



## **Sommaire**

Objectifs du TP

Architecture du projet (MVC)

3.1 Modèle

3.2 Main

3.3 Contrôleur

Réalisation

4.1 Création du projet JavaFX

4.2 Partie Modèle

4.3 Partie Vue (Fenetre.fxml)

4.4 Partie Contrôleur (Rep Contrôleur.java)

Problèmes rencontrés et solutions

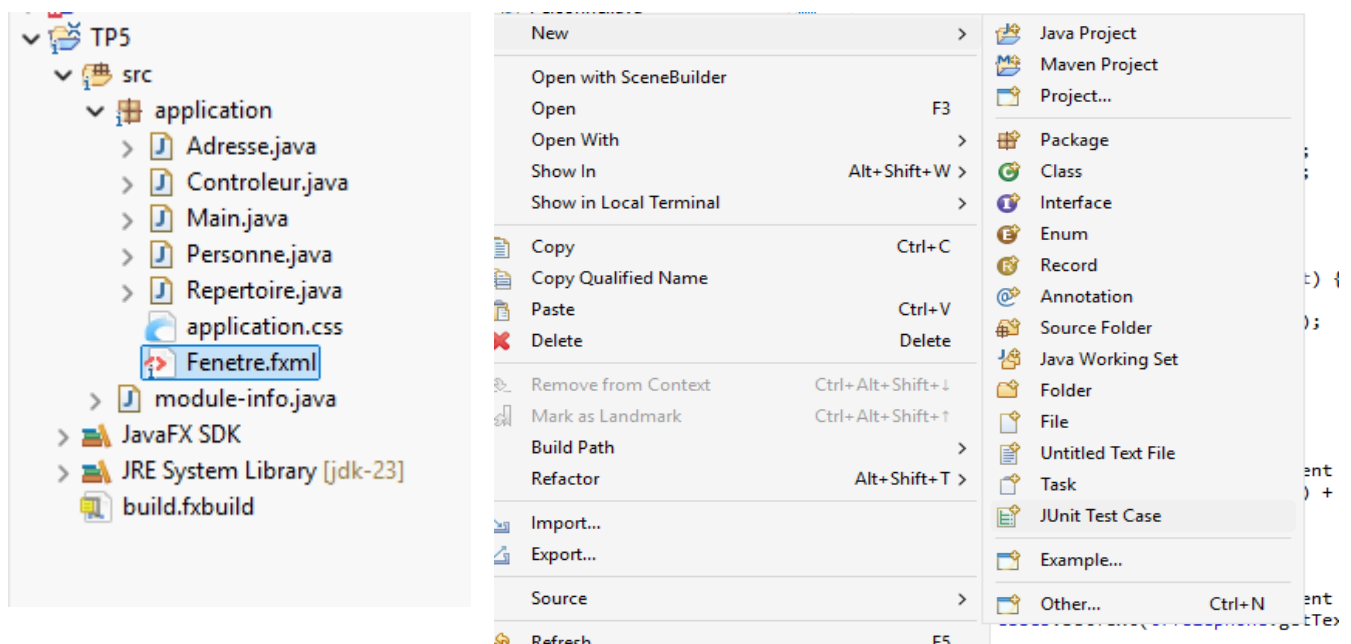
## Conclusion

## Objectifs du TP

Les objectifs du TP sont de créer une interface graphique avec JavaFX pour gérer un répertoire de personnes, de séparer le code selon l'architecture Modèle-Vue-Contrôleur, de gérer les événements utilisateurs tels que les clics sur les boutons et la navigation dans le répertoire, et de manipuler des objets représentant les personnes et leurs adresses.

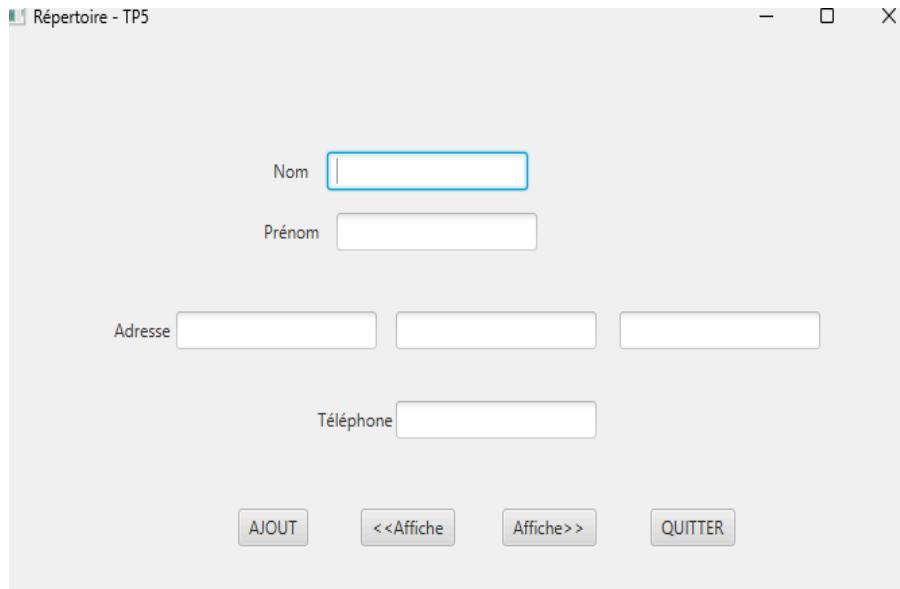
## Architecture du projet (MVC)

### Structure



Voici la structure du TP5 et comment créer un projet en JAVA FXML new, other et on sélectionne le projet en FXML.

je fait des copier coller de class de TP1 comme par exemple Repertoire, Adresse, Personne et je modifier le package défaut par application.



Voici l'interface graphique du TP5 on peut retrouver plusieurs élément important les champ et le bouton qui est un élément important pour faire le lien avec le code fenetre.fxml et l'interface graphique.

### Fenetre.fxml

```
<?xml version="1.0" encoding="UTF-8"?>
<?import javafx.scene.control.Button?>
<?import javafx.scene.control.Label?>
<?import javafx.scene.control.TextField?>
<?import javafx.scene.layout.AnchorPane?>
<AnchorPane prefHeight="400.0" prefWidth="674.0"
    xmlns="http://javafx.com/javafx/24.0.1"
    xmlns:fx="http://javafx.com/fxml/1"
    fx:controller="application.Controleur">
    <children>
        <Label layoutX="207.0" layoutY="78.0" text="Nom" />
        <TextField fx:id="tfNom" layoutX="247.0" layoutY="74.0"/>
        <Label layoutX="200.0" layoutY="118.0" text="Prénom" />
        <TextField fx:id="tfPrenom" layoutX="254.0" layoutY="114.0"/>
        <Label layoutX="89.0" layoutY="183.0" text="Adresse" />
```

```

<TextField fx:id="tfAdresse1" layoutX="135.0" layoutY="179.0"/>
<TextField fx:id="tfAdresse2" layoutX="298.0" layoutY="179.0"/>
<TextField fx:id="tfAdresse3" layoutX="464.0" layoutY="179.0"/>
<Label layoutX="240.0" layoutY="242.0" text="Téléphone" />
<TextField fx:id="tfTelephone" layoutX="298.0" layoutY="238.0"/>
<!-- Boutons -->
<Button fx:id="btnAjout" layoutX="181.0" layoutY="309.0" text="AJOUT" onAction="#onAjout"/>
<Button fx:id="btnAfficheGauche" layoutX="272.0" layoutY="309.0" text="&lt;&lt;Affiche"
onAction="#onAfficheGauche"/>
<Button fx:id="btnAfficheDroite" layoutX="377.0" layoutY="309.0" text="Affiche&gt;&gt;"
onAction="#onAfficheDroite"/>
<Button fx:id="btnQuitte" layoutX="487.0" layoutY="309.0" text="QUITTER"
onAction="#onQuitte"/>
</children>
</AnchorPane>

```

### 3.3 Contrôleur

voici le code controleur dans laquelle je mets des id dans l'interface graphique, le code controleur.java fait le lien avec le bouton et le champ de texte.

```

package application;
import javafx.fxml.FXML;
import javafx.scene.control.TextField;
import javafx.scene.control.Button;
import javafx.stage.Stage;
import javafx.scene.Node;
import javafx.event.ActionEvent;
public class Controleur {
    @FXML private TextField tfNom;
    @FXML private TextField tfPrenom;
    @FXML private TextField tfAdresse1;
    @FXML private TextField tfAdresse2;
    @FXML private TextField tfAdresse3;
    @FXML private TextField tfTelephone;
    @FXML private Button btnAjout;
    @FXML private Button btnAfficheGauche;
    @FXML private Button btnAfficheDroite;
    @FXML private Button btnQuitte;

    @FXML
    private void onAjout(ActionEvent event) {
        String nom = tfNom.getText();
        String prenom = tfPrenom.getText();
        tfAdresse1.setText(nom);
        tfAdresse2.setText(prenom);
    }

    @FXML
    private void onAfficheGauche(ActionEvent event) {
        tfAdresse3.setText(tfNom.getText() + " " + tfPrenom.getText());
    }
}

```

```

}
@FXML
private void onAfficheDroite(ActionEvent event) {
    tfAdresse3.setText(tfTelephone.getText());
}
@FXML
private void onQuitte(ActionEvent event) {
    Node source = (Node) event.getSource();
    Stage stage = (Stage) source.getScene().getWindow();
    stage.close();
}
}

```

La partie Vue est définie dans le fichier `Fenetre.fxml`. Elle comprend des champs `TextField` pour le nom, le prénom, la rue, le code postal, la ville et le téléphone (`tfNom`, `tfPrenom`, `tfRue`, `tfCp`, `tfVille`, `tfTel`). Les boutons présents sont `bAjout`, `bAffprec`, `bAffsuiv` et `bQuit`. Le layout utilise un `AnchorPane` ou des conteneurs `VBox/HBox` pour organiser correctement les champs et les boutons. Chaque élément possède un `fx:id` correspondant à un attribut du contrôleur afin de permettre l'interaction entre la vue et le code Java.

## Problèmes rencontrés et solutions

Le problème que j'ai pu rencontrer est que dans mon code `fenetre.fxml` il y a une fonction `ONaffiche gauche et droite` et cette même fonction n'était pas définie dans le code Contrôleur, j'étais vraiment bloqué là.

sinon dans l'ensemble les codes marchent donc tout va pour le mieux.

## Conclusion

Ce TP a permis de créer une interface graphique fonctionnelle pour gérer un répertoire de personnes. L'utilisation de l'architecture MVC a assuré la lisibilité et la maintenabilité du code. L'application permet d'ajouter, parcourir et afficher les personnes de manière simple et efficace. Les erreurs courantes rencontrées, comme l'utilisation de caractères spéciaux dans le FXML ou les méthodes manquantes dans le contrôleur, ont été corrigées, rendant le projet stable et opérationnel.