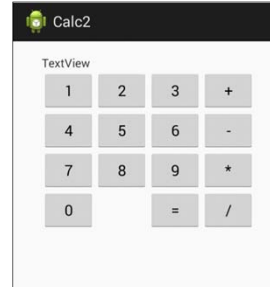


Programmation sur Smartphone

TP2

- Créer une calculatrice avec les touches



TP2

- Pour manipuler les deux nombres
 - Utiliser deux chaînes de caractères
 - L'appui des boutons pour les chiffres permet d'ajouter des chiffres au premier nombre puis au deuxième nombre
- 10 boutons pour les chiffres :
 - Utiliser un seul callback qui
 - Lit le titre du bouton
 - Ajoute cette valeur à la fin de la chaîne de caractères du nombre.

TP2

- Lire le titre du bouton dans la méthode servant de callback pour le bouton :

```
public void calculate(View view)
{
    Button but=(Button)view;
    String operator=but.getText().toString();
    if (operator.equals("+"))
    {
        ...
    }
}
```

Trois façons d'ajouter un callback aux boutons

- Méthode 1 :
 - Cliquer avec le bouton droit de la souris sur le bouton dans l'éditeur de ressources
 - Other Properties -> Inherited from View -> On Click...
 - Entrer le nom de la méthode pour le callback
 - Ajouter la méthode dans le code java
- Méthode 2 (classe anonyme) :
 - Ajouter un callback au bouton "b1" avec le code :


```
b1.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v)
    {
        // Réagir au clic
    }
});
```

Trois façons d'ajouter un callback aux boutons

- Méthode 3 :
 - Créer une classe pour le callback


```
private OnClickListener clickListenerBoutons = new
View.OnClickListener()
{
    @Override
    public void onClick(View v)
    {
        /* Réagir au clic pour les boutons 1 et 2 */
    }
};
```
 - Ajouter la classe pour le callback du bouton "b1"


```
b1.setOnClickListener(clickListenerBoutons);
```

TP2

- Réécrire le programme de la calculatrice avec les méthodes "setOnClickListener".

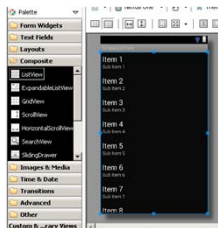
Manipulation des listes

- Une liste permet d'afficher un ensemble d'éléments :



Manipulation des listes

- Dans le fichier ressource du layout qui affichera la liste :
 - Insérer le widget ListView



Manipulation des listes

- Dans les ressources :
 - Insérer le widget ListView dans le layout qui affichera la liste.
 - Son ID : list
 - Créer un nouveau layout pour l'affichage des éléments de la liste :
 - Créer un fichier dans le dossier res/layout
 - Nom en minuscules : listitem.xml
 - Extension : xml
 - Supprimer le layout créé par défaut
 - Lui ajouter le widget TextView

Manipulation des listes

- Dans la classe du layout affichant la liste :
 - Créer deux variables :


```
private ListView mainListView ;
private ArrayAdapter<String> listAdapter ;
```
 - Dans la méthode


```
void onCreate(Bundle savedInstanceState)
```

 - Ajouter le code :


```
mainListView = (ListView) findViewById( R.id.list );
listAdapter = new ArrayAdapter<String>(this, R.layout.listitem);
mainListView.setAdapter( listAdapter );
```

Manipulation des listes

- Dans la classe du layout affichant la liste :
 - Pour ajouter un élément dans la liste


```
listAdapter.add( "Exemple" );
```
 - Pour supprimer un élément dans la liste


```
listAdapter.remove( "Exemple" );
```
 - Pour insérer un élément dans la liste


```
int position=0;
listAdapter.insert( "Exemple", position );
```
 - Pour modifier un élément dans la liste


```
Faire une suppression suivie d'une insertion
```
 - Pour lire un élément dans la liste


```
int position=0;
String s=listAdapter.getItem(position);
```

Manipulation des listes

- Dans la classe du layout affichant la liste :
 - Pour ajouter un callback pour le click d'un élément de la liste :

```
mainListView.setOnItemClickListener(new
AdapterView.OnItemClickListener()
{
    @Override
    public void onItemClick(AdapterView<?> adapterView, View view, int
                                position, long id)
    {
        // Insérer le code ici
    }
});
```

TP3

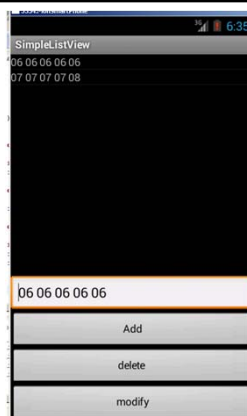
- Carnet de numeros de telephone
 - Une liste pour afficher les numéros
 - Un champ texte éditable
 - Trois boutons :
 - Ajouter
 - Supprimer
 - Modifier

TP3

- Ajout d'un numéro :
 - L'utilisateur tape le numéro dans le champ texte
 - Il clique le bouton "ajouter"
- Modification d'un numéro :
 - L'utilisateur clique sur l'élément de la liste à modifier
 - L'application copie le texte de l'élément dans le champ texte
 - L'utilisateur modifie le numéro dans le champ texte.
 - Il clique sur le bouton "modifier"

TP3

- Suppression d'un numéro :
 - L'utilisateur clique sur l'élément de la liste à modifier
 - L'application copie le texte de l'élément dans le champ texte
 - L'utilisateur clique sur le bouton "supprimer"



TP3

- Facultatif : Ajouter une icône à côté de chaque numéro de la liste
 - Créer une classe dérivée de la classe `ArrayAdapter<String>`
 - Écrire la méthode « `public View getView(int position, View convertView, ViewGroup parent)` » qui affiche un élément de la liste.

Utiliser l'exemple:

<http://www.mkyong.com/android/android-listview-example/>