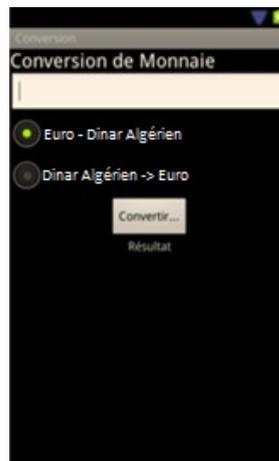


N.B. : Ce TP fera l'objet d'un compte-rendu individuel en format PDF, qui doit être envoyé avant le 23/08/2020 à l'enseignant^{e1,e2} responsable de ce TP. Le compte rendu doit contenir les programmes développés (contenu du fichier java + contenu du fichier xml). Vous êtes libres d'ajouter tout élément d'information qui vous semblera judicieux de donner.

But

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 l3si.conversion. Créer les éléments nécessaires pour que l'interface soit comme la figure suivante :



I. Boutons, boutons radios

a. 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"
```

e1^{e1} G1, G3 : rimadehana@gmail.com

e2^{e2} G2, G4, G5, G6, G7 : rayene.bennecer@gmail.com

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.

b. 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 / 1.47);
    }
private float euroToDinar(float valeurEuro)
    { return (float) (valeurEuro *1.47); }
```

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) ;
```