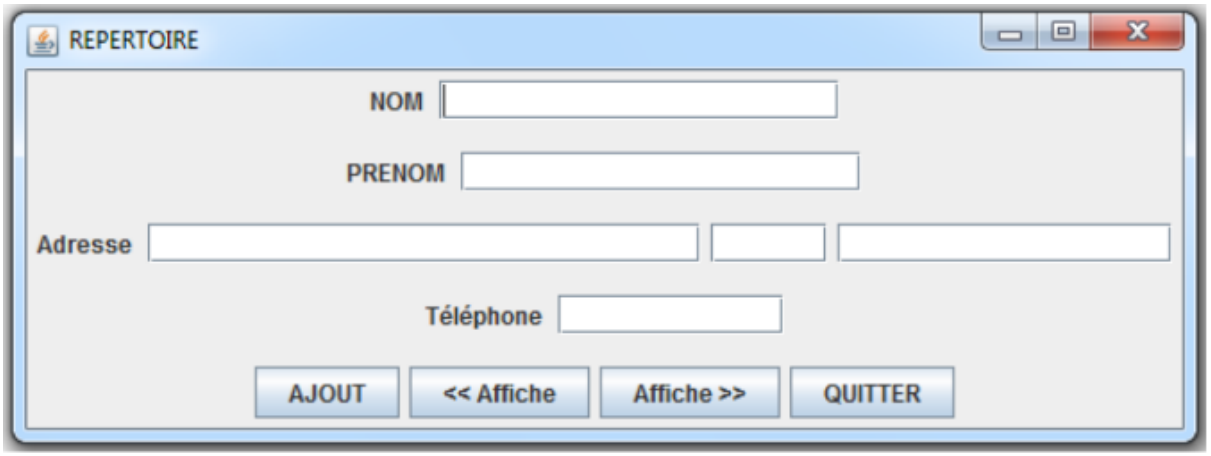
Compte rendu JAVA

Anthony Afonso

Objectif à atteindre :



Ici, L'objectif est créer un répertoire possédant 4 boutons : "AJOUT", permettant d'ajouter un nouveau compte grâce à NOM, PRÉNOM, Adresse, et Téléphone qui seront eux mêmes à créer, un bouton "<<Affiche" et "Affiche>>" qui vont détailler les comptes précédents et suivants, un bouton "QUITTER" qui sert à sortir de la page. Ce travail devra être personnalisé à la manière de chacun avec des changements de couleurs ou encore des éléments supplémentaires.

Mon objectif de résultat :

Ici, j'ai essayé d'être un peu original dans cette interface afin de faire un travail qui se distingue un peu notamment avec de la couleur.

Cette interface a été produite depuis le logiciel Scene builder.

Le code produit :

Partie controller

```java
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 Controller {

    @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();
    }
}
```

Ici, nous sommes dans la classe répertoire qui va me permettre d’afficher tous les comptes déjà enregistrés.

Partie répertoire

```java
package application;

public class Personne implements Comparable<Personne> {
    private String nom, prenom, telephone;
    private Adresse adresse;

    public Personne(String nom, String prenom) {
        this.nom = nom;
        this.prenom = prenom;
    }

    public Personne(String nom, String prenom, String telephone, Adresse adresse) {
        this.nom = nom;
        this.prenom = prenom;
        this.telephone = telephone;
        this.adresse = adresse;
    }

    public String getNom() { return nom; }
    public void setNom(String nom) { this.nom = nom; }

    public String getPrenom() { return prenom; }
    public void setPrenom(String prenom) { this.prenom = prenom; }

    public String getTelephone() { return telephone; }
    public void setTelephone(String telephone) { this.telephone = telephone; }

    public Adresse getAdresse() { return adresse; }
    public void setAdresse(Adresse adresse) { this.adresse = adresse; }


    public void affiche() {
        System.out.println(nom + " " + prenom + " " + adresse + " " + telephone);
    }

    @Override
    public String toString() {
        return nom + ";" + prenom + ";" + adresse + ";" + telephone;
    }

    @Override
    public int compareTo(Personne p) {
        int resultat = this.nom.compareTo(p.getNom());
        if (resultat == 0) {
            resultat = this.prenom.compareTo(p.getPrenom());
        }
        return resultat;
    }
}
```

```java
import java.util.ArrayList;
import java.util.Collections;

public class Repertoire {

    private ArrayList<Personne> liste;

    public Repertoire() {
        super();
        this.liste = new ArrayList<Personne>();
    }

    public String toString() {
        String laListe = "";
        for(Personne p : liste) {
            laListe += p + "\n";
        }
        return laListe;
    }

    public void ajoutePersonne(Personne p) {
        liste.add(p);
        Collections.sort(liste); // Assure-toi que Personne implémente Comparable
    }


    public Personne rechercheNom(String unNom) {
        for(Personne p : liste) {
            if(p.getNom().equals(unNom)) {
                return p;
            }
        }
        return null;
    }

    public Personne rechercheNomPrenom(String unNom,String unPrenom) {
        for(Personne p : liste) {
            if(p.getNom().equals(unNom) && p.getPrenom().equals(unPrenom)) {
                return p;
            }
        }
        return null;
    }

    // recherche
    public Personne recherche_personne(int index) {
        if (index >= 0 && index < liste.size()) {
            return liste.get(index);
        }
        return null;
    }

    // Nouvelle méthode : taille de la collection
    public int taille() {
        return liste.size();
    }
```

Dans le code présent, on peut retrouver les nominations de chaques éléments dont nous avons besoin pour enregistrer les comptes, pouvoir les lister, les rechercher et les afficher dans l'ordre alphabétique et en fonction des noms, prénoms, adresse, ou encore numéro de téléphone.

Conclusion

Finalement par manque d'efficacité dans le temps prévu, je n'ai pas réussi à avoir un résultat au niveau de l'affichage. Malgré ça, ce TP m'a fait découvrir Scene Builder que j'aime beaucoup utiliser, et m'a permis de comprendre que je manque encore d'efficacité et donc que j'ai encore beaucoup à apprendre.