جامعة طرابلس
كلية تقنية المعلومات

البرمجة المرئية
Visual Programming

ITSE423

إعداد: حسن علي حسن إبراهيم
المحاضر: فاطمة علي بن الأشهر

المحاضرة الرابعة - التخطيط Layout Pane

الجزء الأول

مواضيع المحاضرة

- ما هو جزء التخطيط Layout Pane
- إضافة Children إلى Layout Pane
- الفئات Class داخل Layout Pane
- الفئة المجموعة Group Class
  - بناء الفئة المجموعة Group Class
- الفئة Hbox Class
  - خصائص الفئة Hbox Class
  - بناء الفئة Hbox Class
- الفئة Vbox Class
  - خصائص الفئة Vbox Class
  - بناء الفئة Vbox Class
- الفئة GridPane Class
  - إنشاء الفئة GridPane Class
  - اضافة الابناء إلى الفئة GridPane Class

ما هو جزء التخطيط Layout Pane

- جزء التخطيط *layout pane* هو عقدة node تحتوي على عقد أخرى، تعرف باسم الابن children أو child nodes.
- تتمثل مسؤولية layout pane في تخطيط children عند الحاجة.
- يعرف جزء التخطيط layout pane أيضًا باسم الحاوية *container* أو حاوية التخطيط *layout container*.
- يحتوي جزء التخطيط على سياسة التخطيط *layout policy* والتي تتحكم في كيفية وضع children. على سبيل المثال ، قد يتم وضع العقد أفقياً أو رأسياً أو بأي شكل آخر.

Hassan Ebrahem

3

جزء التخطيط Layout Pane

- تحتوي JavaFX على العديد من الفئات classes المتعلقة بالتخطيط.
- **يؤدي جزء التخطيط layout pane وظيفتين:**
- يحسب موضع position العقدة node باستخدام إحداثيات x و y داخل parent.
- يحسب حجم size العقدة node (العرض width والارتفاع height).

Hassan Ebrahem

4

إضافة Children إلى Layout Pane

- الحاوية container هي مخصصة لاحتواء children. يمكن إضافة العقدة children إلى الحاوية عند إنشاء كائن object الحاوية أو بعد إنشائه.
- توفر بعض الحاويات دوال إنشاء constructors وذلك لإضافة مجموعة من children وكذلك تعيين خصائص properties للحاويات.
- جميع أنواع التخطيط تحتاج الى
- **javafx.scene.layout**

إضافة Children إلى Layout Pane

- **لإنشاء تخطيط Layout، تحتاج إلى :**

1. إنشاء العقد Node.

- Button stopButton = new Button("stop");

2. إنشاء فئة معينة من التخطيط المطلوب.

- HBox hbox = new HBox();

3. تعيين خصائص التخطيط.

- hbox.setSpacing(10);

4. أضف جميع العقد التي تم إنشاؤها إلى التخطيط.

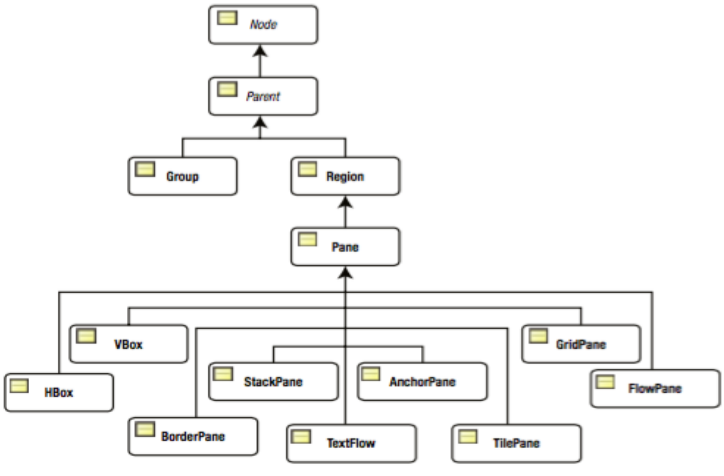
- HBox hbox = new HBox(10, stopButton);

إضافة Children إلى Layout Pane

- يمكن أيضًا إضافة الابناء children إلى الحاوية container في أي وقت بعد إنشاء الحاوية.
- تخزن الحاويات أبنائها في قائمة تسمى observable list، والتي يمكن استرجاعها باستخدام getChildren(). مثال
  - root.getChildren().add(okBtn);
- تلميح Tip: عندما تحتاج إلى إضافة عدة عقد nodes إلى حاوية، يمكن استخدم الطريقة addAll() من ObservableList بدلاً من استخدام الطريقة add() عدة مرات.
  - root.getChildren().addAll(smallBtn, bigBtn);

الفئات Class داخل Layout Pane

A class diagram for container classes in JavaFX



مخطط فئة لفئات الحاوية container classes في JavaFX

الفئات Class في Layout Pane

- تحتوي JavaFX على العديد من فئات الحاويات container classes نعرض منها:
- Group هذا التخطيط يقوم يضع العقد على بعض.
- Hbox هذا التخطيط يرتب أبنائه أفقياً في صف واحد.
- Vbox هذا التخطيط يرتب أبنائه عموديًا في عمود واحد.
- GridPane يرتب أبنائه في شبكة من الخلايا المتغيرة الحجم، صفوف وأعمدة.

فئة المجموعة Group Class

- المجموعة Group هي عقدة جماعية تحتوي على قائمة بالعقد الفرعية.
- المجموعة لها ميزات الحاوية؛ على سبيل المثال ، لديها سياسة التخطيط layout policy الخاصة بها، ونظام الإحداثيات، وهي فئة فرعية من فئة الأصل Parnet.
- ومع ذلك ، ينعكس معناها بشكل أفضل من خلال تسميتها مجموعة من العقد *collection of nodes* أو المجموعة *group*، بدلاً من كونها حاوية *container*.
- أي تطبيق من التحولات والتأثيرات والخصائص على مجموعة group يتم على جميع العقد في المجموعة.

فئة المجموعة Group Class

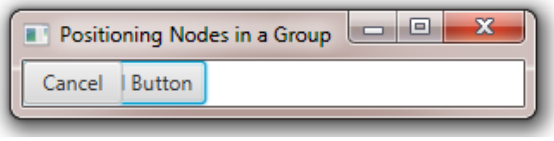
- تمتلك المجموعة Group سياسة تخطيط خاصة بها، والتي لا توفر أي تخطيط محدد لأبنائها، باستثناء منحهم الحجم المفضل لديهم:
- يعرض العقد nodes بالترتيب الذي تمت إضافتهم به.
- يتم وضع جميع children عند الاحداثيات (0، 0) بشكل افتراضي.
- يستخدم خصائص layoutX وlayoutY لوضع children داخل المجموعة Group.
- بشكل افتراضي، يقوم بتغيير حجم جميع الابناء إلى الحجم المفضل لديهم.

انشاء الكائن المجموعة Group Object

- يتم استخدام دالة الانشاء constructor ليتم انشاء كائن Object من group باستخدام عدة طرق منها:
- انشاء Group بدون تحديد الابناء.
  - Group emptyGroup = new Group();
- إنشاء Group بعد انشاء الابناء واضافتهم 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;

مثال لفئة المجموعة Group Class

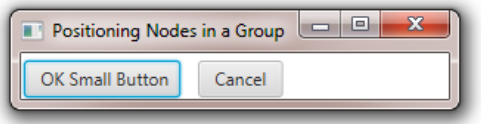
```
public void start(Stage stage) {
// Create two buttons
    Button okBtn = new Button(" OK Small Button");
    Button cancelBtn = new Button(" Cancel ");
    Group root = new Group();
    root.getChildren().addAll(okBtn, cancelBtn);
    Scene scene = new Scene(root);
    stage.setScene(scene);
  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



إذا كنت لا تريد أن تتداخل العقد في المجموعة ، فأنت بحاجة إلى تعيين مواضعها.
وذلك باستخدام خصائص layoutX و layoutY .

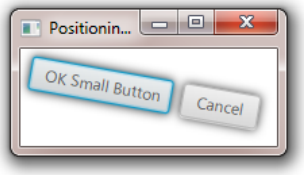
مثال لفئة المجموعة Group Class

```
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();
}
}
```



مثال لفئة المجموعة Group Class

```
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(6);
    okBtn.setLayoutY(20);
    cancelBtn.setLayoutX(120);
    cancelBtn.setLayoutY(20);
    Group root = new Group();
    root.setEffect(new DropShadow()); // Set a drop shadow effect
    root.setRotate(10); // Rotate by 10 degrees clockwise
    root.setOpacity(0.80);
    root.getChildren().addAll(okBtn, cancelBtn);
    Scene scene = new Scene(root);
    stage.setScene(scene);
    stage.setTitle("Positioning Nodes in a Group");
    stage.show();
}
```



Hassan Ebrahem

16

الفئة HBox Class

- تضع الفئة HBox أبنائها children في صف أفقي واحد.
- يتيح امكانية ضبط التباعد الأفقي بين الابناء المتجاورة، وكذلك الهوامش لأي أبناء children.
- يستخدم القيمة 0px كتباعد افتراضي بين الأبناء المتجاورة.
- العرض width الافتراضي للمنطقة التي بداخله كافيه لعرض جميع الابناء الخاصة به في العرض المفضل لديهم، والارتفاع height الافتراضي هو أكبر من ارتفاع جميع الابناء.
- لا يمكنك تعيين المواقع للابناء في HBox. حيث يتم حسابها تلقائيًا بواسطة HBox.

Hassan Ebrahem

17

خصائص الفئة HBox Class

- توجد في HBox ثلاث خصائص كالتالي:
- الخاصية Alignment نوعها ObjectProperty<Pos>
  - تحدد محاذاة الابناء بالنسبة إلى منطقة المحتوى، ويتم تجاهل الخاصية fillHeight إذا تم ضبط المحاذاة الرأسية إلى BASELINE.
  - القيمة الافتراضية لها هي Pos.TOP_LEFT.
- الخاصية fillHeight نوعها BooleanProperty
  - وهي تحدد ما إذا كان الابناء يمكن تغيير حجمهم لملأ الارتفاع الكامل Hbox أو يتم منحهم ارتفاعاتهم المفضلة.
  - يتم تجاهل هذه الخاصية إذا تم ضبط المحاذاة الرأسية إلى BASELINE، القيمة الافتراضية هي True.
- الخاصية **setSpacing()** نوعها DoubleProperty
  - تحدد التباعد الأفقي بين الابناء المتجاورين. القيمة الافتراضية هي صفر.

إنشاء الكائن HBox Object

- 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.

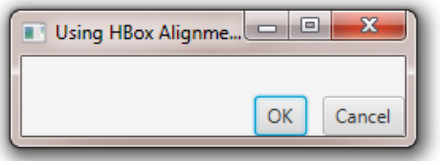
```
HBox hbox2 = new HBox(10);
```

- 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);
```

مثال عن الفئة HBox Class

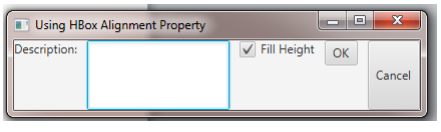
```
public void start(Stage stage) {
    Button okBtn = new Button("OK");
    Button cancelBtn = new Button("Cancel");
    //Hbox with spacing 10 between button
    HBox hbox = new HBox(10);
    hbox.setPrefSize(100, 50);
    hbox.getChildren().addAll(okBtn, cancelBtn);
    // Set the alignment to bottom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



تذكر اضافة المكتبة import javafx.scene.layout.HBox;

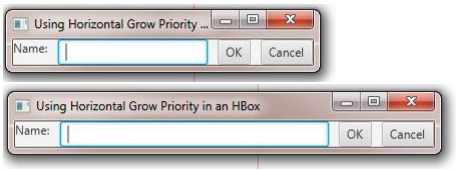
مثال عن الفئة HBox Class

```
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 fillHeightCbx = new CheckBox("Fill Height");
    fillHeightCbx.setSelected(true);
    root.getChildren().addAll( descLbl, desc, fillHeightCbx, okBtn,cancelBtn);
    Scene scene = new Scene(root);
    stage.setScene(scene);
    stage.setTitle("Using HBox Alignment Property");
    stage.show();
}
```



مثال عن الفئة HBox Class

```
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();
}
```



Hassan Ebrahem

22

الفئة VBox Class

- 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 0px كمسافة افتراضية بين الابناء المجاورة.
- الارتفاع الافتراضي لمساحة المحتوى في VBox مرتفع بما يكفي لعرض جميع الابناء به بالارتفاعات المفضلة لديهم.
- العرض الافتراضي هو أكبر من عرض لجميع الابناء.

Hassan Ebrahem

23

خصائص الفئة VBox Class

- توجد في VBox ثلاث خصائص كالتالي:
- الخاصية Alignment نوعها ObjectProperty<Pos>
  - تحدد محاذاة الابناء بالنسبة إلى منطقة المحتوى.
  - القيمة الافتراضية لها هي Pos.TOP_LEFT.
- الخاصية fillWidth نوعها BooleanProperty
  - وهي تحدد ما إذا كان الابناء يمكن تغيير حجمهم لملأ العرض الكامل VBox أو يتم منحهم ارتفاعاتهم المفضلة.
  - القيمة الافتراضية هي True.
- الخاصية **setSpacing()** نوعها DoubleProperty
  - تحدد التباعد الأفقي بين الابناء المتجاورين. القيمة الافتراضية هي صفر.

الفئة VBox Class

- لا يمكنك تعيين المواقع الابناء في VBox. يتم حسابها تلقائيًا بواسطة VBox.
- يشبه العمل مع VBox العمل مع HBox مع اختلاف أنهما يعملان في اتجاهين متعاكسين.
- يتيح لك HBox تعيين قيود hgrow على الابناء بينما يتيح لك VBox تعيين قيد vgrow.

تذكر اضافة المكتبة ;import javafx.scene.layout.VBox

إنشاء الكائن VBox Object

- يتم استخدام الدالة Constructor لانشاء الكائن VBox Object باستخدام عدة طرق منها:
- انشاء VBox بدون تحديد المسافة بين الابناء.

```
VBox vbox1 = new VBox();
```

- انشاء VBox مع تحديد المسافة 10 بين الابناء.

```
VBox vbox2 = new VBox(10);
```

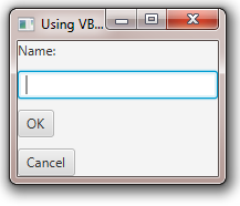
- انشاء VBox مع اضافة الابناء و تحديد المسافة 10 بينهم.

```
Button okBtn = new Button("OK");
Button cancelBtn = new Button("Cancel");
VBox Vbox3 = new VBox(10, okBtn, cancelBtn);
```


مثال الفئة VBox Class

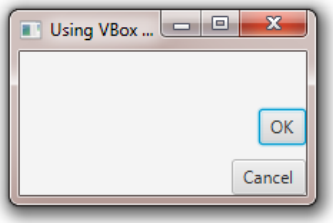
```
import javafx.application.Application;
import javafx.scene.Scene;
import javafx.scene.control.Button;
import javafx.scene.control.Label;
import javafx.scene.control.TextField;
import javafx.scene.layout.VBox;
import javafx.stage.Stage;

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();
}
}
```



VBox Class مثال الفئة

```
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();
}
```

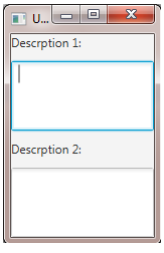


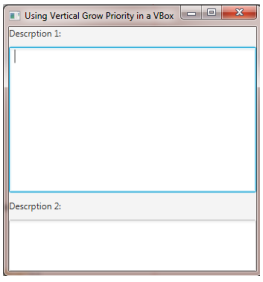
Hassan Ebrahem

28

VBox Class مثال الفئة

```
public void start(Stage stage) {
    Label descLbl = new Label("Descrption 1:");
    TextArea desc = new TextArea();
    desc.setPrefColumnCount(10);
    desc.setPrefRowCount(3);
    Label descLb2 = new Label("Descrption 2:");
    TextArea desc1 = new TextArea();
    desc1.setPrefColumnCount(10);
    desc1.setPrefRowCount(3);
    VBox root = new VBox(10);
    root.getChildren().addAll(descLbl, 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();
}
```





Hassan Ebrahem

30

GridPane Class الفئة

- تعد الفئة GridPane أحد أقوى أجزاء التخطيط layout panes، ومع القوة يأتي التعقيد. لذلك ، فإن تعلمها معقد بعض الشيء.
- يضع GridPane ابنائه في شبكة متحركة dynamic grid من الخلايا cells مرتبة في صفوف وأعمدة.
- تعتبر الشبكة متحركة لأن عدد الخلايا وحجمها في الشبكة يحدد بناء على عدد الابناء.
- يتم تحديد كل خلية Cell في الشبكة من خلال موضعها في العمود والصف.

GridPane Class الفئة

- تبدأ فهارس الأعمدة والصفوف من 0.
- يمكن وضع الابن في أي مكان في الشبكة التي تمتد لأكثر من خلية واحدة.
- جميع الخلايا في صف لها نفس الارتفاع. قد يكون للخلايا الموجودة في صفوف مختلفة ارتفاعات مختلفة.

إنشاء الكائن GridPane Object

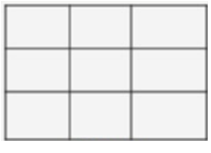
- يتم استخدام الدالة constructor لإنشاء الكائن GridPane بدون ارسال args كالتالي:
  - GridPane gpane = new GridPane();
- حيث يتم إنشاء GridPane فارغة بمسافة 0px بين الصفوف والأعمدة، مع وضع الابناء التي سيتم إضافتها لاحقًا في الزاوية العلوية اليسرى داخل منطقة المحتوى الخاصة بها.

Hassan Ebrahem

33

عرض خطوط الفئة GridPane Class

- تحتوي فئة GridPane على الخاصية gridLinesVisible من نوع Boolean ، لتتحكم في رؤية خطوط الشبكة.
- بشكل افتراضي، يتم تعيين القيمة False، وتكون خطوط الشبكة غير مرئية.
- يتم عرض الخطوط لأغراض التصحيح فقط وذلك في حالة رؤية أماكن الابناء في الشبكة.
  - GridPane gpane = new GridPane();
  - gpane.setGridLinesVisible(true); // Make grid lines visible



Hassan Ebrahem

34

اضافة الابناء إلى الفئة GridPane Class

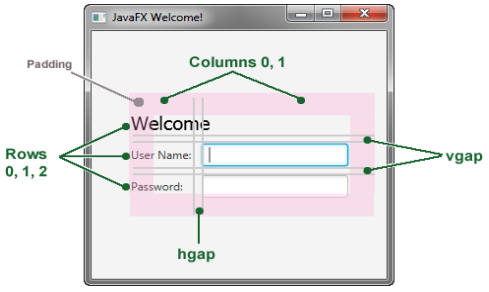
- مثل معظم أجزاء التخطيط الأخرى، حيث يقوم GridPane بتخزين الابناء الخاصة به في ObservableList<Node> التي يتم إرجاع الابناء منها بواسطة getChildren().
- في GridPane، بشكل افتراضي ، يتم إضافة جميع الابناء في الخلية الأولى (c0, r0) التي تمتد على عمود واحد وصف واحد، وبالتالي تتداخل الابناء مع بعضها البعض.

اضافة الابناء إلى الفئة GridPane Class

- تقوم الخاصية **setAlignment()** بوضع الشبكة بالكامل في موضع معين وتأخذ عدة قيم منها.center, top-center, bottom-left
- والخاصية **hgap** تقوم بوضع مسافة افقية بين العقد.
- الخاصية **vgap** تقوم بوضع مسافة عمودية بين العقد.

JavaFX GridPane Class في الفئة form انشاء

- يعد إنشاء نموذج نشاطًا شائعًا عند تطوير تطبيق.



Hassan Ebrahem

37

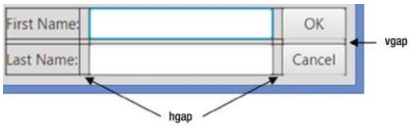
JavaFX GridPane Class

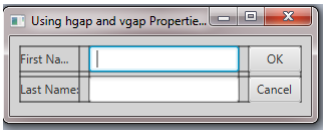

```
import javafx.application.Application;
import javafx.geometry.Insets;
import javafx.geometry.Pos;
import javafx.scene.Scene;
import javafx.scene.control.Button;
import javafx.scene.control.Label;
import javafx.scene.control.PasswordField;
import javafx.scene.control.TextField;
import javafx.scene.layout.GridPane;
import javafx.scene.layout.HBox;
import javafx.scene.text.Font;
import javafx.scene.text.Text;
import javafx.stage.Stage;
public class GridPaneForm extends Application {
public static void main(String[] args) {
    Application.launch(args);
}
@Override
public void start(Stage stage) {
    GridPane root = new GridPane();
    root.setAlignment(Pos.CENTER);
    root.setHgap(5);
    root.setVgap(5);
    root.setPadding(new Insets(20, 20, 20, 20));
    Label username = new Label("User Name:");
    TextField textfield1 = new TextField("Your User Name");
    Label password = new Label("Password:");
    PasswordField passwordfield1 = new PasswordField();
    Text text1 = new Text("Wellcome");
    text1.setFont(Font.font("Tahoma", 24));
    root.add(text1,0,0);
    HBox hb1 = new HBox();
    hb1.setAlignment(Pos.BOTTOM_RIGHT);
    hb1.setSpacing(5);
    Button btn = new Button("Login");
    Button btn1 = new Button("Cancel");
    hb1.getChildren().add(btn);
    hb1.getChildren().add(btn1);
    root.add(hb1,1,3);
    root.getChildren().add(username);
    GridPane.setConstraints(username, 0, 1);
    root.getChildren().add(textfield1);
    GridPane.setConstraints(textfield1, 1, 1);
    root.add(password,0,2);
    root.add(passwordfield1, 1,2);
    root.setGridLinesVisible(false);
    Scene scene = new Scene(root, 400, 200);
    stage.setScene(scene);
    stage.setTitle("Login Page");
    stage.show();
}
}
```

Hassan Ebrahem

مثال في الفئة GridPane Class

```
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();
}
```



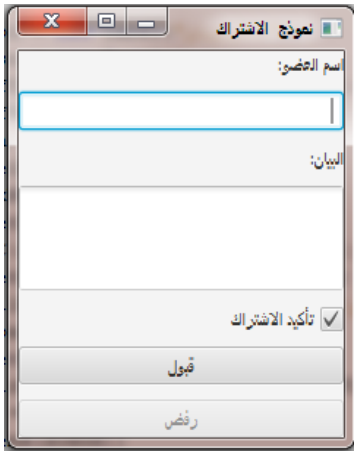


Hassan Ebrahem

41

واجب

- اكتب التعليمات البرمجية لتنفيذ واجهة المستخدم التالية باستخدام التخطيط Layout المناسب.



ملاحظة: 1- اسم البرنامج يكون باسم HomeWork ملحق برقم قيد الطالب.

2- تستبدل كلمة «تأكيد الاشتراك» برقم قيد الطالب داخل البرنامج.

Hassan Ebrahem

42

ملخص المحاضرة

- تحتوي الجافا اف اكس على مجموعة من العقد التي تحتوي على مجموعة من العقد الأخرى. تسمى هذه العقد بالحاوية *container*.
- هذه العقد الحاوية تستخدم في تنسيق الشكل.
- تم توضيح استخدامها ببعض الأمثلة.

نهاية المحاضرة
الجزء الأول