**Android Activity 跳转课程作业**

**作业目标**

1. 掌握**显式跳转**（Explicit Intent）的用法，能够从一个Activity启动另一个指定的Activity。
2. 掌握**隐式跳转**（Implicit Intent）的用法，能够通过Action和Category启动符合条件的Activity。
3. 掌握**带返回结果的跳转**（Start Activity for Result），能够从子Activity返回数据给父Activity。

**作业内容**

请完成一个名为 **"Activity Navigator"** 的Android应用。该应用包含三个核心Activity：

* **MainActivity**：应用的入口。它将作为所有跳转的起点和终点。
* **SecondActivity**：一个用于演示显式和隐式跳转的中间Activity。
* **ThirdActivity**：一个用于演示带返回结果跳转的子Activity。

图形用户界面, 文本, 应用程序

AI 生成的内容可能不正确。

**作业详细步骤**

**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。

文本

AI 生成的内容可能不正确。

1. 在 MainActivity 的代码中，为该按钮设置点击事件监听器。

图形用户界面, 文本

AI 生成的内容可能不正确。

1. 在点击事件中，使用**显式Intent**（Intent(Context packageContext, Class<?> cls)）启动 SecondActivity。

图形用户界面, 文本

AI 生成的内容可能不正确。

1. 在 SecondActivity 的布局文件中，添加一个名为 **"返回到主页"** 的按钮。

文本

AI 生成的内容可能不正确。

1. 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。

图形用户界面, 文本

AI 生成的内容可能不正确。

**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。

文本

AI 生成的内容可能不正确。

1. 在 SecondActivity 的代码中，为该按钮设置点击事件监听器。

文本

AI 生成的内容可能不正确。

1. 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。

文本

AI 生成的内容可能不正确。

1. 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。

文本

AI 生成的内容可能不正确。

1. 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。

文本

AI 生成的内容可能不正确。

1. 在 MainActivity 的代码中，为该按钮设置点击事件监听器。

文本

AI 生成的内容可能不正确。

1. 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. 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。

文本

AI 生成的内容可能不正确。

1. 在 ThirdActivity 的代码中，为\*\*“返回结果”\*\*按钮设置点击事件

文本

AI 生成的内容可能不正确。

1. 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")。
2. 使用 **setResult(int resultCode, Intent data)** 方法设置返回结果。
   * resultCode 设置为 Activity.RESULT\_OK。
   * 文本

     AI 生成的内容可能不正确。
3. 调用 finish() 方法结束 ThirdActivity。
4. 回到 MainActivity，重写 **onActivityResult(int requestCode, int resultCode, Intent data)** 方法。文本

   AI 生成的内容可能不正确。
5. 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。

**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。
* 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。

**作业提交要求**

* 提交完整的Android Studio项目文件夹。
* 项目代码应清晰、有注释，遵循良好的编码规范。
* 提供一个简短的文档（可以是README文件），说明你如何实现了上述功能，并附上应用的运行截图。