Cours R2.02
Introduction à l'Interaction Humain-Machine

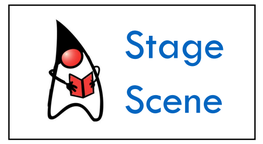
Cours 1 : gestionnaires de placement

Géry Casiez https://gery.casiez.net
Département informatique - IUT de Lille - Université de Lille

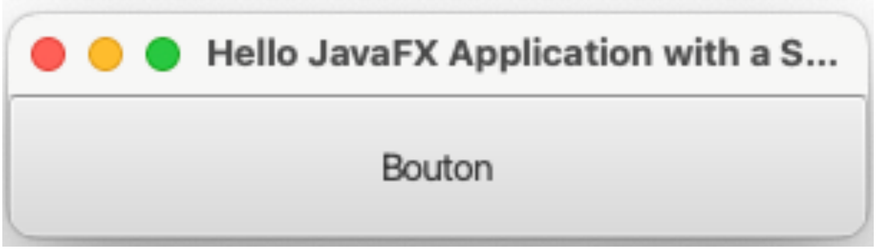
Plan du cours en 9 semaines

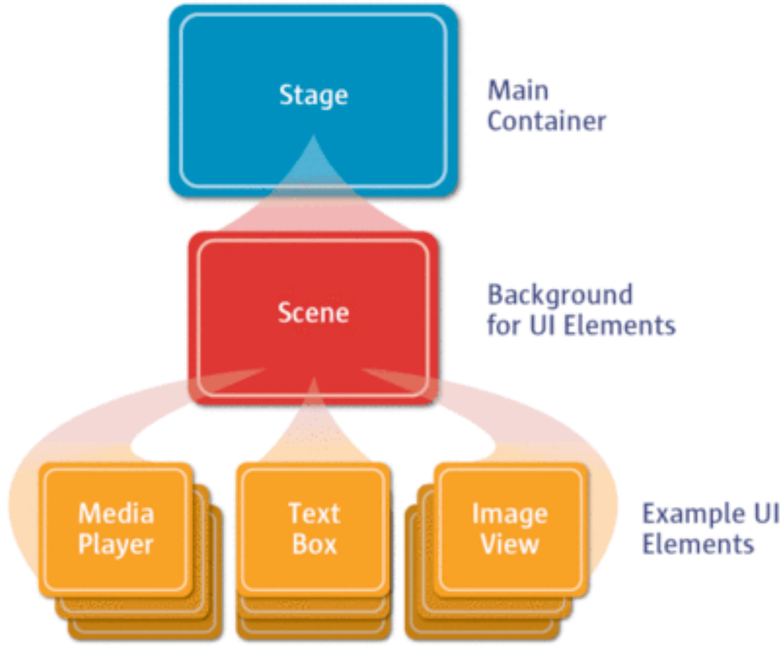
1. **Introduction à l'interaction, placement**
2. Programmation événementielle
3. Widgets et événements (1/2)
4. Widgets et événements (2/2)
5. Conception et prototypage (1/2)
6. Conception et prototypage (2/2)
7. Heuristiques et recommandations
8. Modèles et théories
9. Méthodes d'évaluation des IHM

Retour sur le dernier cours

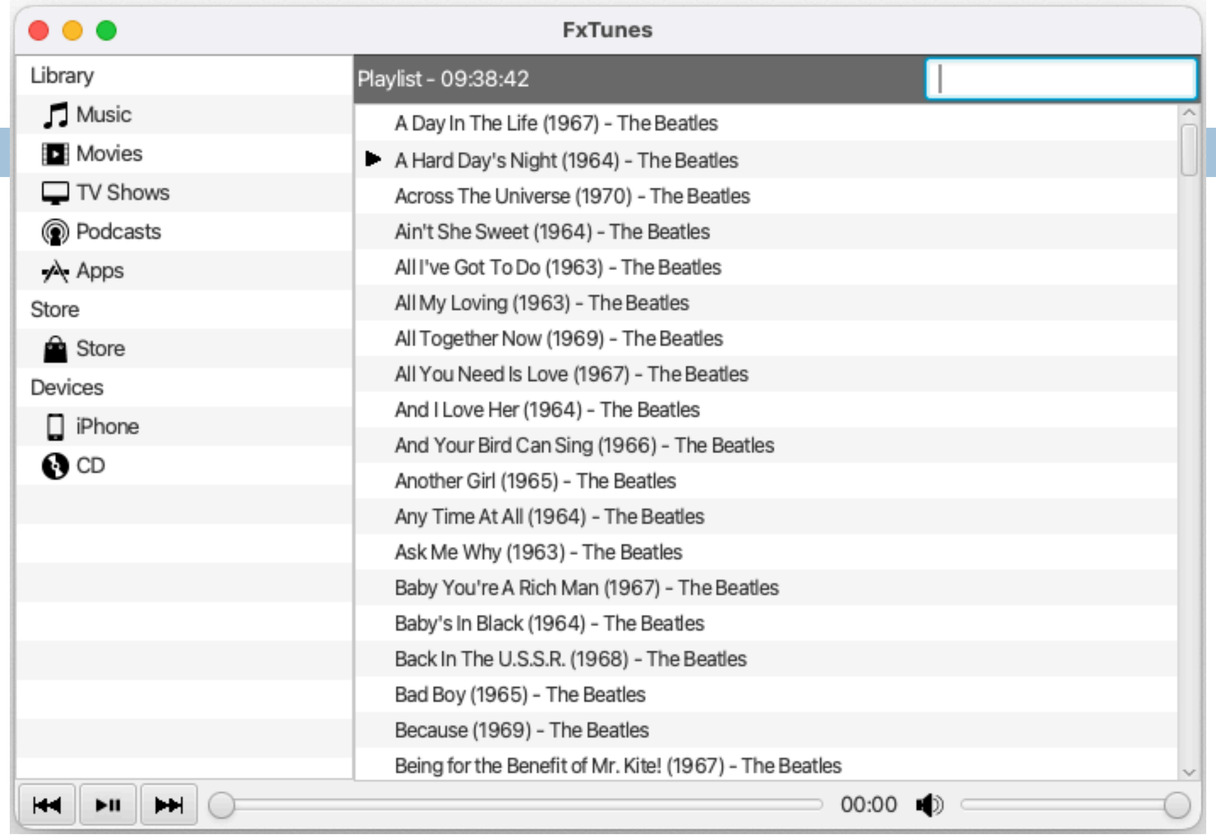



```
public class SimpleScene extends Application {

    public void start(Stage stage) {
        Button bouton = new Button("Bouton");

        Scene scene = new Scene(bouton, 300, 50);
        stage.setScene(scene);

        stage.setTitle("Hello JavaFX Application with a Scene");
        stage.show();
    }

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





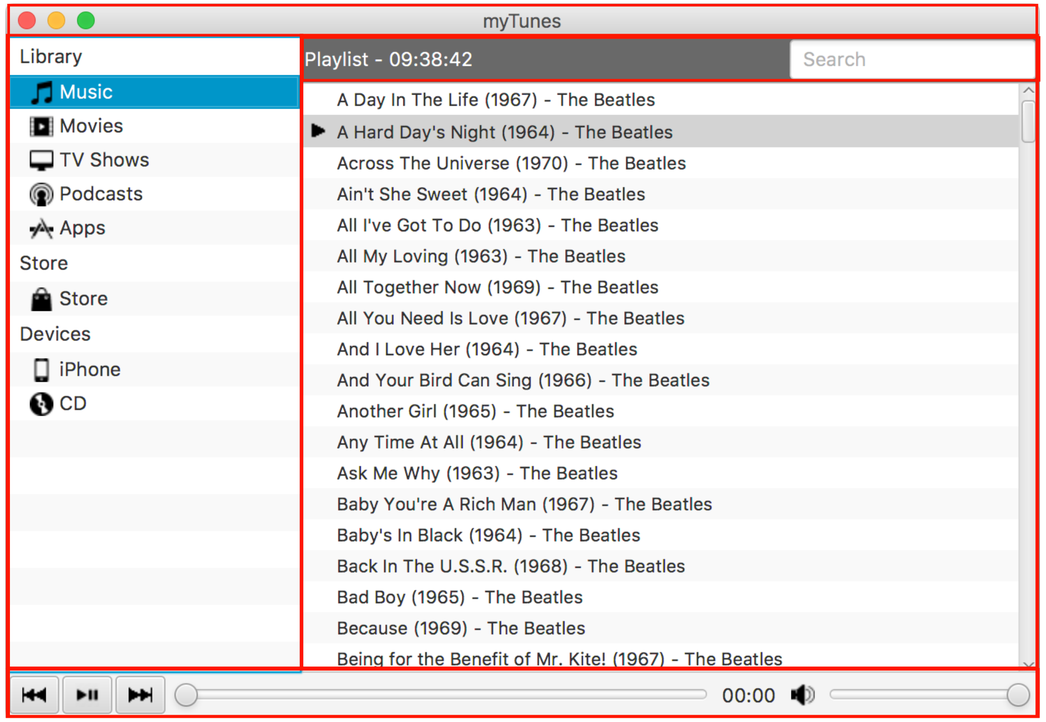
FxTunes

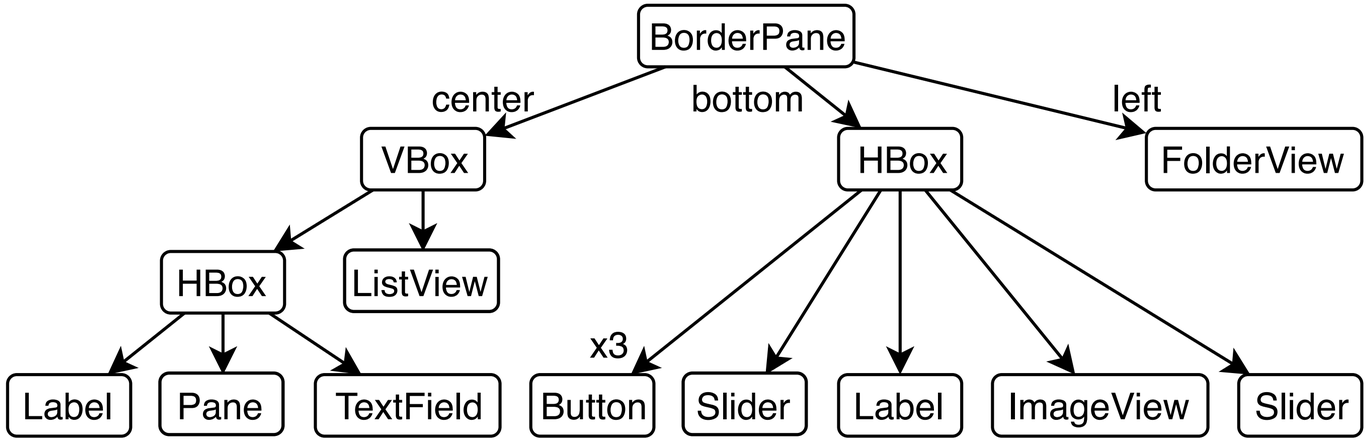

FxTunes
Library
Music
Movies
TV Shows
Podcasts
Apps
Store
Store
Devices
iPhone
CD
Playlist - 09:38:42
A Day In The Life (1967) - The Beatles
A Hard Day's Night (1964) - The Beatles
Across The Universe (1970) - The Beatles
Ain't She Sweet (1964) - The Beatles
All I've Got To Do (1963) - The Beatles
All My Loving (1963) - The Beatles
All Together Now (1969) - The Beatles
All You Need Is Love (1967) - The Beatles
And I Love Her (1964) - The Beatles
And Your Bird Can Sing (1966) - The Beatles
Another Girl (1965) - The Beatles
Any Time At All (1964) - The Beatles
Ask Me Why (1963) - The Beatles
Baby You're A Rich Man (1967) - The Beatles
Baby's In Black (1964) - The Beatles
Back In The U.S.S.R. (1968) - The Beatles
Bad Boy (1965) - The Beatles
Because (1969) - The Beatles
Being for the Benefit of Mr. Kite! (1967) - The Beatles
00:00


FxTunes
Library
Music
Movies
TV Shows
Podcasts
Apps
Playlist - 09:38:42
A Day In The Life (1967) - The Beatles
A Hard Day's Night (1964) - The Beatles
Across The Universe (1970) - The Beatles
Ain't She Sweet (1964) - The Beatles
All I've Got To Do (1963) - The Beatles
All My Loving (1963) - The Beatles
00:00

Principe général

Les widgets doivent être placés dans des conteneurs.





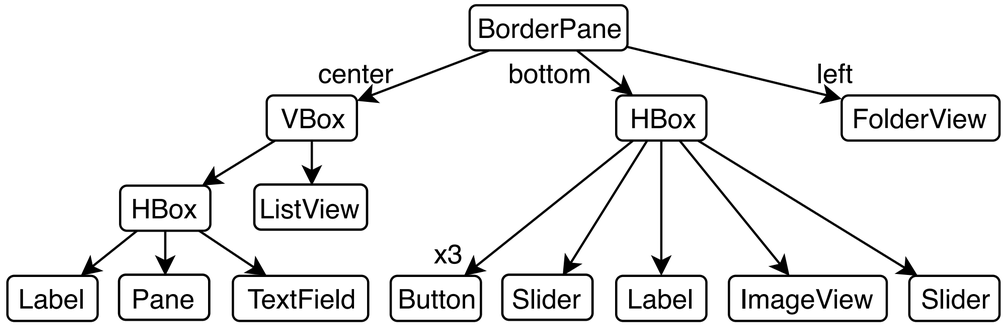
Graphe de scène

Un graphe de scène structure la représentation spatiale de l'interface

Représentation sous forme d'arbre : chaque noeud peut avoir plusieurs enfants mais un seul parent

Possède une racine

Composé de widgets et conteneurs



Les conteneurs peuvent contenir des widgets et des conteneurs

Les feuilles de l'arbre sont widgets, les noeuds sont les conteneurs

L'objet Scene contient la racine du graphe de scène

Conteneurs

Un conteneur regroupe un ensemble de widgets et de conteneurs

Il possède un gestionnaire de placement (layout) qui détermine la position et les dimensions de chacun des éléments qu'il contient en suivant une stratégie donnée

Stratégies de placement

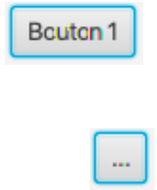
Comment un conteneur agence-t-il visuellement les widgets qu'il contient ?

Que se passe-t-il quand on agrandit la fenêtre ? Les widgets doivent-ils s'agrandir ? Ajoute-t-on de l'espace ? Où ?

Minimum, maximum, preferred sizes

Chaque noeud a quatre tailles :

- La taille idéale (preferred size)
- La taille minimale (minimum size)
- La taille maximale (maximum size)
- La taille réelle



Un layout va essayer de rendre la taille réelle la plus proche possible de la taille préférée compte tenu des contraintes dues à sa stratégie et dues à la taille réelle du conteneur

Gestionnaires de placement (layout manager)

Gestionnaires de placement : placement dynamique des enfants d'un nœud du graphe de scène

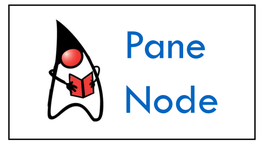
- Calcul de position et des dimensions de chaque Node
- Généralement, imbrication géométrique d'un widget dans son parent

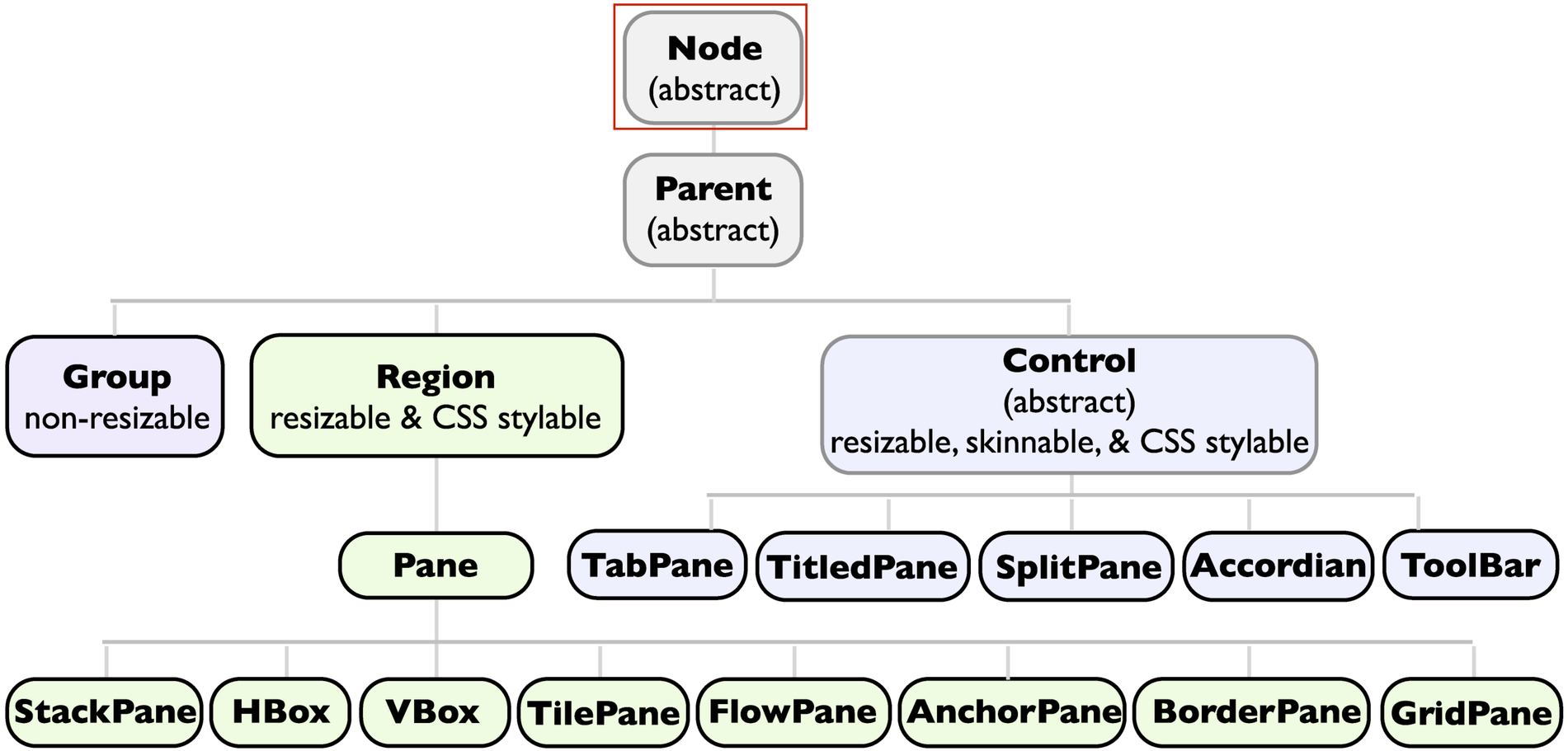
Contraintes

- Taille « naturelle » de chaque fils 

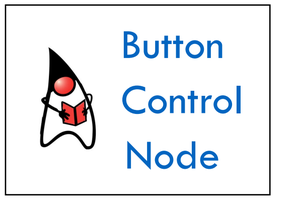
- Taille imposée par le parent
- Contraintes de placement spécifiées par le programmeur

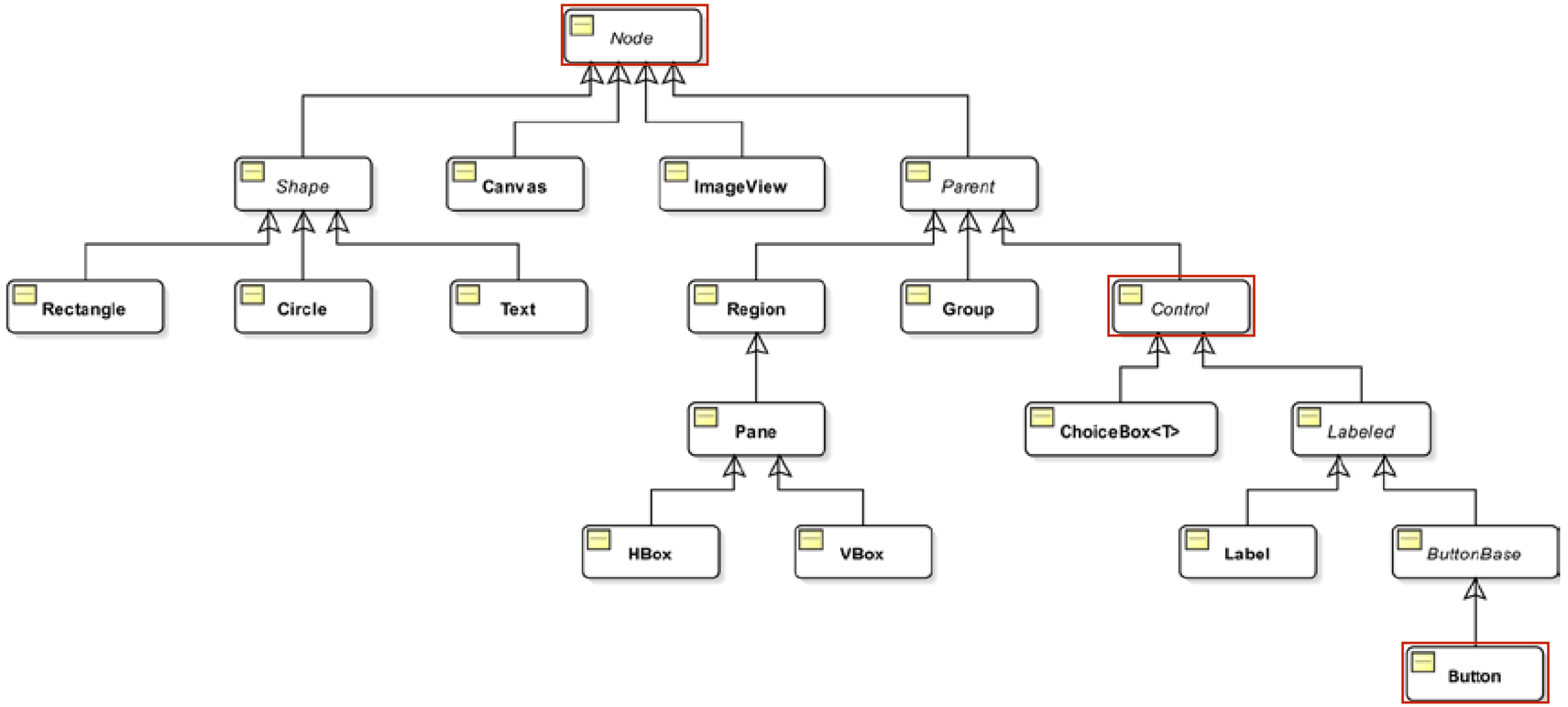
Les conteneurs héritent de Node





Les widgets héritent de Node





Node

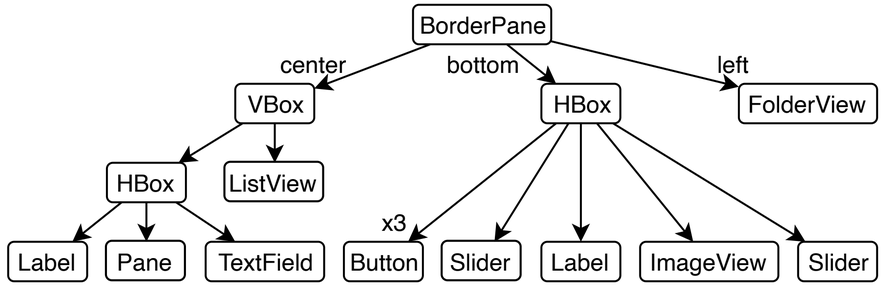
Par héritage, les conteneurs et les widgets sont des Node

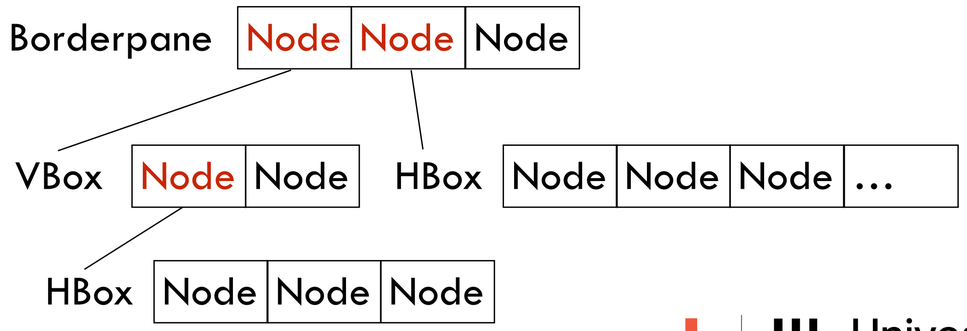
Arbre de Node(s)

Au sein des conteneurs les widgets et les conteneurs sont stockés sous la forme d'une *ObervableList<Node>*

Parmi l'ensemble des noeuds que contient un conteneur peut se trouver un conteneur (rouge)

On crée ainsi une hiérarchie de noeuds qui est appelée graphe de scène





ObservableList<Node>

Possède une méthode add(Node e) pour ajouter un Node à la liste

Possède aussi une méthode addAll(Node… Elements) pour ajouter un nombre variable de Node à la liste

Java varargs …

Syntaxe qui permet de définir un nombre variable d'arguments **de même type**

Sans varargs :

```
public String myMethod()
public String myMethod(String value)
public String myMethod(String val1, String val2)
```

ou

```
public String myMethod(String[] val)
```

avec

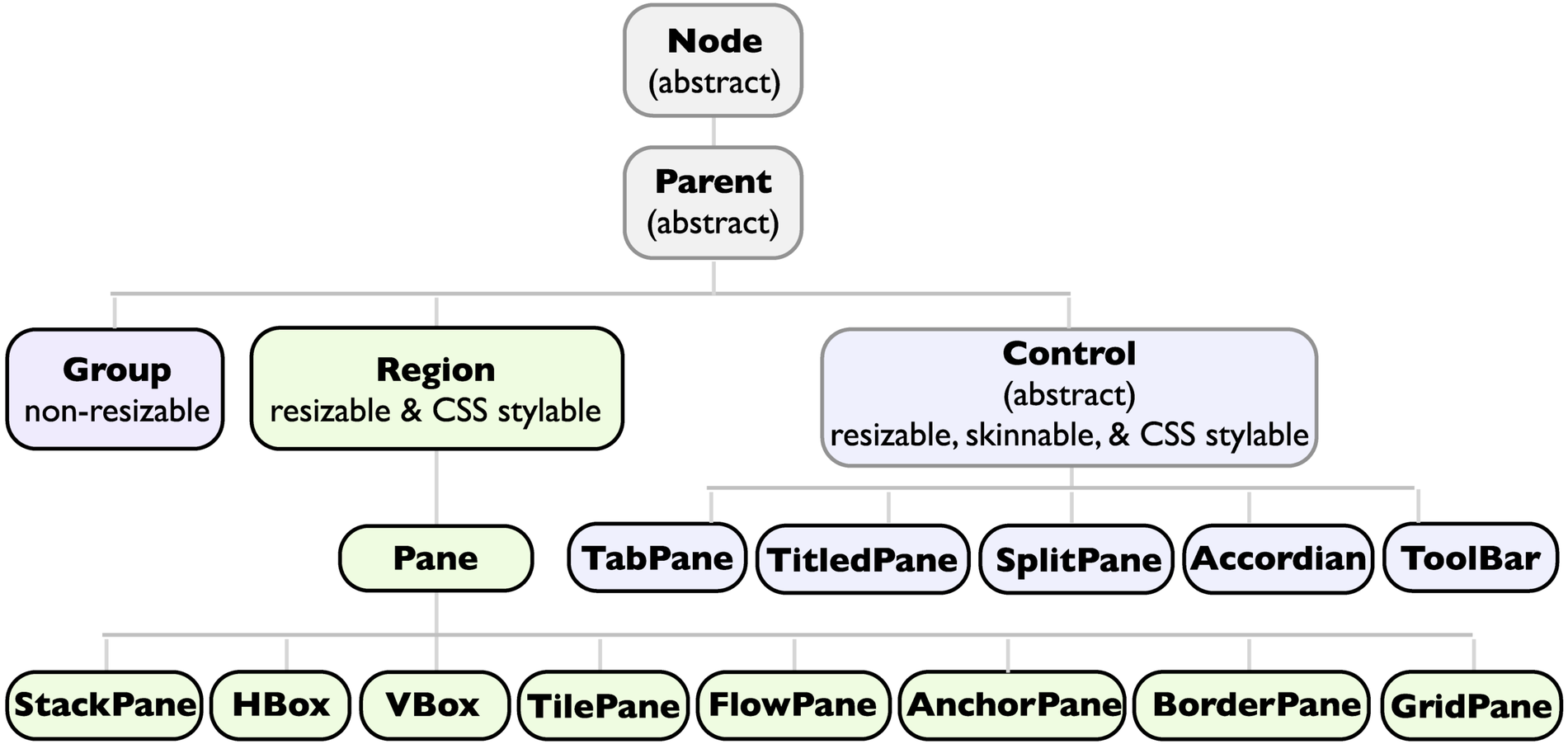
```
public String myMethod(String… val)
```

Java varargs …

Règles :

- une méthode ne peut avoir qu'un seul paramètre varargs
- le paramètre varargs doit être le dernier

Gestionnaires qui héritent de Pane



ObservableList<Node>

L'accès à l'ObservableList<Node> se fait avec la méthode **getChildren()** (définie dans Pane)

Positionnement absolu

cours1 Layout/PositionnementAbsolu.java

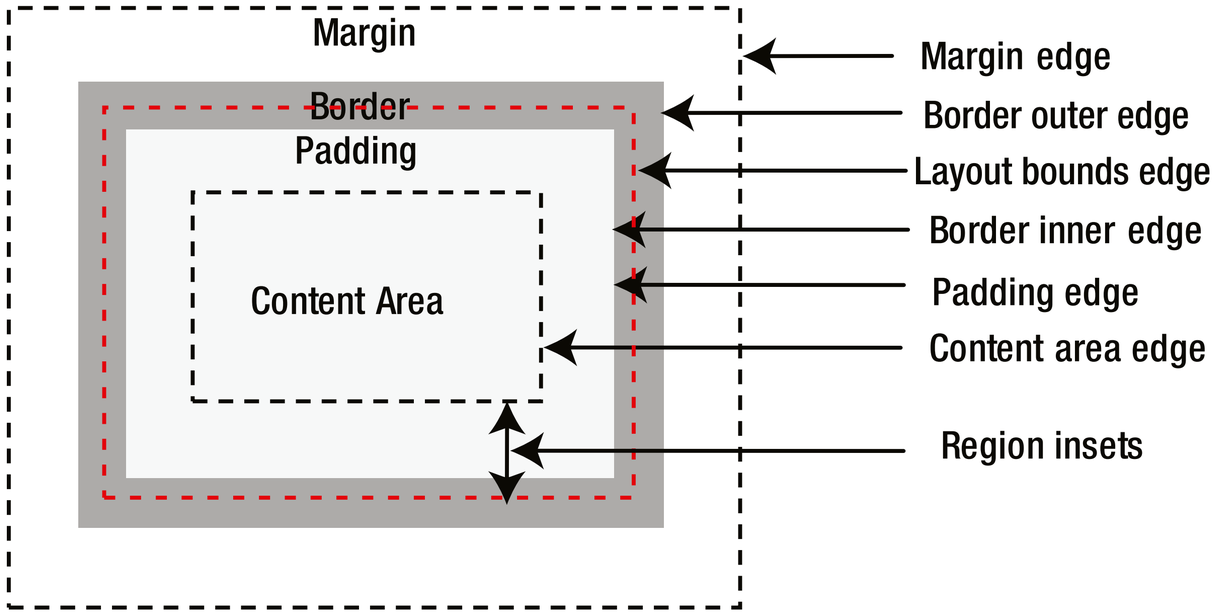
Définition de la taille et de la position de chaque composant

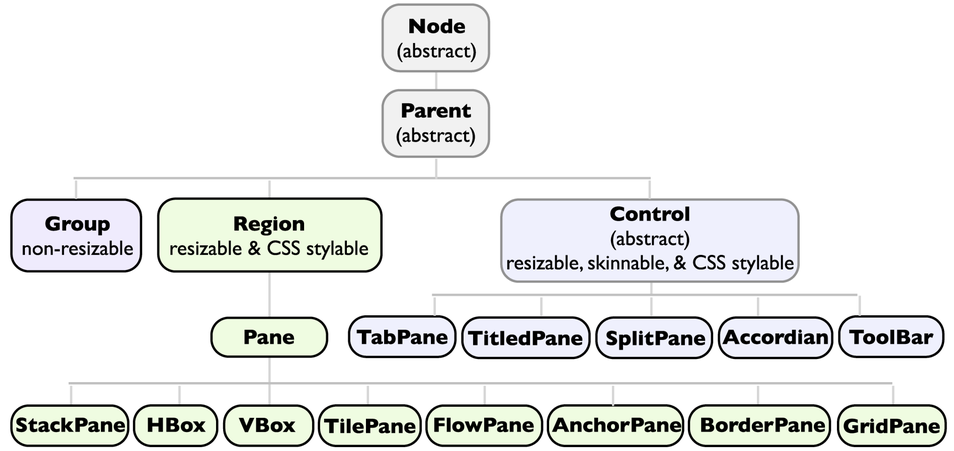
L'origine est le coin supérieur gauche du conteneur


```
Button bouton1 = new Button("Bouton 1");
Button bouton2 = new Button("Bouton 2");
Button bouton3 = new Button("Bouton 3");

Pane pane = new Pane();
bouton1.relocate(30, 20);
bouton2.relocate(150, 15);
bouton2.setPrefSize(80, 50);
bouton3.relocate(200, 70);

pane.getChildren().addAll(bouton1, bouton2, bouton3);

Scene scene = new Scene(pane, 300, 100);
```

Region

Classe dont hérite les gestionnaires de placement qui héritent de Pane



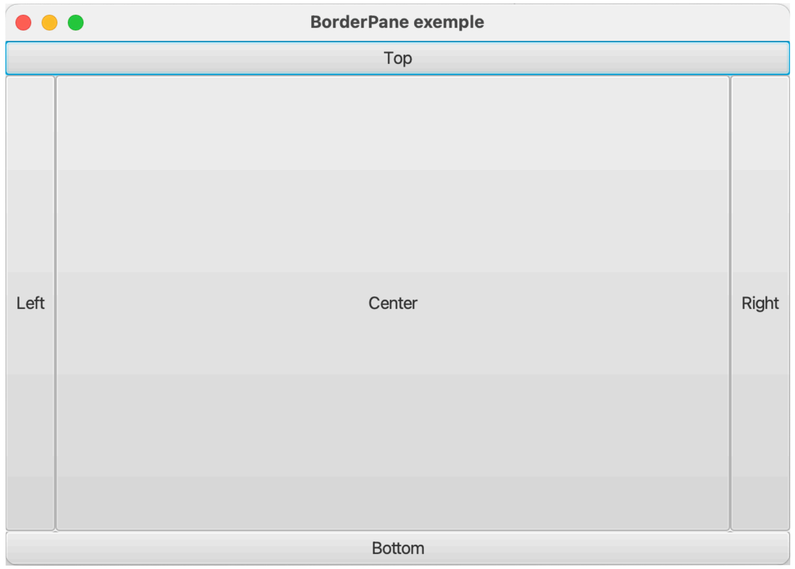


BorderPane

cours1Layout/BorderPaneExample.java

Satisfait les exigences de la plupart des applications de bureau.

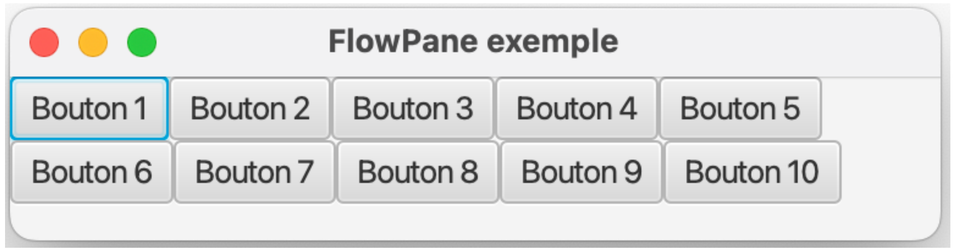
```
Button top = new Button("Top");
top.setMaxSize(Double.MAX_VALUE, Double.MAX_VALUE);
…

BorderPane bp = new BorderPane();
bp.setTop(top);
bp.setBottom(bottom);
bp.setLeft(left);
bp.setRight(right);
bp.setCenter(center);
```



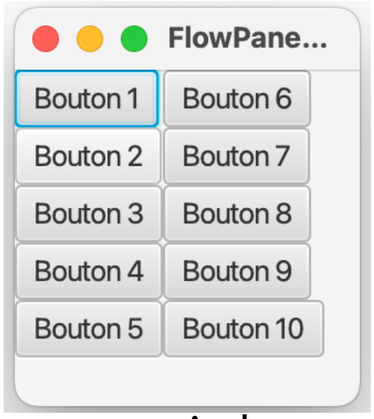
FlowPane

cours1 Layout/FlowPaneExample.java

Place les noeuds les uns à côté des autres ou les uns en dessous des autres



horizontal



vertical

```
FlowPane fp = new FlowPane(Orientation.HORIZONTAL);

for (int i=0; i<10; i++) {
    Button b = new Button("Bouton "+(i+1));
    fp.getChildren().add(b);
}
Scene scene = new Scene(fp);
```

HBox

cours1Layout/HBoxExample.java

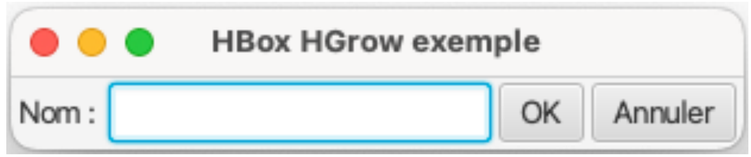
Place les noeuds sur une ligne horizontale

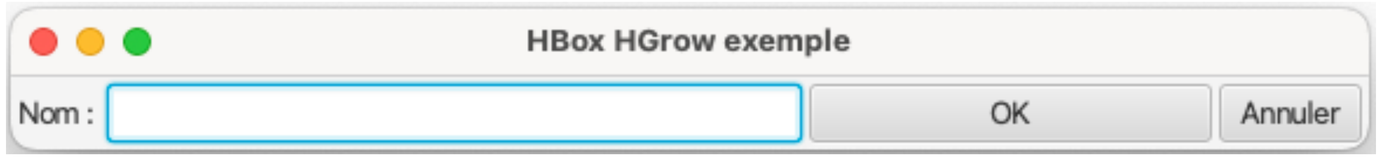
```
Label lbl_nom = new Label("Nom : ");
TextField tf_nom = new TextField();
Button bt_ok = new Button("OK");
Button bt_cancel = new Button("Annuler");

HBox hb = new HBox();
hb.setSpacing(3);   ajout d'un espace de 3 pixels entre les widgets
hb.setAlignment(Pos.CENTER_LEFT);  pour centrer verticalement les widgets et aligner à gauche horizontalement
hb.setPadding(new Insets(3, 3, 3, 3));   ajout d'un espace de 3 pixels autour de la HBox
hb.getChildren().addAll(lbl_nom, tf_nom, bt_ok, bt_cancel);
```

HBox - Hgrow

cours1Layout/HBoxHGrowExample.java

Possibilité de contrôler priorité d'agrandissement de certains noeuds






```
// ...
Button bt_ok = new Button("OK");
bt_ok.setMaxWidth(Double.MAX_VALUE);
// ...

HBox hb = new HBox();
// ...
hb.getChildren().addAll(lbl_nom, tf_nom, bt_ok, bt_cancel);
HBox.setHgrow(tf_nom, Priority.ALWAYS);
HBox.setHgrow(bt_ok, Priority.ALWAYS);
```

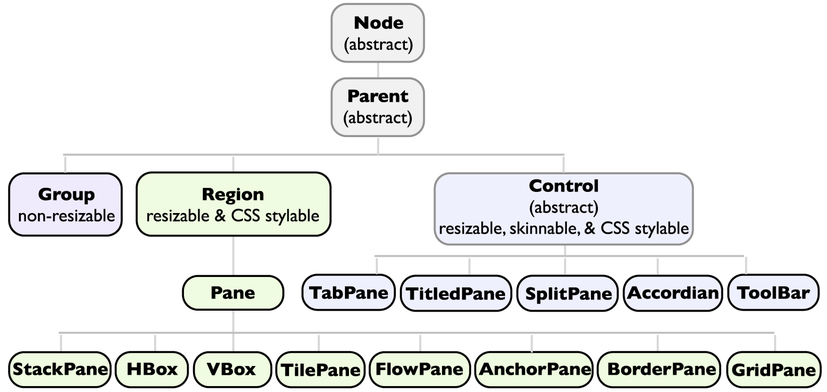
la taille des widgets ne peut pas augmenter au delà de leur taille maximale

setHBox est une méthode statique de HBox

HBox - ajout d'espace

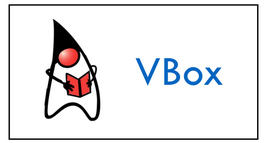
cours1Layout/HBoxHGrowSpaceExample.java

Ajout d'espace entre les widgets : utilisation de Region



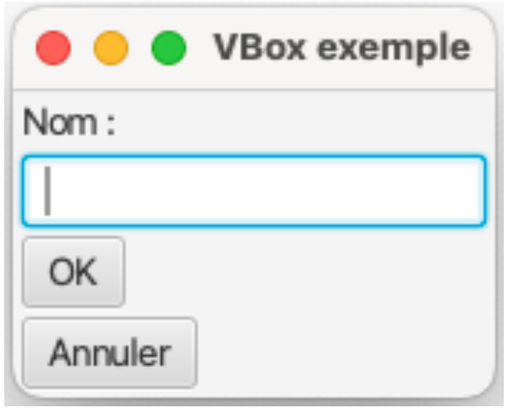

```
Region space = new Region();
// ...

HBox hb = new HBox();
// ...
hb.getChildren().addAll(lbl_nom, tf_nom, space, bt_ok, bt_cancel);
HBox.setHgrow(space, Priority.ALWAYS);
```

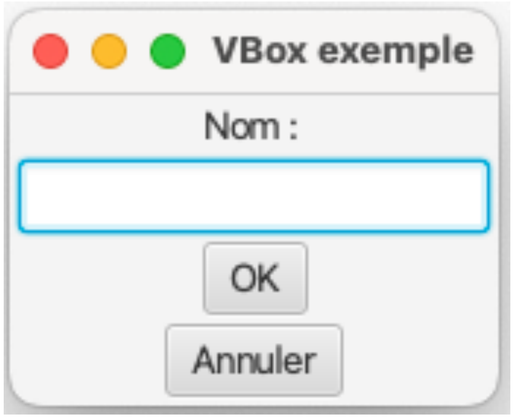
VBox



cours1Layout/VBoxExample.java

Place ses noeuds suivant une seule colonne






```
Label lbl_nom = new Label("Nom : ");
TextField tf_nom = new TextField();
Button bt_ok = new Button("OK");
Button bt_cancel = new Button("Annuler");

VBox vb = new VBox();
vb.setSpacing(3);
vb.setAlignment(Pos.CENTER_LEFT);  permet de jouer sur l'alignement
vb.setPadding(new Insets(3, 3, 3, 3));
vb.getChildren().addAll(lbl_nom, tf_nom, bt_ok, bt_cancel);
```

TilePane

cours1 Layout/TilePaneExample.java

Disposition des noeuds selon une grille dont toutes les cellules ont la même taille

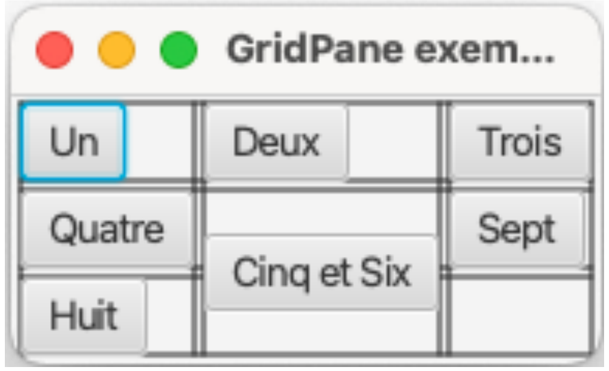

```
TilePane tp = new TilePane(Orientation.HORIZONTAL);

tp.getChildren().addAll(
    new Button("janvier"),
    new Button("février"),
    // ...
    new Button("décembre"));
```

GridPane

cours1Layout/GridPaneExample.java

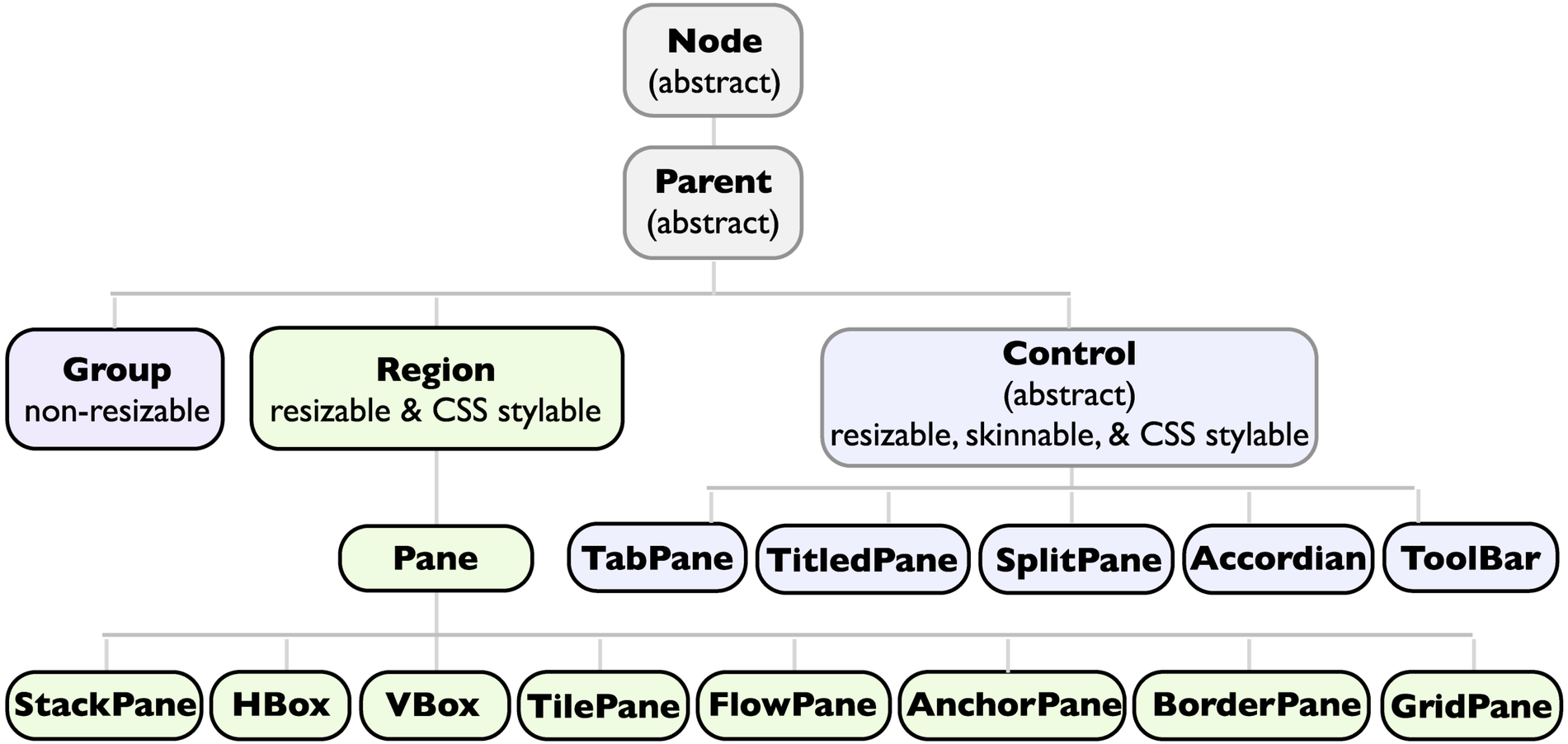
Utilisation d'une grille

Un noeud peut occuper plusieurs cellules



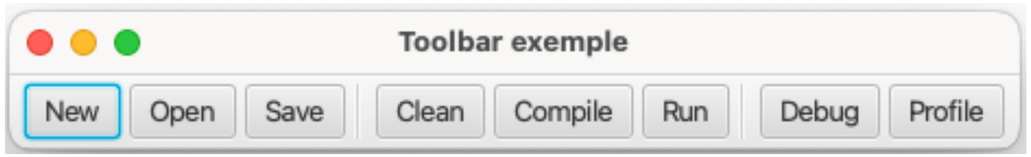

```
GridPane gp = new GridPane();
gp.setGridLinesVisible(true);  pour afficher la grille
gp.setHgap(3.0);
gp.setVgap(3.0);

Button b = new Button("Un");
gp.add(b, 0, 0);

b = new Button("Deux");
gp.add(b, 1, 0);
//...

b = new Button("Cinq et Six");
gp.add(b, 1, 1, 1, 2);  2e ligne, 2e colonne, occupe 1 colonne et 2 lignes
```

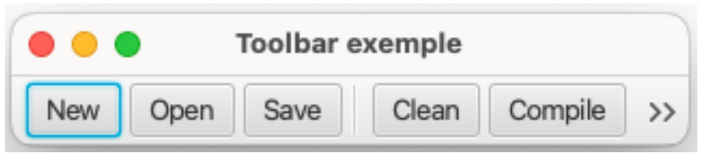
Gestionnaires qui n'héritent pas de Pane

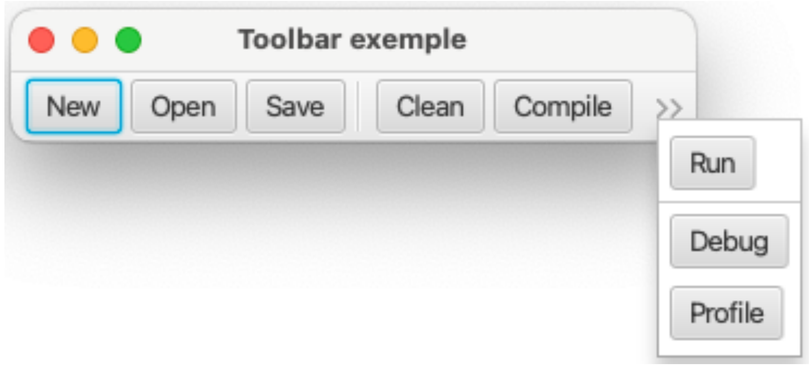


Toolbar

cours1 Layout/ToolbarExample.java



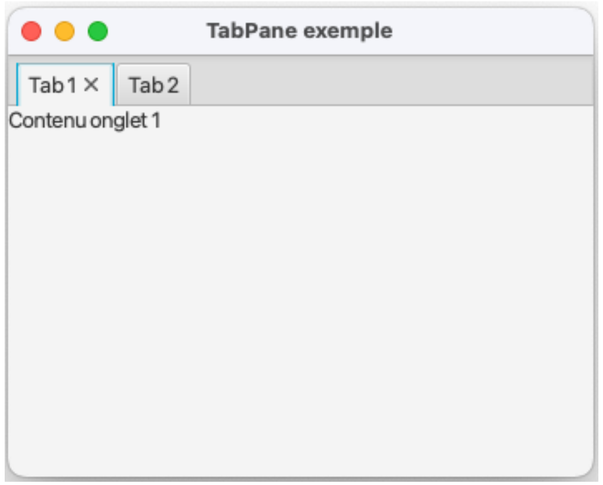





```
ToolBar tb = new ToolBar(
    new Button("New"),
    new Button("Open"),
    new Button("Save"),
    new Separator(),
    new Button("Clean"),
    new Button("Compile"),
    new Button("Run"),
    new Separator(),
    new Button("Debug"),
    new Button("Profile")
);

Scene scene = new Scene(tb);
```

TabPane

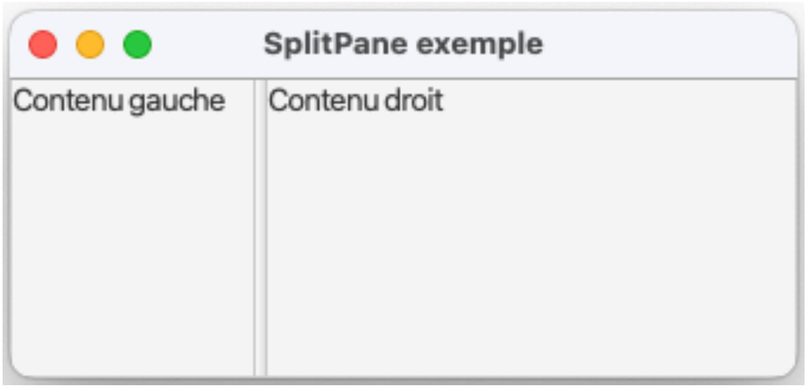
cours1 Layout/TabPaneExample.java




```
TabPane tabPane = new TabPane();

Tab tab1 = new Tab();
tab1.setText("Tab 1");
Label lbl1 = new Label("Contenu onglet 1");
tab1.setContent(lbl1);

Tab tab2 = new Tab();
tab2.setText("Tab 2");
Label lbl2 = new Label("Contenu onglet 2");
tab2.setContent(lbl2);

tabPane.getTabs().addAll(tab1, tab2);
```

SplitPane

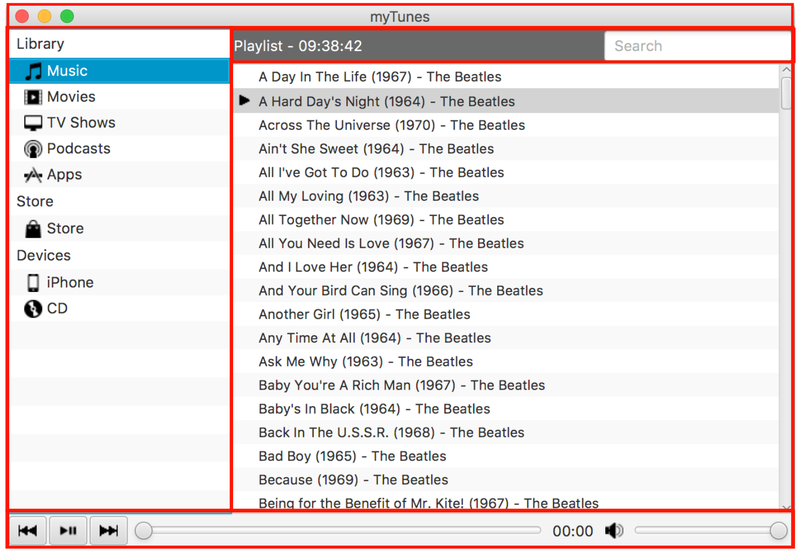
cours1Layout/SplitPaneExample.java

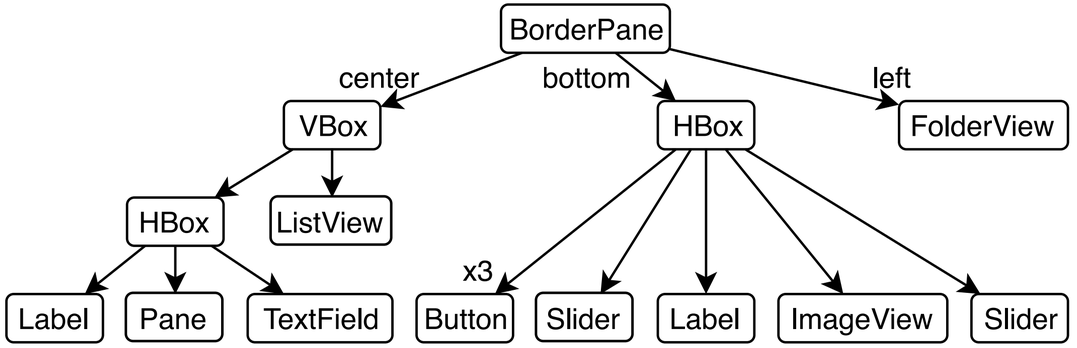



```
SplitPane sp = new SplitPane();

HBox hb1 = new HBox();
Label lbl1 = new Label("Contenu gauche");
hb1.getChildren().add(lbl1);

HBox hb2 = new HBox();
Label lbl2 = new Label("Contenu droit");
hb2.getChildren().add(lbl2);

sp.getItems().addAll(hb1, hb2);
sp.setDividerPositions(0.3);
```

chacun des items doit être un gestionnaire de placement pour que le(s) séparateur(s) soi(en)t manipulable(s)

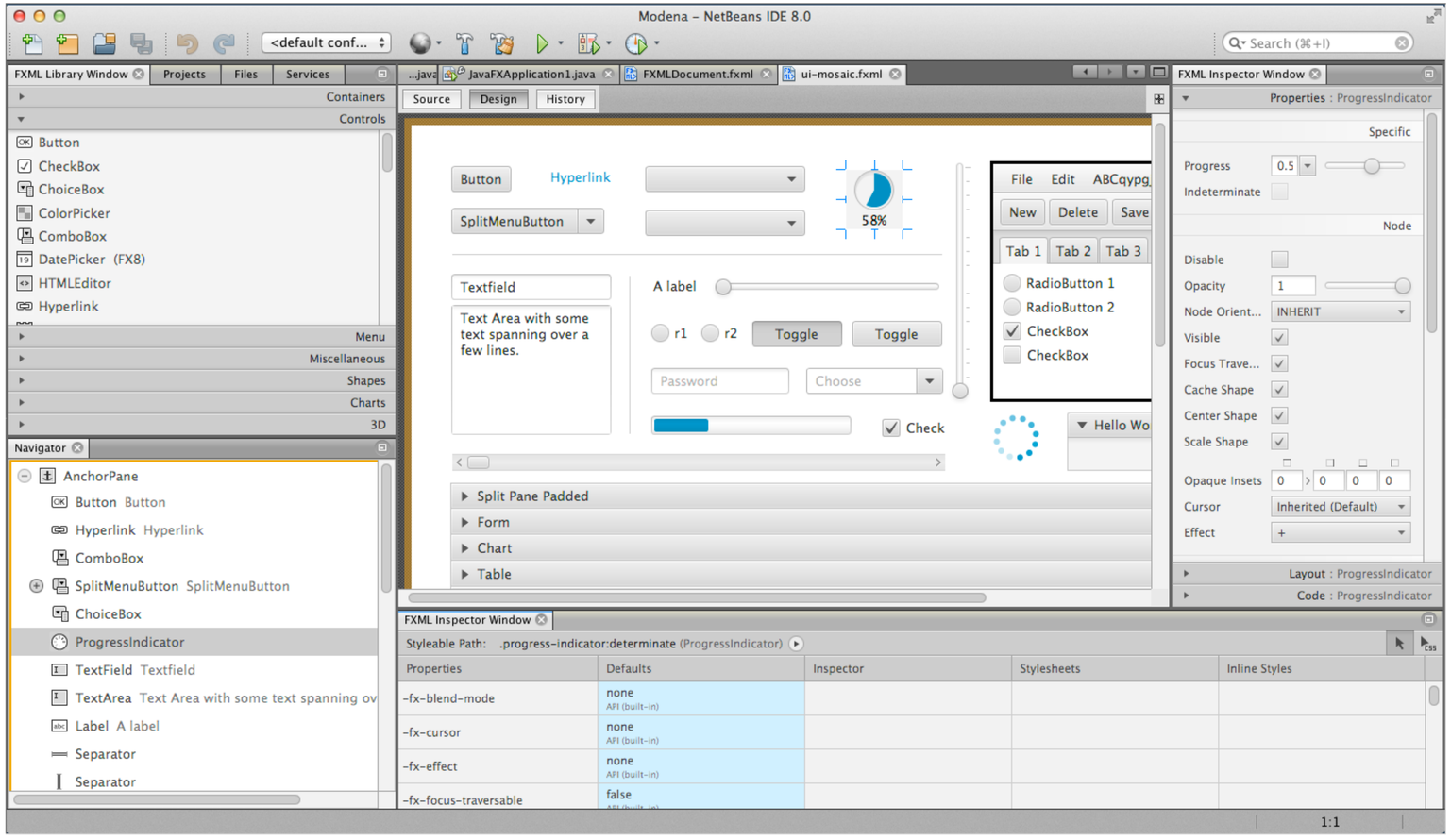
Exemple

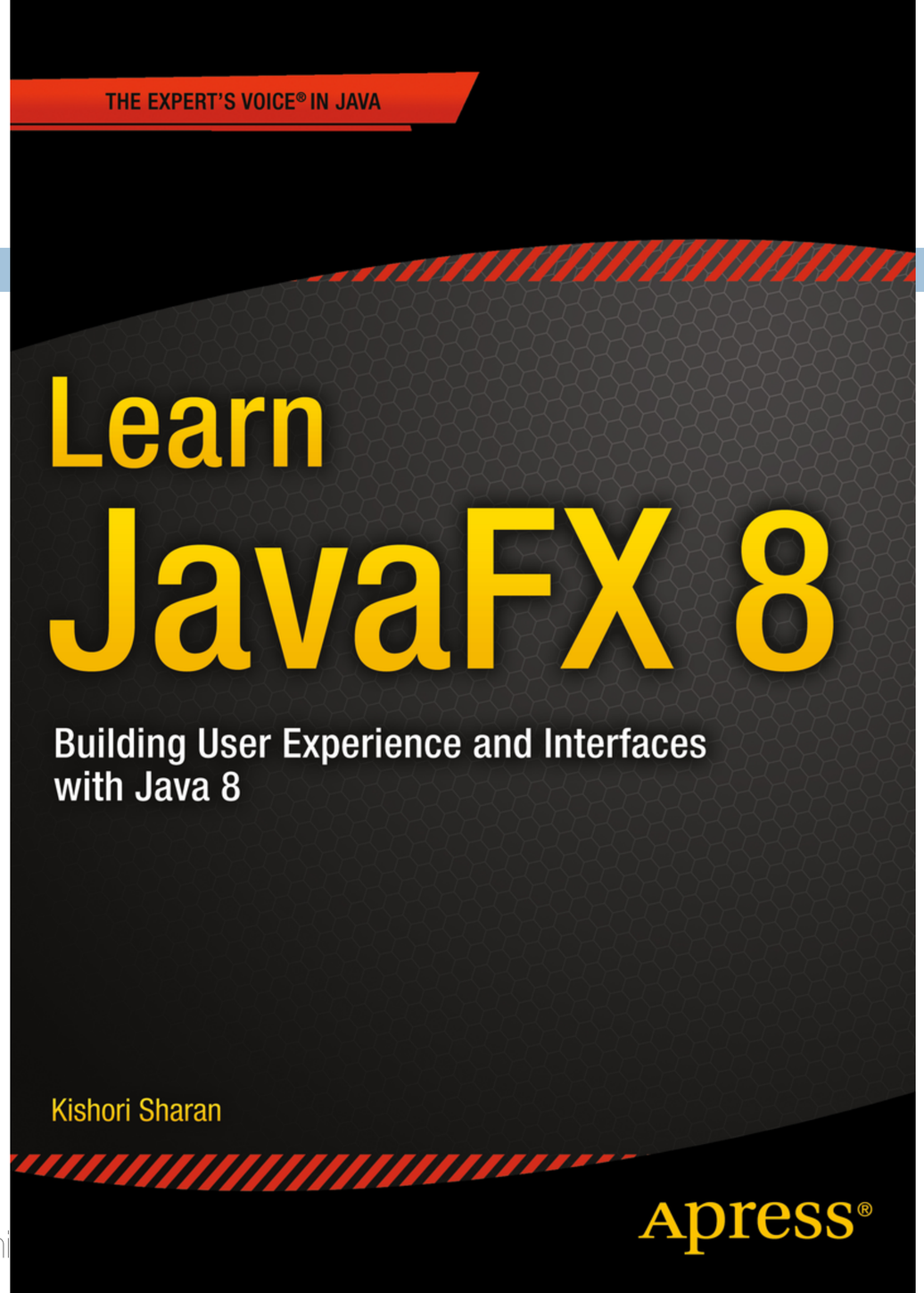

myTunes
Library
Music
Movies
TV Shows
Podcasts
Apps
Store
Store
Devices
iPhone
CD
Playlist - 09:38:42
Search
A Day In The Life (1967) - The Beatles
A Hard Day's Night (1964) - The Beatles
Across The Universe (1970) - The Beatles
Ain't She Sweet (1964) - The Beatles
All I've Got To Do (1963) - The Beatles
All My Loving (1963) - The Beatles
All Together Now (1969) - The Beatles
All You Need Is Love (1967) - The Beatles
And I Love Her (1964) - The Beatles
And Your Bird Can Sing (1966) - The Beatles
Another Girl (1965) - The Beatles
Any Time At All (1964) - The Beatles
Ask Me Why (1963) - The Beatles
Baby You're A Rich Man (1967) - The Beatles
Baby's In Black (1964) - The Beatles
Back In The U.S.S.R. (1968) - The Beatles
Bad Boy (1965) - The Beatles
Because (1969) - The Beatles
Being for the Benefit of Mr. Kite! (1967) - The Beatles
00:00


BorderPane
center
bottom
left
VBox
HBox
FolderView
HBox
ListView
x3
Label
Pane
TextField
Button
Slider
Label
ImageView
Slider

Générateurs d'interfaces

JavaFX Scene Builder




THE EXPERT'S VOICE® IN JAVA
Learn
JavaFX 8
Building User Experience and Interfaces with Java 8
Kishori Sharan
Apress®

A retenir

Utilisation de conteneurs pour regrouper des Nodes

Fenêtre = Stage

Différentes stratégies de positionnement