

Programmation Orientée Objet – TP7

S.l. Alexandru Mitrea

S.l. Iulia-Cristina Stănică

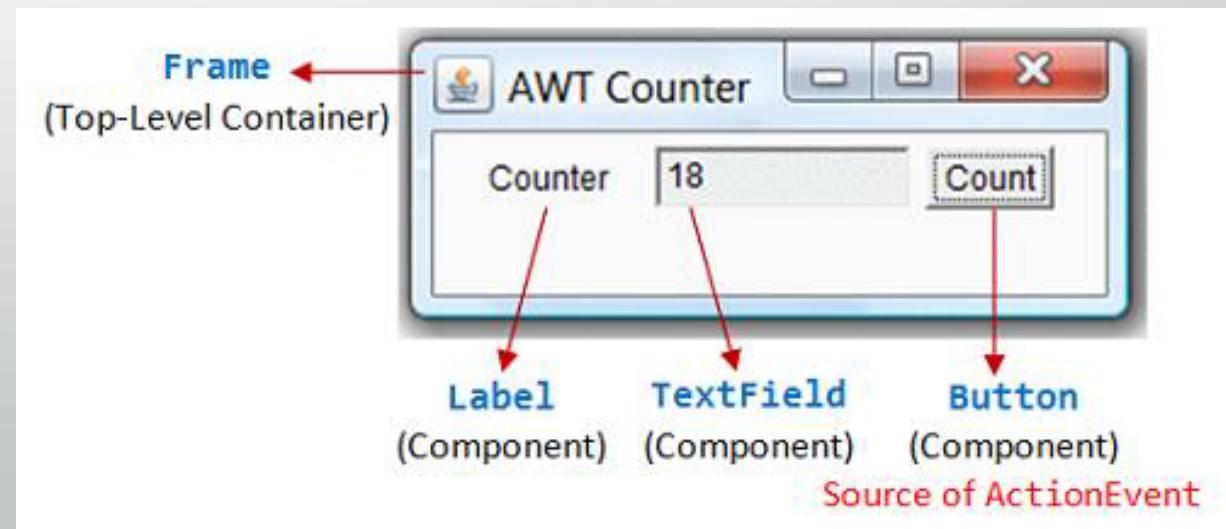
iulia.stanica@gmail.com

Objectifs pour aujourd'hui

- Interfaces graphiques pour applications desktop en Java – Java Swing, Java FX

Historique

- **AWT – Abstract Window Toolkit**
 - Apparue au moment de la création du Java (1995)
 - Bas niveau d'abstraction sur l'interface native de chaque système d'exploitation => aspect différent pour chaque OS (« lourd »)
 - ...mais AWT est assez ancien (obsolète), donc il a été remplacé



Historique

- **Swing**

- Créer des interfaces graphiques identiques quel que soit le système d'exploitation (« léger »)
- Plusieurs choix de design pour chaque composant standard
- Composants: JButton, JLabel, JRadioButton, JTree, JTable, JCheckbox, JMenu....
- Conteneurs: JFrame, JPanel, JScrollPane, JDialog,

WindowTitle

MenuWidget1

MenuWidget2

ToolbarButton

☒ ToolbarCheckBox

PanelCaption

Panel

Item 1

Item 2

Item 3

Item 4

Item 5

☒ RadioButton1

☐ RadioButton2

☐ RadioButton3

☐ InactiveRadio

Button

SelectedTab

OtherTab

☐ UncheckedCheckBox

☒ CheckedCheckBox

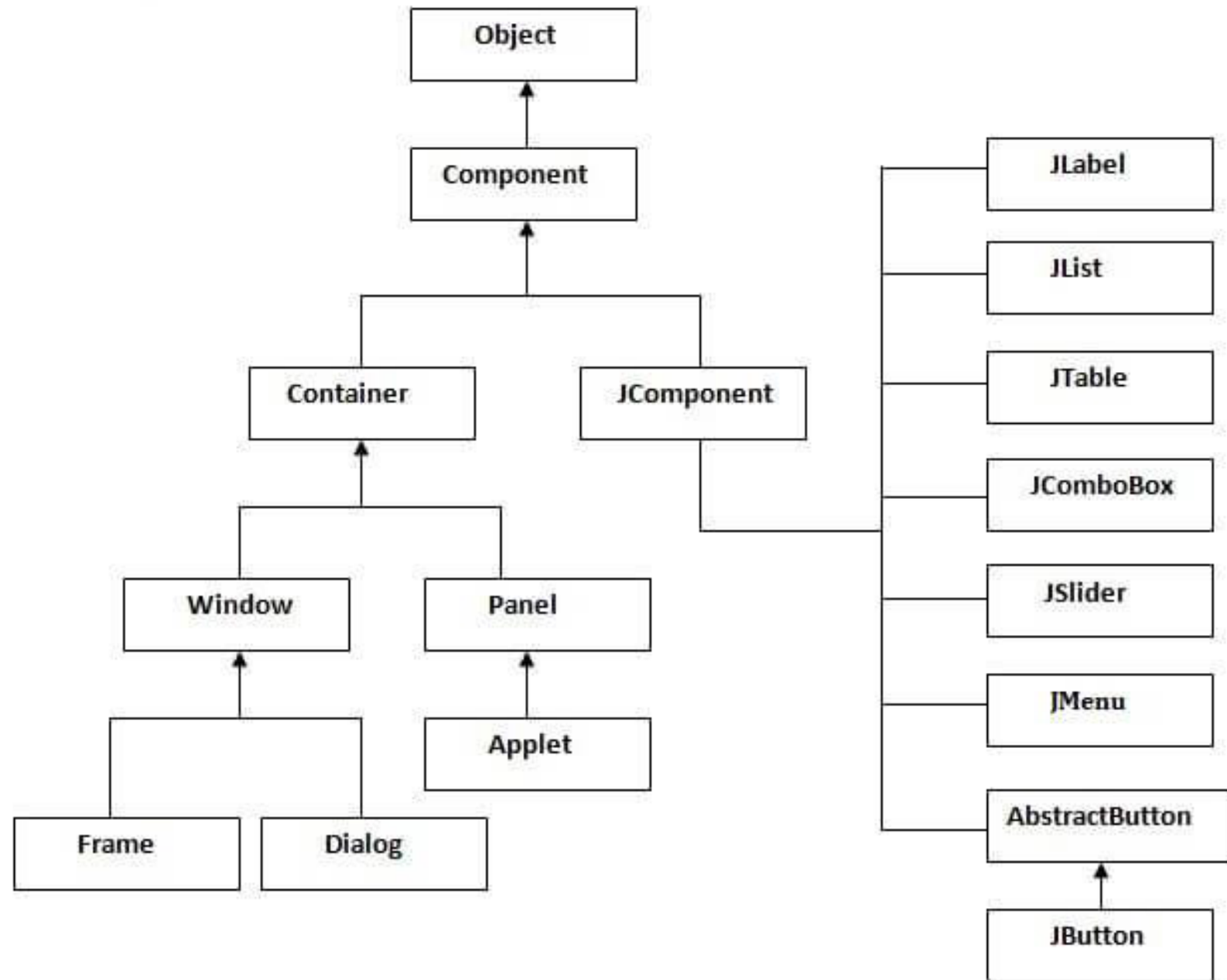
☐ InactiveCheckBox

TextField

.....

Item 1

TextArea



Java Swing

- Pour créer une application en Java Swing, on a besoin d'une classe de type « Container » (classe qui peut contenir d'autres composants):
 - **Panel:** conteneur proprement dit, pas une fenêtre; son but: organiser les composantes dans une fenêtre
 - **Frame:** fenêtre fonctionnelle, avec titre, icônes etc.
- On peut ajouter ensuite toutes les composantes (JComponent) dans le conteneur.
- On peut le faire directement du code ou en utilisant “drag and drop” dans un JFrame Form
- Plus d'infos:
 - <https://www.javatpoint.com/java-swing>
 - <https://www.guru99.com/java-swing-gui.html>

Java Swing – Méthodes usuelles

Method	Description
<code>public void add(Component c)</code>	add a component on another component.
<code>public void setSize(int width,int height)</code>	sets size of the component.
<code>public void setLayout(LayoutManager m)</code>	sets the layout manager for the component.
<code>public void setVisible(boolean b)</code>	sets the visibility of the component. It is by default false.

Java Swing – Premier exemple

```
import javax.swing.*;
public class TestJavaSwing {
    public static void main(String[] args) {

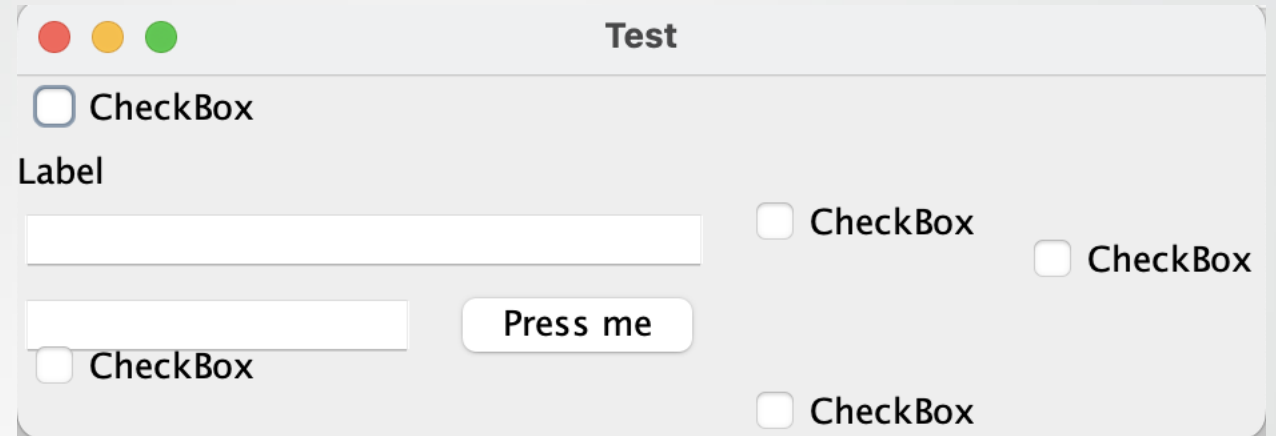
        JFrame frame = new JFrame("My First GUI"); //creating instance of JFrame
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setSize(400,500); //400 width and 500 height

        JButton button = new JButton("Press"); // creating instance of JButton
        button.setBounds(130,100,100,40); // x axis, y axis, width, height

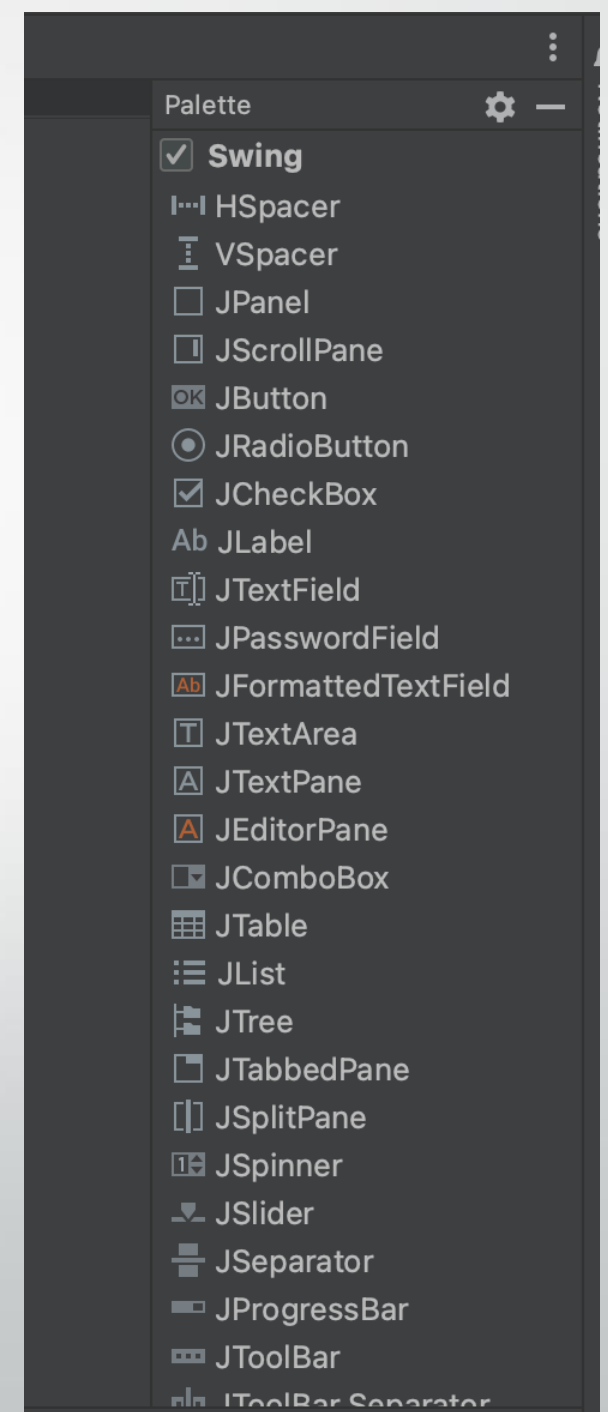
        frame.add(button); // Adds Button to content pane of frame
        frame.setLayout(null); //using no layout managers
        frame.setVisible(true); //making the frame visible

    }
}
```

Java Swing – Observations Intellij



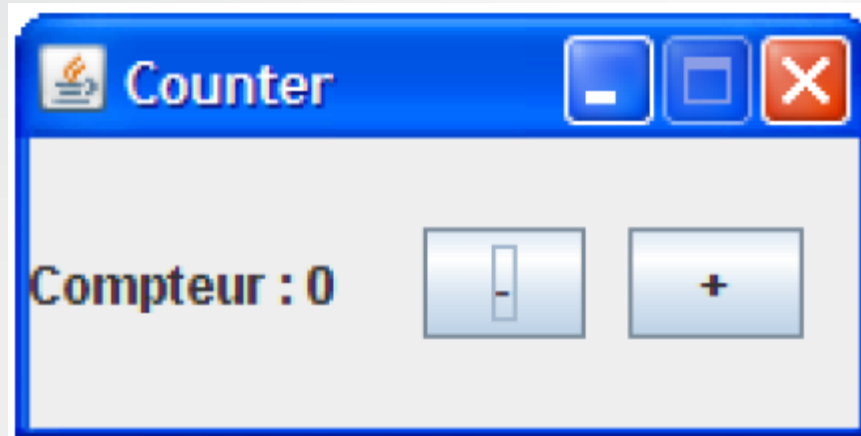
- Possible d'utiliser le designer visuel, pas seulement générer du code l'interface
- New -> Swing UI Designer -> Form
 - Choisissez le type de layout
 - Cochez « Create bound class »
 - Dans la classe, right click, Generate Form main()



Exercice 1 – Java Swing

- Écrire un programme « Counter » qui crée une fenêtre comme celle présentée dans la figure suivante. La fenêtre n'est pas redimensionnable et à la dimension 400 x 300. Elle contient une / deux étiquettes (label) et deux boutons.
- En appuyant sur les boutons + ou -, on incrémente ou décrémente la valeur du compteur et on met à jour le label.

Exercice 1 – Java Swing



- Hint: On peut utiliser un « action listener » - déclenche une action si le bouton est appuyé:

```
button.addActionListener(new ActionListener() {  
    public void actionPerformed(ActionEvent evt) {  
        // action when we press button  
    }  
});
```

Historique...mais

- Swing – pas le standard, aussi assez ancien =>
- **JavaFX**
 - Successeur de Swing
 - Plateforme plus légère, performance élevée
 - Utilisée pour desktop, web, mobile etc.
 - Version 1: 2008, version 2: 2011, ...
 - Installation: incluse par défaut pour Java SE JDK entre 7 et 10
- !!! Doc officielle:
<https://docs.oracle.com/javase/8/javafx/api/javafx>

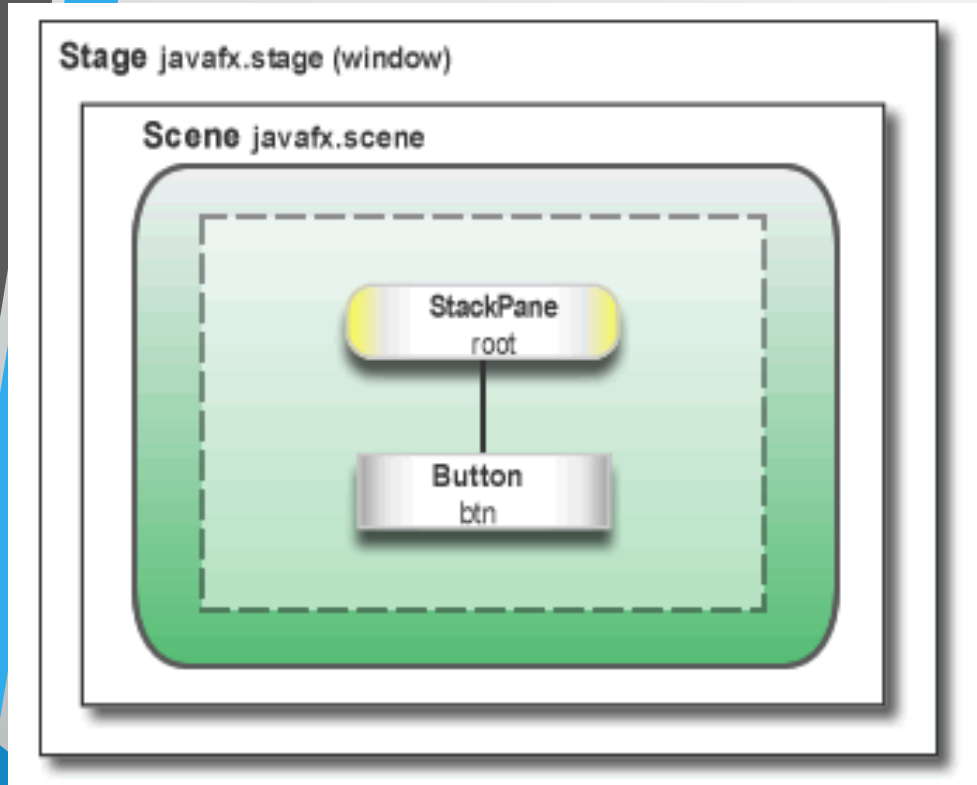
JavaFX – premier exemple (hello world)

- La classe principale d'une application JavaFX étend la classe **javafx.application.Application**. La méthode `start()` est le point d'entrée principal de toutes les applications JavaFX.
- Une application JavaFX définit le conteneur du GUI en utilisant les classes **JavaFX Stage** (top level) et **JavaFX Scene** (basic level)
- Tout le contenu va être ajouté dans la scene.

JavaFX – premier exemple (hello world)

- En JavaFX, le contenu de la scène est représenté sous la forme d'un **graphe de nœuds** (scène hiérarchique). Dans cet exemple, le nœud racine est un objet **StackPane**, qui est un nœud de présentation (**layout**) redimensionnable (« stack » des éléments l'un sur le précédent). On peut avoir GridPane, BorderPane etc.
- Le nœud racine contient un nœud enfant, un bouton avec du texte, ainsi qu'un gestionnaire d'événements (**event handler**) pour imprimer un message lorsque le bouton est appuyé.

JavaFX – premier exemple (hello world)



- La méthode **main()** n'est pas requise pour les applications JavaFX
- Cependant, il est utile d'inclure la méthode `main()` afin que vous puissiez exécuter les fichiers JavaFX (appeler le *launch()* pour lancer l'application graphique).



JavaFX – créer un formulaire

- On utilise **GridPane Layout** (lignes et colonnes)

```
GridPane grid = new GridPane();
```

```
//default position
```

```
grid.setAlignment(Pos.CENTER);
```

```
//space between rows / columns
```

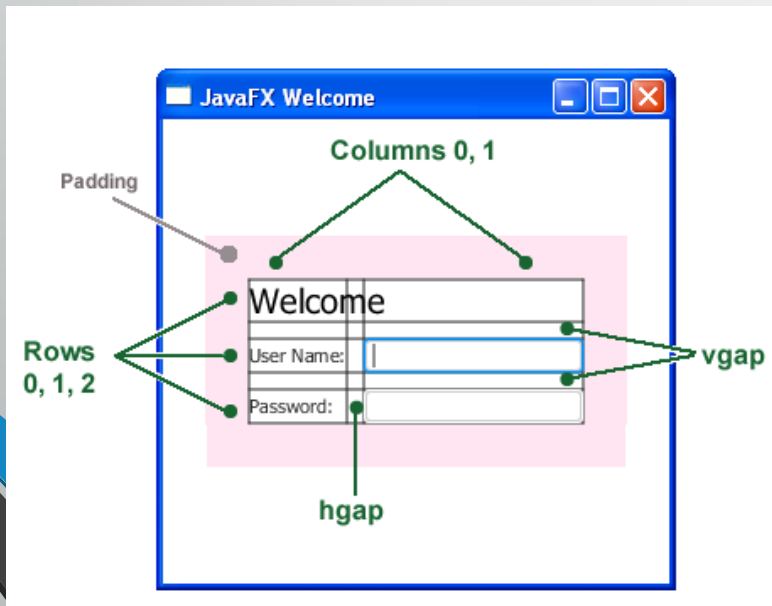
```
grid.setHgap(10);
```

```
grid.setVgap(10);
```

```
//space around edges of grid pane
```

```
grid.setPadding(new Insets(25, 25, 25, 25));
```

```
 //(top, right, bottom, left)
```



JavaFX – créer un formulaire

- Éléments utiles: Text, Label, TextField, PasswordField, Bouton

- Créer du texte:

```
Text scenetitle = new Text("Welcome");
```

```
scenetitle.setFont(Font.font("Tahoma",FontWeight.NORMAL, 20));
```

- Ajouter un element dans le grid:

```
grid.add(scenetitle, 0, 0, 2, 1); //element name, column, row, column span, row span
```

- Ajouter un event handler:

```
btn.setOnAction(new EventHandler<ActionEvent>() {
```

```
    @Override
```

```
        public void handle(ActionEvent e) {
```

```
            //do smth when pressing button }    });
```

JavaFX – exercice 2

- Créer un formulaire qui ressemble le plus possible avec l'image présentée dans le slide 18.
- Quand l'utilisateur appuie le bouton de sign in, une autre fenêtre va s'ouvrir avec un message de succès.
 - N'oubliez pas:

