

# Compte rendu : Gestion d'un répertoire par interface graphique

Nom : Selahattin Unal  
Classe : SIO2



# Objectif

Le but de ce TP était de reprendre le répertoire développé lors du **TP3** et de lui ajouter une **interface graphique en JavaFX**.

L'application devait permettre :

- d'ajouter une personne avec ses informations (nom, prénom, adresse, téléphone),
- de naviguer dans les personnes enregistrées grâce aux boutons **Précédent** et **Suivant**,
- de quitter l'application proprement.

L'ensemble devait être structuré selon l'architecture **MVC (Modèle – Vue – Contrôleur)**.

## Architecture utilisée

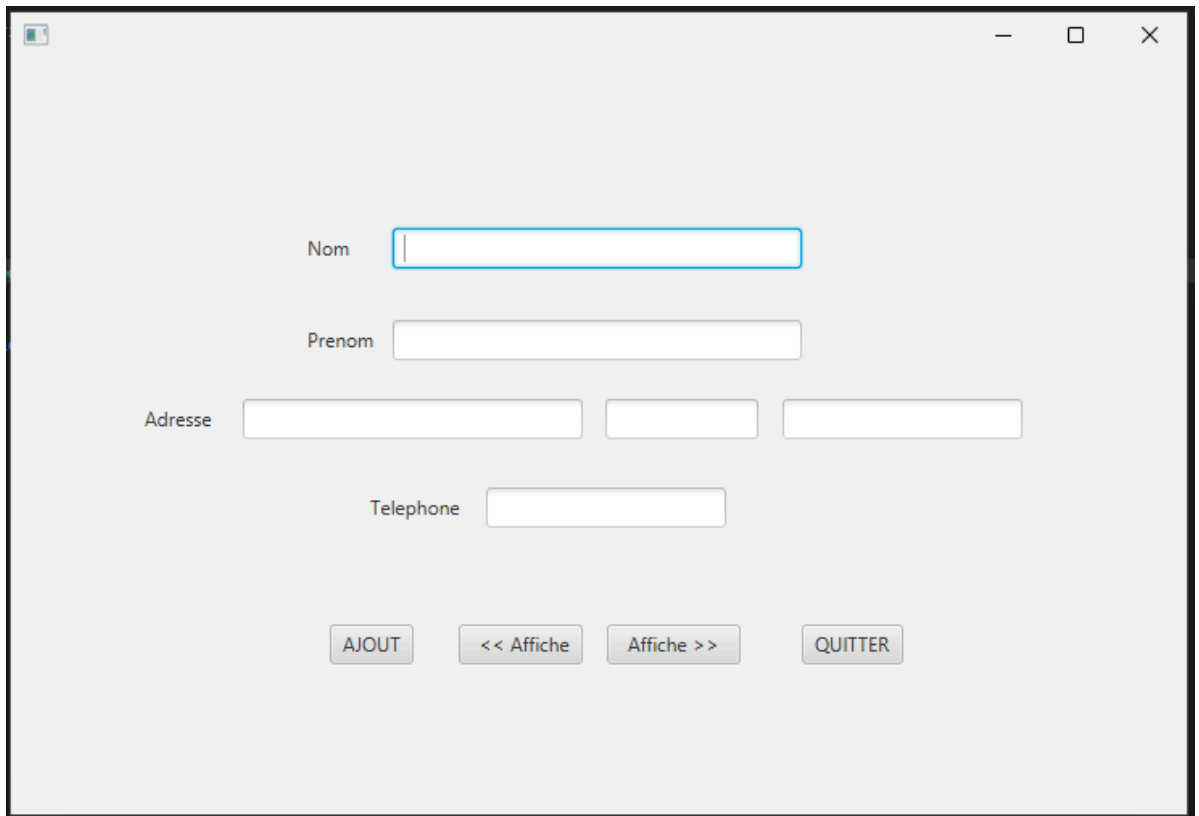
### 1. Le Modèle

- Le modèle est représenté par la classe **Répertoire**.
- Elle contient une collection de personnes (**ArrayList<Personne>**).
- J'ai ajouté deux méthodes importantes :
  - **recherche\_personne(int indice)** : retourne la personne à l'indice donné.
  - **taille()** : retourne le nombre total de personnes dans le répertoire.

Cela permet de séparer les données de la logique d'affichage.

## 2. La Vue

- La vue est définie dans le fichier Fenetre.fxml.
- J'ai utilisé Scene Builder pour créer l'interface facilement.
- Les éléments ajoutés :
  - Champs texte (**TextField**) :
    - **nom, prenom, rue, cp, ville, tel.**
  - Boutons (**Button**) :
    - **bAjout** → ajouter une personne,
    - **bAffprec** → afficher la personne précédente,
    - **bAffsuiv** → afficher la personne suivante,
    - **bQuit** → fermer l'application.



The screenshot shows a Java Swing window with a light gray background. At the top, there are standard window control buttons (minimize, maximize, close). The form contains the following elements:

- Nom**: A single text input field.
- Prenom**: A single text input field.
- Adresse**: Three text input fields arranged horizontally.
- Telephone**: A single text input field.
- Buttons**: Four buttons at the bottom: "AJOUT", "<< Affiche", "Affiche >>", and "QUITTER".

### 3. Le Contrôleur

- Le contrôleur est géré par la classe **Rep\_Contrôleur**.
- Il fait le lien entre la **vue** et le **modèle**.
- Attributs principaux :
  - **rep** de type Répertoire → contient toutes les personnes.
  - **numCourant** de type int → indique quelle personne est affichée.
- Gestion des événements :
  - **AJOUT** → récupère le contenu des **TextField**, crée une **Personne** et l'ajoute au **Répertoire**.
  - **Affiche <<** → décrémente **numCourant** et affiche la personne précédente.
  - **Affiche >>** → incrémente **numCourant** et affiche la personne suivante.
  - **QUITTER** → ferme la fenêtre et arrête l'application.

```
1 package application;
2
3 import javafx.fxml.FXML;
4 import javafx.scene.control.TextField;
5 import javafx.scene.control.Button;
6 import javafx.stage.Stage;
7 import javafx.scene.Node;
8 import javafx.event.ActionEvent;
9
10 public class SampleController {
11
12     @FXML private TextField nom;
13     @FXML private TextField prenom;
14     @FXML private TextField rue;
15     @FXML private TextField cp;
16     @FXML private TextField ville;
17     @FXML private TextField tel;
18
19     @FXML private Button bAjout;
20     @FXML private Button bAffprec;
21     @FXML private Button bAffsuiv;
22     @FXML private Button bQuit;
23
24     private int numCourant = -1; // index courant dans le répertoire
25     private Répertoire rep = new Répertoire(); // modèle
26
27     @FXML
28     private void onAjout(ActionEvent event) {
29         // Récupération des données
30         String n = nom.getText();
31         String p = prenom.getText();
32         String r = rue.getText();
33         String c = cp.getText();
34         String v = ville.getText();
35         String t = tel.getText();
36
37         // Création d'une personne
38         Adresse adresse = new Adresse(n, c, v);
39         Personne pers = new Personne(n, p, adresse, t);
40
41         // Ajout au répertoire
42         rep.ajouter_personne(pers);
43         numCourant = rep.taille() - 1;
44
45         // Affichage console
46         System.out.println("AJOUT -> " + pers);
47     }
48 }
```

## Réalisation pas à pas

1. J'ai créé un **nouveau projet JavaFX** nommé *TP5*.
2. J'ai copié les classes **Repertoire.java**, **Adresse.java** et **Personne.java** du TP3.
3. J'ai complété la classe **Répertoire** avec les nouvelles méthodes (**recherche\_personne**, **taille**).
4. J'ai conçu la **Vue** dans *Scene Builder* en ajoutant les champs texte et les boutons.
5. J'ai relié les boutons aux méthodes du **Contrôleur** dans le fichier **.fxml**.
6. J'ai testé progressivement chaque bouton pour vérifier leur bon fonctionnement.
- 7.

## Conclusion

Ce TP m'a permis de :

- Mettre en pratique l'architecture **MVC** en JavaFX,
- Apprendre à utiliser **Scene Builder** pour créer une interface graphique,
- Manipuler des **collections de personnes** via le modèle,
- Gérer des **événements utilisateurs** (clic sur un bouton).

L'application est fonctionnelle et répond aux objectifs fixés dans le sujet.

