



# TP ANDROID

## INTERFACES COMPLEXES



### ArrayList

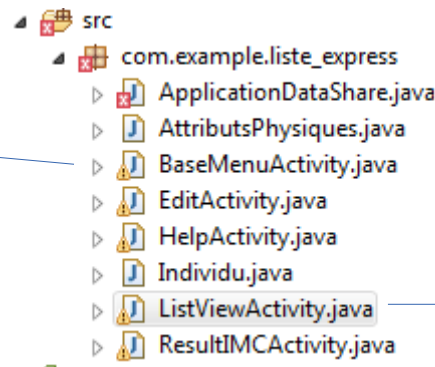
- Ouvrir le projet ListeIndividu
- Surcharger la méthode toString de la classe individu afin de réaliser un affichage de ce type. Tester l'affichage en créant dans le main un individu et en invoquant un println : Exemple : `System.out.println(etudiant);`
- Créer une liste d'individu et peuplez-la. Itérez la liste (parcourir) et afficher tous les individus.

NOM: Amel PRENOM:Oscar ATTRIBUTS 55kg 177cm  
 NOM: Copter PRENOM:Eli ATTRIBUTS 90kg 165cm  
 NOM: Sime PRENOM:Emille ATTRIBUTS 40kg 180cm

### Liste android version express

- Ouvrir le projet ListeExpressIMC.

Toutes les activités hériteront de cette classe.  
 Objectif : mise en place d'un menu commun par héritage



Classe permettant de définir des variables globales (alternative à la transmission de Bundle entre activités)

Activité rajout utilisateur à la liste

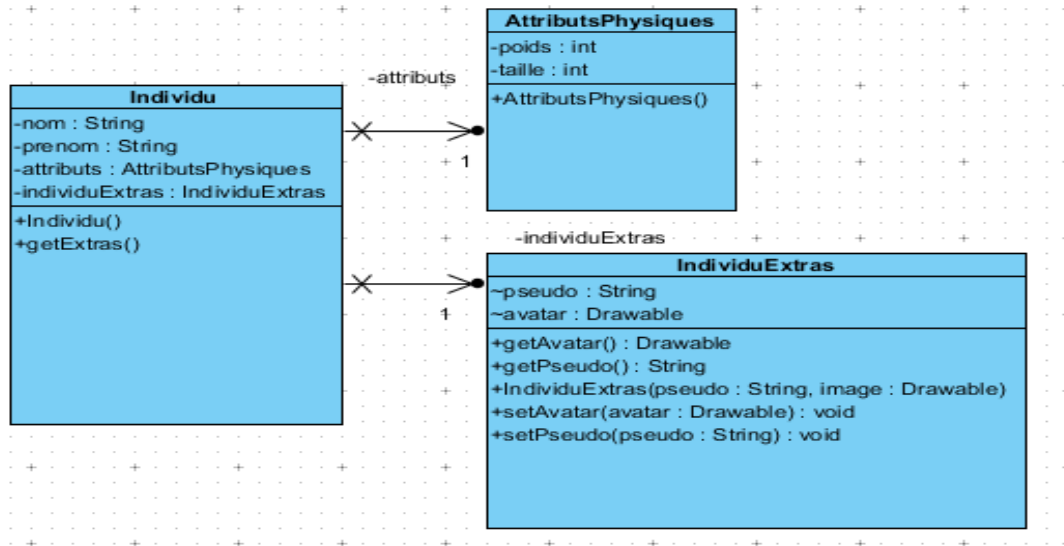
Activité principale : liste des individus suivis

- Etape 1 : Mise en place d'un liste sans interaction
  - peupler 'à la main' la liste d'individu définie dans la classe ApplicationDataShare. Les champs privés de cette classe seront accessibles depuis les activités par le biais des getter/setter. Exemple :  
`ArrayList<Individu> liste=ApplicationDataShare.getInstance().getListeIndividu();`
  - Mettre en place le ListView dans l'activité principale ListViewActivity. A quoi sert `updateList()` ?
  - Testez
- Etape 2 : gestion d'un clic sur la liste
  - Un clic sur un item doit lancer l'activité resultIMCActivity. Plutôt que d'utiliser un Bundle une variable globale est utilisée. Ouvrez ResultIMCActivity et observez la méthode onCreate pour comprendre le passage de donnée.
  - Testez
- Etape 3 : gestion du rajout d'un utilisateur depuis l'activité EditActivity
  - Ouvrez le fichier EditActivity et aller au niveau du handler de click. Crée un individu à partir des données saisies depuis l'interface graphique et rajouter le à la liste globale (pensez à récupérer une référence sur cette liste en utilisant la technique vue au-dessus)
  - Testez

### Liste android mise en forme complexe

L'objectif est de réaliser l'affichage d'une liste avec les noms et prénoms des individus accompagné d'un avatar et d'un pseudo.

Une classe IndividuExtras est rajouté comme suit :



- Compléter la classe IndividuExtras
- Compléter le layout de disposition de la liste afin de produire un résultat analogue à ceci :



```

RelativeLayout: @+id/list_view_adapter
    ImageView: @+id/iv_list_avatar
    TextView: @+id/tv_list_prenom
    TextView: @+id/tv_list_nom
    TextView: @+id/tv_list_pseudo
    
```

- Compléter le code de l'adaptateur de liste : IndividuListAdapter ( remarquez au passage la ligne Individu record=getItem(position) qui permet de récupérer les infos des individus lors de l'itération de la liste)
- Pour finir compléter la mise en place du ListView dans l'activité principale ListViewActivity
- Testez

### Fragments statiques

La mise en place est identique à celle du cours

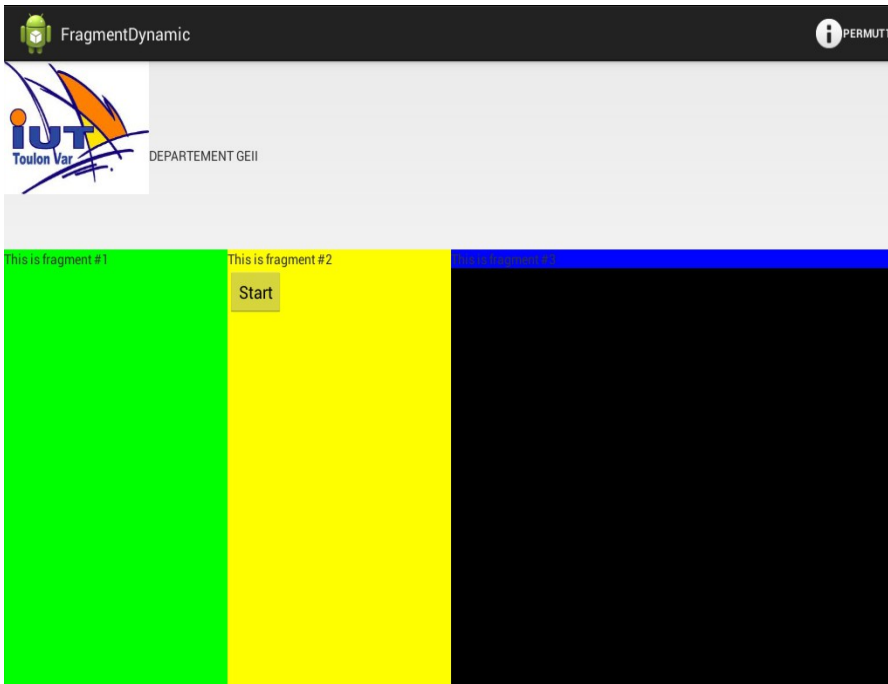


- Ouvrez le projet FragmentsStatic et complétez les portions manquantes. Le changement de couleur du texte du fragment 2 se fera au run-time (programmation). Testez.

### Fragment dynamique

L'objectif est la réalisation d'une interface graphique contenant 1 bandeau en haut commun et de 3 fragments,

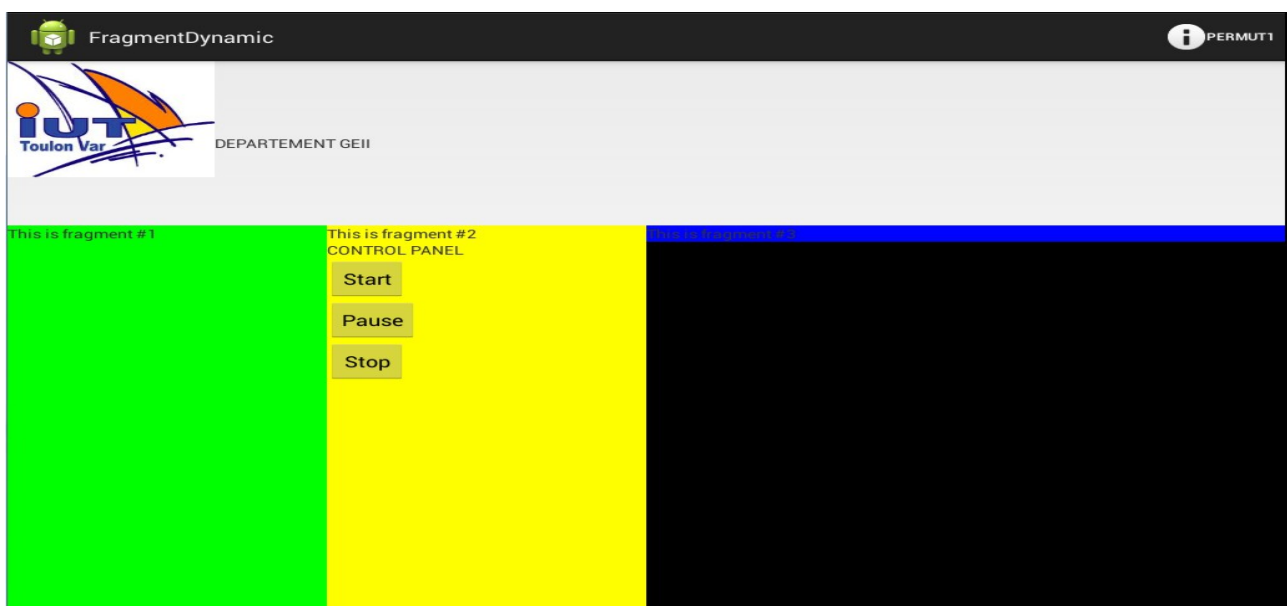
dont 1 contiendra une video. Le projet est FragmentDynamique



- Télécharger dans votre tablette , répertoire /sdcard/Movies/ ,le fichier robot.mp4.
- Rajouter dans le layout du fragment3 la balise `VideoView`
- Compléter le code de l'activité principale de telle manière à rajouter le fragment 3
- Testez la rotation. Testez la permutation des fragments 1 et 2
- Etudier l'ensemble du code

### Communication inter fragments

L'interface présente est reprise . L'idée est de rajouter des boutons de contrôles sur le fragment 2 qui piloteront la video.



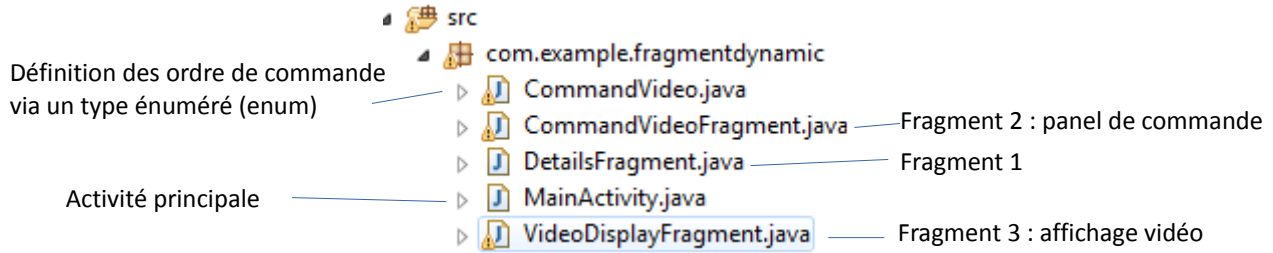


# TP ANDROID

## INTERFACES COMPLEXES



- Ouvrez le projet FragmentDynamicListener



- compléter les portions de code manquantes dans le fragment 2 (c'est le sujet : Observable) .  
Le contrat de l'écouteur est :

```
public interface OnCommandVideoListener {  
    public void onClickVideoCommand(CommandVideo command);  
}
```

L'abonné (Observer) est l'activité parente qui est chargée de relayer l'info vers le fragment 3 (celui qui contient la vidéo!).

- compléter les portions de code manquantes dans le fragment 3. Testez