
Interaction Homme-Machine

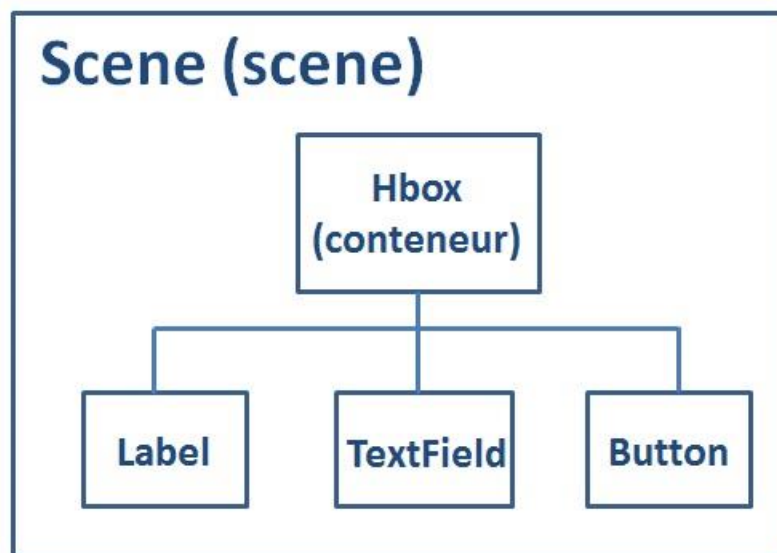
TP 1 : Premières Interfaces Graphiques

Objectifs :

- *Savoir créer et ajouter des nœuds graphiques en JavaFX*
- *Savoir programmer un événement suite à une action de l'utilisateur*

EXERCICE 1 : Première application en JavaFX

Dans cet exercice vous allez créer votre première interface graphique JavaFX. Cette interface graphique est composée de la structure suivante :



Afin de programmer cette structure. Suivez les indications suivantes :

- 1.- Lancez Eclipse et créez un nouveau projet « TP1 ». Dans ce projet créez une classe nommée « MyFirstJavaFX. »
- 2.- Notre classe « MyFirstJavaFX » devra hériter de la classe **Application**.

```
public class MyFirstJavaFX extends Application{
```

- 3.- Au début du fichier, importez les classes suivantes :

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

4. Ajoutez la méthode suivante dans la classe « MyFirstJavaFX »

```
@Override
    public void start(Stage primaryStage) throws Exception {
        // TODO Auto-generated method stub

    }
```

5. Créez dans votre classe une méthode main qui appellera la méthode launch() qui lancera la méthode start() créé précédemment.

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

6. Dans votre méthode start(), créez et ajoutez les nœuds graphiques suivantes :

- a. Créez un nouveau conteneur (Layout) de type **Hbox**

```
HBox hbox = new HBox();
```

Note : à partir d'ici, à chaque fois que vous ajouterez un nouveau nœud graphique (Hbox, Label, Button, etc.), Eclipse vous demandera d'importer la classe JavaFX correspondante. Vous devrez importer la classe correspondante au fur et à mesure.

- b. Créez un nœud de type **Label** avec le texte « Message : ».

```
Label label1 = new Label("Message:");
```

- c. Créez un nœud de type **TextField**

- d. Créez un nœud de type **Button** avec le texte « Afficher »

- e. Ajoutez tous les trois nœuds graphiques précédemment créés (**Label**, **TextField**, **Button**) dans votre gestionnaire de mise en page **Hbox**.

```
hbox.getChildren().addAll(nœud1, nœud2, nœud3);
```

- f. Créez un nouveau objet de type **Scène** et ajoutez le nœud graphique **Hbox** à cette scène. Donnez à votre scène une largeur de 350 et une hauteur de 150.

- g. Ajoutez la scène à votre stage

```
primaryStage.setScene(scene);
```

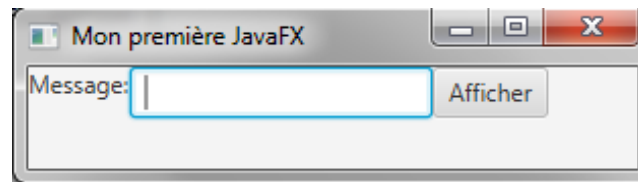
- h. Spécifiez le titre « Mon première JavaFx » de votre fenêtre principale (Stage).

```
primaryStage.setTitle("Mon première JavaFX");
```

- i. Finalement, affichez votre application JavaFX avec la ligne de code suivante :

```
primaryStage.show();
```

Si toutes les indications sont bien respectées, l'exécution du code doit montrer l'interface graphique de la manière suivante :



Ajoutez une action utilisateur à votre Button « afficher »

Afin d'ajouter une action utilisateur au bouton « afficher », ajoutez les lignes de code suivantes après de la création de votre bouton.

```
boutton1.setOnAction( e -> {  
    text1.setText("Hello World!");  
});
```

Exécutez votre application JavaFX. Maintenant si vous cliquez sur votre bouton « afficher », le message « Hello World ! » sera affichée dans le champ de texte (TextField).

EXERCICE 2.1 : Interface pour l'addition de deux valeurs

Considérer l'exemple de l'interface suivant :

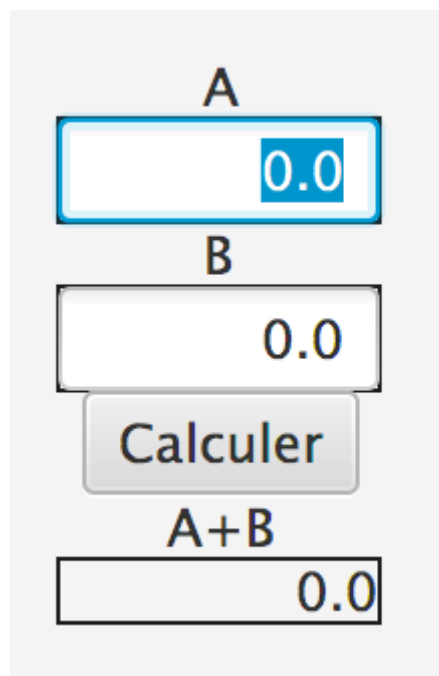


Figure 1 : Interface graphique pour additionner deux valeurs.

Modifier votre fichier `MyFirstJavaFX` pour qu'il crée une interface pour additionner deux valeurs (figure 1). Afin de créer un tableau d'éléments, étudiez la composante de mise en page **GridPane** (<https://docs.oracle.com/javafx/2/api/javafx/scene/layout/GridPane.html>).

Respectez les consignes suivantes :

1. Rajouter des champs de textes pour le masque d'entrée (TextField), et des étiquettes (Label) pour la solution.
2. Utiliser la méthode `setPrefWidth(80)` pour les labels et les textfields pour qu'ils aient la même largeur.
3. Ajoutez également un bouton « Calculer » qui lancera le calcul.
4. Soignez la mise en page pour avoir une interface le plus conviviale possible.

EXERCICE 2.2 : Interface pour l'addition de deux vecteurs

L'objectif de cet exercice est d'une part de créer une interface plus complexe, et d'autre part d'utiliser une boucle pour rajouter plusieurs éléments à la fois dans l'interface, selon un paramètre.

Considérer l'exemple de l'interface suivant :

Vector A			
0.0	0.0	0.0	0.0

Vector B			
0.0	0.0	0.0	0.0

Calculer

A+B			
0.0	0.0	0.0	0.0

Figure 2 : Interface graphique pour additionner deux vecteurs.

Toujours dans votre fichier `MyFirstJavaFX`, modifiez votre programme précédent pour qu'il crée une interface pour additionner deux vecteurs (figure 2). Utiliser un boucle pour rajouter plusieurs éléments à la fois dans l'interface, selon un paramètre de nombre de colonnes (`int nbColonnes = 4`).

EXERCICE 2.3 : Interface pour l'addition de deux matrices

Considérer l'exemple de l'interface suivant :

The interface displays two 3x4 matrices, Matrix A and Matrix B, each containing the value 0.0 in all cells. A button labeled 'Calculer' is positioned below the matrices. Below the button is a table labeled 'A+B' which also contains the value 0.0 in all cells.

0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0

0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0

Calculer

0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0

Modifier votre fichier MyFirstJavaFX pour qu'il crée une interface pour additionner deux matrices de taille N x M. Utiliser des boucles pour rajouter plusieurs éléments à la fois dans l'interface, selon deux paramètres : nombre de colonnes (int nbColonnes = 4;) et nombre de lignes (int nbLignes = 3;).

Pour aller plus loin :

Modifiez le style, la couleur et les dimensions des nœuds graphiques **Label** et **Button**. Vous pouvez regarder comment modifier ces paramètres dans les références suivantes :

Pour modifier le style, la couleur et la taille du Label :

http://docs.oracle.com/javafx/2/ui_controls/label.htm

Pour modifier le style, la couleur et la taille du Button :

http://docs.oracle.com/javafx/2/ui_controls/button.htm

Finalement, vous pouvez créer des autres nœuds graphiques (RadioButton, CheckBox, Combo Box, ListView) dans votre interface à partir des exemples fournis dans la documentation officielle d'Oracle.

Documentation nœuds graphiques :

http://docs.oracle.com/javafx/2/ui_controls/jfxpub-ui_controls.htm