

---

## Interaction Homme-Machine

### TP 3 : Calculatrice fonctionnelle MVC

---

#### Objectifs :

- *Utilisation de l'architecture Modèle – Vue – Contrôleur (MVC)*
- *Utilisation de Scene Builder pour programmer l'interface graphique d'une calculatrice*

Dans cet exercice vous allez développer une calculatrice fonctionnelle en utilisant l'architecture MVC. L'architecture Modèle-Vue-Contrôleur (MVC) est une façon d'organiser une interface graphique d'un programme. Elle consiste à distinguer trois entités distinctes qui sont, le modèle, la vue et le contrôleur ayant chacun un rôle précis dans l'interface.

- Modèle : gestion et manipulation des données
- Vue : interface graphique utilisateur (entrées et sorties des données)
- Contrôleur : gestion des événements et synchronisation

Le modèle MVC permet une conception plus claire et efficace grâce à la séparation des données de la vue et du contrôleur.

La calculatrice à développer dans cet exercice doit être affichée de la manière suivante :

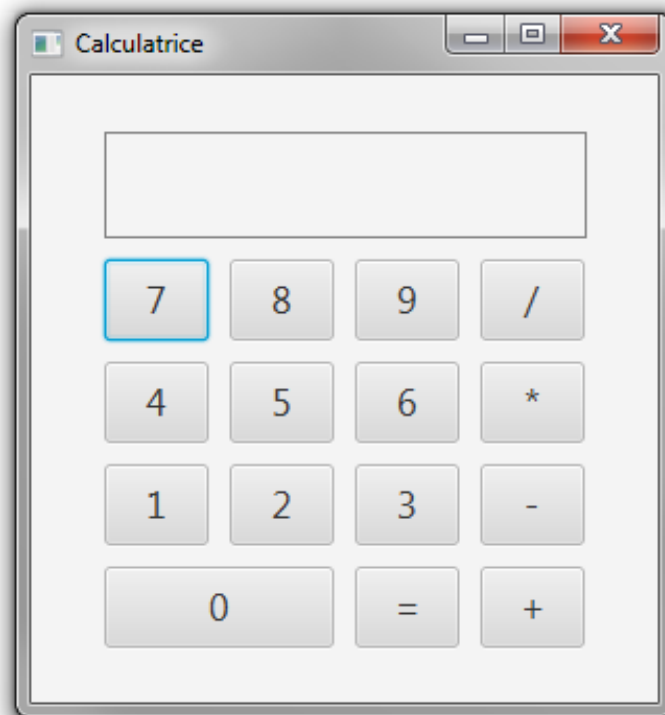


Figure 1 : Interface graphique « Calculatrice »

Dans cette interface graphique, les nœuds sont contenus en utilisant un **VBox**. Les nœuds graphiques seront positionnés de la manière suivante :

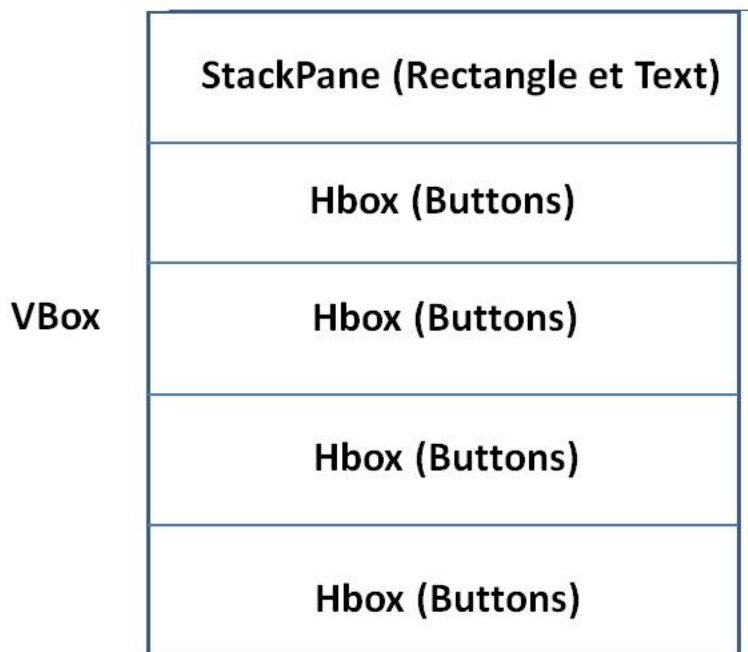


Figure 2 : Nœuds positionnés dans le VBox

Suivez les indications suivantes pour programmer l'interface graphique de la figure 1 :

1. Créez un nouveau projet TP3
2. Dans le projet TP3, créez un package nommé **calculatrice**
3. Dans le package, créez les 4 fichiers suivants :
  - a. Une classe principale JavaFX nommée **CalculatriceApp**
  - b. Une classe nommée **CalculatriceController**
  - c. Une classe nommée **CalculatriceModele**
  - d. Un document FXML nommé **CalculatriceGUI**.
4. Dans la méthode `start()` de la classe principale (**CalculatriceApp**), chargez votre fichier FXML (**CalculatriceGUI**) et mettez le titre « Calculatrice » dans votre Stage.
5. Utilisez **Scene Builder** pour construire une interface graphique similaire à celle de la figure 1 en respectant la structure de la figure 2. Pour installer **Scene Builder** et savoir comment l'utiliser, suivez le tutoriel « **Introduction à Scène Builder** » qui se trouve sur le serveur Moodle. Pour ajouter les composants graphiques, respectez les consignes suivantes :
  - a. Créez (glissez et déposez) la racine (**VBox**) en définissant un écart entre les nœuds graphiques de 10 pixels (`spacing="10"`), une longueur et une hauteur de 300 (`prefWidth="300" prefHeight="300"`). Spécifiez votre classe contrôleur (`calculatrice.CalculatriceController`).

- b. Créez (glisser et déposer) un **StackPane** avec un alignement centrée (`alignment="CENTER"`).
  - c. Dans le **StackPane**, créez un nœud **Rectangle** transparente (`fill="TRANSPARENT"`), de couleur gris (`stroke="GRAY"`) de longueur de 230 (`width="230"`) et une hauteur de (`height="50"`).
  - d. Dans le **StackPane**, créez un nœud **Text** avec un id (`fx:id`). .
  - e. Créez un **HBox** qui contiendra les boutons 7, 8, 9, /
  - f. Créez un **HBox** qui contiendra les boutons 4, 5, 6, \*
  - g. Créez un **HBox** qui contiendra les boutons 1, 2, 3, -
  - h. Créez un **HBox** qui contiendra les boutons 0, =, +
  - i. Chaque HBox aura un écart entre les nœuds graphique de 10 pixels (`spacing="10"`) et un alignement centrée (`alignment="CENTER"`).
  - j. Les boutons contenant de numéros feront appelle (onAction) à une méthode nommée *processNumpad* dans la classe contrôleur (**CalculatriceController**).
  - k. Les boutons contenant des opérateurs mathématiques (=, +)feront appelle (onAction) à une méthode nommée *processOperator* dans la classe contrôleur (**CalculatriceController**).
6. Dans la classe **CalculatriceModele** définir une méthode *Calculate* qui prendra comme paramètre deux numéros flottants et un opérateur (String). Le paramètre opérateur contiendra le symbole du calcul à réaliser ("/","\*","-","+"). La méthode *Calculate* retournera le résultat de ce calcul.

7. Dans la classe **CalculatriceController** définir les deux méthodes suivantes :

```
@FXML
private void processNumpad(ActionEvent event) {
    // S'il s'agit d'un nouveau calcul, effacez le texte dans l'écran de la
//calculatrice
    //Récupérer la valeur du numéro du texte du bouton cliqué par
    //l'utilisateur
    // Mettez le numéro dans l'écran de la calculatrice
}

@FXML
private void processOperator(ActionEvent event) {
    //Récupérer la valeur de l'operateur

    //Si l'opérateur n'est pas "=", récupérer l'opérateur cliqué et le
//numéro sur l'écran, ensuite effacez le texte dans l'écran

    // Si l'opérateur est "=", appeler la méthode Calculate dans la classe
// CalculatriceModele pour réaliser l'opération spécifié par
//l'utilisateur
}
```

Astuce : pour récupérer la valeur le texte d'un événement (click) sur un bouton, utilisez l'instruction `((Button)event.getSource()).getText()`