

# TP 2 - exercice 5

---

- ✓ Comment réaliser graphiquement les 10 opérations de la table de multiplication ?

## 4 solutions !

1. Tout décrire dans le XML (10 LinearLayout)
2. Faire une itération dans l'activité et création Java
3. Faire une itération dans l'activité et ***inflate***\* XML
4. Utiliser une layout spécifique associé à un objet de type Adapter

\*gonfler

*Sur Chamilo, le projet android generationList reprend les exemples présentés dans ce document*

# 1. Tout décrire dans le XML

---

- ✓ Solution correcte
- ✓ MAIS fastidieuse
- ✓ Beaucoup de code dans le XML
- ✓ Beaucoup de code dans l'activité
- ✓ Possibilité de BUG plus grande
- ✓ Maintenabilité faible

## 2. Itération et création Java

```
// 1. Boucler pour générer la table  
for (Multiplication multiplication : tableMultiplication.getMultiplications()) {
```

```
// 2. Création de la ligne temporaire
```

```
LinearLayout linearTMP = new LinearLayout( context: this);  
linearTMP.setOrientation(LinearLayout.HORIZONTAL);
```

*Création d'objets graphique  
Comme en Swing*

```
// 3. Création du texte décrivant l'opération
```

```
TextView calcul = new TextView( context: this);  
calcul.setText(multiplication.getOperande1()  
+ " x " + multiplication.getOperande2() + " = ");
```

```
// 4. Création de l'editText permettant d'interagir avec l'utilisateur
```

```
EditText resultat = new EditText( context: this);  
resultat.setInputType(InputType.TYPE_CLASS_NUMBER);  
resultat.setHint("?");
```

```
// 5. Ajout à la ligne
```

```
linearTMP.addView(calcul);  
linearTMP.addView(resultat);
```

Même description graphique  
qu'en XML mais syntaxe Java

```
// 6. Ajout au linear principal
```

```
linear.addView(linearTMP);
```

```
}
```

### 3. Itération et inflate XML

On fait une petite vue, faite pour afficher une ligne.

```
// 1. Boucler pour générer la table
for (Multiplication multiplication : tableMultiplication.getMultiplications()) {

    // 2. Création de la ligne temporaire
    LinearLayout linearTMP = (LinearLayout) getLayoutInflater().inflate(R.layout.template_calcul, root: null);

    // 3. Création du texte décrivant l'opération
    TextView calcul = (TextView) linearTMP.findViewById(R.id.template_calcul);
    calcul.setText(multiplication.getOperande1() + " x " + multiplication.getOperande2() + " = ");

    // 4. Création de l'editText permettant d'interagir avec l'utilisateur
    EditText resultat = (EditText) linearTMP.findViewById(R.id.template_resultat);
    resultat.setText(Integer.toString( multiplication.getOperande1() * multiplication.getOperande2()));

    // 5. Ajout au linear principal
    linear.addView(linearTMP);
}
```

**Comme le chargement  
du layout principal**

Par rapport à la solution 2, La description graphique se fait sous forme XML (séparation entre le traitement et l'interface)

template\_calcul.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="horizontal">

    <TextView
        android:id="@+id/template_calcul"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />

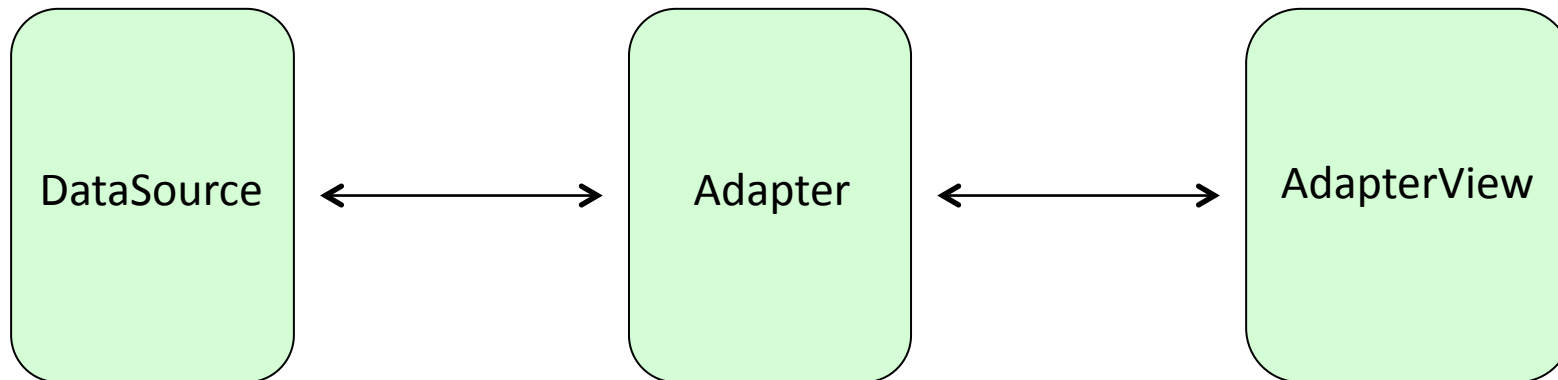
    <EditText
        android:id="@+id/template_resultat"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:inputType="number"
        android:hint="\?" />

</LinearLayout>
```

## 4. Layout spécifique et son adapter

### ✓ Layout spécifique : ListView, RecyclerView.

- <https://developer.android.com/reference/android/widget/ListView>
  - <https://developer.android.com/guide/topics/ui/layout/recyclerview>
  - RecyclerView plus avancée et plus flexible que ListView mais aussi plus difficile à utiliser.
- RecyclerView est bien pour l'interaction



*Par rapport à l'exemple dans generationList :*

List<Multiplication>

ArrayAdapter

(Inflate XML comme dans la solution 3)

ListView

*L'adapter sert de pivot entre les données et l'élément graphique.*

# IMPORTANT

---

- ✓ Dans tous les cas pensez au **MODÈLE** !
  - Classe TableDeMultiplication
  - Classe Multiplication
  
- ✓ Comment récupérer les résultats ?
  - Vous pouvez conserver les objets de type `EditText` créés dans une liste de l'activité.

# Conseil pour l'exercice 5 du TP2

---

✓ Utilisez plutôt la solution 3 : itération et inflate XML

■ Solution intéressante

- ◆ La description graphique se fait sous forme XML (séparation entre le traitement et l'interface)
- ◆ Pour comprendre les layout spécifiques (solution 4)
- ◆ Pour comprendre l'inflate de l'activité avec son XML associé

✓ Vous utiliserez la solution 4 dans le projet à l'étape 3 : gestion de compte

- Solution plus élégante et utilisant les mécanismes android