBox(10);`
 - إنشاء VBox مع إضافة الأبناء و تحديد المسافة 10 بينهم.
`Button okBtn = new Button("OK");`
`Button cancelBtn = new Button("Cancel");`
`VBox vbox3 = new VBox(10, okBtn, cancelBtn);`

19



خصائص VBox Layout

- **alignment**: محاذاة الأبناء في حدود VBox Layout ويمكن تخصيص قيمة لها باستخدام `setAlignment()` حيث أن القيمة الافتراضية هي `Pos.TOP_LEFT`
- **fillWidth**: هذه الخاصية تحدد ما إذا كان على الأبناء ان يتم تغيير حجمهم لملائمة حجم layout ام يبقون على احجامهم المحددة. يمكن ضبط قيمة هذه الخاصية باستخدام `setFillWidth()` حيث أن القيمة الافتراضية هي `Boolean.TRUE`
- **spacing**: تمثل المسافة بين الأبناء ويمكن تعيين قيمة لهذه الخاصية باستخدام `setSpacing()` والقيمة الافتراضية لها هو صفر
- **setVgrow()**: ضبط أولوية الزيادة العمودية للأبناء بحيث تقبل الدالة اسم الابن وقيمة الأولوية.
- **setMargin()**: لتعيين هوامش VBox Layout.

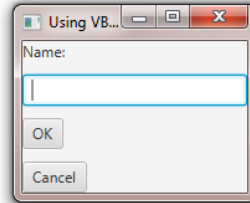
20



مثال

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

public class VBoxTest extends Application {
    public static void main(String[] args) {
        Application.launch(args);
    }
    @Override
    public void start(Stage stage) {
        Label nameLbl = new Label("Name:");
        TextField nameFld = new TextField();
        Button okBtn = new Button("OK");
        Button cancelBtn = new Button("Cancel");
        VBox root = new VBox(10); // 10px spacing
        root.getChildren().addAll(nameLbl, nameFld, okBtn, cancelBtn);
        Scene scene = new Scene(root);
        stage.setScene(scene);
        stage.setTitle("Using VBox");
        stage.show();
    }
}
```



21

```
import javafx.application.Application;
import javafx.collections.ObservableList;
import javafx.geometry.Insets;
import javafx.scene.Scene;
import javafx.scene.control.Button;
import javafx.scene.control.TextField;
import javafx.stage.Stage;
import javafx.scene.layout.VBox;
```

```
public class Lect4Ex2 extends Application {
    public void start(Stage stage) {

        TextField textField = new TextField();
        Button playButton = new Button("Play");
        Button stopButton = new Button("stop");
```

```
VBox vbox = new VBox();

vbox.setSpacing(10);
```

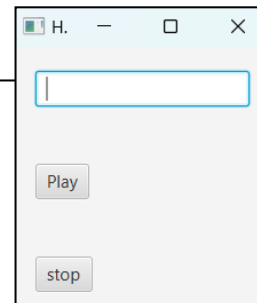
```
vbox.setMargin(textField, new Insets(20, 20, 20, 20));
vbox.setMargin(playButton, new Insets(20, 20, 20, 20));
vbox.setMargin(stopButton, new Insets(20, 20, 20, 20));
```

```
ObservableList list = vbox.getChildren();

list.addAll(textField, playButton, stopButton);
```

```
Scene scene = new Scene(vbox);
stage.setTitle("Hbox Example");
stage.setScene(scene);
stage.show();
}
```

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

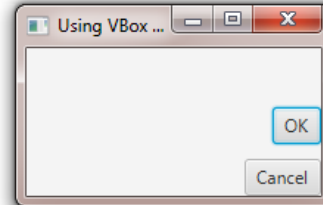


مثال



مثال

```
public void start(Stage stage) {
    Button okBtn = new Button("OK");
    Button cancelBtn = new Button("Cancel");
    VBox vbox = new VBox(10);
    vbox.setPrefSize(200, 100);
    vbox.getChildren().addAll(okBtn, cancelBtn);
    // Set the alignment to bottom right
    vbox.setAlignment(Pos.BOTTOM_RIGHT);
    Scene scene = new Scene(vbox);
    stage.setScene(scene);
    stage.setTitle("Using VBox Alignment Property");
    stage.show();
}
```

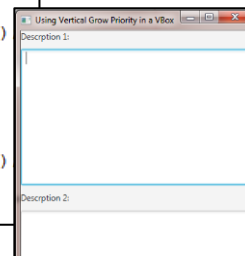
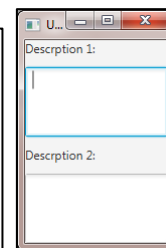


24



مثال

```
public void start(Stage stage) {
    Label descLb1 = new Label("Description 1:");
    TextArea desc = new TextArea();
    desc.setPrefColumnCount(10);
    desc.setPrefRowCount(3);
    Label descLb2 = new Label("Description 2:");
    TextArea desc1 = new TextArea();
    desc1.setPrefColumnCount(10);
    desc1.setPrefRowCount(3);
    VBox root = new VBox(10);
    root.getChildren().addAll(descLb1, desc, descLb2, desc1);
    // Let the TextArea always grow vertically
    VBox.setVgrow(desc, Priority.ALWAYS);
    Scene scene = new Scene(root);
    stage.setScene(scene);
    stage.setTitle("Using Vertical Grow Priority in a VBox");
    stage.show();
}
```



25



BorderPane Layout

- عند استخدام BorderPane فان جميع العقد تنظم في المواضع الأعلى واليسار واليمين والوسط.
- الفئة BorderPane تتبع الحزمة javafx.scene.layout
- يمكن استخدام الدوال setBottom() و setCenter() و setLeft() و setRight() و setTop() لتخصيص قيم center و bottom و left و right و top.

26

```
import javafx.scene.Scene;
import javafx.scene.control.TextField;
import javafx.scene.layout.BorderPane;
import javafx.stage.Stage;

public class Lect4Ex3 extends Application {

    @Override
    public void start(Stage stage) {

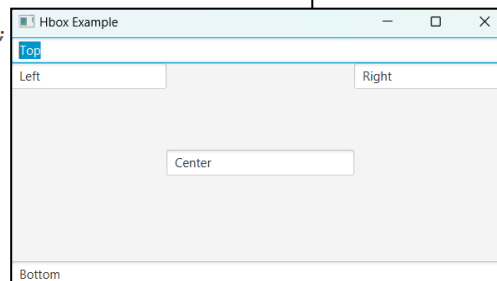
        BorderPane bPane = new BorderPane();

        //Setting the top, bottom, center, right and left nodes to the pane
        bPane.setTop(new TextField("Top"));
        bPane.setBottom(new TextField("Bottom"));
        bPane.setLeft(new TextField("Left"));
        bPane.setRight(new TextField("Right"));
        bPane.setCenter(new TextField("Center"));

        Scene scene = new Scene(bPane, 600, 300);
        stage.setTitle("Hbox Example");
        stage.setScene(scene);
        stage.show();
    }

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

مثال





GridPane Layout

- يضع GridPane ابنائه في شبكة من الخلايا مرتبة في صفوف وأعمدة.
- فئة GridPane تتبع الحزمة `javafx.scene.layout`
- لها الخصائص `alignment` و `hgap` (الفراغ الافقي بين الاعمدة) و `vgap` (الفراغ العمودي بين الصفوف) و `gridLinesVisible` و (ظهور الخطوط) التي لها القيمة الافتراضية `false`

28



GridPane Layout

- في GridPane، بشكل افتراضي، يتم إضافة جميع الابناء في الخلية الأولى $(c0, r0)$ التي تمتد على عمود واحد وصف واحد، وبالتالي تتداخل الابناء مع بعضها البعض.



29



إنشاء الكائن GridPane Object

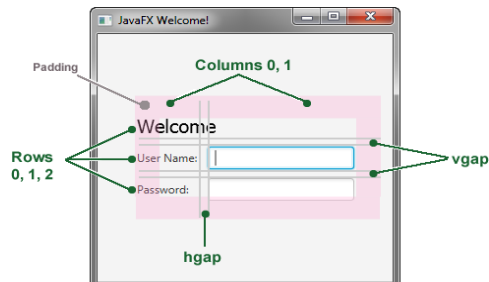
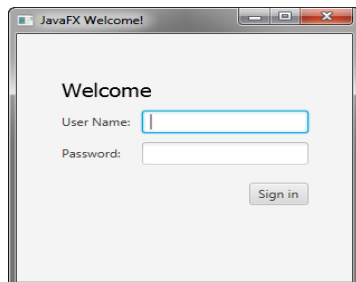
- يتم استخدام الدالة constructor لإنشاء الكائن GridPane بدون ارسال معاملات كالتالي:
- `GridPane gpane = new GridPane();`
- حيث يتم إنشاء GridPane فارغة بمسافة صفر بين الصفوف والأعمدة، مع وضع الابناء التي سيتم إضافتهم لاحقاً في الزاوية العلوية اليسرى داخل منطقة المحتوى الخاصة بها.

30



GridPane Layout

- يعد إنشاء نموذج form نشاطاً شائعاً عند تطوير تطبيق ما



31



```

/* Set constraints for children to customize their resizing behavior */
// The max width of the OK button should be big enough,
// so it can fill the width of its cell
okBtn.setMaxWidth(Double.MAX_VALUE);
// The name field in the first row should grow horizontally
GridPane.setHgrow(nameFld, Priority.ALWAYS);
// The description field in the third row should grow vertically
GridPane.setVgrow(descText, Priority.ALWAYS);
// The status bar in the last should fill its cell
statusBar.setMaxWidth(Double.MAX_VALUE);
Scene scene = new Scene(root);
stage.setScene(scene);
stage.setTitle("Creating Forms Using a GridPane");
stage.show();

```

```


public void start(Stage stage) {
    // A Label and a TextField
    Label nameLbl = new Label("Name:");
    TextField nameFld = new TextField();
    // A Label and a TextArea
    Label descLbl = new Label("Description:");
    TextArea descText = new TextArea();
    descText.setPrefColumnCount(20);
    descText.setPrefRowCount(5);
    // Two buttons
    Button okBtn = new Button("OK");
    Button cancelBtn = new Button("Cancel");
    // A Label used as a status bar
    Label statusBar = new Label("Status: Ready");
    // Create a GridPane and set its background color to lightgray
    GridPane root = new GridPane();
    // Add children to the GridPane
    root.add(nameLbl, 0, 0, 1, 1); // (c0, r0, colspan=1, rowspan=1)
    root.add(nameFld, 1, 0, 1, 1); // (c1, r0, colspan=1, rowspan=1)
    root.add(descLbl, 0, 1, 3, 1); // (c0, r1, colspan=3, rowspan=1)
    root.add(descText, 0, 2, 2, 1); // (c0, r2, colspan=2, rowspan=1)
    root.add(okBtn, 2, 0, 1, 1); // (c2, r0, colspan=1, rowspan=1)
    root.add(cancelBtn, 2, 1, 1, 1); // (c2, r1, colspan=1, rowspan=1)
}

```



مثال

33

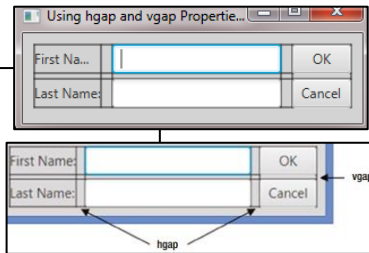


مثال

```

public void start(Stage stage) {
    Label fnameLbl = new Label("First Name:");
    TextField fnameFld = new TextField();
    Label lnameLbl = new Label("Last Name:");
    TextField lnameFld = new TextField();
    Button okBtn = new Button("OK");
    Button cancelBtn = new Button("Cancel");
    // The Ok button should fill its cell
    okBtn.setMaxWidth(Double.MAX_VALUE);
    // Create a GridPane and set its background color to lightgray
    GridPane root = new GridPane();
    root.setGridLinesVisible(true); // Make grid lines visible
    root.setHgap(10); // hgap = 10px
    root.setVgap(5); // vgap = 5px
    root.setStyle("-fx-padding: 10;-fx-background-color: lightgray;");
    // Add children to the GridPane
    root.addRow(0, fnameLbl, fnameFld, okBtn);
    root.addRow(1, lnameLbl, lnameFld, cancelBtn);
    Scene scene = new Scene(root);
    stage.setScene(scene);
    stage.setTitle("Using hgap and vgap Properties for a GridPane");
    stage.show();
}

```



34



تمارين

نموذج الاشتراك

اسم العضو:

البريد:

تأكيد الاشتراك

قبول

إلغاء

نموذج إدخال البيانات

الاسم:

العنوان:

تأكيد

موافق

إلغاء

Registration Form (FXML)

Full Name:

Email ID:

Password:

Hbox Example

35

نهاية المحاضرة الجزء الأول



36