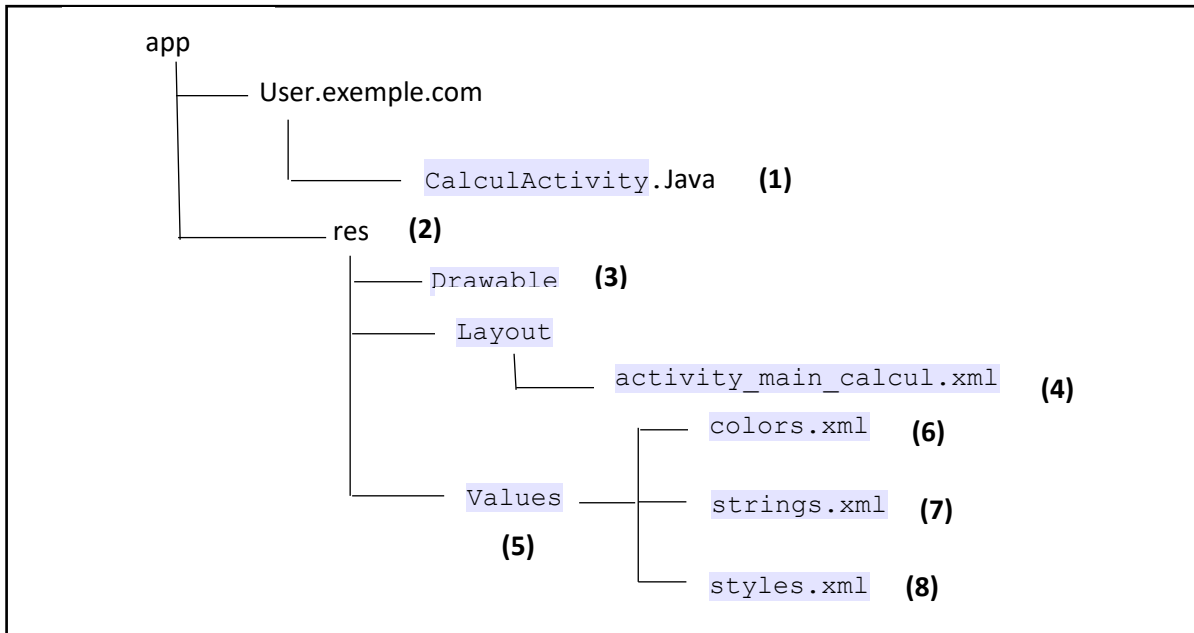


## Série d'exercices N° 2

### Exercice 1

En créant une nouvelle application, intitulée Calcul, vous obtenez l'arborescence suivante :



1. Donner le nom associé à chacun des numéros.

*la class principale (CalculActivity.java), (2) : res : ressource de l'application (3) : image de l'application comme .png, .jpg, .gif, (4) : l'interface de l'activité principale, (5) : values : dossier regroupe les valeurs simple comme les entiers et les chaines de caractères utilisés par l'application comme suit : (6) colors.xml pour les différentes valeur de couleurs utilisé par l'application, (7) strings.xml : contient les textes utilisés par l'application comme les nom des boutons, des labels, des textes par défauts ..., (8) styles.xml pour les styles*

2. Quelle est la class java générée à partir de l'élément (2) et quel est son rôle ?

*La class java générée à partir de l'élément (2) res est la class R, elle représente les ressources de l'application sous forme d'éléments Java, pour qu'ils soient utilisés dans le code Java des activités*

### Exercice 2

Dans votre application, vous avez commencé par modifier l'interface.

1. Quel fichier devez-vous modifier pour définir l'interface.

*Le fichier activity\_main\_calcul.xml*

2. Vous avez écrit le code suivant :

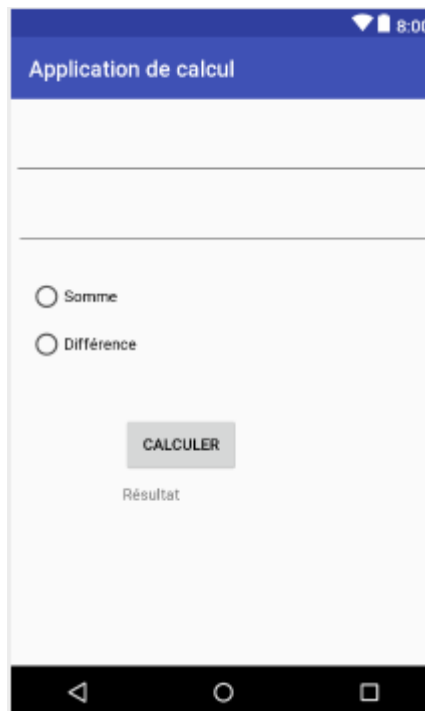
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout (...) >
  <TextView
```

```

        android:id="@+id/titre"
        android:text="Application de calcul"/>
<EditText
    android:id="@+id/val1"
    android:inputType="number"/>
<EditText
    android:id="@+id/val2"
    android:inputType="number" />
<RadioGroup
    android:id="@+id/operations">
    <RadioButton
        android:id="@+id/somme"
        android:checked="true"
        android:text="Somme" />
    <RadioButton
        android:id="@+id/diff"
        android:text="Différence" />
</RadioGroup>
<Button
    android:id="@+id/calculer"
    android:layout_gravity="center"
    android:text="Calculer"
    android:onClick="calculer" />
<TextView
    android:id="@+id/res"
    android:layout_gravity="center"
    android:text="Résultat" />
</LinearLayout>

```

a. Dessiner l'interface que vous obtenez avec ce code.



b. A quoi sert l'élément *RadioGroup* ?

*RadioGroup permet de rassembler les RadioButtons dans un groupe, pour les associer et permettre qu'un seul bouton radio soit coché à la fois.*

c. A quoi sert l'attribut `android:layout_gravity="center"`?

*Cet attribut permet de centrer l'élément graphique auquel il est associé.*

### Exercice 3

Vous allez modifier le fichier `CalculActivity.java`

1. Qu'est-ce qu'une activité en Android ? De quelle classe doit-elle hériter ?

*Une activité est la composante principale pour une application Android. Elle représente l'implémentation métier dans une application Android, permettant de gérer l'ensemble des vues et ressources. Elle hérite de la classe Activity.*

2. Quelle est la différence entre une activité et une application ?

*Une application peut contenir plusieurs activités.*

3. Votre application va avoir le comportement suivant :

- L'utilisateur saisit deux entiers dans les champs d'édition `val1` et `val2`

- Il choisit l'opération à faire : *somme* ou *différence* en cliquant sur le bouton radio associé

- Il clique sur le bouton *calculer*. Le résultat est affiché dans le champs de texte *res*.

a. donner la ligne de code nécessaire pour associer l'interface définie avec l'activité `CalculActivity`.

*`setContentView(R.layout.activity_main_calcul)`*

b. Vous avez créé une méthode dans le code pour définir le comportement de votre bouton. Donner la signature de cette méthode .

*`public void onClick(View v)`*

c. vous avez créé deux attributs représentant les deux boutons radios :

`private RadioButton b1;`

`private RadioButton b2;`

Vous voulez que `b1` soit associé au bouton radio *somme* et `b2` soit associé à *diff*.

- Dans quelle méthode proposez-vous de faire ces initialisations ? pourquoi ?

***Dans la méthode onCreate pour qu'on puisse l'utiliser directement une fois lancer l'application.***

- Donner les lignes de code nécessaires pour faire cela.

*`b1 = (RadioButton) findViewById(R.id.somme) ;`  
`b2 = (RadioButton) findViewById(R.id.diff) ;`*