

## TP 3 « Développement Android »

### Création d'un Fragment et communication entre Activity et Fragment

#### 1) Création d'un nouveau projet

Créer un nouveau projet Android vide nommé « **Tp3** » du domaine « **fsm.isi2** ». Ajouter l'activité vide suivante (**Empty Activity**) **SANS générer des fichiers Layout** :

- « **InfoActivity** » : activité principale (« **Launcher Activity** »)

Créer un fichier Layout nommé « **activity\_info.xml** » contenant un LinearLayout ayant l'identifiant « **llInfoContainer** ». Voir code suivant :

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical" android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:id="@+id/llInfoContainer">

</LinearLayout>
```

Redéfinir la fonction **onCreate()** de **InfoActivity** pour afficher le layout « **activity\_info.xml** ».

Créer l'objet « **ArrayList listEtudiants** » et remplir le avec des vecteurs de votre choix, tel que chaque vecteur contient les informations suivantes :

- Nom de l'étudiant ;
- Prénom de l'étudiant ;
- Formation de l'étudiant ;

#### 2) Création d'un Fragment

Créer un Fragment nommé « **InfoFragment** » qui hérite de la classe Fragment. Créer ensuite un Layout nommé « **fragment\_info.xml** » et qui affiche l'interface suivante :

The screenshot shows a mobile application interface titled "Info Etudiant" in a blue header. Below the header, there are three input fields with labels "Nom :", "Prenom :", and "Formation :". Each label is followed by a placeholder text in angle brackets: "< nom >", "< prenom >", and "< formation >". At the bottom of the interface, there are two buttons: "Supprimer" and "Suivant".

Redéfinir la fonction « `onCreateView()` » du fragment « `InfoFragment` » pour afficher le layout « `fragment_info.xml` ».

Le rôle du fragment « `InfoFragment` » est d'afficher le contenu de la liste « `listEtudiants` » (un seul étudiant est affiché à la fois). Le fragment permet aussi de parcourir les différents éléments de la liste grâce au bouton « suivant » et peut supprimer un élément grâce au bouton « supprimer ».

La liste « `listEtudiants` » appartient à l'activité, donc l'activité doit être capable de communiquer les détails d'un étudiant au fragment.

Les boutons « supprimer » et « suivant » appartiennent au fragment, donc le fragment doit être capable de communiquer les événements d'appui sur ces boutons à l'activité :

- L'appui sur le bouton « supprimer » permet au fragment de demander à l'activité de supprimer l'étudiant actuel de la liste « `listEtudiants` » et d'afficher l'étudiant suivant (si la liste n'est pas vide).
- L'appui sur le bouton « suivant » permet au fragment de demander à l'activité d'afficher l'étudiant suivant de la liste « `listEtudiants` ».

Implémenter la fonction publique suivante dans « `InfoFragment` » :

- **afficherEtudiant (String nom, String prenom, String formation)** : permet à l'activité de demander au fragment d'afficher les détails d'un étudiant.

Créer l'interface « `OnEtudiantEventListener` » à l'intérieur de « `InfoFragment` » avec les deux fonctions **`onSupprimerClicked()`** et **`onSuivantClicked()`**, comme suit :

```
public class InfoFragment extends Fragment {  
    ...  
    public interface OnEtudiantEventListener {  
        public void onSupprimerClicked();  
        public void onSuivantClicked();  
    }  
    ...  
}
```

Implémenter cette interface dans « `InfoActivity` » pour gérer les événements des boutons du fragment. **Remarque** : l'interface est déclarée dans le fragment, donc il faut l'implémenter comme suit : « `... implements InfoFragment.OnEtudiantEventListener` »

Implémenter les listeners des boutons « Supprimer » et « Suivant » dans « `InfoFragment` » : ces listeners permettent de notifier l'activité à l'intermédiaire des deux méthodes de l'interface.