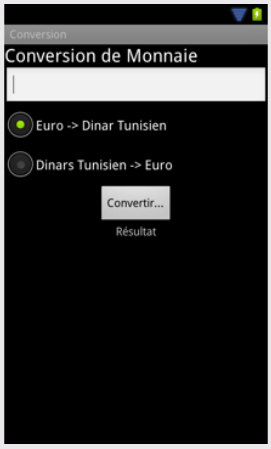
TP3 : Composants Android

Objectifs du TP

Ce TP a pour objectif de vous initier aux différents composants importants d'Android.

Nous allons dans ce TP réaliser progressivement une petite application de conversion.

TAF-1 **:** Créer un nouveau projet intitulé *Conversion*, qui contient le package *isi.conversion*. Créer les éléments nécessaires pour que l'interface soit comme la figure suivante :



I. Boutons, boutons radios et cases à cocher

I. 1. Associer un comportement à un bouton

1. Créer une méthode dans le code Java de l'activité qui définit le comportement du bouton. On l'appellera

```
public void auClicMethode(View v){…}
```

   Cette méthode doit obligatoirement être publique, retourner *void* et avoir un paramètre de type *android.view.View*.

2. Créer un bouton dans le fichier *layout* en utilisant la partie graphique.
3. Modifier son identifiant et son texte à votre guise

Remarque : Il vous est possible de modifier ces informations graphiquement. Pour cela, ouvrir la partie graphique du fichier *layout*, clic-droit sur l'élément à configurer, et choisir *Edit Text*, *Edit ID*…

4. Dans le code xml de votre bouton, ajouter l'attribut :

```
android:onClick = "@string/auClic"
```

5. Créer dans le fichier *strings.xml* un nouveau string dont le nom est *auClic* et la valeur est *auClicMethode* (qui est le nom de la méthode que vous avez créé dans 1.)

TAF-2 **:** Créer une méthode appelée *convertir* et l'associer au bouton *Convertir* de votre interface.

I. 2. Gérer les boutons radios

Un bouton radio est un bouton à deux états qui peut être soit coché (*checked*) ou décoché *(unchecked)*. Les boutons radios sont en général utilisés dans un groupe *RadioGroup*. Au sein d'un même groupe, un seul bouton radio peut être coché.

Pour gérer l'état d'un bouton radio, il faut suivre les étapes suivantes :

1. Créer un attribut de type *RadioButton* dans votre activité (par exemple *radio1*).
2. L'associer au bouton radio approprié de votre interface en utilisant la méthode *findViewById*.
3. Pour tester l'état de votre bouton radio, appeler la méthode *isChecked()*. Par exemple :

```
if (radio1.isChecked() ){
        //traitement
}
```

***TAF-3* :**

1. Créer deux méthodes : dinarToEuro et euroToDinar, qui prennent de convertir une valeur en entrée :

```
private float dinarsToEuro(float valeurDinar) {
        return (float) (valeurDinar * 1.9919);
}

private float euroToDinar(float valeurEuro) {
        return (float) (valeurEuro * 0.5020);
}
```

2. Implémenter la méthode *convertir* pour qu'elle fasse la conversion nécessaire, selon le bouton radio qui est coché. Mettre le résultat dans le champs de texte *Resultat*.

*Indication : La valeur lue dans le champs de saisie (ici appelé **edt**) doit être convertie en float pour être manipulée. Pour cela, utiliser le code suivant :*

```
EditText edt = (EditText) findViewById(R.id.edit_float);
float number = Float.valueOf(edt.getText().toString());
```

*D'autre part, pour extraire la chaîne de caractères associée à une variable float (appelée ici **floatVar**), utiliser le code suivant :*

```
String s = String.valueOf(floatVar) ;
```

I. 3. Gérer les cases à cocher

Tout comme les boutons radio, les cases à cocher ont deux états : coché ou décoché. Cependant, on peut avoir plusieurs cases qui sont cochées en même temps, et elle sont la plupart du temps indépendantes.

Pour gérer l'état d'une case à cocher, il faut suivre les étapes suivantes :

1. Créer un attribut de type *CheckBox* dans votre activité (par exemple *check1*).
2. L'associer à la case à cocher appropriée de votre interface en utilisant la méthode *findViewById*.
3. Pour tester l'état de votre case à cocher, appeler la méthode *isChecked()*. Par exemple :

```
if (check1.isChecked() ){
          //traitement
}
```

4. Pour modifier l'état de la case à cocher, utiliser la méthode *setChecked(boolean etat)*. Par exemple :

```
check1.setChecked(false) ;        //pour décocher la case
check1.setChecked(true) ;         //pour cocher la case
```