**安卓第一次作业 实验报告**

42312233 徐欣瑶

**一、实验目的**

通过开发一个名为 "Activity Navigator" 的 Android 应用，掌握 Android 开发中 Activity 之间的三种核心跳转方式：显式跳转、隐式跳转和带返回结果的跳转。

**二、功能实现**

**2.1 显式跳转功能实现**

**2.1.1 实现原理**

显式跳转通过直接指定目标 Activity 的类名来实现页面跳转，是最常用的 Activity 启动方式。

**2.1.2 代码实现位置**

布局文件: res/layout/activity\_main.xml 包含跳转按钮

Java 代码: MainActivity.java 按钮点击事件处理

**2.1.3 核心代码说明**

java

// MainActivity.java

btnToSecond.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

// 创建显式 Intent，直接指定 SecondActivity.class

Intent intent = new Intent(MainActivity.this, SecondActivity.class);

startActivity(intent);

}

});

创建了一个显式 Intent，通过 SecondActivity.class 直接指定目标 Activity

**2.1.4 返回功能实现**

java

// SecondActivity.java

btnBackToMain.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

finish(); // 结束当前 Activity，返回上一个 Activity

}

});

使用 finish() 方法结束当前 Activity，系统自动回退到上一个 Activity（MainActivity）

**2.2 隐式跳转功能实现**

**2.2.1 实现原理**

隐式跳转不直接指定目标 Activity，而是通过 Action 和 Category 等条件让系统匹配合适的 Activity。

**2.2.2 核心代码说明**

java

// SecondActivity.java

btnImplicitJump.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

Intent intent = new Intent();

// 设置自定义 Action

intent.setAction("com.example.action.VIEW\_THIRD\_ACTIVITY");

// 添加默认 Category

intent.addCategory(Intent.CATEGORY\_DEFAULT);

if (intent.resolveActivity(getPackageManager()) != null) {

startActivity(intent);

} else {

// 备选方案：使用显式跳转

Intent explicitIntent = new Intent(SecondActivity.this, ThirdActivity.class);

startActivity(explicitIntent);

}

}

});

**2.2.3 Manifest 配置**

xml

<! AndroidManifest.xml

<activity android:name=".ThirdActivity">

<intentfilter>

<action android:name="com.example.action.VIEW\_THIRD\_ACTIVITY" />

<category android:name="android.intent.category.DEFAULT" />

</intentfilter>

</activity>

这段配置告诉系统：ThirdActivity 能够响应 Action 为 com.example.action.VIEW\_THIRD\_ACTIVITY 的隐式 Intent。

**2.3 带返回结果的跳转功能实现**

**2.3.1 实现原理**

通过 startActivityForResult() 启动 Activity，并在目标 Activity 完成后通过 setResult() 返回数据，父 Activity 在 onActivityResult() 中接收处理。

**2.3.2 代码实现位置**

布局文件:

res/layout/activity\_main.xml 启动按钮和结果显示 TextView

res/layout/activity\_third.xml 输入框和返回按钮

Java 代码:

MainActivity.java 启动和结果接收

ThirdActivity.java 数据返回处理

**2.3.3 核心代码说明**

启动带结果的跳转:

java

// MainActivity.java

btnStartForResult.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

Intent intent = new Intent(MainActivity.this, ThirdActivity.class);

startActivityForResult(intent, REQUEST\_CODE\_THIRD);

}

});

返回结果处理:

java

// ThirdActivity.java

btnReturnResult.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

String inputText = etInput.getText().toString().trim();

Intent resultIntent = new Intent();

// 将输入文本作为 Extra 数据放入 Intent

resultIntent.putExtra("result\_data", inputText.isEmpty() ? "默认文本" : inputText);

setResult(RESULT\_OK, resultIntent);

finish();

}

});

接收返回结果:

java

// MainActivity.java

@Override

protected void onActivityResult(int requestCode, int resultCode, Intent data) {

super.onActivityResult(requestCode, resultCode, data);

if (requestCode == REQUEST\_CODE\_THIRD) {

if (resultCode == RESULT\_OK) {

String result = data.getStringExtra("result\_data");

tvResult.setText("返回结果: " + result);

} else if (resultCode == RESULT\_CANCELED) {

tvResult.setText("用户取消了操作");

}

}

}

3.4 加分项功能实现

3.4.1 长按提示功能

java

// MainActivity.java

btnStartForResult.setOnLongClickListener(new View.OnLongClickListener() {

@Override

public boolean onLongClick(View v) {

Toast.makeText(MainActivity.this, "长按启动了带返回结果的跳转！", Toast.LENGTH\_SHORT).show();

return true; // 返回 true 表示消费事件，不触发点击

}

});

3.4.2 取消返回功能

java

// ThirdActivity.java

btnReturnCancel.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

setResult(RESULT\_CANCELED);

finish();

}

});