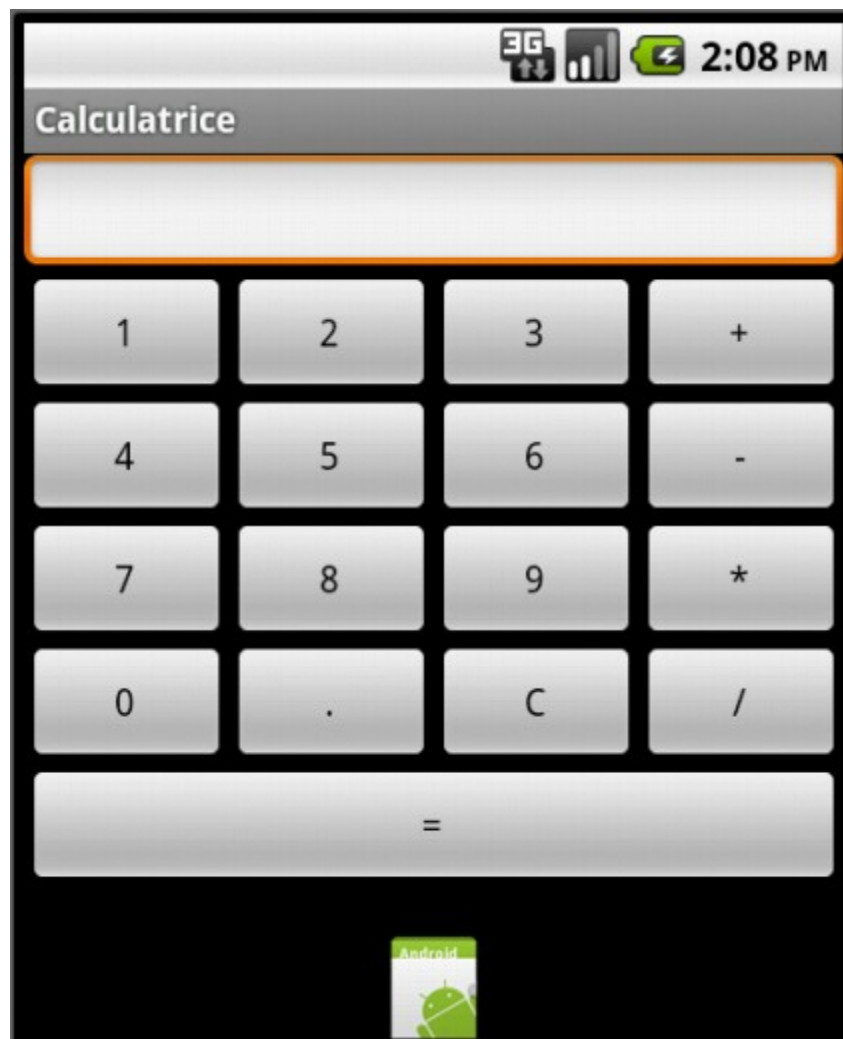


Projet Android

Exercice 1

On souhaite développer une application Android « calculatrice » qui permet à l'utilisateur d'effectuer les opérations arithmétiques de base (+, -, x, /).

Voici l'aperçu que vous devez avoir en final :



Etape 1 : Création de l'interface graphique via le fichier main.xml

- 1) créer un **EditText** qui aura les propriétés suivantes :

- identifiant : EditText01
- il doit s'afficher sur la largeur de l'écran (donc sur la largeur de l'activité)
- son hauteur doit s'ajuster à son contenu
- taille du texte : 20px
- non éditable (l'utilisateur entre du texte avec le clavier virtuel et non directement dans l'EditText)
- le curseur doit être invisible

2) créer la première rangée de boutons.

Les boutons de la 1^{ère} rangée auront les identifiants suivants : button1, button2, button3, buttonPlus.

N.B : Tous les boutons auront les attributs suivants :

```
android:text (qui contient le texte qui s'affiche au dessus du bouton)
android:layout_width
android:layout_height
android:layout_weight (poids du bouton, fixé à 1)
```

3) Créer la deuxième rangée de boutons.

Les boutons de la 2^{ème} rangée auront les identifiants suivants : button4, button5, button6, buttonMoins.

4) Faire pareil pour la rangée 3 et 4.

Les boutons de la 3^{ème} rangée auront les identifiants suivants : button7, button8, button9, buttonMultiplier.

Les boutons de la 4^{ème} rangée auront les identifiants suivants : button0, buttonPoint, buttonC, buttonDivision.

5) Créer un bouton = avec l'identifiant « buttonEgal » sur la largeur de l'écran et insérer vers la fin une image avec l'identifiant « logo » (icon.png par exemple ou une autre image png sur votre disque dur). N'oubliez pas que votre image doit être placée sous le dossier drawable.

Etape 2 : Ecrire les traitements relatifs aux actions de l'utilisateur sur les composants de l'interface graphique

Compléter le code de l'activité, sachant que :

- la méthode **public void plusClick()** est exécutée lorsqu'on clique sur le bouton +
- la méthode **public void moinsClick()** est exécutée lorsqu'on clique sur le bouton -
- la méthode **public void mulClick()** est exécutée lorsqu'on clique sur le bouton *
- la méthode **public void divClick()** est exécutée lorsqu'on clique sur le bouton /
- la méthode **public void egalClick()** est exécutée lorsqu'on clique sur le bouton =
- la méthode **public void resetClick()** est exécutée lorsque l'on clique sur le bouton C
- la méthode **private void calcul()** fait le calcul qui a été demandé par l'utilisateur

Le code de ces méthodes a été donné, on vous demande seulement de programmer les événements.

```
package tp.android.calculatrice;

import android.app.Activity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;

public class Tutoriel4_Android extends Activity {

    //On déclare toutes les variables dont on aura besoin
    Button button0;
    Button button1;
    Button button2;
    Button button3;
    Button button4;
```

```
Button button5;
Button button6;
Button button7;
Button button8;
Button button9;
Button buttonPlus;
Button buttonMoins;
Button buttonDiv;
Button buttonMul;
Button buttonC;
Button buttonEgal;
Button buttonPoint;
EditText resultat;

//chiffre contient le résultat temporaire avant clic sur =
private double chiffre;

// clicOperateur indique si on a déjà cliqué sur au moins un opérateur ou // pas. Si
false, c'est qu'on n'a encore cliqué sur rien
private boolean clicOperateur = false;

//update permet de contrôler la saisie, prend true si on saisit un //opérateur, reste à
false tant qu'on saisit un nombre (qui peut comporter //plus qu'un chiffre ou un «. »(3.5
par exemple)
private boolean update = false;

//contient le dernier opérateur sur lequel nous avons cliqué
private String operateur = "";

/** Called when the activity is first created. */
@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.main);
```

//récupérez ici tous les éléments de votre interface graphique grâce aux ID

```
button0 = (Button) findViewById(R.id.button0);
```

//complétez ici

...

...

//attribuez un écouteur d'évènement à tous les boutons

```
button0.setOnClickListener(new View.OnClickListener() {  
    public void onClick(View v) {  
        chiffreClick("0");  
    }  
});
```

//complétez ici

...

...

```
}
```

```
public void chiffreClick(String str) {
```

```
    String s=str;
```

```
    if(update){
```

```
        //on est en train de saisir un chiffre après avoir saisi un opérateur
```

```
        update = false;
```

```
    }
```

```
    else
```

```
    {
```

```
        //on est en train de saisir un nombre qui comporte plus qu'un chiffre (12 //par exemple)
```

```
        if(!resultat.getText().equals("0"))
```

```
            s = resultat.getText() + str;
```

```
    }
```

```
        resultat.setText(s);
    }

    public void plusClick() {

        if(clicOperateur){
//on a déjà cliqué sur d'autres opérateurs auparavant, on évalue alors ce //qui précède
            calcul();
            resultat.setText(String.valueOf(chiffre));
        }
        else
        {
            //c'est le premier clic
            chiffre = Double.valueOf(resultat.getText().toString()).doubleValue();
            clicOperateur = true;
        }
        operateur = "+";
        update = true;
    }

    public void moinsClick() {
        if(clicOperateur){
            calcul();
            resultat.setText(String.valueOf(chiffre));
        }
        else
        {
            chiffre = Double.valueOf(resultat.getText().toString()).doubleValue();
            clicOperateur = true;
        }
        operateur = "-";
        update = true;
    }
}
```

```
public void mulClick() {
    if(clicOperateur){
        calcul();
        resultat.setText(String.valueOf(chiffre));
    }
    else
    {
        chiffre = Double.valueOf(resultat.getText().toString()).doubleValue();
        clicOperateur = true;
    }
    operateur = "*";
    update = true;
}

public void divClick() {
    if(clicOperateur){
        calcul();
        resultat.setText(String.valueOf(chiffre));
    }
    else
    {
        chiffre = Double.valueOf(resultat.getText().toString()).doubleValue();
        clicOperateur = true;
    }
    operateur = "/";
    update = true;
}

public void egalClick() {
    calcul();
    update = true;
    clicOperateur = false;
}
```

```
}

public void resetClick() {
    clicOperateur = false;
    update = true;
    chiffre = 0;
    operateur = "";
    resultat.setText("");
}

private void calcul() {
    if(operateur.equals("+")){
        chiffre = chiffre + Double.valueOf(resultat.getText().toString()).doubleValue();
        resultat.setText(String.valueOf(chiffre));
    }

    if(operateur.equals("-")){
        chiffre = chiffre - Double.valueOf(resultat.getText().toString()).doubleValue();
        resultat.setText(String.valueOf(chiffre));
    }

    if(operateur.equals("*")){
        chiffre = chiffre * Double.valueOf(resultat.getText().toString()).doubleValue();
        resultat.setText(String.valueOf(chiffre));
    }

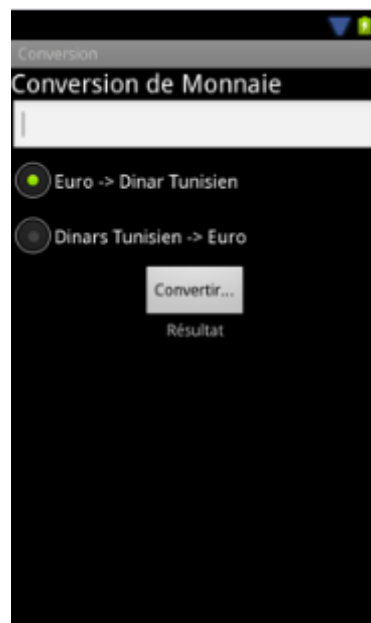
    if(operateur.equals("/")){
        try{
            chiffre = chiffre / Double.valueOf(resultat.getText().toString()).doubleValue();
            resultat.setText(String.valueOf(chiffre));
        } catch(ArithmeticException e) {
            resultat.setText("0");
        }
    }
}
```



```
}  
}
```

Exercice 2

1. Créer un nouveau projet intitulé *Conversion*. Créer les éléments nécessaires pour que l'interface ressemble à ce qui suit :



La zone de saisie ne doit accepter que les nombres décimaux (avec virgule, soit par exemple 15.6). Identifiez l'attribut XML qui permet de répondre à cette contrainte.

- 2- Créer une méthode appelée *convertir* qui définit le comportement du bouton "Convertir" (indication : utiliser l'attribut android:OnClick)

- 3- Implémenter la méthode *convertir* pour qu'elle fasse la conversion nécessaire, selon le bouton radio qui est coché et met le résultat dans le champs de texte

Resultat.

Si le bouton radio "Euro -> Dinar Tunisien" est coché, on appelle une méthode *euroToDinar* qui effectue la conversion de la valeur en entrée sinon on appelle une autre méthode *dinarToeuro*.

Sachant que :

1 euro = 2.5 DT

Créer ces deux méthodes.

4- Rajouter en bas de l'interface, un TextView qui affiche le message suivant "Cours de change des devises à la date du *notre_date_du_jour* : 1 euro=2.293DT"

5- Créer une alerte appelée si l'utilisateur clique sur le bouton *conversion* alors que le champs en entrée est vide.