**Bài tập 13: Thực hành về ListView trong Android**

Trong các bài tập trước các bạn đã được làm quen với nhiều control cơ bản, bài tập này bạn sẽ được làm quen với control nâng cao, cụ thể là ListView.

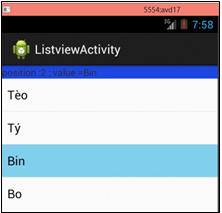
Trong ứng dụng cần lưu trữ và hiển thị danh sách các thông tin đa phần chúng ta sài control ListView. Hiện tại bạn chỉ cần biết sử dụng ListView có sẵn của Android là được rồi, trong các bài tập tiếp theo Tôi sẽ hướng dẫn các bạn Custom Layout lại ListView (tự làm mới ListView theo ý mình).

– Bài tập này Tôi sẽ cung cấp nhiều cách hành xử với ListView, ứng với mỗi cách là có các ví dụ mẫu khác nhau, vì vậy các bạn nên cố gắng theo dõi và thực hành lại những ví dụ.

– Bạn hãy thực hành tốt 5 trường hợp Tôi trình bày dưới đây:

**1) Trường hợp 1:**– Sử dụng ListView control với mảng dữ liệu định sẵn.

-Trường hợp này Tôi đưa ra một ví dụ đơn giản là cho phép hiển thị mảng dữ liệu lên trên ListView, bạn xem hình minh họa:

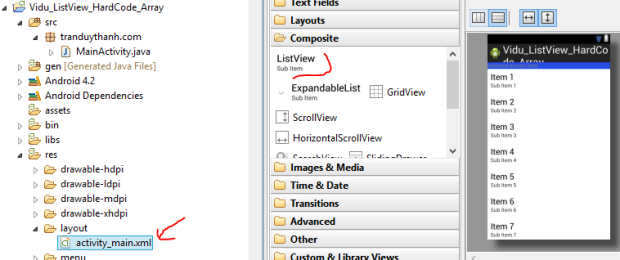
[](https://duythanhcse.files.wordpress.com/2013/04/13_lv_01.png)

– Giao diện trên có 2 control:

+**ListView** : dùng để hiển thị mảng dữ liệu

+**TextView** có màu xanh lục: Dùng để hiển thị vị trí và giá trị của phần tử được chọn trong ListView

– Bạn tạo một Android Project tên là : **Vidu\_ListView\_HardCode\_Array, chọn**layout phù hợp và kéo thả các control vào giao diện:

[](https://duythanhcse.files.wordpress.com/2013/04/13_lv_1.png)

– Dưới đây là nội dung của **activity\_main.xml**:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19 | <LinearLayout xmlns:android="<http://schemas.android.com/apk/res/android>"   xmlns:tools="<http://schemas.android.com/tools>"   android:id="@+id/LinearLayout1"   android:layout\_width="match\_parent"   android:layout\_height="match\_parent"   android:orientation="vertical"   tools:context=".MainActivity" >   <TextView   android:id="@+id/txtselection"   android:layout\_width="match\_parent"   android:layout\_height="wrap\_content"   android:background="#dd0230dd"   android:hint="Selected person here" />   <ListView   android:id="@+id/lvperson"   android:layout\_width="match\_parent"   android:layout\_height="wrap\_content" >   </ListView>  </LinearLayout> |

– Đặt **id** cho *Listview* là **lvperson (nhìn dòng lệnh 15 ở trên)**, bạn có thể định dạng thêm một số đặc tính khác nhưng trong bài tập này thì chưa cần thiết, chỉ cần hiển thị được dữ liệu lên giao diện là đã đạt yêu cầu.

– Bây giờ bạn mở **MainActivity.java** lên để viết code:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38 | package tranduythanh.com;    import android.os.Bundle;  import android.app.Activity;  import android.view.View;  import android.widget.AdapterView;  import android.widget.ArrayAdapter;  import android.widget.ListView;  import android.widget.TextView;    public class MainActivity extends Activity {    protected void onCreate(Bundle savedInstanceState) {   super.onCreate(savedInstanceState);   setContentView(R.layout.activity\_main);   //1. Khởi tạo dữ liệu cho mảng arr (còn gọi là data source)   final String arr[]={"Teo","Ty","Bin","Bo"};   //2. Lấy đối tượng Listview dựa vào id   ListView lv=(ListView) findViewById(R.id.lvperson);   //3. Gán Data source vào ArrayAdapter   ArrayAdapter<String>adapter=new ArrayAdapter<String>   (this, android.R.layout.simple\_list\_item\_1, arr);   //4. Đưa Data source vào ListView   lv.setAdapter(adapter);   final TextView txt=(TextView) findViewById(R.id.txtselection);   //5. Thiết lập sự kiện cho Listview, khi chọn phần tử nào thì hiển thị lên TextView   lv.setOnItemClickListener(   new AdapterView.OnItemClickListener() {   public void onItemClick(AdapterView<?> arg0,   View arg1,   int arg2,   long arg3) {   //đối số arg2 là vị trí phần tử trong Data Source (arr)   txt.setText("position :"+arg2+" ; value ="+arr[arg2]);   }   });   }  } |

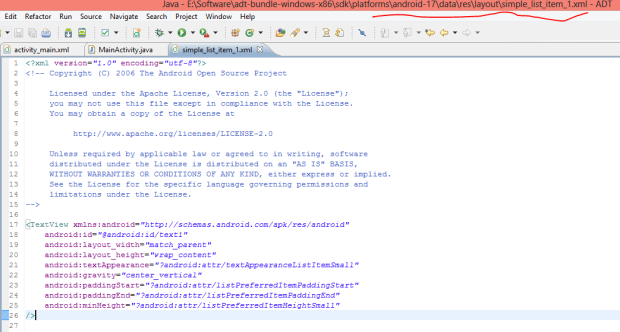
– Tôi đã giải thích từng dòng lệnh ở bên trong code, giờ Tôi giải thích thêm về ArrayAdapter, bạn nhìn vào dòng lệnh 21.

ArrayAdapter<**String**>adapter=new ArrayAdapter<**String**>(**this**, android.R.layout.simple\_list\_item\_1, **arr**);

– Dữ liệu từ Data source (arr) sẽ được gắn vào  ArrayAdapter, ArrayAdapter sẽ gắn vào ListView.

– Bạn nhìn vào đối số đầu tiên của constructor ArrayAdapter : **this**, chính là context của Activity hiện tại, bạn có thể viết **MainActivity.this** (nếu bạn viết như thế này thì ở bất kỳ vị trí nào nó cũng hiểu là context của MainActivity, do đó các bạn nên viết như thế này để bạn có thể copy paste nó tới bất kỳ vị trí nào thì nó cũng hiểu)

– Đối số thứ 2 **android.R.layout.simple\_list\_item\_1** : bạn để ý  android Tôi tô màu xanh, đây chính là layout Listview mà được Android xây dựng sẵn, các bài tập kế tiếp ta sẽ tự xây dựng mà không sử dụng cái có sẵn này. Như vậy thì **simple\_list\_item\_1**lưu ở đâu? và bên trong nó như thế nào?. Nó được lưu trong SDK/platforms/android-api (x)/data/res/layout/**simple\_list\_item\_1.xml.** Bạn có thể xem nội dung và vị trí của layout này một cách nhanh chóng bằng  đè phím Ctrl + click chuột vào dòng lệnh này, bạn sẽ thấy như bên dưới:

[](https://duythanhcse.files.wordpress.com/2013/04/13_lv_2.png)– Đối số thứ 3: chính là arr (data source), bạn có thể truyền vào ArrayList.

– Nhìn vào dòng lệnh 27 chỗ gán sự kiện cho ListView (bạn nhớ là chỉ cần gõ một vài ký tự đầu rồi nhấn Ctrl+ Space Bar thì các lệnh đằng sau sẽ tự động xuất hiện ra cho bạn):

+ Ta có interface **AdapterView.OnItemClickListener**, nó dùng để thiết lập sự kiện cho ListView, interface này có 1 phương thức trừu tượng là **onItemClick** nên ta override nó về xử lý trong này. Bạn cũng nhớ là chỗ này không có gõ bằng tay mà chỉ cần nhấn tổ hợp phím **Ctrl + 1** chọn add unimplement method là nó tự xuất hiện. Ngoài ra nó còn nhiều sự kiện khác các bạn tự tìm hiểu thêm.

Bạn có thể tải code đầy đủ ở đây:<http://www.mediafire.com/?uwy4lp0e1jt0mik>

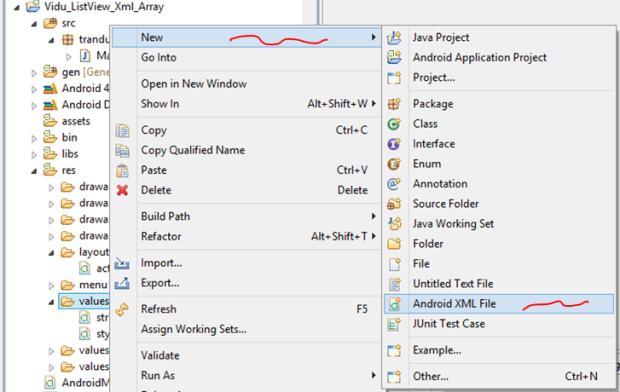
**2) Trường  hợp 2: Sử dụng ListView với mảng dữ liệu được lưu trong Xml:**

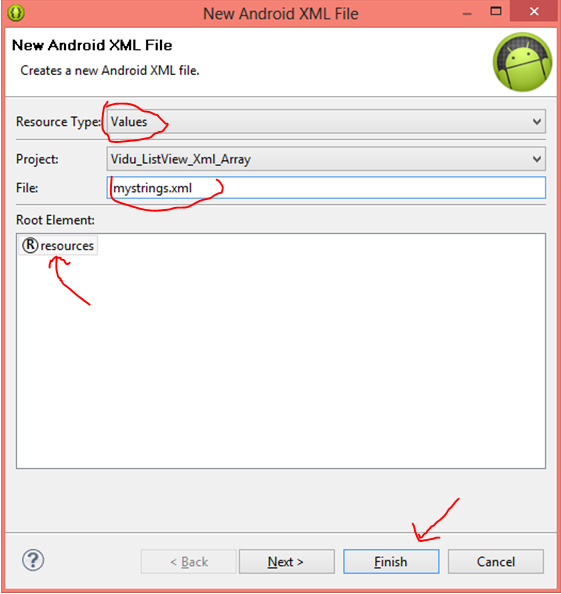
– Giao diện và xử lý sự kiện trong trường hợp này là y xì trường hợp 1. Chỉ khác ở chỗ là dữ liệu sẽ được load từ XML, nên Tôi chỉ hướng dẫn **cách tạo String – Array** trong XML và  **cách load String-Array** trong coding như thế nào.

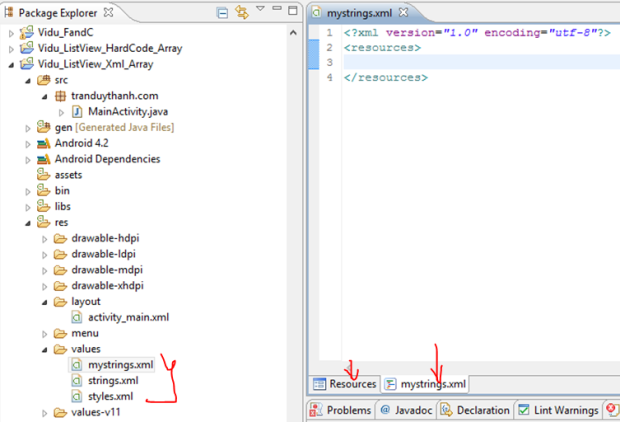
– Bạn tạo một Android Project tên là: **Vidu\_ListView\_Xml\_Array**

– **Để tạo String – Array trong XML** bạn làm như sau:

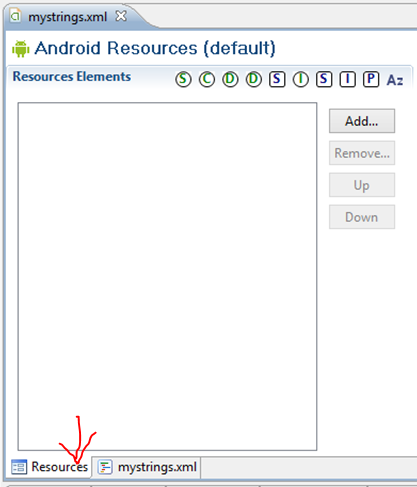
**Bước 1:** Bấm chuột phải vào thư mục **values** của Project/ chọn New/Android XML File:

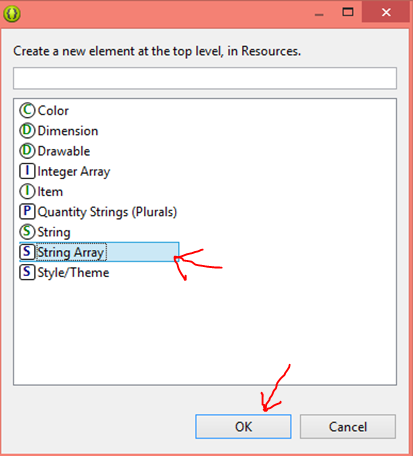
[](https://duythanhcse.files.wordpress.com/2013/04/13_lv_3.png)– **Bước 2:** Màn hình New Android XML hiển thị lên, bạn chọn thông số giống như hình bên dưới, đặt tên tập tin là **mystrings.xml** rồi nhấn nút **Finish**:

[](https://duythanhcse.files.wordpress.com/2013/04/13_lv_41.png)Xem kết quả khi bấm nút **Finish và** quan sát cho nhận xét:

[](https://duythanhcse.files.wordpress.com/2013/04/13_lv_5.png)Chú ý là ta cũng có thể thêm dữ liệu vào tập tin **strings.xml**, nhưng đây là ý đồ của Tôi, Tôi muốn hướng dẫn các bạn cách tạo 1 tập tin XML mới và thêm dữ liệu vào tập tin mới này luôn.

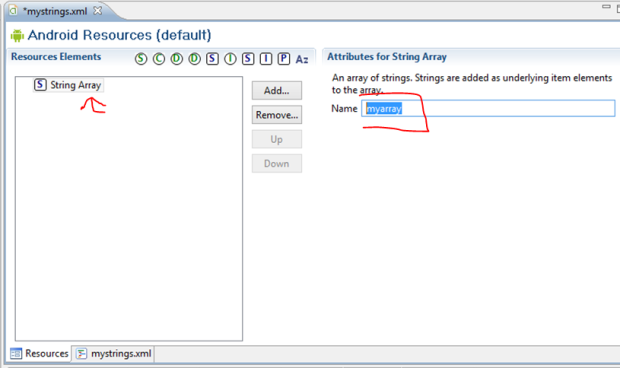
**Bước 3:** chọn tab Resource ở hình trên:

[](https://duythanhcse.files.wordpress.com/2013/04/13_lv_6.png)– **Bước 4**: Bấm nút “Add…” ở màn hình trên để thêm **String-Array**:

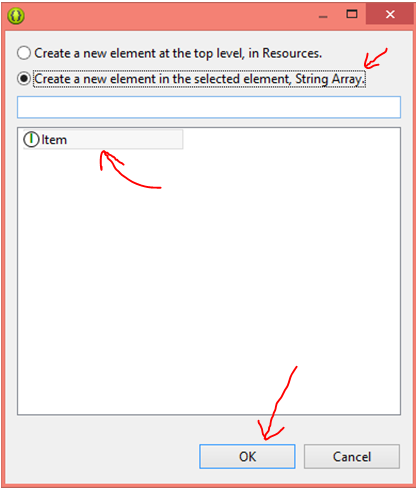
[](https://duythanhcse.files.wordpress.com/2013/04/13_lv_7.png)

Ở màn hình trên bạn chọn String Array rồi nhấn OK:

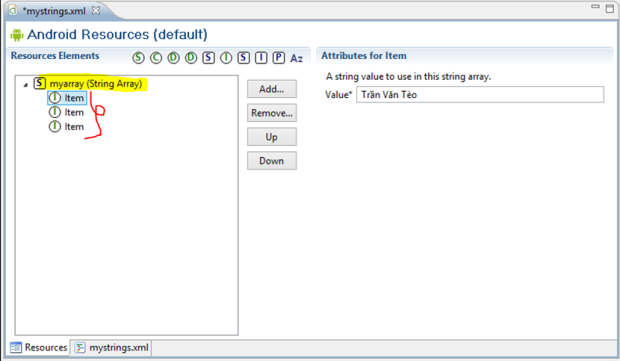
– **Bước 5**: Đặt tên cho String Array, sau khi nhấn OK thì màn hình bên dưới hiển thị ra, bạn đặt tên String Array này là **myarray rồi nhấn ctrl+s để lưu**:

[](https://duythanhcse.files.wordpress.com/2013/04/13_lv_8.png)

– **Bước 6**: Thêm các phần tử vào String Array, tiếp tục bấm nút **Add** ở màn hình bên trên:

[](https://duythanhcse.files.wordpress.com/2013/04/13_lv_9.png)-Sau khi nhấn OK thì nó sẽ cho phép bạn nhập giá trị cho phần tử.

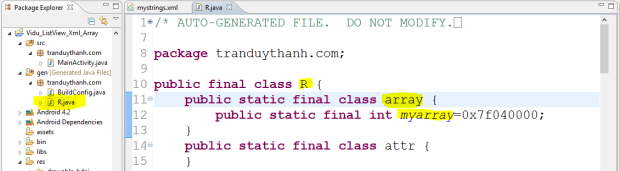
– Bạn cứ lặp liên tục thao tác ở bước 6 này, thêm bao nhiêu phần tử thì click từng đó lần Add, ở đây là Tôi thêm 3 phần tử, bạn xem hình:

[](https://duythanhcse.files.wordpress.com/2013/04/13_lv_10.png)

– Xem nội dung XML:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8 | <?xml version="1.0" encoding="utf-8"?>  <resources>   <string-array name="myarray">   <item >Trần Văn Tèo</item>   <item >Nguyễn Thị Tẹt</item>   <item >Hồ Văn Hiến</item>   </string-array>  </resources> |

– Như vậy bạn đã biết cách tao 1 tập tin XML và biết cách tạo String Array cũng như tạo các phần tử nằm bên trong nó, Bây giờ trong Coding ta sẽ dựa vào **myarray** để đọc toàn bộ dữ liệu từ XML ra. Bạn cũng chú ý là myarray khi bạn tạo ra ở trên thì nó cũng được tạo ra trong **gen**:

[](https://duythanhcse.files.wordpress.com/2013/04/13_lv_11.png)

– Như bạn thấy đó, trong coding ta sẽ dựa vào đây để lấy ra: **R.array.myarray**.

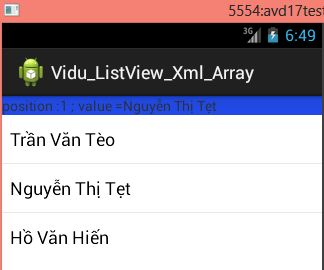
\* Mở MainActivity.java lên:

– Như Tôi đã nói là mọi thứ nó y xì như trường hợp số 1, dó đó trong trường hợp này Tôi chỉ viết dòng lệnh đọc dữ liệu từ XML đổ về một Mảng (thay vì trường hợp 1 Tôi hardcode mấy phần tử khi khai báo mảng), còn các phần khác bạn tự làm giống trường hợp 1:

**final** String arr[]=**getResources**().**getStringArray**(**R.array.myarray**);

Tức là bạn thay thế dòng lệnh thứ 17 trong trường hợp 1 của MainActivity.java bằng dòng lệnh trên.

Thực thi ứng dụng bạn sẽ được kết quả như hình bên dưới:

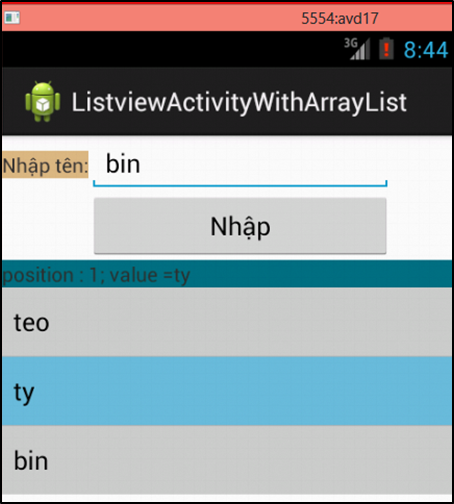
[](https://duythanhcse.files.wordpress.com/2013/04/13_lv_13.png)

– Tôi nghĩ tới đây bạn đã thật sự hiểu trường hợp 2.

– Bạn có thể tải toàn bộ coding mẫu ở đây: <http://www.mediafire.com/?168jsno9iq85qsc>

**3) Trường hợp 3: Sử dụng   ArrayList  và Listview control:**

– Trường hợp này Tôi muốn hướng dẫn các bạn cách sử dụng ArrayList để lưu trữ dữ liệu và đổ lên ListView như thế nào, bạn xem giao diện của chương trình:

[](https://duythanhcse.files.wordpress.com/2013/04/13_lv_12.png)**– Mô tả:**

+ Nhập dữ liệu và nhấn nút “Nhập” thì sẽ đưa vào ArrayList và hiển thị lên ListView

+ Nhấn vào phần tử nào thì hiển thị vị trí và giá trị của phần tử đó lên TextView

+ Nhấn thật lâu (long click ) vào phần tử nào đó trên ListView thì sẽ xóa phần tử đó.

\* Tạo Android Project tên: **Vidu\_ListView\_ArrayList**,

Xem Layout XML của ứng dụng (**activity\_main.xml**):

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49 | <RelativeLayout xmlns:android="<http://schemas.android.com/apk/res/android>"   xmlns:tools="<http://schemas.android.com/tools>"   android:layout\_width="match\_parent"   android:layout\_height="match\_parent"   tools:context=".MainActivity" >   <EditText   android:id="@+id/txtTen"   android:layout\_width="wrap\_content"   android:layout\_height="wrap\_content"   android:layout\_alignParentTop="true"   android:inputType="text"   android:layout\_toRightOf="@+id/textView1"   android:ems="10" />   <TextView   android:id="@+id/textView1"   android:layout\_width="wrap\_content"   android:layout\_height="wrap\_content"   android:layout\_alignBaseline="@+id/txtTen"   android:layout\_alignBottom="@+id/txtTen"   android:layout\_alignParentLeft="true"   android:background="#deb887"   android:text="Nhập tên:" />   <Button   android:id="@+id/btnNhap"   android:layout\_width="wrap\_content"   android:layout\_height="wrap\_content"   android:layout\_alignRight="@+id/txtTen"   android:layout\_below="@+id/txtTen"   android:layout\_toRightOf="@+id/textView1"   android:textAlignment="center"   android:text="Nhập" />    <TextView   android:id="@+id/txtselection"   android:layout\_width="match\_parent"   android:layout\_height="wrap\_content"   android:layout\_alignParentLeft="true"   android:layout\_below="@+id/btnNhap"   android:background="#007380" />    <ListView   android:id="@+id/lvperson"   android:layout\_width="match\_parent"   android:layout\_height="wrap\_content"   android:layout\_alignParentLeft="true"   android:layout\_below="@+id/txtselection"   android:background="#cccccc" >   </ListView>  </RelativeLayout> |

Xem MainActivity.java:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71 | package tranduythanh.com;    import java.util.ArrayList;  import android.os.Bundle;  import android.app.Activity;  import android.view.View;  import android.widget.AdapterView;  import android.widget.ArrayAdapter;  import android.widget.Button;  import android.widget.EditText;  import android.widget.ListView;  import android.widget.TextView;    public class MainActivity extends Activity {   EditText txtten;   TextView txtchon;   Button btn;   ListView lv;   ArrayList<String>arrList=null;   ArrayAdapter<String> adapter=null;   @Override   protected void onCreate(Bundle savedInstanceState) {   super.onCreate(savedInstanceState);   setContentView(R.layout.activity\_main);     txtten=(EditText) findViewById(R.id.txtTen);   txtchon=(TextView) findViewById(R.id.txtselection);     lv=(ListView) findViewById(R.id.lvperson);   //1. Tạo ArrayList object   arrList=new ArrayList<String>();   //2. Gán Data Source (ArrayList object) vào ArrayAdapter   adapter=new ArrayAdapter<String>   (this,   android.R.layout.simple\_list\_item\_1,   arrList);   //3. gán Adapter vào ListView   lv.setAdapter(adapter);     btn=(Button) findViewById(R.id.btnNhap);   //4. Xử lý sự kiện nhấn nút Nhập   btn.setOnClickListener(new View.OnClickListener() {   public void onClick(View arg0) {   arrList.add(txtten.getText()+"");   adapter.notifyDataSetChanged();   }   });   //5. Xử lý sự kiện chọn một phần tử trong ListView   lv.setOnItemClickListener(new AdapterView   .OnItemClickListener() {   public void onItemClick(   AdapterView<?> arg0,View arg1,   int arg2,long arg3) {     txtchon.setText("position : "+ arg2+   "; value ="+arrList.get(arg2));   }   });   //6. xử lý sự kiện Long click   lv.setOnItemLongClickListener(new AdapterView   .OnItemLongClickListener() {   @Override   public boolean onItemLongClick(AdapterView<?> arg0, View arg1,   int arg2, long arg3) {   arrList.remove(arg2);//xóa phần tử thứ arg2   adapter.notifyDataSetChanged();   return false;   }   });   }  } |

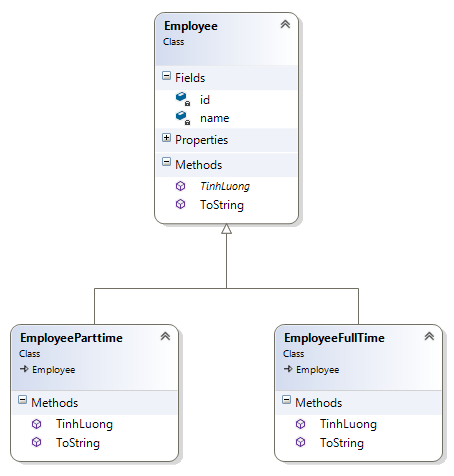
Tôi giải thích thêm về coding:  
ArrayList bạn đã được học trong môn Java 1 rồi. Ở đây Tôi nói hàm **adapter.notifyDataSetChanged()**; Bạn chú ý là ArrayList được gán vào adapter nên mọi sự thay đổi trong ArrayList thì adapter đều nhận biết được. Khi có sự thay đổi trong ArrayList bạn chỉ cần gọi **notifyDataSetChanged thì**ListView sẽ được cập nhật (bởi vì Adapter được gắn vào ListView).

– Sự kiện **setOnItemLongClickListener**, được gắn cho ListView Item, khi nhấn lâu từ 2.5 tới 3 giây thì sự kiện này sẽ sảy ra. Tương tự như **setOnItemClickListener** , đối số có tên **arg2** được dùng để xác định được vị trí của phần tử nằm trong ArrayList.

– Bạn có thể vào đây để tải coding mẫu: <http://www.mediafire.com/?64k77e8yiazar8i>

**4) Trường hợp 4: Sử dụng ArrayList và ListView nhưng từng phần tử trong ArrayList là các Object bất kỳ:**

– Tôi có một ví dụ về hiển thị danh sách nhân viên theo mô hình sau:

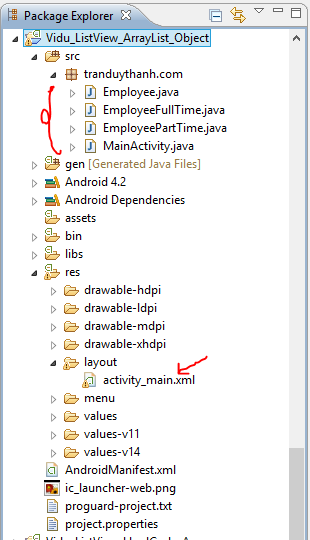
[](https://duythanhcse.files.wordpress.com/2013/04/13_lv_14.png)– Có 2 loại nhân viên : Nhân viên chính thức (EmployeeFullTime ) và nhân viên thời vụ (EmployeePartime).

– Mỗi nhân viên sẽ có cách tính lương khác nhau (tên phương thức tính lương giống nhau)

– Mỗi nhân viên có phương thức toString để xuất thông tin, Nội dung xuất khác nhau. Thêm FullTime đằng sau Id và Name đối với nhân viên chính thức. Thêm Partime đằng sau Id và Name đối với nhân viên thời vụ.

– Xem giao diện chương trình:

[](https://duythanhcse.files.wordpress.com/2013/04/13_lv_15.png)-Tạo một Android Project tên: **Vidu\_ListView\_ArrayList\_Object**, cấu trúc như bên dưới:

[](https://duythanhcse.files.wordpress.com/2013/04/13_lv_16.png)

– Layout XML (**activity\_main.xml**):

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99  100  101  102  103  104  105  106  107  108  109  110  111  112  113  114  115  116  117  118  119  120  121  122  123  124  125 | <LinearLayout xmlns:android="<http://schemas.android.com/apk/res/android>"   xmlns:tools="<http://schemas.android.com/tools>"   android:id="@+id/LinearLayout1"   android:layout\_width="match\_parent"   android:layout\_height="match\_parent"   android:orientation="vertical"   tools:context=".MainActivity" >    <TextView   android:id="@+id/textView1"   android:layout\_width="match\_parent"   android:layout\_height="wrap\_content"   android:background="#008000"   android:gravity="center"   android:text="Quản lý nhân viên"   android:textColor="#FFFFFF"   android:textSize="20sp" />    <TableLayout   android:layout\_width="match\_parent"   android:stretchColumns="\*"   android:layout\_height="wrap\_content" >    <TableRow   android:id="@+id/tableRow1"   android:layout\_width="wrap\_content"   android:layout\_height="wrap\_content" >    <TextView   android:id="@+id/textView2"   android:layout\_width="wrap\_content"   android:layout\_height="wrap\_content"   android:text="Mã NV:" />    <EditText   android:id="@+id/editMa"   android:layout\_width="wrap\_content"   android:layout\_height="wrap\_content"   android:layout\_span="2"   android:ems="10" >    <requestFocus />   </EditText>    </TableRow>    <TableRow   android:id="@+id/tableRow2"   android:layout\_width="wrap\_content"   android:layout\_height="wrap\_content" >    <TextView   android:id="@+id/textView3"   android:layout\_width="wrap\_content"   android:layout\_height="wrap\_content"   android:text="Tên NV:" />    <EditText   android:id="@+id/editTen"   android:layout\_width="wrap\_content"   android:layout\_height="wrap\_content"   android:layout\_span="2"   android:ems="10" />    </TableRow>    <TableRow   android:id="@+id/tableRow3"   android:layout\_width="wrap\_content"   android:layout\_height="wrap\_content" >    <TextView   android:id="@+id/textView4"   android:layout\_width="wrap\_content"   android:layout\_height="wrap\_content"   android:text="Loại NV:" />    <RadioGroup   android:id="@+id/radiogroud1"   android:orientation="horizontal" >    <RadioButton   android:id="@+id/radChinhthuc"   android:layout\_width="wrap\_content"   android:layout\_height="wrap\_content"   android:checked="true"   android:text="Chính thức" />    <RadioButton   android:id="@+id/radThoivu"   android:layout\_width="wrap\_content"   android:layout\_height="wrap\_content"   android:text="Thời vụ" />   </RadioGroup>    </TableRow>    <TableRow   android:id="@+id/tableRow4"   android:layout\_width="wrap\_content"   android:layout\_height="wrap\_content" >    <Button   android:id="@+id/btnnhap"   android:layout\_width="wrap\_content"   android:layout\_height="wrap\_content"   android:layout\_column="1"   android:text="Nhập NV" />    </TableRow>   </TableLayout>    <TextView   android:id="@+id/textView5"   android:layout\_width="match\_parent"   android:layout\_height="wrap\_content"   android:background="#008000" />    <ListView   android:id="@+id/lvnhanvien"   android:layout\_width="match\_parent"   android:layout\_height="wrap\_content" >   </ListView>    </LinearLayout> |

– Xem nội dung từng class:

**– Abstract class Employee:**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29 | package tranduythanh.com;    public abstract class Employee {   private String id;   private String name;     public String getId() {   return id;   }    public void setId(String id) {   this.id = id;   }    public String getName() {   return name;   }    public void setName(String name) {   this.name = name;   }    public abstract double TinhLuong();   @Override   public String toString() {   // TODO Auto-generated method stub   return this.id+" - "+this.name;   }  } |

**– class EmployeeFullTime:**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14 | package tranduythanh.com;    public class EmployeeFullTime extends Employee {    @Override   public double TinhLuong() {   return 500;   }   @Override   public String toString() {   // TODO Auto-generated method stub   return super.toString() +" -->FullTime="+TinhLuong();   }  } |

**– Class EmployeePartTime:**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15 | package tranduythanh.com;    public class EmployeePartTime extends Employee {    @Override   public double TinhLuong() {   // TODO Auto-generated method stub   return 150;   }   @Override   public String toString() {   // TODO Auto-generated method stub   return super.toString() +" -->PartTime="+TinhLuong();   }  } |

– Bạn thấy là 2 class dẫn xuất từ Employee Tôi làm đơn giản là cách tính lương khác nhau. Đối với FullTime thì lương 500, Partime lương 150. và hàm Xuất toString() cũng khác nhau 1 xí.

– Ở đây ta sẽ áp dụng tính đa hình thông qua thừa kế, chỉ cần dùng một biến có kiểu Employee, nhưng nó có thể hiểu FullTime hoặc Partime và xuất ra thông tin đúng như mình mong đợi.

**– Xem class MainActivity.java:**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76 | package tranduythanh.com;    import java.util.ArrayList;  import android.os.Bundle;  import android.app.Activity;  import android.view.View;  import android.view.View.OnClickListener;  import android.widget.ArrayAdapter;  import android.widget.Button;  import android.widget.EditText;  import android.widget.ListView;  import android.widget.RadioGroup;    public class MainActivity extends Activity {    EditText editId,editName;   Button btnNhap;   RadioGroup radgroup;   ListView lvNhanvien;   ArrayList<Employee>arrEmployee=new ArrayList<Employee>();   ArrayAdapter<Employee>adapter=null;   //Khai báo 1 employee object   Employee employee=null;     @Override   protected void onCreate(Bundle savedInstanceState) {   super.onCreate(savedInstanceState);   setContentView(R.layout.activity\_main);   editId=(EditText) findViewById(R.id.editMa);   editName=(EditText) findViewById(R.id.editTen);   btnNhap=(Button) findViewById(R.id.btnnhap);   radgroup=(RadioGroup) findViewById(R.id.radiogroud1);   lvNhanvien=(ListView) findViewById(R.id.lvnhanvien);   //đưa Data Source là các employee vào Adapter   adapter=new ArrayAdapter<Employee>(this,   android.R.layout.simple\_list\_item\_1,   arrEmployee);   //đưa adapter vào ListView   lvNhanvien.setAdapter(adapter);     btnNhap.setOnClickListener(new OnClickListener() {     @Override   public void onClick(View arg0) {   // TODO Auto-generated method stub   processNhap();   }   });   }   //Xử lý sự kiện nhập   public void processNhap()   {   //Lấy ra đúng id của Radio Button được checked   int radId=radgroup.getCheckedRadioButtonId();   String id=editId.getText()+"";   String name=editName.getText()+"";   if(radId==R.id.radChinhthuc)   {   //tạo instance là FullTime   employee=new EmployeeFullTime();   }   else   {   //Tạo instance là Partime   employee=new EmployeePartTime();   }   //FullTime hay Partime thì cũng là Employee   //nên có các hàm này là hiển nhiên   employee.setId(id);   employee.setName(name);   //Đưa employee vào ArrayList   arrEmployee.add(employee);   //Cập nhập giao diện   adapter.notifyDataSetChanged();   }  } |

– Chương trình sẽ tự động hiển thị đúng loại Employee mà ta chọn lựa trên giao diện, ở đây bạn lại được ôn tập thêm về tính đa hình trong java. Ứng với mỗi loại Employee nó sẽ tự động gọi hàm toString của loại đó.

Ví dụ: Nếu Employee đó là FullTime thì nó gọi toString của FullTime và ngược lại với PartTime cũng vậy nó sẽ lấy toString của PartTime.

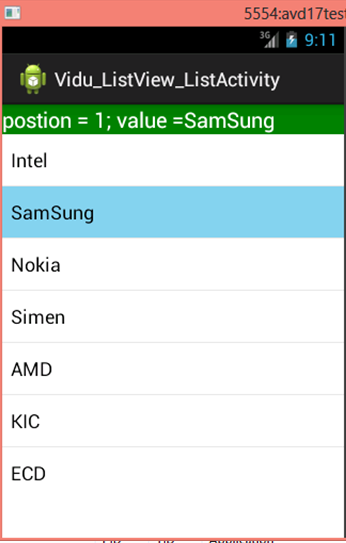
– Đến đây bạn đã hiểu tương đối sâu về ListView, bạn có thể tải coding mẫu ở đây: <http://www.mediafire.com/?o1we6e0mw8cpbba>

**5)Trường hợp 5: Sử dụng ListView nhưng dưới dạng ListActivity:**

– Thay vì kế thừa từ **Activity**, ta sẽ cho kế thừa từ **ListActivity**.

– Và dĩ nhiên cách đặt Id cho ListView cũng có sự khác biệt.

– Bạn xem giao diện bên dưới:

[](https://duythanhcse.files.wordpress.com/2013/04/13_lv_17.png)– Xem cách làm XML layout (**activity\_main.xml**):

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29 | <LinearLayout xmlns:android="<http://schemas.android.com/apk/res/android>"   xmlns:tools="<http://schemas.android.com/tools>"   android:id="@+id/LinearLayout1"   android:layout\_width="match\_parent"   android:layout\_height="match\_parent"   android:orientation="vertical"   tools:context=".MainActivity" >    <TextView   android:id="@+id/selection"   android:layout\_width="match\_parent"   android:layout\_height="wrap\_content"   android:background="#008000"   android:textColor="#FFFFFF"   android:textSize="20sp" />    <ListView   android:id="@android:id/list"   android:layout\_width="match\_parent"   android:layout\_height="wrap\_content" >   </ListView>    <TextView   android:id="@android:id/empty"   android:layout\_width="wrap\_content"   android:layout\_height="wrap\_content"   android:text="Không có gì cả" />    </LinearLayout> |

– Bạn nhìn vào dòng lệnh thứ 18:

**android:id=”@android:id/list“, bạn viết y xì như thế này.**Bởi vì nó là id được định nghĩa sẵn bên trong Android.

– Tiếp tục nhìn vào dòng lệnh 24:

**android:id=”@android:id/empty“** , cũng là có sẵn của Android. Nó có tác dụng tự động thông báo khi ListView của bạn không có bất kỳ một phần tử nào cả.

**– Xem class MainActivity.java:**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39 | package tranduythanh.com;    import android.os.Bundle;  import android.app.ListActivity;  import android.view.View;  import android.widget.ArrayAdapter;  import android.widget.ListView;  import android.widget.TextView;    public class MainActivity extends ListActivity {    TextView selection;   String arr[]={"Intel","SamSung",   "Nokia","Simen","AMD",   "KIC","ECD"};   ArrayAdapter<String >adapter=null;   @Override   protected void onCreate(Bundle savedInstanceState) {   super.onCreate(savedInstanceState);   setContentView(R.layout.activity\_main);   //Thiết lập Data Source cho Adapter   adapter=new ArrayAdapter<String>   (this,   android.R.layout.simple\_list\_item\_1,   arr);   //Gán Adapter vào ListView   //Nhớ là phải đặt id cho ListView theo đúng quy tắc   setListAdapter(adapter);     selection=(TextView) findViewById(R.id.selection);   }   @Override   protected void onListItemClick(ListView l, View v, int position, long id) {   // TODO Auto-generated method stub   super.onListItemClick(l, v, position, id);   String txt="postion = "+position +"; value ="+arr[position];   selection.setText(txt);   }  } |

– Bạn nhìn vào dòng lệnh số 10: bạn thấy đấy, Tôi cho kế thừa từ **ListActivity** chứ không phải **Activity**. (Android đã viết class ListActivity kế thừa từ Activity rồi). Tức là ListActivity cũng chính là Activity.

– Bạn nhìn vào dòng lệnh số 28:

**setListAdapter(adapter);**

Ở đây ta hoàn toàn không “Lôi” control ListView ra. Mà ta chỉ cần gọi hàm **setListAdapter thì**ListView cũng tự động cập nhập dữ liệu. Bạn phải làm chính xác 2 nơi thì mới được như vậy:

1. Kết thừa từ **ListActivity**,

2. đặt id cho ListView theo đúng quy tắc **android:id=”@android:id/list”**

– Bạn load coding đầy đủ ở đây: <http://www.mediafire.com/?mjxgs1yv0mvujuz>

\*\*\* Như vậy bạn đã thực hành xong 5 trường hợp mà Tôi đã nêu. Tôi hi vọng các bạn hãy làm lại nhiều lần để có thể hiểu sâu hơn về nó.

Bài tập kế tiếp bạn sẽ được thực hành về cách Custom layout lại ListView theo ý thích của bạn.