جامعة طرابلس
كلية تقنية المعلومات

البرمجة المرئية
Visual Programming

ITSE423

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

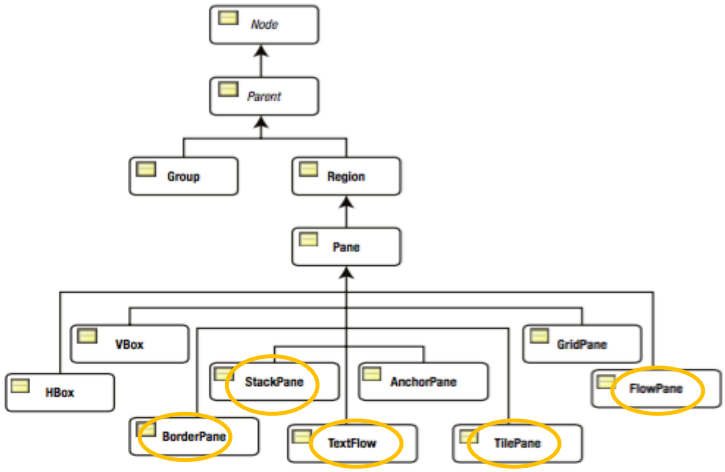
المحاضرة الرابعة ـ التخطيط Layout Pane

الجزء الثاني

مواضيع المحاضرة

- الفئة Insets Class
- الفئة FlowPane Class
  - إنشاء الفئة FlowPane Class
- الفئة TitledPane Class
  - إنشاء الفئة TitledPane Class
- الفئة Accordion
- الفئة Border Pane
  - إنشاء الفئة Border Pane
- الفئة StackPane
  - إنشاء الفئة StackPane
- الفئة TextFlow
  - إنشاء الفئة TextFlow

Layout Pane داخل Class الفئات

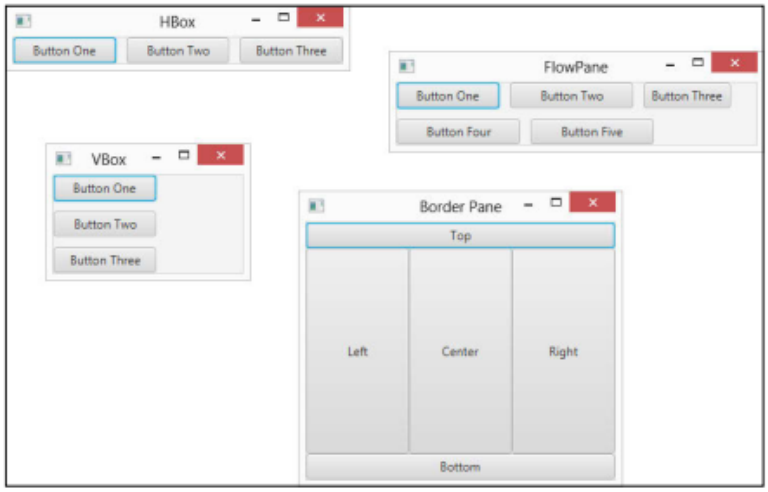


مخطط فئة لفئات الحاوية JavaFX في container classes

Layout Pane في Class الفئات

- تحتوي JavaFXعلى العديد من فئات الحاويات **container classes**
- FlowPane يرتب أبنائه أفقياً أو رأسياً في صفوف أو أعمدة. إذا لم يتم احتواؤها في صف أو عمود واحد، يتم التفافها بالعرض أو الارتفاع المحدد.
- TilePane يرتب أبنائه في شبكة من الخلايا ذات الحجم الموحد.
- BorderPane يقسم منطقة التخطيط الخاصة به إلى خمسة مناطق top, right, bottom, left, center ويضع كل من Children في إحدى المناطق الخمس.
- StackPane يرتب أبنائه في Stack من الأسفل إلى الأعلى أي فوق بعض.
- TextFlow يضع نصًا منسقًا قد يتكون محتواه من عدة عقد nodes نصية.

بعض واجهات Layout Pane



Hassan Ebrahem

5

الفئة Insets Class

- أحيانا أثناء العمل مع أجزاء من layout panes، تحتاج إلى استخدام عدة فئات Classes مرتبطة بالمسافات والاتجاهات.
- هذه الفئات ليست مفيدة عند استخدامها بشكل مستقل، يتم استخدامها دائمًا كخصائص للعقد Nodes.
- الفئة Insets:
- تمثل فئة Insets الإزاحات الداخلية لمساحة مستطيلة من أربعة اتجاهات:
  - top, right, bottom, and left.

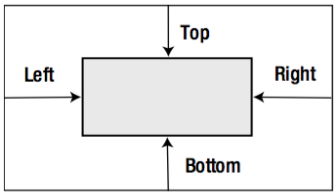
Hassan Ebrahem

6

إنشاء كائن Insets Object

- يتم استخدام دالة الانشاء constructor لانشاء كائن Object من هذه الفئة باستخدام طريقتين:
- أحدهما تتيح تعيين نفس الإزاحة لجميع الاتجاهات الأربعة بقيمة واحدة.
- الآخرى تتيح تعيين إزاحات مختلفة لكل اتجاه.

- Insets(double topRightBottomLeft)
- Insets(double top, double right, double bottom, double left)

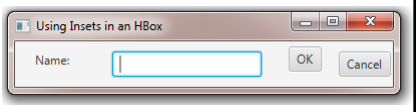


Hassan Ebrahem

7

مثال: الفئة Insets 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);
    // Set a margin for all children:
    // 10px top, 12px right, 10px bottom, and 20px left
    Insets margin = new Insets(10, 12, 10, 20);
    HBox.setMargin(nameLbl, margin);
    HBox.setMargin(nameFld, margin);
    // Set a margin of 5px around all four sides
    HBox.setMargin(okBtn, new Insets(5));
    HBox.setMargin(cancelBtn, new Insets(10, 2, 10, 2));
    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 Insets in an HBox");
    stage.show();
}
```

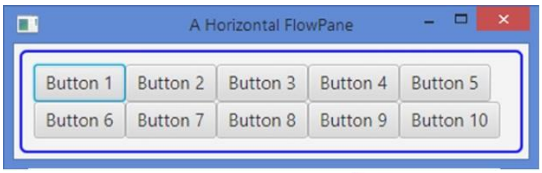


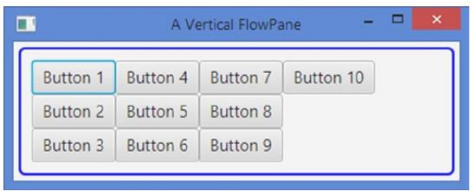
Hassan Ebrahem

8

FlowPane Class الفئة

- الفئة FlowPane هي عبارة عن جزء تخطيط layout pane بسيط يضع أبنائه في صفوف rows أو أعمدة columns تلتف بعرض أو ارتفاع محدد.
- يسمح لأبنائه بالتدفق أفقيًا أو رأسيًا كما في الشكلين التاليين:
- بشكل افتراضي يتم الترتيب بشكل أفقي. تستخدم عندما يكون الترتيب الابناء غير مهم.





Hassan Ebrahem

9

FlowPane Class الفئة

- الابناء في horizontal FlowPane ترتب في صفوف من اليسار إلى اليمين أو العكس، القيمة المبدئية من اليسار إلى اليمين. يتم تطبيقها على جميع الابناء في الصف كما يتم تطبيقه على العقد (HBox, TilePane, etc.).
- يمكن تغيير الاتجاه الابناء من اليمين الى اليسار باستخدام
  - NodeOrientation.RIGHT_TO_LEFT
- يمكن تعيين مسافات بين الابناء افقيا وعموديا باستخدام hgap, vgap.
- يمكن تحديد موضع الابناء في الصف أو العمود باستخدام alignment، القيمة المبدئية لها Pos.TOP_LEFT

Hassan Ebrahem

10

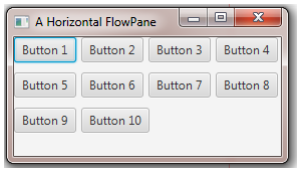
إنشاء الكائن FlowPane Object

- توفر الفئة FlowPane العديد من دوال الإنشاء للكائنات objects منها ذات اتجاه محدد (أفقي horizontal أو رأسي vertical). مع تحديد المسافة الأفقية والرأسية بين الابناء.
- إنشاء FlowPane فارغة بشكل افقي من دون مسافات 0px
- FlowPane fpane1 = new FlowPane();
- إنشاء FlowPane فارغة بشكل راسي من دون مسافات 0px
- FlowPane fpane2 = new FlowPane(Orientation.VERTICAL);
- إنشاء FlowPane فارغة بشكل افقي مع وجود 5px افقية و وجود 10px رأسية
- FlowPane fpane3 = new FlowPane(5, 10);
- إنشاء FlowPane فارغة بشكل رأسي مع وجود 5px افقية و وجود 10px رأسية
- FlowPane fpane4 = new FlowPane(Orientation.VERTICAL, 5, 10);
- إنشاء FlowPane بشكل افقي مع وجود زرين Buttons ومسافة 0px
- FlowPane fpane5 = new FlowPane(new Button("Button 1"), new Button("Button 2"));

Hassan Ebrahem

مثال: الفئة FlowPane Class

```
public void start(Stage stage) {
    double hgap = 5;
    double vgap = 10;
    FlowPane root = new FlowPane(hgap, vgap);
    // Add ten buttons to the flow pane
    for(int i = 1; i <= 10; i++) {
    root.getChildren().add(new Button("Button " + i));
    }
    // Change the orientation of the FlowPane to vertical
    //root.setOrientation(Orientation.VERTICAL);
    Scene scene = new Scene(root);
    stage.setScene(scene);
    stage.setTitle("A Horizontal FlowPane");
    stage.getScene().setNodeOrientation(NodeOrientation.RIGHT_TO_LEFT);
    stage.show();
}
```



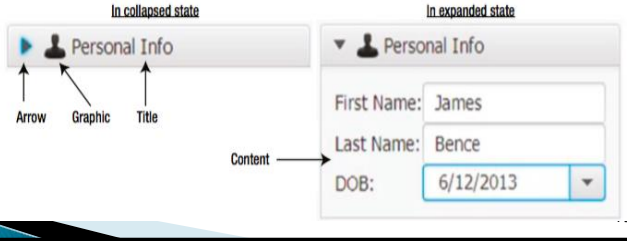
Hassan Ebrahem

الفئة TitledPane Class

- الفئة TitledPane ترث من الفئة Labeled class، وتحتوي على نص ورسم ، بحيث يمكن أن تعرض النص كعنوانه. كما يظهر الرسم في شريط العنوان.
- إلى جانب النص والرسم، يحتوي TitledPaneعلى محتوى وهو عبارة عن عقدة node.
- عادةً ما يتم وضع مجموعة من عناصر التحكم في حاوية وتتم إضافة الحاوية كمحتوى لجزء العنوان.

الفئة TitledPane Class

- يمكن أن تكون TitledPane في حالة مطوية collapsed أو موسعة expanded.
- في الحالة المطوية، يعرض شريط العنوان فقط ويخفي المحتوى.
- في الحالة الموسعة ، يعرض شريط العنوان والمحتوى، وفي شريط العنوان الخاص به يعرض سهمًا يشير إلى ما إذا كان موسعًا أو مطويًا. يؤدي النقر فوق أي مكان في شريط العنوان إلى توسيع المحتوى أو تصغيره.



TitledPane Class الفئة

- تكون TitledPaneبشكل مبدئي في حالة موسعة. أي تأخذ الدالة القيمة true.
- في حالة تصغيرها برمجيا نستخدم الدالة setExpanded(false)
- يمكن عرض صورة مع TitledPane كالتالي:

```
String imageStr = "resources/picture/privacy_icon.png";
URL imageUrl = getClass().getClassLoader().getResource(imageStr);
Image img = new Image(imageUrl.toExternalForm());
ImageView imgView = new ImageView(img);
infoPane2.setGraphic(imgView);
```


TitledPane Object إنشاء الكائن

- يمكن استخدام دالة البناء constructor لبدء بأنشاء TitledPane بدون عنوان ولا محتوى.
- كما يمكن ضبطه لاحقًا باستخدام setText() و setContent().
- انشاء TitledPane ثم اضافة العنوان والمحتوى.

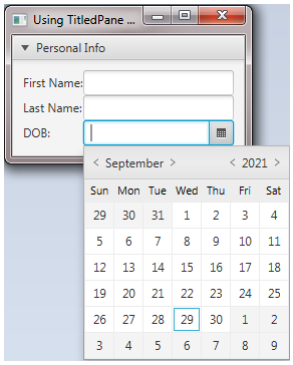
```
TitledPane infoPane1 = new TitledPane();
infoPane1.setText("Personal Info");
infoPane1.setContent(new Label("Here goes the content."));
```

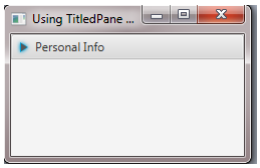
- إنشاء TitledPane مع تحديد العنوان والمحتوى بداخله.

```
TitledPane infoPane2 = new TitledPane("Personal Info", new Label("Content"));
```

مثال: الفئة TitledPane Class

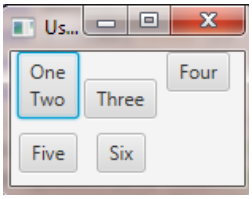
```
public void start(Stage stage) {
    TextField firstNameFld = new TextField();
    firstNameFld.setPrefColumnCount(8);
    TextField lastNameFld = new TextField();
    lastNameFld.setPrefColumnCount(8);
    DatePicker dob = new DatePicker();
    dob.setPrefWidth(150);
    GridPane grid = new GridPane();
    grid.addRow(0, new Label("First Name:"), firstNameFld);
    grid.addRow(1, new Label("Last Name:"), lastNameFld);
    grid.addRow(2, new Label("DOB:"), dob);
    TitledPane infoPane = new TitledPane();
    infoPane.setText("Personal Info");
    infoPane.setContent(grid);
    HBox root = new HBox(infoPane);
    root.setSpacing(10);
    Scene scene = new Scene(root);
    stage.setScene(scene);
    stage.setTitle("Using TitledPane Controls");
    stage.show();
}
```

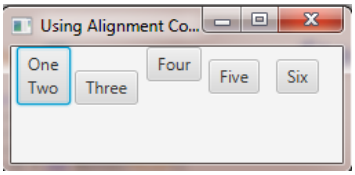




مثال : الفئة TiledPane

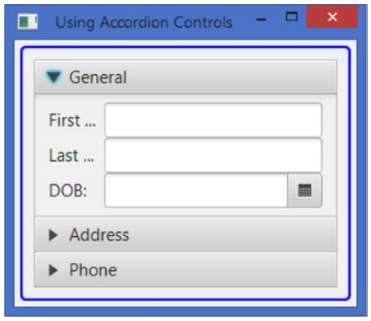
```
public void start(Stage stage) {
    Button b12 = new Button("One\nTwo");
    Button b3 = new Button("Three");
    Button b4 = new Button("Four");
    Button b5 = new Button("Five");
    Button b6 = new Button("Six");
    // Set the tile alignment constraint on b3 to BOTTOM_RIGHT
    TilePane.setAlignment(b3, Pos.BOTTOM_RIGHT);
    TilePane.setAlignment(b4, Pos.TOP_RIGHT);
    TilePane.setAlignment(b5, Pos.CENTER);
    TilePane root = new TilePane(b12, b3, b4, b5, b6);
    root.setPrefColumns(3);
    /*root.setStyle("-fx-padding: 10;" +
    "-fx-border-style: solid inside;" +
    "-fx-border-width: 2;" +
    "-fx-border-insets: 5;" +
    "-fx-border-radius: 5;" +
    "-fx-border-color: blue;");*/
    Scene scene = new Scene(root);
    stage.setScene(scene);
    stage.setTitle("Using Alignment Constraints in TilePane");
    stage.show();
}
```





Accordion Class الفئة

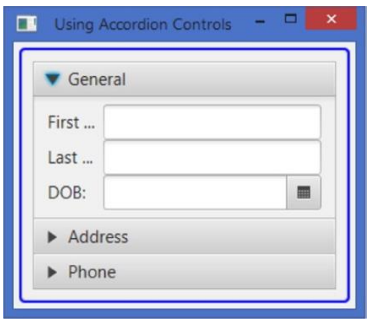
- هو عنصر تحكم بسيط، يعرض مجموعة من TitledPane حيث يكون واحد منهم فقط في الحالة الموسعة expanded في كل مرة.
- يتم استخدام constructor لانشاء الكائن Object بطريقة واحدة.
- Accordion root = new Accordion();



Accordion Class الفئة

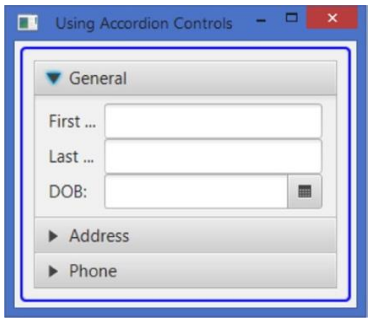
- تقوم الدالة getPanes() بإرجاع قائمة من TitledPane.
- تستخدم القائمة لإضافة أو إزالة أي من TitledPane إلى Accordion، كما هو موضح في الكود التالي:

```
TitledPane generalPane = new TitledPane();
TitledPane addressPane = new TitledPane();
TitledPane phonePane = new TitledPane();
...
Accordion root = new Accordion();
root.getPanes().addAll(generalPane, addressPane, phonePane);
```



مثال الفئة Accordion

```
public void start(Stage stage) {
    TitledPane generalPane = getGeneralPane();
    TitledPane addressPane = getAddressPane();
    TitledPane phonePane = getPhonePane();
    Accordion root = new Accordion();
    root.getPanes().addAll(generalPane, addressPane, phonePane);
    root.setExpandedPane(generalPane);
    Scene scene = new Scene(root);
    stage.setScene(scene);
    stage.setTitle("Using Accordion Controls");
    stage.show();
    }
public TitledPane getGeneralPane() {
    GridPane grid = new GridPane();
    grid.addRow(0, new Label("First Name:"), new TextField());
    grid.addRow(1, new Label("Last Name:"), new TextField());
    grid.addRow(2, new Label("DOB:"), new DatePicker());
    TitledPane generalPane = new TitledPane("General", grid);
    return generalPane;
    }
```

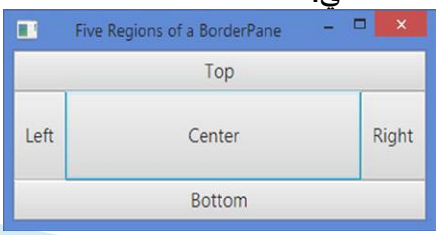



```
public TitledPane getAddressPane() {
    GridPane grid = new GridPane();
    grid.addRow(0, new Label("Street:"), new TextField());
    grid.addRow(1, new Label("City:"), new TextField());
    grid.addRow(2, new Label("State:"), new TextField());
    grid.addRow(3, new Label("ZIP:"), new TextField());
    TitledPane addressPane = new TitledPane("Address", grid);
    return addressPane;
    }
public TitledPane getPhonePane() {
    GridPane grid = new GridPane();
    grid.addRow(0, new Label("Home:"), new TextField());
    grid.addRow(1, new Label("Work:"), new TextField());
    grid.addRow(2, new Label("Cell:"), new TextField());
    TitledPane phonePane = new TitledPane("Phone", grid);
    return phonePane;
    }
```

Hassan Ebrahem

الفئة Border Pane Class

- تقوم الفئة بتقسيم منطقة التخطيط layout الخاصة به إلى خمس مناطق:
  - top, right, bottom, left, and center
- يمكن وضع عقدة واحدة على الأكثر في كل منطقة من المناطق الخمس.
- تسمح BorderPane بتعيين قيود المحاذاة والهامش على الابناء بشكل فردي.
- يتم تحديد محاذاة الابناء حسب منطقتها، تكون القيمة المبدئية للتصفيف كالتالي:
  - Pos.TOP_LEFT for the top child node
  - Pos.BOTTOM_LEFT for the bottom child node
  - Pos.TOP_LEFT for the left child node
  - Pos.TOP_RIGHT for the right child node
  - Pos.CENTER for the center child node



Hassan Ebrahem

23

إنشاء الكائن Border Pane Object

- يتم انشاء constructor الكائنات Objects من الفئة BorderPane بعدة طرق:
- إنشاء كائن BorderPane فارغ.
- BorderPane bpane1 = new BorderPane();
- إنشاء كائن BorderPane يحتوي على TextArea في المنتصف
- TextArea center = new TextArea();
  BorderPane bpane2 = new BorderPane(center);
- إنشاء كائن BorderPane يحتوي على عقدة نصية في المناطق الخمسة
- Text center = new Text("Center");
  Text top = new Text("Top");
  Text right = new Text("Right");
  Text bottom = new Text("Bottom");
  Text left = new Text("Left");

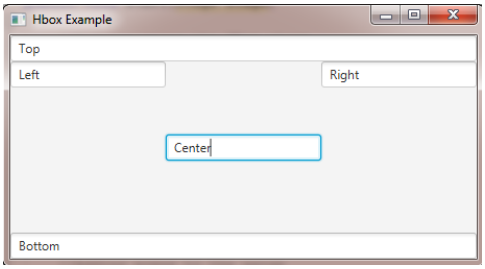
إنشاء الكائن Border Pane Object

- يتم انشاء الكائنات Objects من الفئة BorderPane بعدة طرق:
- إنشاء كائن BorderPane وفي نفس الوقت يتم اضافة عقد نصية اليه في المناطق الخمس

- BorderPane bpane = new BorderPane();
  bpane.setTop(new Text("Top"));
  bpane.setRight(new Text("Right"));
  bpane.setBottom(new Text("Bottom"));
  bpane.setLeft(new Text("Left"));
  bpane.setCenter(new Text("Center"));

مثال : الفئة Border Pane

```
public void start(Stage stage)
{
    //Instantiating the HBox class
    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"));
    //Creating a scene object
    Scene scene = new Scene(bPane);
    //Setting title to the Stage
    stage.setTitle("Hbox Example");
    //Adding scene to the satge
    stage.setScene(scene);
    //Displaying the contents of the stage
    stage.show();
}
```

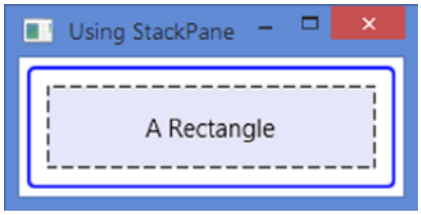


الفئة StackPane Class

- يضع StackPane أبنائه Nodes بداخله حسب ترتيب اضافتهم إليه. بمعنى يتم رسم الابن الاول أولاً ؛ يتم رسم الابن الثاني الذي تم اضافته بعد ذلك أعلى منه، وهكذا.
- على سبيل المثال، يكون وضع النص على الشكل سهلاً باستخدام StackPane. أي أضف الشكل أولاً ثم اضف النص ثانياً. يتم رسم الشكل أولاً متبوعا بالنص ، مما يجعله يبدو كما لو كان النص جزءا من الشكل.

الفئة StackPane

- في الشكل التالي تم تعيين StackPane كعقدة رئيسية root node. وتم إضافة شكل مستطيل ونص "A Rectangle" إلى StackPane.
- النص تمت إضافته الاخير.
- الخط المتصل هو الحد الخارجي الذي يخص StackPane، بينما الحد الداخلي المتقطع هو حدود المستطيل.



إنشاء الكائن StackPane Object

- يتم استخدام الدالة constructor لانشاء الكائن من StackPane

- إنشاء كائن من StackPane فارغ
- StackPane spane1 = new StackPane();

- اضافة مستطيل ونص إلى StackPane
- Rectangle rect = new Rectangle(200, 50);
  rect.setFill(Color.LAVENDER);
  Text text = new Text("A Rectangle");
  spane1.getChildren().addAll(rect, text);

JavaFx

مثال : الفئة StackPane

```
public void start(Stage stage) {
    //Drawing a Circle
    Circle circle = new Circle(300, 135, 100);
    circle.setFill(Color.DARKSLATEBLUE);
    circle.setStroke(Color.BLACK);
    Circle circle1 = new Circle(150, 70, 50);
    circle1.setFill(Color.GAINSBORO);
    circle1.setStroke(Color.BLACK);
    //creating a text
    Text text = new Text("السلام عليكم ورحمة الله وبركاته");
    //Setting the font of the text
    text.setFont(Font.font(null, FontWeight.BOLD, 15));
    //Setting the color of the text
    text.setFill(Color.CRIMSON);
    //setting the position of the text
    text.setX(20);
    text.setY(50);
    //Creating a Stackpane
    StackPane stackPane = new StackPane();
    //Setting the margin for the circle
    stackPane.setMargin(circle, new Insets(50, 50, 50, 50));
    stackPane.getChildren().addAll(circle, circle1, text);
    //Creating a scene object
    Scene scene = new Scene(stackPane);
    //Setting title to the Stage
    stage.setTitle("Stack Pane Example");
    //Adding scene to the satge
    stage.setScene(scene);
    //Displaying the contents of the stage
    stage.show();
}
```

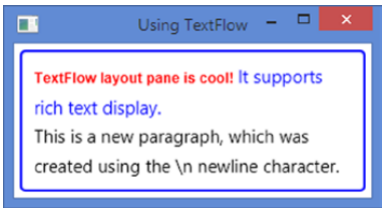
Hassan Ebrahem

31

JavaFx

الفئة TextFlow Class

- تم تصميم TextFlow لعرض نص منسق. يتكون النص المنسق من عدة عقد نصية Nodes.
- يدمج TextFlow النص في جميع عقد النص لعرضه في نص واحد.
- يشير n\ داخل النص إلى بداية فقرة جديدة. التفاف النص يكون حسب عرض TextFlow.
- كل عقدة نصية لها موضعها وحجمها وعرض التفافها. ومع ذلك عند إضافته إلى TextFlow يتم تجاهل هذه الخصائص.



Hassan Ebrahem

35

TextFlow Class الفئة

- تدعم TextFlow الخاصيتين التاليتين:
- الخاصية **lineSpacing** التي تحدد المسافة بين الكائنات النصية وتستخدم الدالة **setLineSpacing()**.
- الخاصية **textAlignment** التي تحدد ضبط الكائنات النصية. وتستخدم الدالة **setTextAlignment()**. **تأخذ هذه الدالة القيم:**
- CENTER, JUSTIFY, LEFT, RIGHT.

TextFlow Object إنشاء الكائن

- يتم انشاء كائنات من TextFlow بعدة طرق:
- يمكن الإنشاء ثم تحديد مجموعة من الابناء واضافتها فيما بعد إلى TextFlow

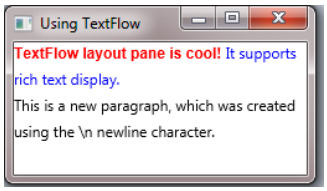
```
TextFlow tflow1 = new TextFlow ();
Text tx1 = new Text("TextFlow layout pane is cool! ");
Text tx2 = new Text("It supports rich text display.");
TextFlow tflow2 = new TextFlow(tx1, tx2);
```

- يمكن تحديد مجموعة من الابناء أولا واضافتها بعد انشاء TextFlow

```
Text tx1 = new Text("TextFlow layout pane is cool! ");
Text tx2 = new Text("It supports rich text display.");
TextFlow tflow3 = new TextFlow();
tflow3.getChildren().addAll(tx1, tx2);
```

مثال: الفئة TextFlow

```
public void start(Stage stage) {
    // Create three Text nodes
    Text tx1 = new Text("TextFlow layout pane is cool! ");
    tx1.setFill(Color.RED);
    tx1.setFont(Font.font("Arial", FontWeight.BOLD, 12));
    Text tx2 = new Text("It supports rich text display.");
    tx2.setFill(Color.BLUE);
    Text tx3 = new Text("\nThis is a new paragraph, which was " +
    "created using the \\n newline character.");
    // Create a TextFlow object with the three Text nodes
    TextFlow root = new TextFlow(tx1, tx2, tx3);
    // Set the preferred width and line spacing
    root.setPrefWidth(300);
    root.setLineSpacing(5);
    Scene scene = new Scene(root);
    stage.setScene(scene);
    stage.setTitle("Using TextFlow");
    stage.show();
}
```

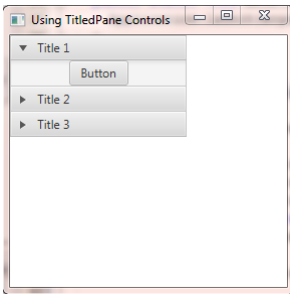


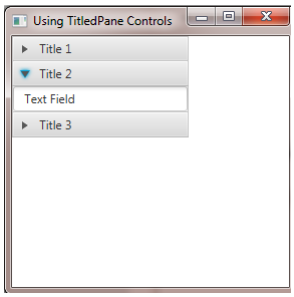
Hassan Ebrahem

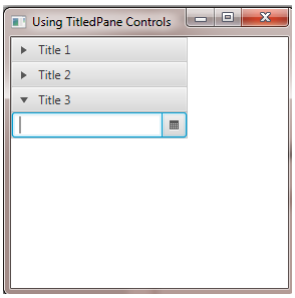
40

الواجب

- اكتب التعليمات البرمجية لتنفيذ التالي باستخدام التخطيط Layout المناسب.







Hassan Ebrahem

42

JavaFx
ملخص المحاضرة
في هذه المحاضرة تم توضيح باقي العقد الحاوية الأخرى container.
هذه العقد الحاوية لها استخدامات مختلفة.
تم توضيح استخدامها ببعض الأمثلة.
Hassan Ebrahem
43

نهاية المحاضرة
JavaFx
Hassan Ebrahem
44