

Bài tập 24: Thực hành về Intent trong Android

- Intent có rất nhiều tính năng từ việc dùng Intent để mở các Activity và truyền dữ liệu qua lại giữa chúng đến dùng Intent ở những mục đích khác nhau sẽ được bàn kỹ hơn ở các bài tập tiếp sau.

- Về mặt tổng quát ta có thể chia Intent ra làm 2 công dụng chính:

- 1) Dùng **Intent** để mở một Activity khác nhưng không kiểm soát kết quả trả về.
- 2) Dùng **Intent** để mở một Activity khác và có kiểm soát kết quả trả về.

Sau đây là những ví dụ cụ thể:

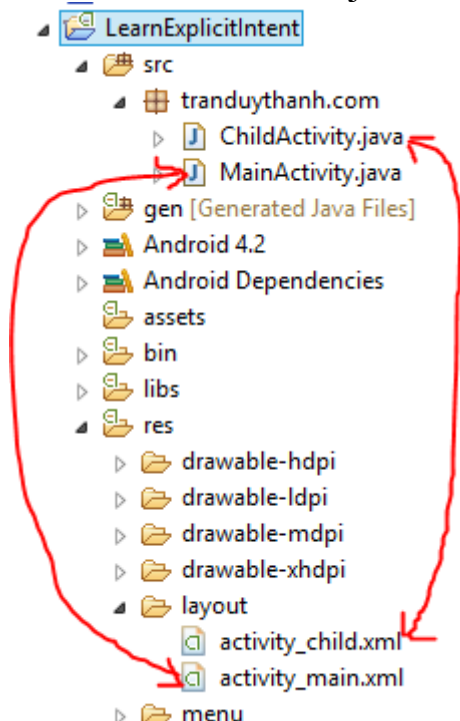
Ví dụ 1: Đơn thuần là mở một Activity khác.



Giải thích hình trên như sau:

- Từ Main Activity có 1 Button, khi nhấn vào Button này thì sẽ mở một Activity khác, tại Activity mới mở này sẽ có 1 Button cho phép quay trở lại MainActivity.

- Tạo một Android Project có cấu trúc như hình bên dưới:



- Chú ý là mọi Activity muốn được triệu gọi trong Android thì phải đăng ký trong Manifest, cụ thể như sau:

AndroidManifest.xml:

```
1 <?xmlversion="1.0"encoding="utf-8"?>
  <manifestxmlns:android="http://schemas.android.com/apk/res/android"
```

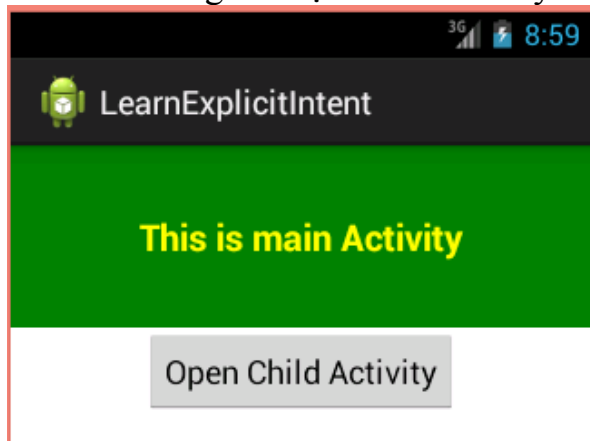
```

2  android:versionCode="1"
3  android:versionName="1.0">
4  <uses-sdk
5  android:minSdkVersion="14"
6  android:targetSdkVersion="17">
7  </uses-sdk>
8  <application
9  android:allowBackup="true"
10 android:icon="@drawable/ic_launcher"
11 android:label="@string/app_name"
12 android:theme="@style/AppTheme">
13     <activity
14         android:name=".MainActivity"
15         android:label="@string/app_name">
16         <intent-filter>
17             <actionandroid:name="android.intent.action.MAIN"/>
18             <categoryandroid:name="android.intent.category.LAUNCHER"/>
19         </intent-filter>
20     </activity>
21     <activity
22         android:name=".ChildActivity"
23         android:label="@string/title_activity_child">
24     </activity>
25 </application>
26 </manifest>

```

- Ta thấy cả **MainActivity** (dòng 13) và **ChildActivity** (Dòng 19) đều phải được đăng ký trong Manifest.

- Giờ ta xem giao diện và cách xử lý coding trong MainActivity:



- Tiến hành xem coding **MainActivity.java**:

```

1  public class MainActivity extends Activity {
2      protected void onCreate(Bundle savedInstanceState) {
3          super.onCreate(savedInstanceState);
4          setContentView(R.layout.activity_main);
5          Button btnOpen=(Button) findViewById(R.id.btnOpenChildActivity);
6          btnOpen.setOnClickListener(new View.OnClickListener() {
7              public void onClick(View arg0) {
8                  doOpenChildActivity();
9              }
10         });
11     }
12     public void doOpenChildActivity()

```

```

10 {
11     Intent myIntent=newIntent(this, ChildActivity.class );
12     startActivity(myIntent);
13 }

```

- Quan sát dòng lệnh 10:

Intent myIntent=new **Intent**(**this**, **ChildActivity.class**);

Đối số 1 là context hiện tại, hoặc tên đầy đủ là:

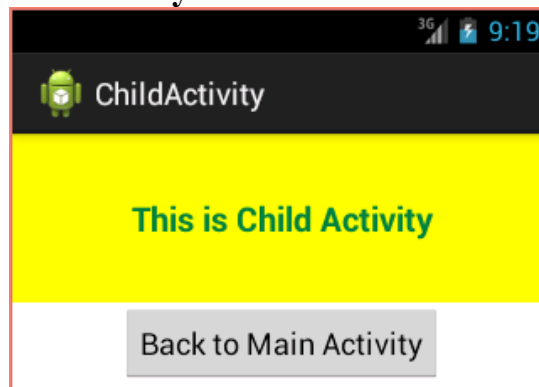
Intent myIntent=new **Intent**(**MainActivity.this**, **ChildActivity.class**);

Với **MainActivity** là một Activity tùy ý (trong Context hiện tại), viết như trên thì bạn sẽ luôn luôn đúng trong mọi trường hợp.

Đối số 2 là : **ChildActivity.class** , Tức là muốn mở Activity nào thì lấy tên Activity đó .class ; trong trường hợp này muốn mở ChildActivity nên ghi là **ChildActivity.class**

- Quan sát dòng lệnh 11: **startActivity**(myIntent); Hàm này có tác dụng mở Activity được truyền vào đối số thứ 2 ở trên.

- Tiến hành khám phá ChildActivity:



-Giao diện của ChildActivity chỉ có một Button cho phép quay lại MainActivity.

- Bạn xem coding của **ChildActivity**:

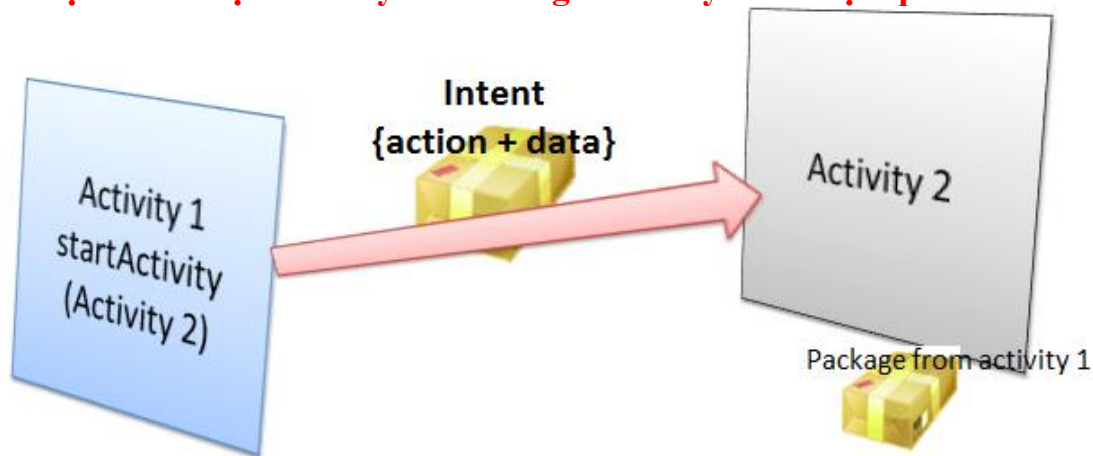
```

1 public class ChildActivity extends Activity {
2     protected void onCreate(Bundle savedInstanceState) {
3         super.onCreate(savedInstanceState);
4         setContentView(R.layout.activity_child);
5
6         Button btnBack= (Button) findViewById(R.id.btnBacktoMainActivity);
7         btnBack.setOnClickListener(new View.OnClickListener() {
8             public void onClick(View v) {
9                 finish();
10            }
11        });
12    }
13 }

```

-Dòng lệnh 8: **finish()** có nhiệm vụ đóng Activity hiện tại.

Ví dụ 2: Mở một Activity khác đồng thời truyền dữ liệu qua.



- Ở đây Ta thường dùng đối tượng **Bundle** để đóng gói dữ liệu để truyền tải qua các Activity khác. Trong Android khi truyền dữ liệu chúng ta cũng có thể truyền từng thông số riêng lẻ nhưng nên thực hiện đóng gói dữ liệu vào 1 Bundle rồi gửi Bundle này qua Activity nào đó. Bên kia chỉ cần lấy Bundle này ra, mọi dữ liệu đều nằm trong Bundle và dễ dàng được xử lý.

```
Bundle bundle=new Bundle();  
bundle.putInt("soa", 5);  
bundle.putDouble("sob", 5.7);  
//...  
int soa=bundle.getInt("soa");  
double sob=bundle.getDouble("sob");
```

Diagram illustrating the Bundle operations: A red arrow labeled 'Put' points to the `bundle.putInt` and `bundle.putDouble` lines. An orange arrow labeled 'get' points to the `bundle.getInt` and `bundle.getDouble` lines.

- Bundle có 2 loại phương thức: **putXXX** dùng để cho bên gửi đi, ứng với kiểu dữ liệu nào thì put đúng kiểu đó. ở ví dụ trên `putInt` và `putDouble` là tương ứng với 2 loại dữ liệu khác nhau (phải đi kèm với **key** để bên nhận có thể xử lý đúng)
- Ở bên nhận Bundle thì dùng các phương thức **getXXX** tương ứng để lấy dữ liệu theo **key** bên gửi.
- key này phải giống nhau cho cả 2 nơi. Nơi nhận đặt `key=soa` thì bên nhận cũng phải dùng `key=soa`.
- Ta cũng có thể truyền Đối Tượng qua lại giữa các Activity, các đối tượng này phải được Serialize

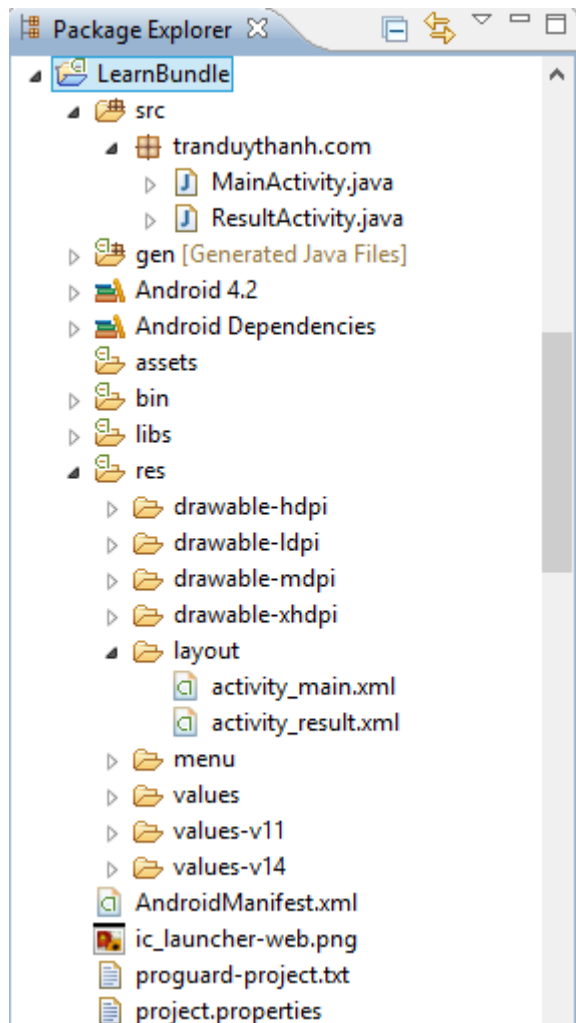
```
class Person implements Serializable
{
    private int id;
    private String name;
    public Person(int id,String name)
    {
        this.id=id;
        this.name=name;
    }
    public String toString() {
        return this.id+"- "+this.name;
    }
}
```

```
Bundle bundle=new Bundle();
Person p=new Person(1, "teo");
bundle.putSerializable("t1", p);
//...
Person t1=(Person) bundle.getSerializable("t1");
```

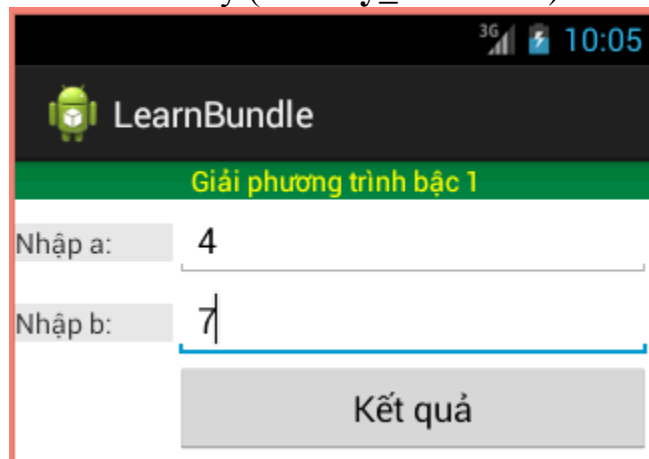
- Trong trường hợp truyền đối tượng thì ta dùng **putSerializable** và **getSerializable**
- Tôi sẽ làm một ví dụ đơn giản với giao diện như bên dưới:



- Từ MainActivity có 2 thông số là a và b. Khi nhấn nút **Kết quả** nó sẽ truyền 2 thông số này qua ResultActivity và tiến hành giải phương trình bậc 1. Muốn trở về để tiếp tục giải phương trình khác khi nhấn nút **Back**.
- Bạn xem cấu trúc chương trình để dễ xử lý:



- Bạn xem giao diện của MainActivity (activity_main.xml):



- MainActivity.java:

```

2  public class MainActivity extends Activity {
3      EditText txta,txtb;
4      Button btnketqua;
5      Protected void onCreate(Bundle savedInstanceState) {
6          super.onCreate(savedInstanceState);
7          setContentView(R.layout.activity_main);
8          txta=(EditText) findViewById(R.id.txta);
9          txtb=(EditText) findViewById(R.id.txtb);
10         btnketqua=(Button) findViewById(R.id.btnketqua);
11
12         btnketqua.setOnClickListener(new View.OnClickListener() {
13             public void onClick(View arg0) {
14                 //Tạo Intent để mở ResultActivity
15                 Intent myIntent=new Intent(MainActivity.this,
16                     ResultActivity.class );
17                 //Khai báo Bundle
18                 Bundle bundle=new Bundle();
19                 int a=Integer.parseInt(txta.getText().toString());
20                 int b=Integer.parseInt(txtb.getText().toString());
21                 //đưa dữ liệu riêng lẻ vào Bundle
22                 bundle.putInt("soa", a);
23                 bundle.putInt("sob", b);
24                 //Đưa Bundle vào Intent
25                 myIntent.putExtra("MyPackage", bundle);
26                 //Mở Activity ResultActivity
27                 startActivity(myIntent);
28             }
29         });
30     }
31 }

```

- Ở dòng 19, key nhận dạng Bundle là **MyPackage**, bên ResultActivity sẽ dựa vào key này để lấy Bundle ra.
- Sau khi có Bundle rồi thì dựa vào **soa, sob** ở dòng lệnh 17,18 để lấy đúng dữ liệu.
- Bây giờ ta qua giao diện **ResultActivity**:



- Tiếp tục xem xử lý coding của **ResultActivity**:

```

1  public class ResultActivity extends Activity {
2      TextView txtketqua;
3      Button btnBack;
4      protected void onCreate(Bundle savedInstanceState) {
5          super.onCreate(savedInstanceState);
6          setContentView(R.layout.activity_result);

```

```

6
7   btnBack=(Button) findViewById(R.id.btnBack);
8   txtketqua=(TextView) findViewById(R.id.txtketqua);
9
10  //lấy intent gọi Activity này
11  Intent callerIntent=getIntent();
12
13  //có intent rồi thì lấy Bundle dựa vào MyPackage
14  Bundle packageFromCaller= callerIntent.getBundleExtra("MyPackage");
15
16  //Có Bundle rồi thì lấy các thông số dựa vào soa, sob
17  inta=packageFromCaller.getInt("soa");
18  intb=packageFromCaller.getInt("sob");
19
20  //tiến hành xử lý
21  giaipt(a, b);
22  btnBack.setOnClickListener(new View.OnClickListener() {
23      public void onClick(View v) {
24          finish();
25      }
26  });
27
28  }
29
30  Public void giaipt(int a,int b)
31  {
32      String kq="";
33      if(a==0&& b==0)
34      {
35          kq="Vô số nghiệm";
36      }
37      else if(a==0&& b!=0)
38      {
39          kq="Vô nghiệm";
40      }
41      else
42      {
43          DecimalFormat dcf = new DecimalFormat("0.##");
44          kq=dcf.format(-b*1.0/a);
45      }
46      txtketqua.setText(kq);
47  }
48  }

```

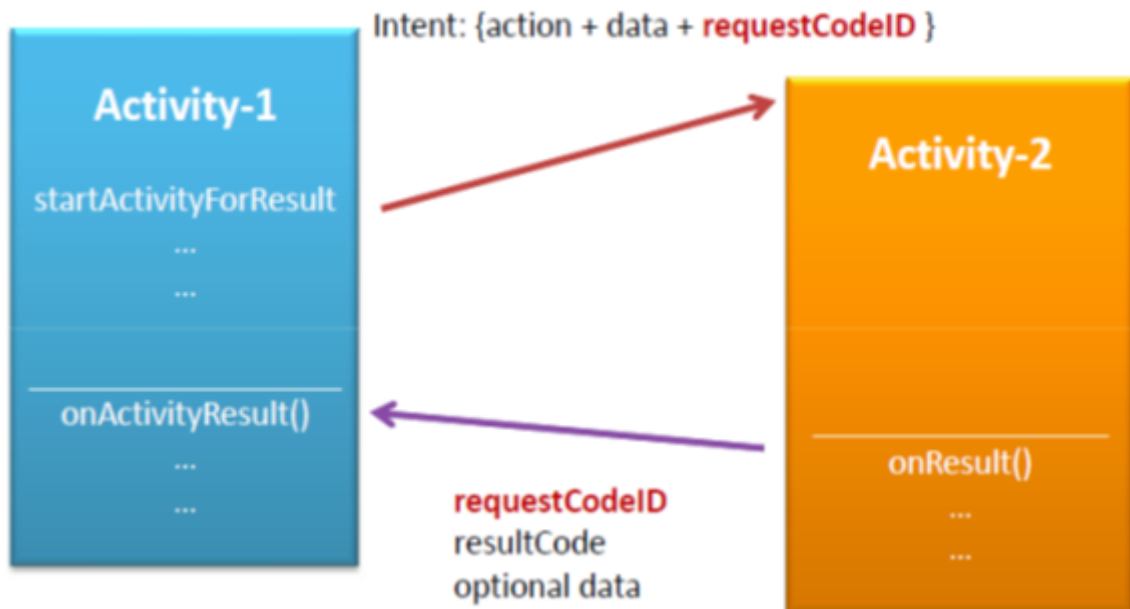
-Bạn xem dòng lệnh 10: **Intent callerIntent=getIntent()**; lệnh này cho phép lấy Intent start Activity này. Tức là lấy Intent mà ta khai báo bên MainActivity để start ResultActivity.

Chú ý khai báo thêm ResultActivity trong fine AndroidMainifest.xml

Ví dụ 3: Mở một Activity khác đồng thời truyền dữ liệu qua và xử lý kết quả trả về.

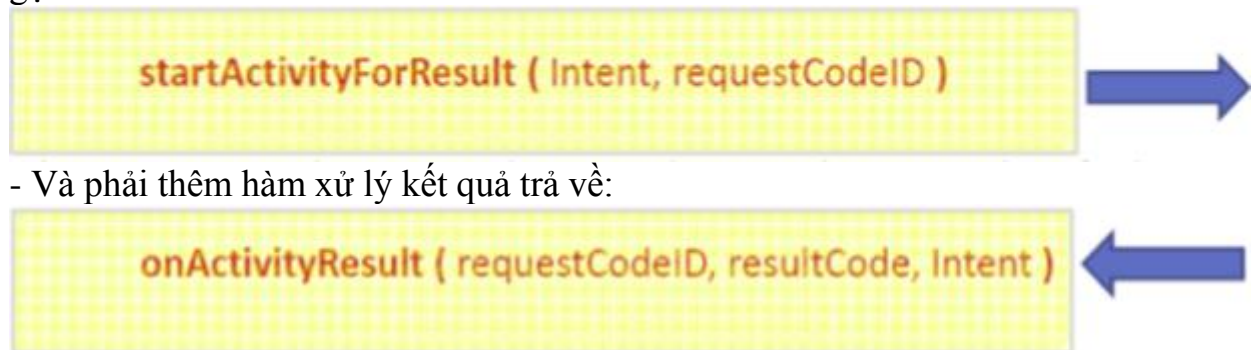
- Ví dụ này bạn sẽ biết cách xử lý kết quả trả về như thế nào.

- Bạn xem hình minh họa:



Chúng ta sẽ dựa vào requestCodeID và resultCode để xử lý.

- Việc tạo Intent trong trường hợp này cũng y xì như trường hợp trước. Nó chỉ khác hàm gọi :

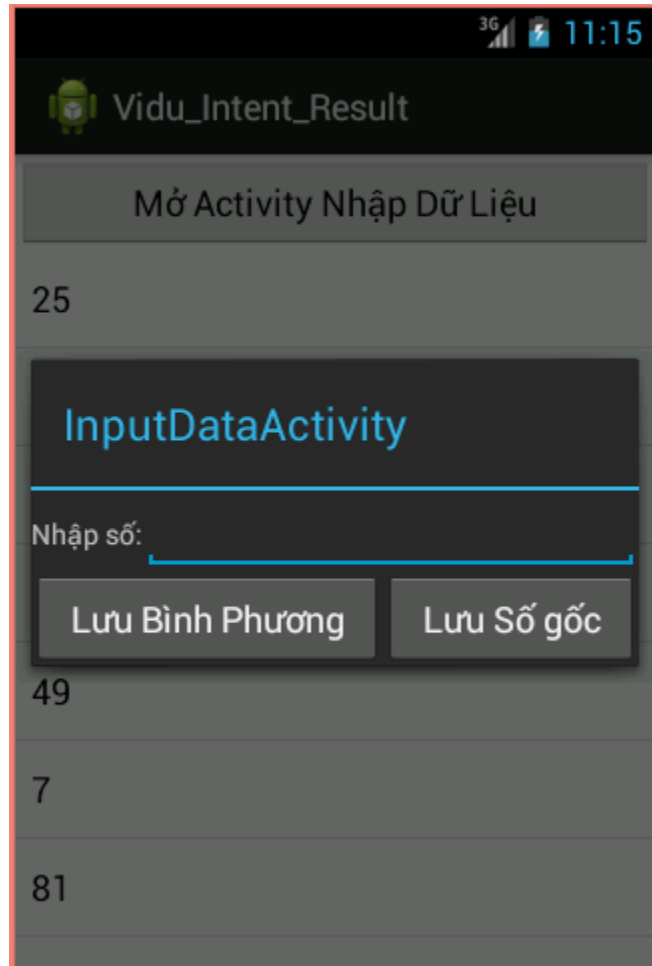


- Và phải thêm hàm xử lý kết quả trả về:

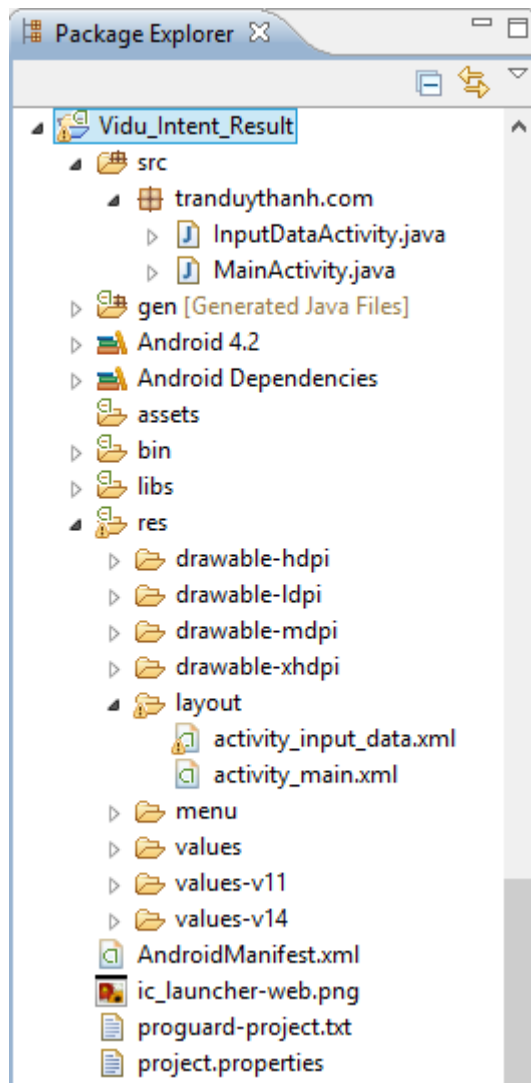
- Ví dụ minh họa: Chương trình đơn giản là MainActivity sẽ có 1 Button và ListView như hình bên dưới:



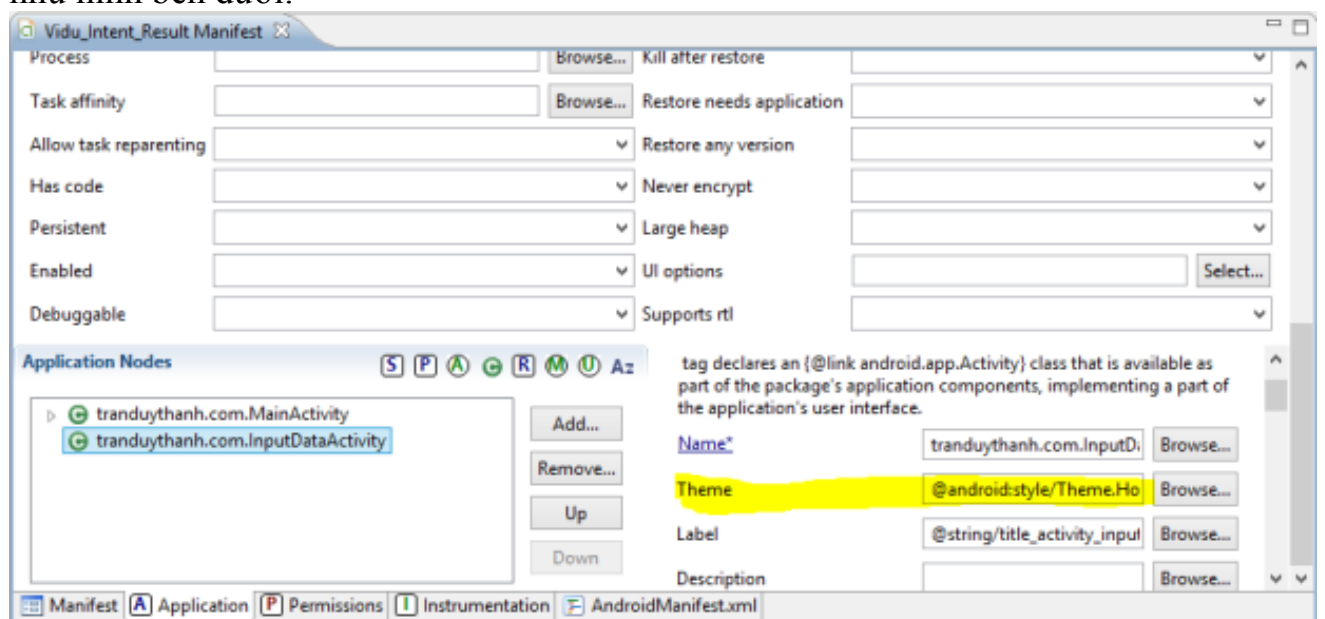
- Khi nhấn “Mở Activity nhập dữ liệu”, chương trình sẽ mở Activity mới dưới dạng Dialog:



- Khi nhập số và nhấn “Lưu bình phương” thì nó sẽ truyền số này qua MainActivity và cập nhật ListView với số này là nhân bình phương.
- Khi nhập số và nhấn “lưu số gốc” thì nó sẽ truyền số này qua MainActivity và cập nhật ListView với đúng số gốc này.
- Chú ý là phải đóng **Dialog** ngay, vì nếu không đóng thì **onActivityResult** sẽ không xảy ra. **onActivityResult** chỉ xảy ra trong **foreground life time**.
- Xem cấu trúc chương trình:

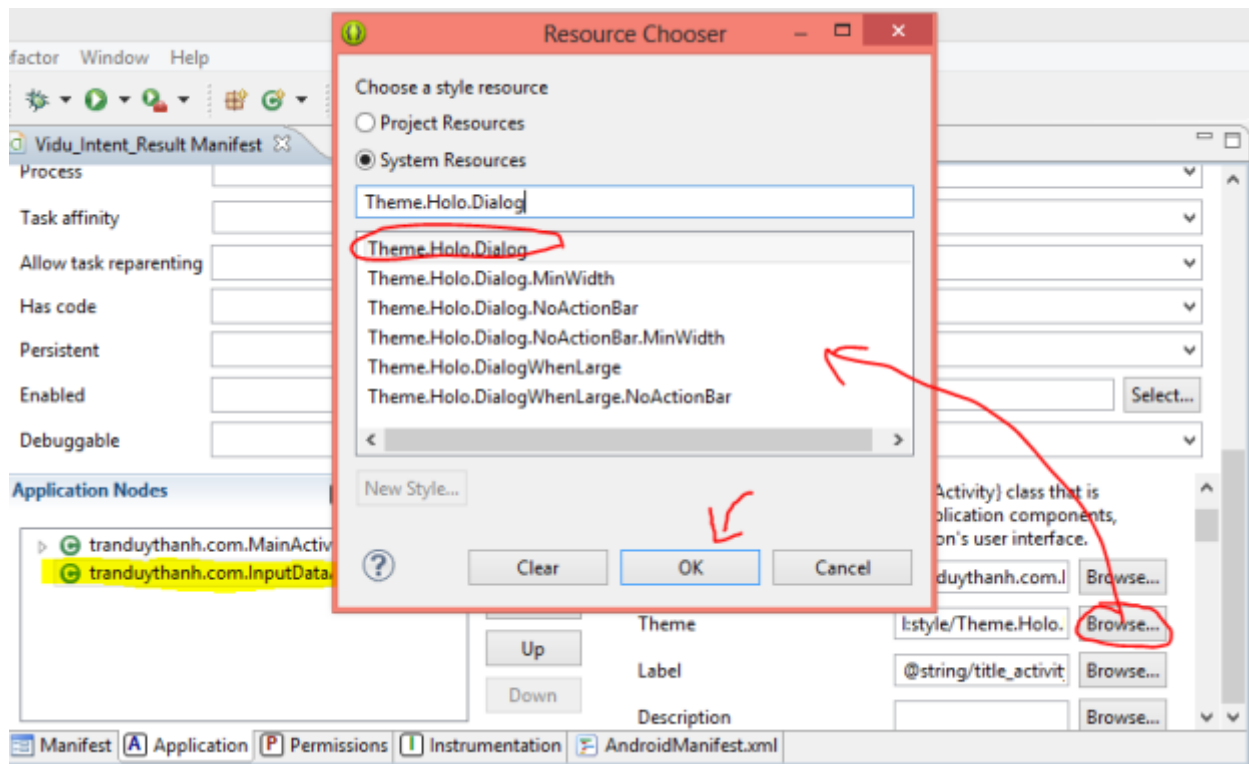


- Nhớ là phải cấu hình InputDataActivity dưới dạng Dialog: Mở Manifest để cấu hình như hình bên dưới:



chọn Theme: **@android:style/Theme.Holo.Dialog**

-Tại màn hình trên ta chọn đúng Activity muốn làm Dialog rồi tìm tới thuộc tính Theme, nhấn vào nút “Browse”:



-Xử lý code của **MainActivity.java**:

```

1
2 public class MainActivity extends Activity {
3
4     public static final int REQUEST_CODE_INPUT=113;
5     public static final int RESULT_CODE_SAVE1=115;
6     public static final int RESULT_CODE_SAVE2=116;
7
8     Button btnInputData;
9     ListView lvData;
10    ArrayList<Integer>arrData = new ArrayList<Integer>();
11    ArrayAdapter<Integer>adapter=null;
12    @Override
13    Protected void onCreate(Bundle savedInstanceState) {
14        super.onCreate(savedInstanceState);
15        setContentView(R.layout.activity_main);
16
17        btnInputData =(Button) findViewById(R.id.btnopenactivity);
18        btnInputData.setOnClickListener(new OnClickListener() {
19
20            @Override
21            public void onClick(View arg0) {
22                //Mở Activity với REQUEST_CODE_INPUT
23                Intent intent =new Intent(MainActivity.this,
24                InputDataActivity.class );
25                //gọi startActivityForResult sẽ yêu cầu ghi đè
26                onActivityResult
27                startActivityForResult(intent, REQUEST_CODE_INPUT);
28            }
29        });
30        //đoạn code dưới này học nhiều rồi, ko nói lại
31        lvData=(ListView) findViewById(R.id.lvdata);
32        adapter=newArrayAdapter<Integer>(this,android.R.layout.simple_li
33        st_item_1,arrData);

```

```

28         lvData.setAdapter(adapter);
29     }
30 /**
31  * Xử lý kết quả trả về ở đây
32  */
33 @Override
34 Protected void onActivityResult(int requestCode, int resultCode,
35 Intent data) {
36     super.onActivityResult(requestCode, resultCode, data);
37     //Kiểm tra có đúng requestCode =REQUEST_CODE_INPUT hay không
38     //Vì ta có thể mở Activity với những RequestCode khác nhau
39     if(requestCode==REQUEST_CODE_INPUT)
40     {
41         //Kiểm tra ResultCode trả về, cái này ở bên InputDataActivity
42         //truyền về
43         //có nó rồi thì xử lý trở lên thông thường
44         switch(resultCode)
45         {
46             case RESULT_CODE_SAVE1:
47                 //giá trị từ InputDataActivity
48                 intv1= data.getIntExtra("data", 0);
49                 arrData.add(v1*v1);
50                 adapter.notifyDataSetChanged();
51                 break;
52             case RESULT_CODE_SAVE2:
53                 //giá trị từ InputDataActivity
54                 intv2= data.getIntExtra("data", 0);
55                 arrData.add(v2);
56                 adapter.notifyDataSetChanged();
57                 break;
58         }
59     }
60 }

```

- Chúng ta nên khai báo hằng số cho **RequestCode** và **ResultCode** như ở trên, dựa vào nó để xử lý kết quả trả về.
- Bạn qua xử lý code của **InputDataActivity.java**:

```

1 public class InputDataActivity extends Activity {
2
3     Button btnSave1, btnSave2;
4     EditText editNumber;
5     @Override
6     Protected void onCreate(Bundle savedInstanceState) {
7         super.onCreate(savedInstanceState);
8         setContentView(R.layout.activity_input_data);
9         btnSave1=(Button) findViewById(R.id.btnSave1);
10        btnSave2=(Button) findViewById(R.id.btnSave2);
11
12        btnSave1.setOnClickListener(new OnClickListener() {
13            @Override
14            public void onClick(View arg0) {
15                //Gửi thông điệp là lưu bình phương
16                sendToMain(MainActivity.RESULT_CODE_SAVE1);
17            }
18        });
19        editNumber=(EditText) findViewById(R.id.editNumber);
20
21        btnSave2.setOnClickListener(new OnClickListener() {
22            @Override
23            public void onClick(View v) {
24                //Gửi thông điệp là lưu số gốc
25                sendToMain(MainActivity.RESULT_CODE_SAVE2);
26            }
27        });
28    }
29    /**
30     * hàm xử lý gửi kết quả về mainactivity
31     * khi hàm này được gọi thì lập tức onActivityResult
32     * ở MainActivity sẽ xảy ra đem theo ResultCode và Intent
33     * @param resultcode
34     */
35    Public void sendToMain(int resultcode)
36    {
37        Intent intent=getIntent();
38        Int value= Integer.parseInt(editNumber.getText()+"");
39        intent.putExtra("data", value);
40        setResult(resultcode, intent);
41        finish();
42    }
43 }

```

Chú ý: hàm **sendToMain**, khi hàm này vừa kết thúc thì sự kiện **onActivityResult** của **MainActivity** sẽ xảy ra

onActivityResult(int requestCode, int resultCode, Intent data)

Ta kiểm tra requestCode và resultCode cho đúng

- **requestCode** là bên MainActivity dùng để triệu gọi một Activity bất kỳ nào đó
- **resultCode** là kết quả trả về trong hàm **setResult(resultcode, **intent**)**; của sub Activity nào đó. Hàm này cho chúng ta biết kết quả trả về là **code** nào và đồng thời cho ta biết luôn **Intent** của nó. Dựa vào **Intent** này mà trong **onActivityResult** ta có thể dễ dàng lấy thông số ra (đối số thứ 3).