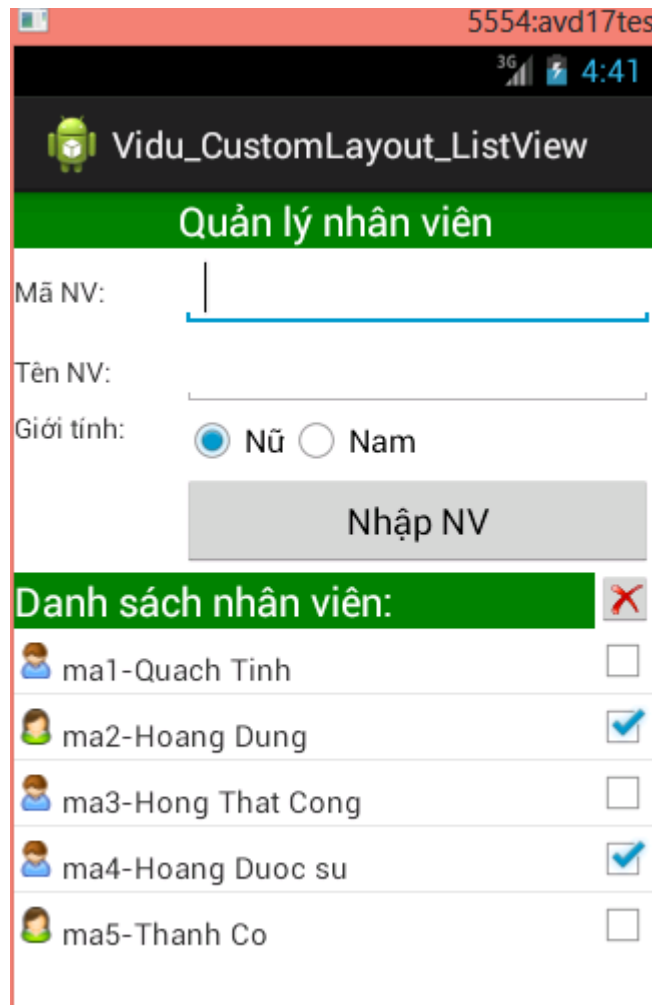


Bài tập 14: Thực hành về Custom Layout cho ListView trong Android

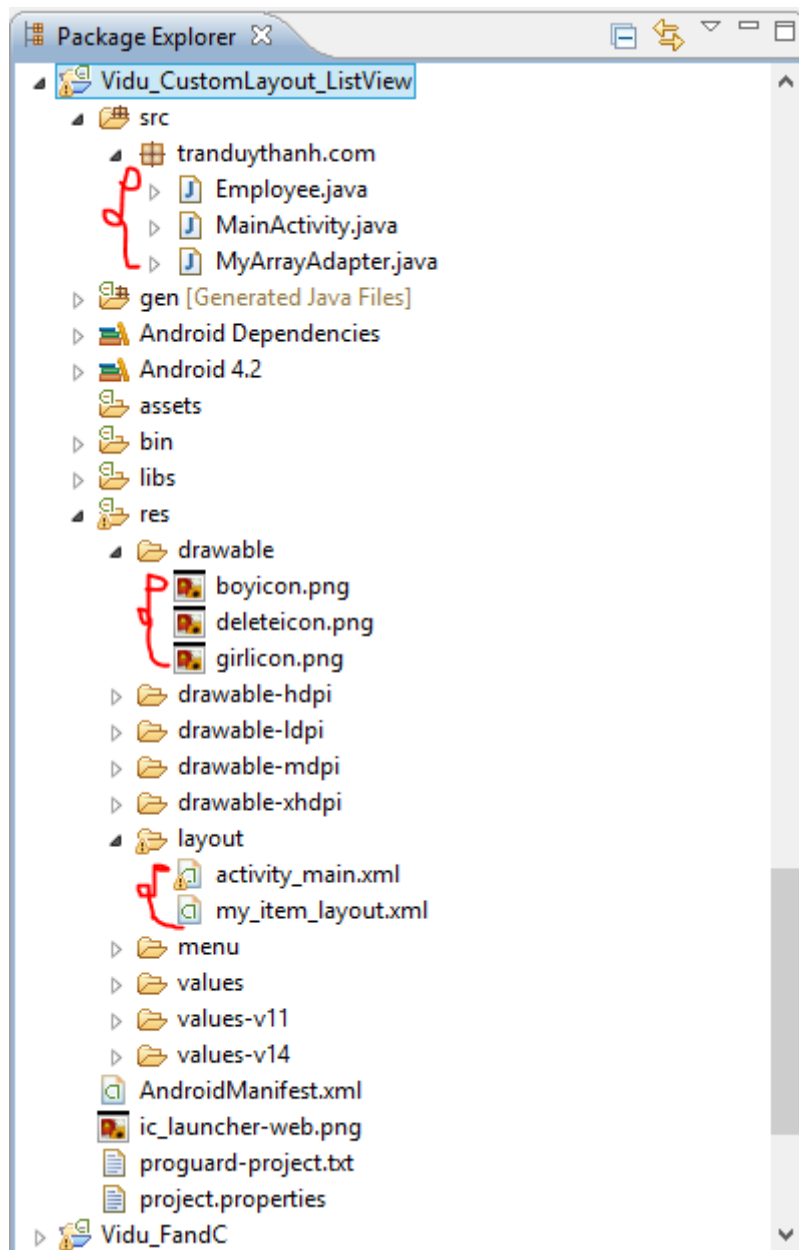
Ở [bài tập 13](#) bạn đã được thực hành với ListView control. Trong bài tập này bạn sẽ học cách Custom lại layout cho ListView trong ứng dụng Android của bạn. Đây là bài tập rất quan trọng.

Bài toán: Quản lý nhân viên với giao diện bên dưới đây:



- Quan sát phần danh sách nhân viên bên dưới là Custom Layout.
- Mỗi dòng trong ListView sẽ có 3 đối tượng: **ImageView**, **TextView** và **Checkbox**.
- Khi nhập nhân viên nếu người sử dụng chọn Nữ thì sẽ hiển thị hình là con gái, nếu chọn Nam thì hiển thị hình là con trai (bạn nhìn danh sách hình nhỏ nhỏ 16×16 ở ListView).
- Mã và tên của nhân viên sẽ được hiển thị vào TextView
- Checkbox cho phép người sử dụng checked (nhằm đánh dấu những nhân viên muốn xóa, ở đây cho phép xóa nhiều nhân viên)
- ImageButton có hình màu Chéo đỏ, nó dùng để xóa tất cả các nhân viên được Checked trong ListView, sau khi xóa thành công thì phải cập nhật lại ListView.

- Để làm được điều trên thì ta sẽ kế thừa từ **ArrayAdapter** và **override** phương thức **getView**, cụ thể:
- Bạn xem Cấu trúc chương trình quản lý nhân viên:



- Tôi tạo thêm thư mục **drawable** và kéo thả 3 icon được sử dụng vào (sv cũng tạo thư mục tên như vậy). Nhớ là tên hình phải viết liền và chữ thường đầu tiên.
- Trong thư mục layout: Tạo **my_item_layout.xml** dùng để Custom lại ListView, dưới đây là cấu trúc XML của nó:

Chú khi thi thiết kế giao diện của Checkbox có đặt 1 thuộc tính quan trọng như sau:

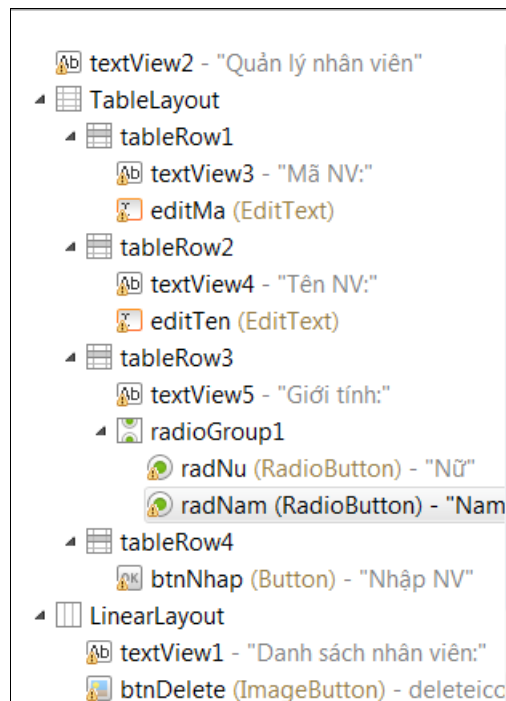
```
<CheckBox
    android:id="@+id/chkitem"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
```

android:focusable="false"//đặt để chọn được nhiều dòng và các dòng hiển thị được trên các ô điều khiển

```
/>
```

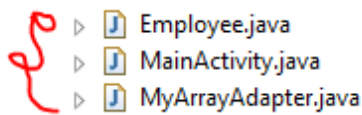
- Ta sẽ dựa vào các id trong này để xử lý trong hàm getView của class mà ta kế thừa từ ArrayAdapter (các id trên là **imgitem** đại diện cho hình là Nữ hay Nam, **txtitem** dùng để hiển thị mã và tên nhân viên, **chkitem** dùng để xử lý Checked)

- Bạn xem **activity_main.xml**:



- Layout main này chính là giao diện chính của ứng dụng.

- Dưới đây là các class hỗ trợ xử lý nghiệp vụ:



- Class **Employee** dùng để lưu trữ thông tin nhân viên: Mã nhân viên, tên nhân viên, giới tính

- Class **MyArrayAdapter** kế thừa từ **ArrayAdapter**, mục đích của nó là giúp chúng ta Custom lại layout cho ListView.

- Cuối cùng **MainActivity**. Chi tiết từng class như sau:

1) class Employee:

```
PublicclassEmployee {
    Private String id;
    Private String name;
    Private boolean gender;
    Public String getId() { return id; }
    // cài đặt các hàm get/set cho các thuộc tính - sv tự làm
    @Override
    Public String toString() {
        Return this.id+"-"+this.name;
    }
}
```

2) class MyArrayAdapter:

```
Public class MyArrayAdapter extends ArrayAdapter<Employee>{
    Activity context=null;
    ArrayList<Employee>myArray=null;
    IntlayoutId;
    /**
     * Constructor này dùng để khởi tạo các giá trị
     * từ MainActivity truyền vào
     * @param context : là Activity từ Main
     * @param layoutId: Là layout custom do ta tạo
     (my_item_layout.xml)
     * @param arr : Danh sách nhân viên truyền từ Main
     */
    Public MyArrayAdapter(Activity
        context, intlayoutId, ArrayList<Employee>arr)
    {
        super(context, layoutId, arr);
        this.context=context;
        this.layoutId=layoutId;
        this.myArray=arr;
    }
    /**
     * hàm dùng để custom layout, ta phải override lại hàm này
     * từ MainActivity truyền vào
     * @param position : là vị trí của phần tử trong danh sách nhân
viên
     * @param convertView: convertView, dùng nó để xử lý Item
     * @param parent : Danh sách nhân viên truyền từ Main
     * @return View: trả về chính convertView
     */
    Public View getView(intposition, View convertView,ViewGroup
parent) {
        LayoutInflater inflater= context.getLayoutInflater();
        convertView=inflater.inflate(layoutId, null);

        //- LayoutInflater là một service của android cho phép chúng ta render (định nghĩa và //hiển thị
lúc runtime) một view dc định nghĩa từ mã XML xác định trước, nó được dùng //trong trường hợp
bạn phải dùng nhiều hơn một kiểu layout trong cùng một module //logic

        if(myArray.size()>0&& position>=0)
        {
```

```

//dòng lệnh lấy TextView ra để hiển thị Mã và tên lên
FinalTextView txtdisplay=(TextView)
convertView.findViewById(R.id.txtitem);
//lấy ra nhân viên thứ position
FinalEmployee emp=myArray.get(position);
//đưa thông tin lên TextView
//emp.toString() sẽ trả về Id và Name
txtdisplay.setText(emp.toString());
//lấy ImageView ra để thiết lập hình ảnh cho đúng
FinalImageView imgitem=(ImageView)
convertView.findViewById(R.id.imgitem);
//nếu là Nữ thì lấy hình con gái
if(emp.isGender())
    imgitem.setImageResource(R.drawable.girlicon);
else//nếu là Nam thì lấy hình con trai
    imgitem.setImageResource(R.drawable.boyicon );
}

//Vì View là Object là dạng tham chiếu đối tượng, nên
//mọi sự thay đổi của các object bên trong convertView
//thì cũng biết sự thay đổi đó

Return    convertView;//trả về View này, tức là trả luôn
//về các thông số mới mà ta vừa thay đổi
}
}

```

- Đây là class quan trọng nhất, mới nhất; dùng để custom layout.

3) class MainActivity:

```

public class MainActivity extends Activity {
ArrayList<Employee> arrEmployee=new ArrayList<Employee>();
//Sử dụng MyArrayAdapter thay thì ArrayAdapter
MyArrayAdapter adapter=null;
ListView lvNhanvien=null;
Button btnNhap;
ImageButton btnRemoveAll;
EditText editMa,editTen;
RadioGroup genderGroup;
@Override
ProtectedvoidonCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
}
}

```

```

        btnNhap=(Button) findViewById(R.id.btnNhap);
        btnRemoveAll=(ImageButton)
        findViewById(R.id.btndelete);
        editMa=(EditText) findViewById(R.id.editMa);
        editTen=(EditText) findViewById(R.id.editTen);
        genderGroup=(RadioGroup)
        findViewById(R.id.radioGroup1);

```

```

        lvNhanvien=(ListView) findViewById(R.id.lvnhanvien);
        arrEmployee=new ArrayList<Employee>();

```

```

        //Khởi tạo đối tượng adapter và gán Data source
        adapter=new MyArrayAdapter(
        this,
        R.layout.my_item_layout,// lấy custom layout
        arrEmployee/*thiết lập data source*/);
        lvNhanvien.setAdapter(adapter);//gán Adapter vào
        Lisview

```

```

btnNhap.setOnClickListener(new OnClickListener() {
    @Override
    public void onClick(View arg0) {
        // TODO Auto-generated method stub
        xulyNhap();
    }
});
btnRemoveAll.setOnClickListener(new OnClickListener() {
    @Override
    public void onClick(View arg0) {
        xulyXoa();
    }
});

```

```

lvNhanVien.setOnItemClickListener(new
    AdapterView.OnItemClickListener() {
        @Override
        public void onItemClick(AdapterView<?> parent, View view, int
            position, long id) {
            xuLyDongChon(position);
        }
});

```

```

    }
public void xuLyDongChon(int    viTri)
    {
        //Lấy dữ liệu lên TextBox ở trên
        Employee myEmpSelected=dsEmployees.get(viTri);
        editMa.setText(myEmpSelected.getMaNV() + "");
    }

```

```

editTen.setText(myEmpSelected.getTenNV() + "");
if (myEmpSelected.isGioiTinh())
{
radNu.setSelected(true);
radNam.setSelected(false);
}
else {
radNam.setSelected(true);
radNu.setSelected(false);
}

View v=lvNhanVien.getChildAt(viTri);
//doi mau nen;

//lay checkbox ra kiểm tra
CheckBox ckDelete=(CheckBox)
v.findViewById(R.id.chkItem);
ckDelete.setChecked(!ckDelete.isChecked());
}

//gọi hàm xử lý nhập thông tin nhân viên
Public void xulyNhap()
{
String ma=editMa.getText()+"";
String ten=editTen.getText()+"";
booleangioitinh=false;//Nam =false
if(genderGroup.getCheckedRadioButtonId()==R.id.radNu)
gioitinh=true;

//Tạo một employee
Employee emp=newEmployee();
emp.setId(ma);
emp.setName(ten);
emp.setGender(gioitinh);

//Đưa vào danh sách
arrEmployee.add(emp);

//gọi hàm cập nhật giao diện
adapter.notifyDataSetChanged();

//Sau khi update thì xóa trắng dữ liệu và cho editma focus
editMa.setText("");
editTen.setText("");
editMa.requestFocus();
}

//hàm xử lý xóa
Public void xulyXoa()

```

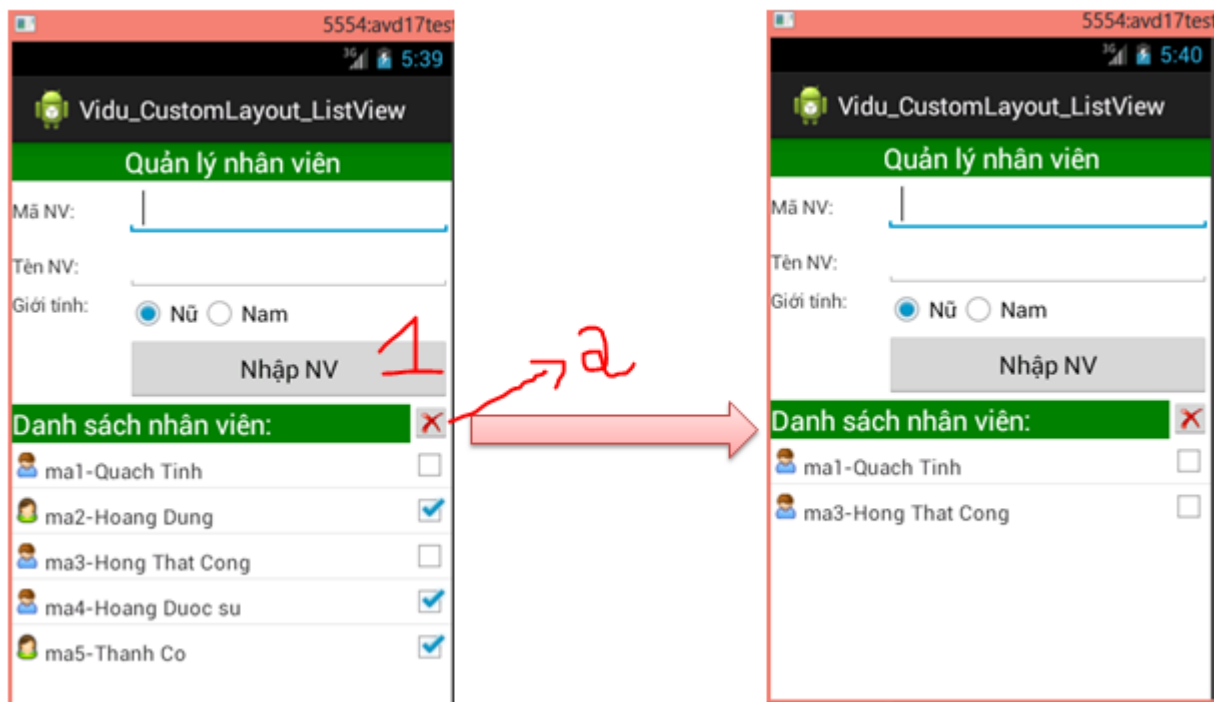
```

{
    //ta nên đi ngược danh sách, kiểm tra phần tử nào checked
    //thì xóa đúng vị trí đó ra khỏi arrEmployee

    for(int i=lvNhanvien.getChildCount()-1;i>=0;i--)
    {
        //lấy ra dòng thứ i trong ListView
        //Dòng thứ i sẽ có 3 phần tử: ImageView, TextView,
        //Checkbox
        View v=lvNhanvien.getChildAt(i);
        //Ta chỉ lấy CheckBox ra kiểm tra
        CheckBox chk=(CheckBox) v.findViewById(R.id.chkitem);
        //Nếu nó Checked thì xóa ra khỏi arrEmployee
        if(chk.isChecked())
        {
            //xóa phần tử thứ i ra khỏi danh sách
            arrEmployee.remove(i);
        }
    }
    //Sau khi xóa xong thì gọi update giao diện
    adapter.notifyDataSetChanged();
}
}

```

- Bây giờ thực hiện chương trình và nhập một số nhân viên, rồi checked rồi nhấn xóa:



- Bạn nên làm tốt bài này vì nó rất quan trọng và hay.