

HAI811I - Programmation Mobile

TD N°1 :

Manifest, Activité, Vue, Layout, Ressource, ...

Exercice 1 : Les différentes parties d'une application Android

- 1) Observez le code suivant d'une application Android.
- 2) Ajoutez le code nécessaire pour la saisie d'un texte.

```
01 <?xml version="1.0" encoding="utf-8"?>
02 <manifest xmlns:android="http://schemas.android.com/apk/res/android"
03     package="com.javacodegeeks.examples.androidViewPagerExample" >
04
05     <application
06         android:allowBackup="true"
07         android:icon="@mipmap/ic_launcher"
08         android:label="@string/app_name"
09         android:theme="@style/AppTheme" >
10         <activity
11             android:name=".AndroidViewPagerExample"
12             android:label="@string/app_name" >
13             <intent-filter>
14                 <action android:name="android.intent.action.MAIN" />
15
16                 <category android:name="android.intent.category.LAUNCHER" />
17             </intent-filter>
18         </activity>
19     </application>
20
21 </manifest>
22 >
```

Fichier Java : la logique métier

```
import android.widget.TextView;
public class HelloAndroid extends Activity {
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        TextView tv = new TextView(this);
        tv.setText("Hello, Android");
        setContentView(tv);
    }
}
```

- 3) Observez maintenant le code suivant de cette même application Android et Indiquez les différences de code et la signification de cette différence par rapport au code précédent.
- 4) Ajoutez le code nécessaire pour la saisie d'un texte.

Fichier : res/layout/layout_file_name.xml → La vue

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical" >
<TextView android:id="@+id/text"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Hello, I am a TextView" />
</LinearLayout>
```

Fichier Java : la logique métier

```
package com.android.helloandroid;
import android.app.Activity;
import android.os.Bundle;
import android.widget.TextView;

public class HelloAndroid extends Activity {
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.layout_file_name);
    }
}
```

Exercice 2 : Les ressources

- 1) Observez le code de l'application suivante et indiquez la différence structurelle avec le code des applications des exercices précédents.
- 2) Ajoutez le code nécessaire pour afficher le texte « Ma première application Android » et pour saisir un texte.

Fichier : res/strings/string.xml :

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
<string name="app_name">AndroidToastExample</string>
<string name="menu_settings">Settings</string>
<string name="button_label">Show message</string>
<string name="image_content">image</string>
</resources>
```

Fichier : res/layout/layout_file_name.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical" >
<Button
    android:id="@+id/mainbutton"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/button_label" />
</LinearLayout>
```

Fichier Java : La logique métier

```
package com.android.helloandroid;
import android.app.Activity;
import android.os.Bundle;
import android.widget.TextView;

public class HelloAndroid extends Activity {

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.layout_file_name);
    }
}
```

Exercice 3 : Les layouts (Gabarits)

- 1) Observez les différents codes de l'application suivante et indiquez l'affichage correspondant.

Layout1:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="horizontal" >
<Button
    android:id="@+id/button1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Button 1" />
<Button
    android:id="@+id/button2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Button 2" />
<Button
    android:id="@+id/button3"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
```

```
        android:text="Button 3"
        android:layout_weight="1"/>
</LinearLayout>
```

Layout2:

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent" >
    <Button
        android:id="@+id/button1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Button 1"/>
    <TextView
        android:id="@+id/textView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/button3"
        android:layout_marginTop="65dp"
        android:text="Username :"
        android:textAppearance="?android:attr/textAppearanceLarge" />
    <EditText
        android:id="@+id/editText"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentRight="true"
        android:layout_alignTop="@+id/textView"
        android:layout_toRightOf="@+id/button2"
        android:inputType="text" />
    <Button
        android:id="@+id/button0"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentRight="true"
        android:layout_below="@+id/editText"
        android:text="Submit" />
    <Button
        android:id="@+id/button3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentLeft="true"
        android:layout_below="@+id/button1"
        android:text="Button 3" />
    <Button
        android:id="@+id/button2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/button1"
        android:layout_toRightOf="@+id/button1"
        android:text="Button 2" />
```

```
</RelativeLayout>
```

Layout3:

```
<?xml version="1.0" encoding="utf-8"?>
<TableLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/tableLayout"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:shrinkColumns="*"
    android:stretchColumns="*" >

    <!-- 2 columns -->

    <TableRow
        android:id="@+id/tableRow1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:padding="5dip" >

        <TextView
            android:id="@+id/textView1"
            android:text="Col 1"
            android:textAppearance="?android:attr/textAppearanceLarge" />

        <Button
            android:id="@+id/button1"
            android:text="Col 2" />

    </TableRow>

    <TableRow
        android:id="@+id/tableRow2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:padding="5dip" >

        <EditText
            android:id="@+id/editText1"
            android:layout_span="2"
            android:text="Col 1 & 2" />

    </TableRow>

    <!-- red line -->
    <View
        android:layout_height="4dip"
        android:background="#FF00" />

    <!-- 4 columns -->
    <TableRow
```

```

        android:id="@+id/tableRow3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:padding="5dip" >
<TextView
    android:id="@+id/textView2"
    android:text="Col 1" />
<Button
    android:id="@+id/button2"
    android:text="Col 2" />
<Button
    android:id="@+id/button3" 6
    android:text="Col 3" />
<Button
    android:id="@+id/button5"
    android:text="Col 4" />
</TableRow>

<TableRow
    android:id="@+id/tableRow4"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:padding="5dip" >
<Button
    android:id="@+id/button4"
    android:layout_column="2"
    android:text="Col 3" />
</TableRow>

<TableRow
    android:id="@+id/tableRow5"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:padding="5dip" >
<Button
    android:id="@+id/button6"
    android:layout_column="1"
    android:text="Col 2" />
</TableRow>
</TableLayout>

```

Exercice 4 : Evènements utilisateurs

1) Observez le code ci-dessous et dire ce qu'il réalise.

- 2) Ajouter un deuxième bouton qui affiche quand il est actionné un message Toast.
- 3) Modifiez ce code pour permettre d'afficher le message « Exercice 4 » sur le même écran (sans sa disparition).
- 4) Les messages affichés sont en français, comment faire pour pouvoir s'adapter à la langue de l'utilisateur ?
- 5) Modifier le code d'un des boutons pour qu'il réagisse après un appui long.

res/layout/main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical" >
    <Button
        android:id="@+id/mainbutton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/mainbutton" />
</LinearLayout>
```

```
import android.app.Activity;
import android.os.Bundle;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.Toast;

public class MainActivity extends Activity {
    private Button button;
    public void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
        button = (Button) findViewById(R.id.mainbutton);
        button.setOnClickListener(new OnClickListener() {
            @Override
            public void onClick(View arg0) {
                Toast.makeText(getApplicationContext(),"Message Bouton 1", Toast.LENGTH_LONG).show();
            }
        });
    }
}
```

Exercice 5 : Différentes vues et actions

- 1) Observez le code ci-dessous et indiquez ce qu'il affiche.

```
<?xml version="1.0" encoding="utf-8"?>
```

```

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical" >
<CheckBox
    android:id="@+id/linux_option"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/linux_box" />
<CheckBox
    android:id="@+id/macos_option"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/macos_box"
    android:checked="true" />
<CheckBox
    android:id="@+id/windows_option"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/windows_box" />
<Button
    android:id="@+id/button"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/display_label" />
</LinearLayout>

```

```

import ...

public class MainActivity extends Activity {
    private CheckBox linux, macos, windows;
    private Button button;
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
        addListenerOnChkWindows();
        addListenerOnButton();
    }
    public void addListenerOnChkWindows() {
        windows = (CheckBox) findViewById(R.id.windows_option);
        windows.setOnClickListener(new OnClickListener() {
            @Override
            public void onClick(View v) {
                if (((CheckBox) v).isChecked()) {
                    Toast.makeText(MainActivity.this, "Bro, try Linux :)", Toast.LENGTH_LONG).show();
                }
            }
        });
    }
}

```



```
public void addListenerOnButton() {  
    linux = (CheckBox) findViewById(R.id.linux_option);  
    macos = (CheckBox) findViewById(R.id.macos_option);  
    windows = (CheckBox) findViewById(R.id.windows_option);  
    button = (Button) findViewById(R.id.button);  
    button.setOnClickListener(new OnClickListener() {  
        @Override  
        public void onClick(View v) {  
            StringBuffer result = new StringBuffer();  
            result.append("Linux check : ").append(linux.isChecked());  
            result.append("\nMac OS check : ").append(macos.isChecked());  
            result.append("\nWindows check : ").append(windows.isChecked());  
            Toast.makeText(MainActivity.this, result.toString(), Toast.LENGTH_LONG).show();  
        }  
    });  
}
```

Exercice 6 : Méta-Modèle Application Android

- 1) Réalisez un méta-modèle d'une application Android