**《移动应用开发》**

**实验报告**

**班级：软件161**

**姓名：胡启明**

**学号：164547**

**实验五 组件通信与广播**

**一、实验目的**

1.使学生学会使用Intent启动Activity的方法。

2．使学生掌握获取Activity返回值的方法。

3. 使学生掌握发送和接收广播消息的方法。

**二、实验内容**

1、使用“IntentDemo”程序来完成如何使用Intent启动新的Activity？

2、使用“ActivityCommunication”说明了如何以Sub-Activity方式启动子Activity，以及使用Intent进行组件间通信？

3、使用“IntentResolutionDemo”说明了如何在AndroidManifest.xml文件中注册Intent过滤器，以及如何设置<intent-filter>节点属性来捕获指定的Intent？

4、使用“BroadcastReceiverDemo”说明了如何在应用程序中注册BroadcastReceiver，并接收指定类型的广播消息？

**三、实验仪器、设备**

硬件：PC 微型计算机、1G以上内存，40G以上硬盘

软件：Windows XP，Eclipse , JDK , Android SDK

**四、实验步骤**

**1. IntentDemo**

1）IntentDemo 页面



2）点击启动ACTIVITY按钮之后跳转到新的页面



3）代码实现如下

IntentDemo.java

**public class** IntentDemo **extends** AppCompatActivity **implements** View.OnClickListener {  
  
 @Override  
 **public void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_main***);  
  
 Button button = findViewById(R.id.***btn***);  
 button.setOnClickListener(**this**);  
 }  
  
 @Override  
 **public void** onClick(View v) {  
 *// 显式启动Activity* Intent intent = **new** Intent(IntentDemo.**this**, ActivityToStart.**class**);  
 startActivity(intent);  
 }  
}

ActivityToStart.java

**public class** ActivityToStart **extends** AppCompatActivity {  
  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_to\_start***);  
 }  
}

activity\_main.xml

<**RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android" android:orientation="vertical"  
 android:layout\_width="fill\_parent"  
 android:layout\_height="fill\_parent"**>  
  
 <**Button  
 android:id="@+id/btn"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_centerVertical="true"  
 android:layout\_centerHorizontal="true"  
 android:text="启动Activity"** />  
</**RelativeLayout**>

activity\_to\_start.xml

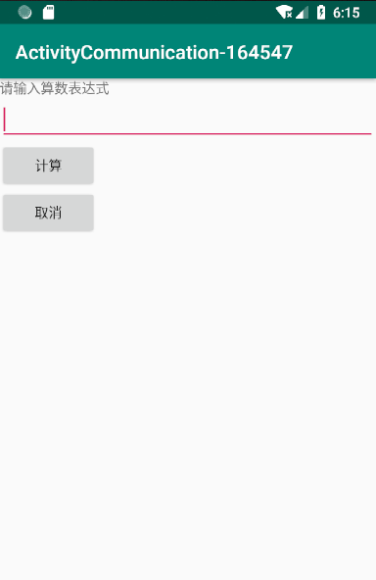
<**LinearLayout xmlns:android="http://schemas.android.com/apk/res/android" android:orientation="vertical"  
 android:layout\_width="fill\_parent"  
 android:layout\_height="fill\_parent"**>  
 <**TextView  
 android:layout\_width="fill\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="@string/hello"** />  
</**LinearLayout**>

**2. ActivityCommunication**

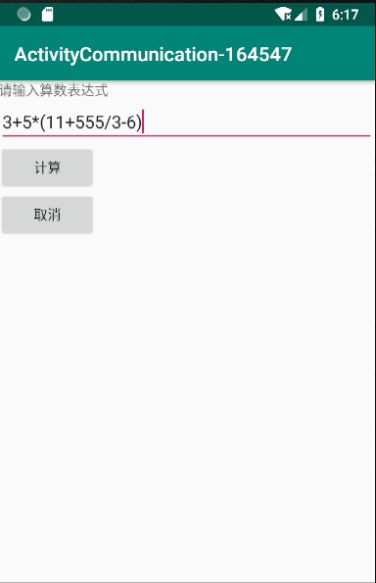
1）ActivityCommunication页面



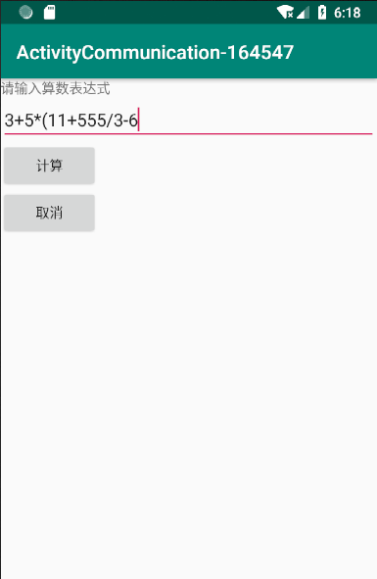
2）点击启动计算页面按钮后跳转到对应页面



3）输入合法的算数表达式后点击计算将数据返回到父活动正确显示计算结果

4）输入非法的算数表达式后点击计算捕获抛出的异常给出用户Toast提示

5）代码实现如下

ActivityCommunication.java

**public class** ActivityCommunication **extends** AppCompatActivity {  
  
 **private static final int *SUBACTIVITY1*** = 1;  
 **private static final int *SUBACTIVITY2*** = 2;  
  
 TextView **textView**;  
  
 */\*\*  
 \* Called when the activity is first created.  
 \*/* @Override  
 **public void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_main***);  
  
 **textView** = findViewById(R.id.***textShow***);  
 **final** Button btn1 = findViewById(R.id.***btn1***);  
 **final** Button btn2 = findViewById(R.id.***btn2***);  
  
 *// 只有一个显示声明的抽象方法的接口就是函数接口，就可以用lambda表达式来代替* btn1.setOnClickListener(view -> {  
 Intent intent = **new** Intent(ActivityCommunication.**this**, SubActivity1.**class**);  
 startActivityForResult(intent, ***SUBACTIVITY1***);  
 });  
  
 btn2.setOnClickListener(view -> {  
 Intent intent = **new** Intent(ActivityCommunication.**this**, SubActivity2.**class**);  
 startActivityForResult(intent, ***SUBACTIVITY2***);  
 });  
 }  
  
 @Override  
 **protected void** onActivityResult(**int** requestCode, **int** resultCode, Intent data) {  
 **super**.onActivityResult(requestCode, resultCode, data);  
  
 **switch** (requestCode) {  
 **case *SUBACTIVITY1***:  
 **if** (resultCode == ***RESULT\_OK***) {  
 Uri uriData = data.getData();  
 **textView**.setText(**"计算结果: "** + uriData.toString());  
 }  
 **break**;  
 **case *SUBACTIVITY2***:  
 **break**;  
 }  
 }  
  
}

SubActivity1.java  
**public class** SubActivity1 **extends** AppCompatActivity {  
  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_sub1***);  
 **final** EditText editText = findViewById(R.id.***edit***);  
 Button btnOK = findViewById(R.id.***btn\_ok***);  
 Button btnCancel = findViewById(R.id.***btn\_cancel***);  
 *// 应用jdk8中的lambda表达式来简化代码* btnOK.setOnClickListener(view -> {  
 String uriString = editText.getText().toString();  
 *// 创建一个js引擎* ScriptEngine engine = **new** ScriptEngineManager().getEngineByName(**"js"**);  
 **try** {  
 *// 将输入的数据使用js引擎计算并转换为字符串封装到data中返回给父活动* Uri data = Uri.*parse*(engine.eval(uriString).toString());  
 Intent result = **new** Intent(**null**, data);  
 setResult(***RESULT\_OK***, result);  
 finish();  
 } **catch** (ScriptException e) {  
 *// 如果输入不满足表达式的标准就会抛出异常，在异常处理中给出相应提示* Toast.*makeText*(**this**, **"请输入正确的表达式"**, Toast.***LENGTH\_SHORT***).show();  
 }  
  
 });  
 btnCancel.setOnClickListener(view -> {  
 setResult(***RESULT\_CANCELED***, **null**);  
 finish();  
 });  
  
 }  
}

SubActivity2.java

**public class** SubActivity2 **extends** AppCompatActivity {  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_sub2***);  
 Button btnReturn = findViewById(R.id.***btn\_return***);  
  
 btnReturn.setOnClickListener(view -> {  
 setResult(***RESULT\_CANCELED***, **null**);  
 finish();  
 });  
  
 }  
}

activity\_main.xml

<**LinearLayout xmlns:android="http://schemas.android.com/apk/res/android" android:orientation="vertical"  
 android:layout\_width="fill\_parent"  
 android:layout\_height="fill\_parent"**>  
 <**TextView android:id="@+id/textShow"  
 android:layout\_width="fill\_parent"  
 android:layout\_height="wrap\_content"  
 android:text=""** />  
 <**Button android:id="@+id/btn1"  
 android:layout\_width="wrap\_content" android:layout\_height="wrap\_content"  
 android:text="启动计算页面"** />  
 <**Button android:id="@+id/btn2"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="启动Activity2"** />  
</**LinearLayout**>

activity\_sub1.xml

<**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/textShow"  
 android:layout\_width="fill\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="请输入算数表达式"** />  
  
 <**EditText  
 android:id="@+id/edit"  
 android:layout\_width="fill\_parent"  
 android:layout\_height="wrap\_content"  
 android:text=""** />  
  
 <**Button  
 android:id="@+id/btn\_ok"  
 android:layout\_width="100dip"  
 android:layout\_height="wrap\_content"  
 android:text="计算"** />  
  
 <**Button  
 android:id="@+id/btn\_cancel"  
 android:layout\_width="100dip"  
 android:layout\_height="wrap\_content"  
 android:text="取消"** />  
</**LinearLayout**>

activity\_2.xml

<**LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:orientation="vertical"  
 android:layout\_width="fill\_parent"  
 android:layout\_height="fill\_parent"** >  
 <**TextView  
 android:id="@+id/textShow"  
 android:layout\_width="fill\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="SubActivity 2"** />  
 <**Button  
 android:id="@+id/btn\_return"  
 android:layout\_width="100dip"  
 android:layout\_height="wrap\_content"  
 android:text="关闭"** />  
</**LinearLayout**>

**3.** **IntentResolutionDemo**

**1）首页**



**2）点击启动ACTIVITY按钮后隐式启动ActivityToStart活动**



**3）代码实现如下**

IntentResolutionDemo.java

**public class** IntentResolutionDemo **extends** AppCompatActivity {  
  
 @Override  
 **public void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_main***);  
  
 Button button = findViewById(R.id.***btn***);  
 button.setOnClickListener(view -> {  
 Intent intent = **new** Intent(Intent.***ACTION\_VIEW***, Uri.*parse*(**"schema164547://ming.hebut/path"**));  
 startActivity(intent);  
 });  
 }  
}

ActivityToStart.java

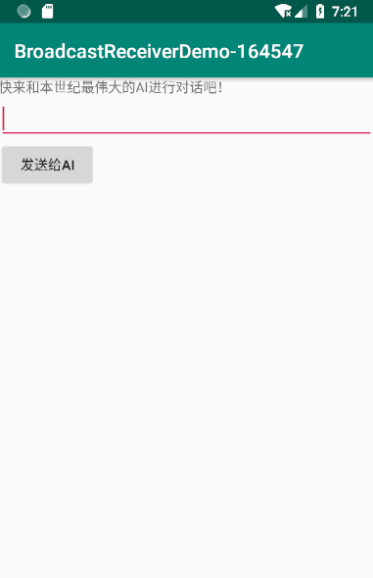
**public class** ActivityToStart **extends** AppCompatActivity {  
 @Override  
 **public void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_to\_start***);  
 }  
}

AndroidMenifest.xml

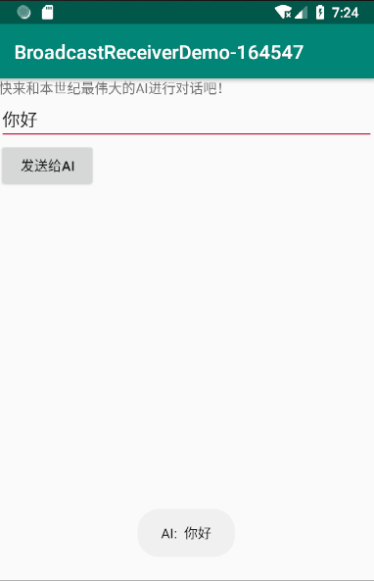
<**manifest xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:tools="http://schemas.android.com/tools"  
 package="com.hebut.intentresolutiondemo"**>  
  
 <**application  
 android:allowBackup="true"  
 android:icon="@mipmap/ic\_launcher"  
 android:label="@string/app\_name"  
 android:roundIcon="@mipmap/ic\_launcher\_round"  
 android:supportsRtl="true"  
 android:theme="@style/AppTheme"  
 tools:ignore="GoogleAppIndexingWarning"**>  
 <**activity android:name=".ActivityToStart"**>  
 <**intent-filter**>  
 <**action android:name="android.intent.action.VIEW"** />  
 <**category android:name="android.intent.category.DEFAULT"** />  
 <**data  
 android:host="ming.hebut"  
 android:scheme="schema164547"** />  
 </**intent-filter**>  
 </**activity**>  
 <**activity android:name=".IntentResolutionDemo"**>  
 <**intent-filter**>  
 <**action android:name="android.intent.action.MAIN"** />  
 <**category android:name="android.intent.category.LAUNCHER"** />  
 </**intent-filter**>  
 </**activity**>  
 </**application**>  
</**manifest**>

**4. BroadcastReceiverDemo**

**1）广播发送页**



**2）输入消息，点击发送给MyBroadcastReceiver，receiver处理后发送Toast到前台**





**3）代码实现如下**

BroadcastReceiverDemo.java

**public class** BroadcastReceiverDemo **extends** AppCompatActivity {  
 **private** EditText **entryText**;  
 **private** Button **button**;  
  
 @Override  
 **public void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_main***);  
 **entryText** = findViewById(R.id.***entry***);  
 **button** = findViewById(R.id.***btn***);  
  
 **button**.setOnClickListener(view -> {  
 String text = **entryText**.getText().toString();  
 Intent intent = **new** Intent(**"com.hebut.BroadcastReceiverDemo"**);  
 *// 安卓8.0之后不再允许通过静态注册的方式接收隐式广播  
 // 现在转用显式广播，在发送前指定接收广播的包名，当然也可以使用动态注册的方式  
 // 就不用关心是隐式广播还是显式广播了* intent.setPackage(**"com.hebut.broadcastreceiverdemo"**);  
 intent.putExtra(**"message"**, text);  
 sendBroadcast(intent);  
 });  
 }  
}

MyBroadcastReceiver.java

**public class** MyBroadcastReceiver **extends** BroadcastReceiver {  
 @Override  
 **public void** onReceive(Context context, Intent intent) {  
 String msg = intent.getStringExtra(**"message"**);  
  
 *// 前一阵很火的估值一个亿的AI核心代码（手动狗头）* msg = msg.replace(**"吗"**, **""**).replace(**"?"**, **"!"**).replace(**"？"**, **"!"**);  
 Toast.*makeText*(context, **"AI: "** + msg, Toast.***LENGTH\_SHORT***).show();  
 }  
}

activity\_main.xml

<**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/textShow"  
 android:layout\_width="fill\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="快来和本世纪最伟大的AI进行对话吧！"** />  
  
 <**EditText  
 android:id="@+id/entry"  
 android:layout\_width="fill\_parent"  
 android:layout\_height="wrap\_content"  
 android:text=""** />  
  
 <**Button  
 android:id="@+id/btn"  
 android:layout\_width="100dip"  
 android:layout\_height="wrap\_content"  
 android:text="发送给AI"** />  
  
</**LinearLayout**>

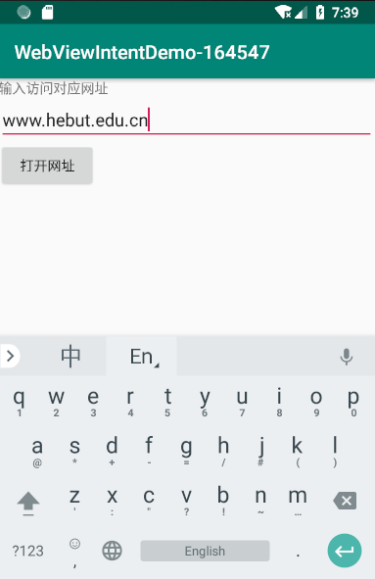
AndroidMenifest.xml

*<?***xml version="1.0" encoding="utf-8"***?>*<**manifest xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:tools="http://schemas.android.com/tools"  
 package="com.hebut.broadcastreceiverdemo"**>  
  
 <**application  
 android:allowBackup="true"  
 android:icon="@mipmap/ic\_launcher"  
 android:label="@string/app\_name"  
 android:roundIcon="@mipmap/ic\_launcher\_round"  
 android:supportsRtl="true"  
 android:theme="@style/AppTheme"  
 tools:ignore="GoogleAppIndexingWarning"**>  
 <**activity android:name=".BroadcastReceiverDemo"**>  
 <**intent-filter**>  
 <**action android:name="android.intent.action.MAIN"** />  
  
 <**category android:name="android.intent.category.LAUNCHER"** />  
 </**intent-filter**>  
 </**activity**>  
 <**receiver android:name=".MyBroadcastReceiver"**>  
 <**intent-filter**>  
 <**action android:name="com.hebut.BroadcastReceiverDemo"** />  
 </**intent-filter**>  
 </**receiver**>  
  
 </**application**>  
  
</**manifest**>

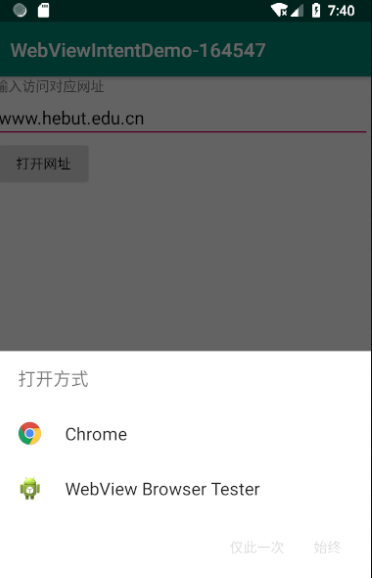
**五、实验思考题**

**1. 使用“WebViewIntentDemo”说明如何隐式启动Activity？**

**1）如下图为网址输入页**



**2）点击之后**



**3）在其中任选一个浏览器之后正确打开输入网址**



**4）代码实现如下**

WebViewIntentDemo.java  
  
**public class** WebViewIntentDemo **extends** AppCompatActivity {  
  
 **private** EditText **entryText**;  
 **private** Button **button**;  
  
 @Override  
 **public void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_main***);  
 **entryText** = findViewById(R.id.***entry***);  
 **button** = findViewById(R.id.***btn***);  
  
 **button**.setOnClickListener(view -> {  
 String url = **entryText**.getText().toString();  
 *// 判断一下输入的网址是否以http开头，如果没有就自动添加协议头，否则Intent将无法识别该数据* url = url.startsWith(**"http"**) ? url : **"http://"** + url;  
  
 *// intent动作为ACTION\_VIEW，该动作会根据Uri类型自动启动处理相应数据的Activity  
 // 通过查看代码可知可处理邮件(mailto:xxx)、电话号码(tel:xxx)等内容* Intent intent = **new** Intent(Intent.***ACTION\_VIEW***, Uri.*parse*(url));  
 startActivity(intent);  
 });  
 }  
}

**activity\_main.xml**

<**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/textShow"  
 android:layout\_width="fill\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="输入访问对应网址"** />  
  
 <**EditText  
 android:id="@+id/entry"  
 android:layout\_width="fill\_parent"  
 android:layout\_height="wrap\_content"  
 android:text=""** />  
  
 <**Button  
 android:id="@+id/btn"  
 android:layout\_width="100dip"  
 android:layout\_height="wrap\_content"  
 android:text="打开网址"** />  
  
</**LinearLayout**>

**2．简述Intent过滤器的原理与匹配机制？**

Intent过滤器主要用于在隐式启动时使Android系统能够根据Intent中的数据信息，找到需要启动的Activity。过滤器可以根据Intent中的动作(action)、类别(category)和数据(data)等内容对适合接收该 Intent 的组件进行匹配和筛选。

在Intent与Intent过滤器进行匹配时，Android系统会将列表中所有Intent过滤器的“动作”和“类别”与Intent进行匹配，任何不匹配的Intent过滤器都将被过滤掉。没有指定“动作”的Intent过滤器可以匹配任何的Intent，但是没有指定“类别”的Intent过滤器只能匹配没有“类别”的Intent。

**3．简述如何获取Activity返回值？**

首先在父活动中通过startActivityForResult(intent, ***SUBACTIVITY1***)的方式打开子活动窗口

然后在子活动的finish()方法之前添加

Intent result = **new** Intent(**null**, data);  
setResult(***RESULT\_OK***, result);

从而将子活动的数据发送给父活动，父活动通过重写onActivityResult(**int** requestCode, **int** resultCode, Intent data) 的回调进而实现对子活动中发送的数据的获取。

**六、实验感想**

1. jdk8中新引入的lambda表达式很好用，可以大大减少重复代码，但同时也会降低代码的可读性

2. js中的eval函数可用来计算字符串形式的数学表达式的值，可以说很好用了，各语言中也都有js的相关引擎可以使用，但是效率较低

3. 安卓8.0之后不再允许通过静态注册的方式接收隐式广播了，现在应转用显式广播，在发送前指定接收广播的包名，或使用动态注册的方式，就不用关心是隐式广播还是显式广播了。

****