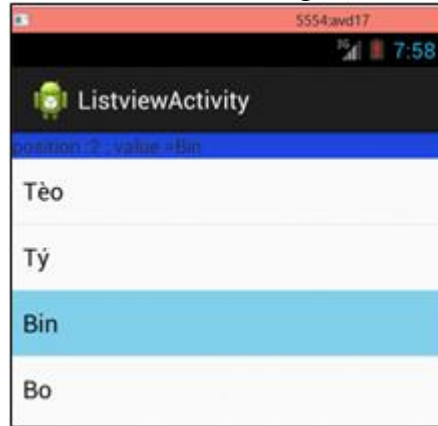


Bài tập 13: Thực hành về ListView trong Android

5 trường hợp tương ứng với 5 cách đưa dữ liệu vào listView

1) Trường hợp 1: - Sử dụng ListView control với mảng dữ liệu định sẵn minh họa:

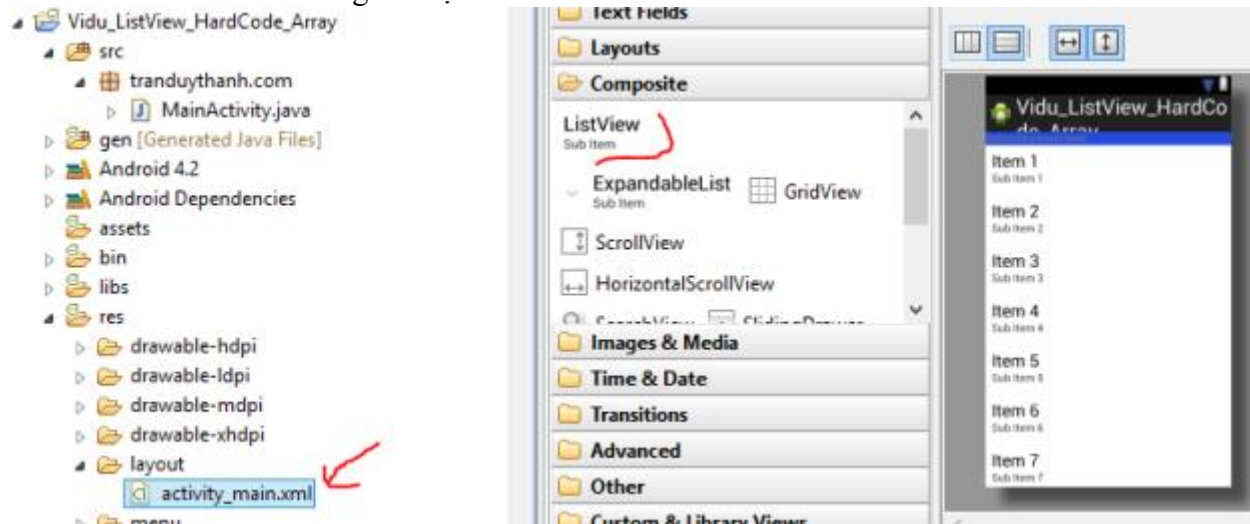


- Giao diện trên có 2 control:

+ **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:



Dưới đây là nội dung của **activity_main.xml**:

-Đặt id cho ListView là **lvperson**, - Bây giờ bạn mở **MainActivity.java** lên để viết code:

```
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
```

```

thì lên TextView
    lv.setOnItemClickListener( new AdapterView.OnItemClickListener() {
        public void onItemClick(AdapterView<?> arg0,
            View arg1,
            Int arg2,
            Long arg3)
        {
            //đổi số arg2 là vị trí phần tử trong Data Source (arr)
            txt.setText("position :"+arg2+" ; value =" +arr[arg2]);
        }
    });
}
}

```

- Giải thích thêm về ArrayAdapter, bạn nhìn vào dòng lệnh:

```

ArrayAdapter<String>adapter=newArrayAdapter<String>(this,
    android.R.layout.simple_list_item_1, arr);

```

- Dữ liệu từ Data source (**arr**) sẽ được gán vào ArrayAdapter, ArrayAdapter sẽ gán vào ListView.

- Đối số đầu tiên của constructor ArrayAdapter : **this**, chính là context của Activity hiện tại, Nếu viết **MainActivity.this** thì ở bất kỳ vị trí nào cũng được hiểu là context của MainActivity, do đó đây là cách viết tường minh.

- Đối số thứ 2 **android.R.layout.simple_list_item_1** : Đây chính là layout Listview mà được Android xây dựng sẵn. Như vậy thì **simple_list_item_1** lưu ở đâu? và bên trong nó như thế nào?.

- o Nó được lưu trong SDK/platforms/android-api (x) /data/res/layout/**simple_list_item_1.xml**. Chi tiết xem nội dung và vị trí của layout này một cách nhanh chóng bằng Ctrl + click chuột vào dòng lệnh này:

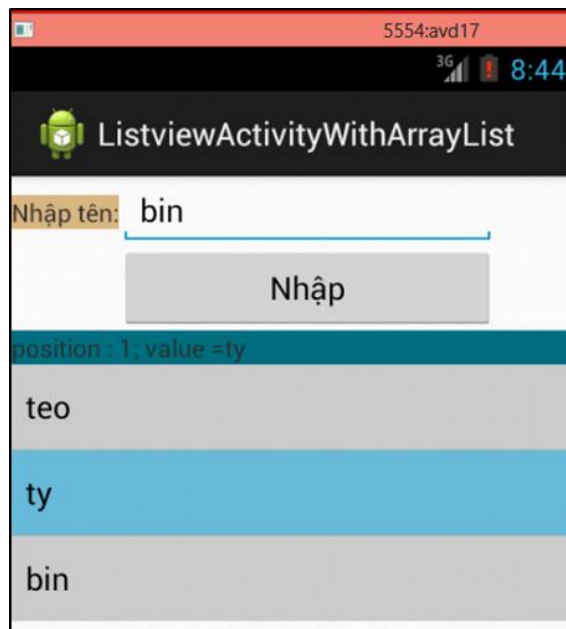
- Đối số thứ 3: chính là **arr** (data source), bạn có thể truyền vào ArrayList hay mảng.

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

+ Ta có interface **AdapterView.OnItemClickListener**, dùng để thiết lập sự kiện cho ListView. Interface này có 1 phương thức trừu tượng là **onItemClick** nên người dùng có thể ghi đè (override) để xử lý. Lưu ý là người viết code không có gõ bằng tay mà chỉ cần nhấn tổ hợp phím **Ctrl + 1** chọn **add unimplement method** sẽ xuất hiện các phương thức này.

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

- Trường hợp sẽ sử dụng **ArrayList** để lưu trữ dữ liệu và đổ lên ListView :



- 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.

Thực hiện:

1. Tạo Android Project tên: **Vidu_ListView_ArrayList**,
 - Xem Layout XML của ứng dụng (**activity_main.xml**):
 - Xem MainActivity.java:

```

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=newArrayAdapter<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() {
            publicvoidonClick(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, intarg2,longarg3) {
                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,
            intarg2, longarg3) {
                arrList.remove(arg2);//xóa phần tử thứ arg2
                adapter.notifyDataSetChanged();
                return false;
            }
        });
    }
}

```

- Hàm **adapter.notifyDataSetChanged();** Do 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 gán vào ListView).
- Sự kiện **setOnItemLongClickListener**, được gán cho ListView Item, khi nhấn lâu từ 2.5 tới 3 giây thì sự kiện này sẽ xảy ra. Tương tự như **setOnItemClickListener**, đối số có tên **arg2** được dùng để xác định được vị trí của phần tử nằm trong ArrayList.

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

- Mở rộng bài 12: Hoàn thành bài 12 với danh sách hàng được lưu trong listView như sau

The screenshot displays a mobile application interface for calculating online book purchase totals. The title bar reads 'Bai13_TongHop_HoaDonBanSach'. Below the title, there's a section titled 'Chương trình tính tiền bán sách online' with a subtitle 'Thông tin hóa đơn:'. The form contains the following elements:

- 'Tên Khách Hàng:' followed by the text 'Vũ Thị Dương'.
- 'Số lượng sách:' followed by the value '113'.
- A checked checkbox labeled 'Khách hàng là VIP'.
- 'Thành tiền:' followed by the value '2034000'.
- Three buttons: 'TÍNH TIỀN', 'NHẬP TIẾP', and 'THỐNG KÊ'.
- A list view below the buttons showing 'Item 1' and 'Sub Item 1'.
- A red power button at the bottom right.

Yêu cầu:

1. Chọn nút tính Tiền thì thông tin khách trên giao diện được bổ sung vào listView
2. Nếu chọn lâu 1 dòng trên listView thì xóa thông tin dòng đó trong listView
3. Nếu chọn 1 dòng trong listview thì thông tin dòng đó được xuất hiện ngược lại giao diện trên các nút nhập liệu phía trên