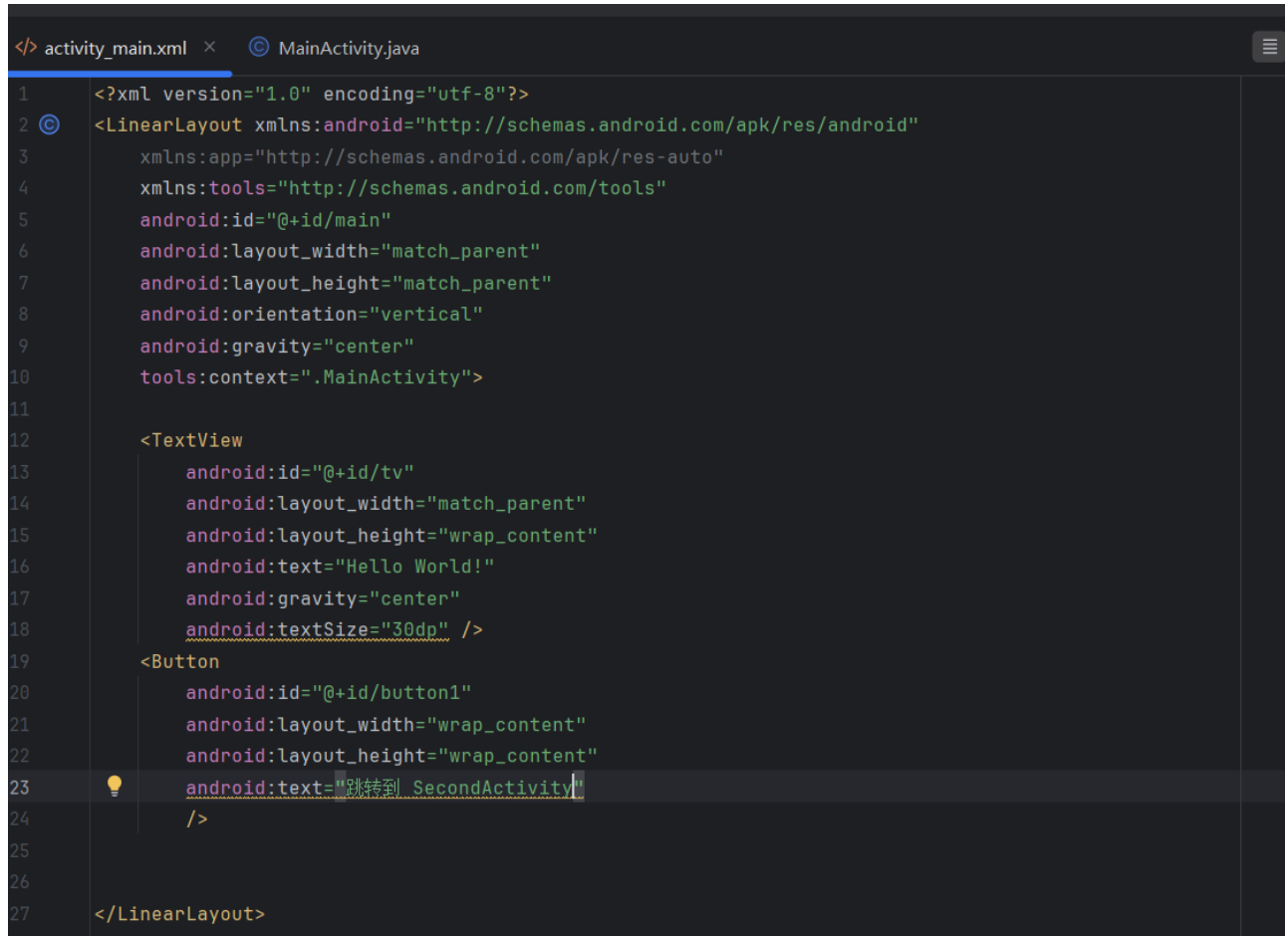


# 课程作业1

## 第一部分：显式跳转

1. 在 `MainActivity` 的布局文件中，添加一个名为 "跳转到 `SecondActivity`" 的按钮。



```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:gravity="center"
    tools:context=".MainActivity">

    <TextView
        android:id="@+id/tv"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Hello World!"
        android:gravity="center"
        android:textSize="30dp" />

    <Button
        android:id="@+id/button1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="跳转到 SecondActivity" />

</LinearLayout>
```

2. 在 `MainActivity` 的代码中，为该按钮设置点击事件监听器。
3. 在点击事件中，使用**显式Intent** (`Intent(Context packageContext, Class<?> cls)`) 启动 `SecondActivity`。

```

public class MainActivity extends AppCompatActivity {
    2 usages
    Button b1;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        EdgeToEdge.enable( $this$enableEdgeToEdge: this);
        setContentView(R.layout.activity_main);
        ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main), ( View v, WindowInsetsCompat insets) -> {
            Insets systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars());
            v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom);
            return insets;
        });
        b1=findViewById(R.id.button1);
        b1.setOnClickListener(new MyListener());
    }
}

1 usage
class MyListener implements View.OnClickListener{
    @Override
    public void onClick(View v){
        if(v.getId()==R.id.button1){
            Intent it =new Intent( packageContext: MainActivity.this,MainActivity2.class);
        }
    }
}
}

```

4. 在 `SecondActivity` 的布局文件中，添加一个名为“返回到主页”的按钮。

```

<Button
    android:id="@+id/bac"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="返回到主页"
/>

```

5. 在 `SecondActivity` 的代码中，为该按钮设置点击事件，并使用 `finish()` 方法结束当前Activity，返回到 `MainActivity`。

```

Button bac,vis;
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    EdgeToEdge.enable( $this$enableEdgeToEdge: this);
    setContentView(R.layout.activity_main2);
    ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main), ( View v, WindowInsetsCompat insets) -> {
        Insets systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars());
        v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom);
        return insets;
    });
    bac = findViewById(R.id.bac);
    bac.setOnClickListener(new View.OnClickListener(){
        @Override
        public void onClick(View view){
            finish();
        }
    });
}

```

## 第二部分：隐式跳转

1. 在 `SecondActivity` 的布局文件中，添加一个名为 "隐式跳转" 的按钮。

```
<Button
    android:id="@+id/vis"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="隐式跳转"
/>
```

2. 在 `SecondActivity` 的代码中，为该按钮设置点击事件监听器。
3. 在点击事件中，创建一个隐式Intent。
  - **Action:** 自定义一个字符串常量，例如 `"com.example.action.VIEW_THIRD_ACTIVITY"`。
  - **Category:** `Intent.CATEGORY_DEFAULT`。

```
vis = findViewById(R.id.vis);
vis.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        // 创建隐式 Intent
        Intent intent = new Intent();
        intent.setAction("com.example.action.VIEW_THIRD_ACTIVITY");
        intent.addCategory(Intent.CATEGORY_DEFAULT);
        startActivityForResult(intent, requestCode: 2);
    }
});

@Override
protected void onActivityResult(int requestCode, int resultCode, @Nullable Intent data) {
    super.onActivityResult(requestCode, resultCode, data);
    if(resultCode == Activity.RESULT_OK){
        if(requestCode==2){
            String result = data.getStringExtra(name: "result");
            Intent resultIntent = new Intent();
            resultIntent.putExtra(name: "result", result);
            setResult(Activity.RESULT_OK, resultIntent);
            finish();
        }
    }
}
```

4. 在 `AndroidManifest.xml` 文件中，为 `ThirdActivity` 的 `<activity>` 标签添加一个 `<intent-filter>`。
  - `action` 标签的 `android:name` 属性设置为你在上一步中定义的 Action。
  - `category` 标签的 `android:name` 属性设置为 `android.intent.category.DEFAULT`。

```
<activity
    android:name=".MainActivity3"
    android:exported="true" >
    <intent-filter>
        <action android:name="com.example.action.VIEW_THIRD_ACTIVITY" />
        <category android:name="android.intent.category.DEFAULT" />
    </intent-filter>
</activity>
```

5. 运行应用，点击"隐式跳转"按钮，确认可以成功启动 `ThirdActivity`。

### 第三部分：带返回结果的跳转

1. 在 `MainActivity` 的布局文件中，添加一个名为 "启动带结果的跳转" 的按钮，以及一个用于显示返回结果的 `TextView`。

```
<TextView
    android:id="@+id/tv"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Hello World!"
    android:gravity="center"
    android:textSize="30dp" />

<Button
    android:id="@+id/button1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="跳转到 SecondActivity"
/>

<Button
    android:id="@+id/outcome"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="启动带结果的跳转"
/>
```

2. 在 `MainActivity` 的代码中，为该按钮设置点击事件监听器。
3. 在点击事件中，使用 `startActivityForResult()` 方法启动 `ThirdActivity`。

- 为这个跳转设置一个唯一的请求码 (`requestCode`)，例如 `101`。

```
2 usages
class MyListener implements View.OnClickListener{
    @Override
    public void onClick(View v){
        if(v.getId()==R.id.button1){
            Intent it =new Intent( packageContext: MainActivity.this,MainActivity2.class);
            startActivityForResult(it, requestCode: 2);
        }
        else if(v.getId()==R.id.outcome){
            Intent it =new Intent( packageContext: MainActivity.this,MainActivity3.class);
            startActivityForResult(it, requestCode: 1);
        }
    }
}

@Override
protected void onActivityResult(int requestCode, int resultCode, @Nullable Intent data) {
    super.onActivityResult(requestCode, resultCode, data);
    if(resultCode == Activity.RESULT_OK){
        String resultstring = data.getStringExtra( name: "result");
        tv.setText(resultstring);
    }
}
```

4. 在 `ThirdActivity` 的布局文件中，添加一个 `EditText` 和一个名为 "返回结果" 的按钮。

```
<EditText
    android:id="@+id/insert"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:textSize="30dp"/>

<Button
    android:id="@+id/back"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="返回结果"
    android:textSize="30dp" />
```

5. 在 `ThirdActivity` 的代码中，为\*\*"返回结果"按钮设置点击事件。
6. 在点击事件中，创建一个新的 `Intent` 对象，并将 `EditText` 中的文本作为额外数据 (Extra) 放入 `Intent` 中。
  - 例如: `intent.putExtra("result_data", "你输入的文本")`。
7. 使用 `setResult(int resultCode, Intent data)` 方法设置返回结果。
  - `resultCode` 设置为 `Activity.RESULT_OK`。

8. 调用 `finish()` 方法结束 `ThirdActivity`。

```

MainActivity.java  </> activity_main.xml  </> activity_main2.xml  </> activity_main3.xml  MainActivity3.java x
public class MainActivity3 extends AppCompatActivity {
    EditText et;
    2 usages
    Button back;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        EdgeToEdge.enable( $this$enableEdgeToEdge: this);
        setContentView(R.layout.activity_main3);
        ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main), ( View v, WindowInsetsCompat insets) -> {
            Insets systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars());
            v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom);
            return insets;
        });
        et = findViewById(R.id.insert);
        back =findViewById(R.id.back);
        back.setOnClickListener(new View.OnClickListener(){
            @Override
            public void onClick(View view){
                Intent it =new Intent();
                it.putExtra( name: "result",et.getText().toString());
                setResult(Activity.RESULT_OK,it);
                finish();
            }
        });
    }
}

```

9. 回到 `MainActivity`，重写 `onActivityResult(int requestCode, int resultCode, Intent data)` 方法。

10. 在 `onActivityResult` 方法中，根据 `requestCode` 和 `resultCode` 判断是否为预期的返回结果。如果是，从 `data Intent` 中获取返回的文本数据，并更新 `MainActivity` 的 `TextView`。

```

@Override
protected void onActivityResult(int requestCode, int resultCode, @Nullable Intent data) {
    super.onActivityResult(requestCode, resultCode, data);
    if(resultCode == Activity.RESULT_OK){
        String resultstring = data.getStringExtra( name: "result");
        tv.setText(resultstring);
    }
}

```

## 加分项

- 在 `MainActivity` 中，为“启动带结果的跳转”按钮添加一个长按监听器。

```

b2.setOnClickListener(new MyListener());
b2.setOnLongClickListener(new View.OnLongClickListener() {
    @Override
    public boolean onLongClick(View v) {
        //显示Toast提示
        Toast.makeText( context: MainActivity.this, text: "长按启动了带返回结果的跳转!",Toast.LENGTH_SHORT).show();
        //启动MainActivity3
        Intent it=new Intent( packageContext: MainActivity.this,MainActivity3.class);
        startActivityForResult(it, requestCode: 1);
        return true;//表示消费了长按事件
    }
});

```

```
@Override
protected void onActivityResult(int requestCode, int resultCode, @Nullable Intent data) {
    super.onActivityResult(requestCode, resultCode, data);
    if(requestCode == 1){//第三页面到第一页面的数据
        if(resultCode == Activity.RESULT_OK){
            String result = data.getStringExtra("result");
            tv.setText(result);
            Toast.makeText(context: this, text: "成功更新文本", Toast.LENGTH_SHORT).show();
        }
        else if(resultCode == Activity.RESULT_CANCELED){
            Toast.makeText(context: this, text: "用户取消了操作", Toast.LENGTH_SHORT).show();
        }
    }
    else if(resultCode == Activity.RESULT_OK && requestCode == 2){
        //处理第二页面返回的数据
        String resultstring = data.getStringExtra("result");
        tv.setText(resultstring);
    }
}
```

- 长按按钮时，弹出一个 `Toast` 提示用户“长按启动了带返回结果的跳转！”。



- 在 `ThirdActivity` 中，添加一个名为 "返回取消" 的按钮。点击后，使用 `setResult(Activity.RESULT_CANCELED)` 并 `finish()`，在 `MainActivity` 的 `onActivityResult` 中处理这个取消操作。



```
<Button
    android:id="@+id/back_cancel"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="返回取消"
    android:textSize="20sp"
    android:layout_margin="16dp" />
```

```
backcancel = findViewById(R.id.back_cancel);
back.setOnClickListener(new View.OnClickListener(){
    @Override
    public void onClick(View view){
        Intent it =new Intent();
        it.putExtra( name: "result",et.getText().toString());
        setResult(Activity.RESULT_OK,it);
        finish();
    }
});
backcancel.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v){
        setResult(Activity.RESULT_CANCELED);
        finish();
    }
});
```

返回结果

返回取消

