

Bài thực hành số 5

Mục tiêu:

- Xây dựng ứng dụng cơ bản
 - Thiết kế giao diện với layout và các điều khiển View
 - Sử dụng Intent để chuyển dữ liệu giữa các Activity
 - Xây dựng menu cho ứng dụng

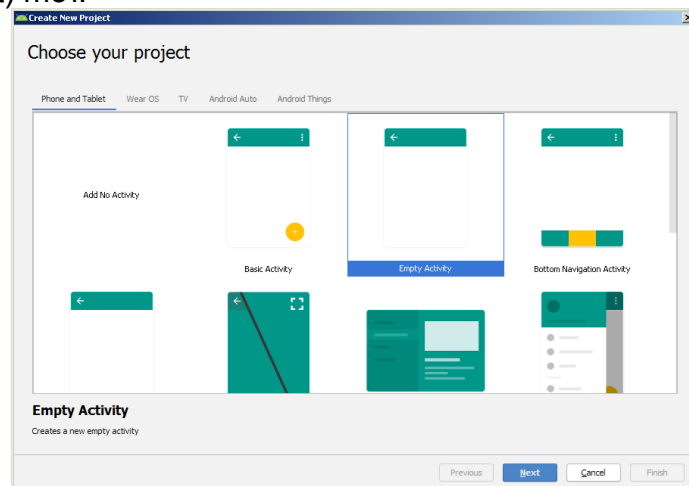
Bài 1. Xây dựng ứng dụng quản lý thông tin sinh viên đơn giản

Mô tả ứng dụng:

- Ứng dụng quản lý sinh viên cho phép quản lý các thông tin về lớp học của sinh viên, quản lý thông tin danh sách sinh viên trong lớp và thiết lập cán bộ lớp (lớp trưởng, lớp phó).

Bước 1: Tạo một dự án mới

- Khởi chạy Android Studio, xuất hiện màn hình “Welcome to Android Studio” → chọn **Start a new Android Studio project** để tạo dự án Android Studio (Android Studio project) mới:



- Chọn kiểu dự án là **Phone and Tablet** → **Empty Activity**, nhấn **Next**.
- Nhập các thông tin

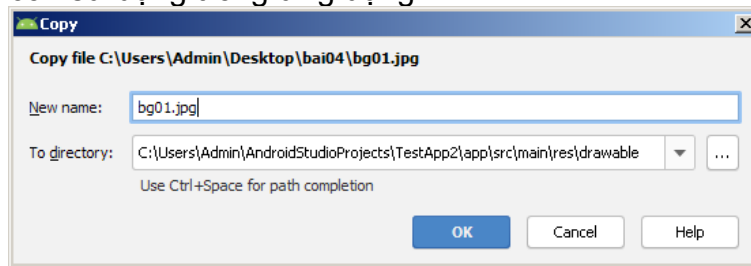
Name
Quan Ly Sinh Vien
Package name
com.thuchanh.quanlysinhvien

- *Application name*: Tên ứng dụng: “**Quan Ly Sinh Vien**”
 - *Package name*: Định danh của ứng dụng: “**com.thuchanh.quanlysinhvien**”
 - *Project location*: Nơi chứa ứng dụng trên đĩa cứng máy tính
 - Chú ý chọn API level phù hợp với chức năng định xây dựng của ứng dụng
- Nhấn **Finish** để hoàn tất tạo một dự án mới.

Bước 2: Thêm (copy) các tài nguyên hình ảnh vào cho ứng dụng

- Hình nền cho các màn hình Activity

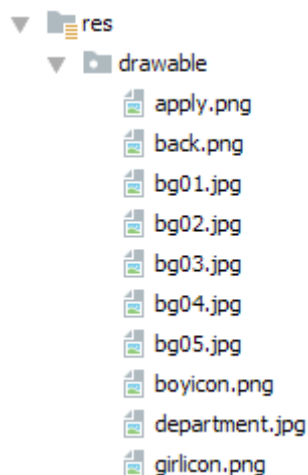
- Các icon sử dụng trong ứng dụng



- **Chú ý: tên tài nguyên không được chứa ký tự viết hoa hoặc dấu khoảng trắng. Thêm vào thư mục “drawable”**

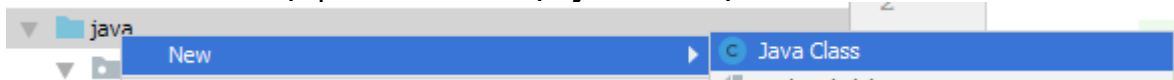


- Thêm các file ảnh và icon vào thư mục **drawable**



Bước 3: Xây dựng các lớp nghiệp vụ cho ứng dụng

- Mã nguồn của ứng dụng được chia thành 3 gói (package):
 - Gói **com.thuchanh.quanlysinhvien** → dùng để chứa các activity của ứng dụng
 - Gói **com.thuchanh.model** → dùng để chứa các lớp nghiệp vụ trong ứng dụng
 - Gói **com.thuchanh.adapter** → dùng để chứa các lớp adapter tạo layout tùy chỉnh (custom layout) cho hiển thị các mục trong ListView
- Tạo một **enum** chứa thông tin về các chức vụ của sinh viên trong lớp:
 - Sinh viên trong lớp có thể là: **Lớp trưởng / Lớp phó / Sinh viên**
 - Nhấn chuột phải vào thư mục **java** → chọn **New** → **Java Class**



- Trong cửa sổ tạo lớp mới, nhập các thông tin như sau:

Name:

Kind: ☒ Enum

Interface(s):

Package:

Visibility: ☒ Public ☐ Package Private

- Nhấn **OK** để tạo mới.
- Viết mã nguồn của **enum ChucVu** như sau:

```

1 package com.thuchanh.model;
2
3 public enum ChucVu {
4     LopTruong("Lớp Trưởng"),
5     LopPho("Lớp Phó"),
6     SinhVien("Sinh Viên");
7     private String chucvu;
8     @ChucVu(String chucvu) {
9         this.chucvu=chucvu;
10    }
11    @public String getChucVu() {
12        return this.chucvu;
13    }
14 }

```

- Xây dựng lớp cha **ThongTin**, lớp này chứa các thông tin chung (như là mã số và tên), các lớp con **LopHoc** và **SinhVien** sẽ kế thừa từ lớp cha này:
 - Nhấn chuột phải vào thư mục **java** → chọn **New** → **Java Class**



- Trong cửa sổ tạo lớp mới, nhập các thông tin như sau:

Name:

Kind: ☒ Class

Superclass:

Interface(s):

Package:

Visibility: ☒ Public ☐ Package Private

Modifiers: ☒ None ☐ Abstract ☐ Final

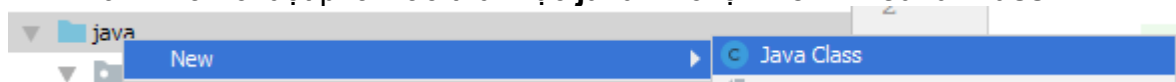
- Nhấn **OK** để tạo mới.
- Viết mã nguồn của **class ThongTin** như sau:

```

1  package com.thuchanh.model;
2
3  import java.io.Serializable;
4
5  public class ThôngTin implements Serializable {
6      private static final long serialVersionUID = 1L;
7      private String ma;
8      private String ten;
9      public ThôngTin(String ma, String ten) {
10         super();
11         this.ma = ma;
12         this.ten = ten;
13     }
14     public ThôngTin() {
15         super();
16     }
17     public String getMa() {
18         return ma;
19     }
20     public void setMa(String ma) {
21         this.ma = ma;
22     }
23     public String getTen() {
24         return ten;
25     }
26     public void setTen(String ten) {
27         this.ten = ten;
28     }
29     @Override
30     public String toString() {
31         return this.ma + " - " + this.ten;
32     }
33 }
34

```

- Xây dựng lớp **SinhVien** mô tả thông tin về các đối tượng Sinh viên:
 - Nhấn chuột phải vào thư mục **java** → chọn **New** → **Java Class**



- Trong cửa sổ tạo lớp mới, nhập các thông tin như sau:

Name:	SinhVien
Kind:	<input checked="" type="radio"/> Class
Superclass:	com.thuchanh.model.ThongTin
Interface(s):	java.io.Serializable
Package:	com.thuchanh.model
Visibility:	<input checked="" type="radio"/> Public <input type="radio"/> Package Private
Modifiers:	<input checked="" type="radio"/> None <input type="radio"/> Abstract <input type="radio"/> Final

- Nhấn **OK** để tạo mới.
- Viết mã nguồn của **class SinhVien** như sau:

```
1 package com.thuchanh.model;
2
3 import java.io.Serializable;
4
5 public class SinhVien extends ThongTin
6     implements Serializable {
7     private static final long serialVersionUID = 1L;
8     private boolean gioitinh;
9     private ChucVu chucvu = ChucVu.SinhVien;
10    private LopHoc lophoc;
11    public SinhVien(String ma, String ten,
12                    boolean gioitinh, ChucVu chucvu,
13                    LopHoc lophoc) {
14        super(ma, ten);
15        this.gioitinh = gioitinh;
16        this.chucvu = chucvu;
17        this.lophoc = lophoc;
18    }
19    public SinhVien(String ma, String ten,
20                    boolean gioitinh) {
21        super(ma, ten);
22        this.gioitinh = gioitinh;
23    }
```

```

24     public SinhVien() {
25         super();
26     }
27     public boolean isGioitinh() {
28         return gioitinh;
29     }
30     public void setGioitinh(boolean gioitinh) {
31         this.gioitinh = gioitinh;
32     }
33     public ChucVu getChucvu() {
34         return chucvu;
35     }
36     public void setChucvu(ChucVu chucvu) {
37         this.chucvu = chucvu;
38     }
39     public LopHoc getLophoc() {
40         return lophoc;
41     }
42     public void setLophoc(LopHoc lophoc) {
43         this.lophoc = lophoc;
44     }
45     @Override
46     public String toString() {
47         return super.toString();
48     }
49 }

```

- Xây dựng lớp **LopHoc** mô tả thông tin về các đối tượng Lớp học:
 - Nhấn chuột phải vào thư mục **java** → chọn **New** → **Java Class**



- Trong cửa sổ tạo lớp mới, nhập các thông tin như sau:

Name:	LopHoc
Kind:	<input checked="" type="radio"/> Class
Superclass:	com.thuchanh.model.ThongTin
Interface(s):	java.io.Serializable
Package:	com.thuchanh.model
Visibility:	<input checked="" type="radio"/> Public <input type="radio"/> Package Private
Modifiers:	<input checked="" type="radio"/> None <input type="radio"/> Abstract <input type="radio"/> Final

- Nhấn **OK** để tạo mới.
- Viết mã nguồn của **class LopHoc** như sau:

```

1  package com.thuchanh.model;
2
3  import java.io.Serializable;
4  import java.util.ArrayList;
5
6  public class LopHoc extends ThôngTin
7      implements Serializable {
8      private static final long serialVersionUID = 1L;
9      //Danh sách sinh viên của lớp học
10     private ArrayList<SinhVien> dssv = new ArrayList<SinhVien>();
11     public LopHoc(String ma, String ten) {
12         super(ma,ten);
13     }
14     public LopHoc() {
15         super();
16     }
17     //Thêm một sinh viên mới,
18     //nếu sinh viên đã có trong danh sách thì cập nhật thông tin
19     public void themSv(SinhVien sv){
20         int i=0;
21         for(i=0 ; i<dssv.size() ; i++){
22             SinhVien nvOld = dssv.get(i);
23             if(nvOld.getMa().trim().equalsIgnoreCase(sv.getMa().trim())){
24                 break;
25             }
26         }
27         if(i<dssv.size())
28             dssv.set(i, sv);
29         else
30             dssv.add(sv);
31     }
32     //Trả về sinh viên tại vị trí index
33     public SinhVien get(int index){
34         return dssv.get(index);
35     }
36     //Số lượng sinh viên trong lớp
37     public int size(){
38         return dssv.size();
39     }

```

```

40 //Trả về sinh viên là lớp trưởng của lớp
41 //Một lớp học chỉ có 1 lớp trưởng
42 public SinhVien getLopTruong() {
43     for(int i=0; i<dssv.size(); i++) {
44         SinhVien sv=dssv.get(i);
45         if(sv.getChucvu()==ChucVu.LopTruong)
46             return sv;
47     }
48     return null;
49 }
50 //Trả về danh sách sinh viên là lớp phó
51 //Một lớp học có thể có nhiều lớp phó
52 public ArrayList<SinhVien>getLopPho() {
53     ArrayList<SinhVien> dsLopPho=new ArrayList<SinhVien>();
54     for(SinhVien sv: dssv)
55     {
56         if(sv.getChucvu()==ChucVu.LopPho)
57             dsLopPho.add(sv);
58     }
59     return dsLopPho;
60 }
61 //Trả về danh sách toàn bộ sinh viên trong lớp
62 public ArrayList<SinhVien> getListSinhVien() {
63     return this.dssv;
64 }
65 @Override
66 public String toString() {
67     // TODO Auto-generated method stub
68     String str=super.toString();
69     if(dssv.size()==0)
70         str += "(Chưa có sinh viên)";
71     else
72         str += "(có "+dssv.size() + " Sinh Viên)";
73     return str;
74 }
75 }

```

➤ Kết quả trong gói **com.thuchanh.model**:

```

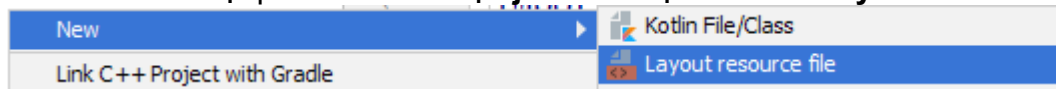
▼ com.thuchanh
  ▼ model
    ● ChucVu
    ● LopHoc
    ● SinhVien
    ● ThongTin

```


Bước 4: Xây dựng các lớp Adapter để gắn kết nguồn dữ liệu và layout hiển thị các mục Item trên các điều khiển ListView

- Xây dựng layout hiển thị các mục Item trên các điều khiển ListView:
“**layout_item_custom.xml**”

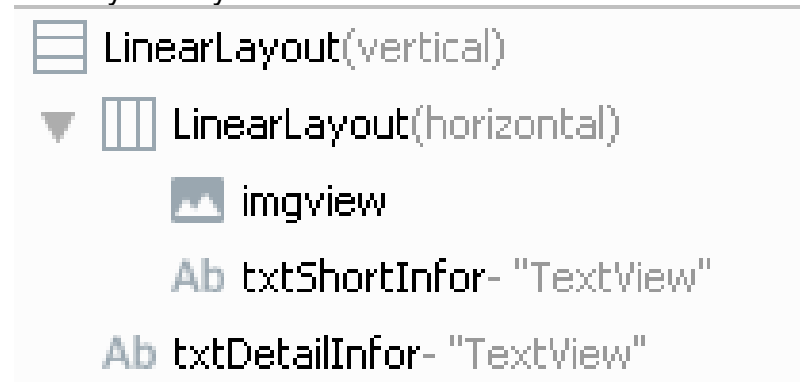
- Nhấn chuột phải vào thư mục **java** → chọn **New** → **Layout resource file**



- Trong cửa sổ tạo lớp mới, nhập các thông tin như sau:

File name:	layout_item_custom
Root element:	LinearLayout
Source set:	main
Directory name:	layout

- Nhấn **OK** để tạo mới.
- Thiết kế layout này như sau:



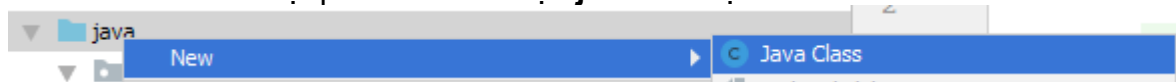
- Nội dung phần XML:

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:orientation="vertical" android:layout_width="match_parent"
    android:layout_height="match_parent">
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal">
        <ImageView
            android:id="@+id/imgview"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content" />
        <TextView
            android:id="@+id/txtShortInfor"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="TextView" />
    </LinearLayout>
    <TextView
        android:id="@+id/txtDetailInfor"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="TextView"
        android:textColor="#000080"
        android:textSize="10sp"
        android:textStyle="italic"/>
</LinearLayout>

```

- Xây dựng lớp **LopHocAdapter** sử dụng custom layout ở trên để hiển thị danh sách các lớp học:
 - Nhấn chuột phải vào thư mục **java** → chọn **New** → **Java Class**



- Trong cửa sổ tạo lớp mới, nhập các thông tin như sau:

Name:	LopHocAdapter		
Kind:	Class		
Superclass:			
Interface(s):			
Package:	com.thuchanh.adapter		
Visibility:	<input checked="" type="radio"/> Public	<input type="radio"/> Package Private	
Modifiers:	<input checked="" type="radio"/> None	<input type="radio"/> Abstract	<input type="radio"/> Final

- Nhấn **OK** để tạo mới.
- Viết mã nguồn của **class LopHocAdapter** như sau:

```

1 package com.thuchanh.adapter;
2
3 import android.widget.ArrayAdapter;
4 import android.app.Activity;
5 import android.view.View;
6 import android.view.ViewGroup;
7 import android.widget.ImageView;
8 import android.widget.TextView;
9 import java.util.ArrayList;
10 import com.thuchanh.model.LopHoc;
11 import com.thuchanh.model.SinhVien;
12 import com.thuchanh.quanlysinhvien.R;
13
14 public class LopHocAdapter extends ArrayAdapter<LopHoc> {
15     Activity context;
16     int layoutId;
17     ArrayList<LopHoc> arrLopHoc;
18     public LopHocAdapter(Activity context, int textViewResourceId,
19         ArrayList<LopHoc> objects) {
20         super(context, textViewResourceId, objects);
21         this.context = context;
22         this.layoutId = textViewResourceId;
23         this.arrLopHoc = objects;
24     }
25     @Override
26     public View getView(int position, View convertView, ViewGroup parent) {
27         //Gán layout vào phần mã nguồn
28         convertView = context.getLayoutInflater().inflate(layoutId, root: null);
29         //Lấy các điều khiển theo id
30         TextView txtlophoc = (TextView) convertView.findViewById(R.id.txtShortInfor);
31         TextView txtmotalophoc = (TextView) convertView.findViewById(R.id.txtDetailInfor);
32         ImageView imgview = (ImageView) convertView.findViewById(R.id.imgview);
33         //Lấy lớp học tại vị trí position, đưa vào phần mô tả ngắn
34         LopHoc lophoc = arrLopHoc.get(position);
35         txtlophoc.setText(lophoc.toString());
36         //Lấy thông tin lớp trưởng, lớp phó đưa vào phần mô tả chi tiết
37         String strMota = "";
38         String loptruong = "Lớp Trưởng: [Chưa có]";
39         SinhVien sv = lophoc.getLopTruong();
40         if(sv!=null){
41             loptruong = "Lớp Trưởng: [" + sv.getTen() + "]";
42         }
43         ArrayList<SinhVien> dsLopPho = lophoc.getLopPho();
44         String loppho = "Lớp Phó: [Chưa có]";
45         if(dsLopPho.size()>0){
46             loppho = "Lớp Phó:\n";
47             for(int i=0; i<dsLopPho.size(); i++){
48                 loppho += (i+1) + ". " + dsLopPho.get(i).getTen() + "\n";
49             }
50         }

```

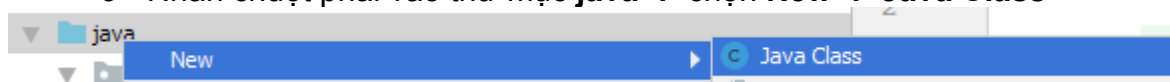
```

51      strMota = loptruong + "\n" + loppho;
52      //gán thông tin cho phần chi tiết
53      txtmotalophoc.setText(strMota);
54      //gán ảnh đại diện cho lớp học
55      imgview.setImageResource(R.drawable.department);
56      return convertView;
57  }
58  }

```

- Xây dựng lớp **SinhVienAdapter** sử dụng custom layout ở trên để hiển thị danh sách các sinh viên trong lớp học:

- Nhấn chuột phải vào thư mục **java** → chọn **New** → **Java Class**



- Trong cửa sổ tạo lớp mới, nhập các thông tin như sau:

Name:	SinhVienAdapter		
Kind:	Class		
Superclass:			
Interface(s):			
Package:	com.thuchanh.adapter		
Visibility:	<input checked="" type="radio"/> Public	<input type="radio"/> Package Private	
Modifiers:	<input checked="" type="radio"/> None	<input type="radio"/> Abstract	<input type="radio"/> Final

- Nhấn **OK** để tạo mới.
- Viết mã nguồn của **class SinhVienAdapter** như sau:

```

1  package com.thuchanh.adapter;
2
3  import android.widget.AdapterView;
4  import android.app.Activity;
5  import android.view.View;
6  import android.view.ViewGroup;
7  import android.widget.AdapterView;
8  import android.widget.ImageView;
9  import android.widget.TextView;
10 import com.thuchanh.quanlysinhvien.R;
11 import com.thuchanh.model.SinhVien;
12 import com.thuchanh.model.LopHoc;
13 import java.util.ArrayList;
14
15 public class SinhVienAdapter extends ArrayAdapter<SinhVien> {
16     Activity context;
17     int layoutId;
18     ArrayList<SinhVien> arrSinhVien;
19     public SinhVienAdapter(Activity context, int textViewResourceId,
20                             ArrayList<SinhVien> objects) {
21         super(context, textViewResourceId, objects);
22         this.context = context;
23         this.layoutId = textViewResourceId;
24         this.arrSinhVien = objects;
25     }
26     @Override
27     public View getView(int position, View convertView, ViewGroup parent) {
28         //Gán layout vào phần mã nguồn
29         convertView = context.getLayoutInflater().inflate(layoutId, root: null);
30         //Lấy các điều khiển theo id
31         TextView txtsinhvien = (TextView) convertView.findViewById(R.id.txtShortInfor);
32         TextView txtmotasinhvien = (TextView) convertView.findViewById(R.id.txtDetailInfor);
33         ImageView img = (ImageView) convertView.findViewById(R.id.imgview);
34         //Lấy sinh viên tại vị trí position, đưa vào phần mô tả ngắn
35         SinhVien sv = arrSinhVien.get(position);
36         txtsinhvien.setText(sv.toString());
37         //Lấy thông tin đưa vào phần mô tả chi tiết
38         String strMota = "";
39         String cv = "Chức vụ: " + sv.getChucvu().getChucVu();
40         String gt = "Giới tính: " + (sv.isGioitinh()?"Nữ":"Nam");
41         //Kiểm tra giới tính để gán cho đúng hình đại diện
42         img.setImageResource(R.drawable.girlicon);
43         if(!sv.isGioitinh())
44             img.setImageResource(R.drawable.boyicon);
45         strMota = cv + "\n" + gt;
46         txtmotasinhvien.setText(strMota);
47         return convertView;
48     }
49 }

```

➤ Kết quả:

- ▼ com.thuchanh
 - ▼ adapter
 - LopHocAdapter
 - SinhVienAdapter

Bước 4: Xây dựng Màn hình chính MainActivity

- Màn hình chính MainActivity: (chú ý thiết lập ID của các điều khiển chính xác)



- Mã nguồn của tệp tin MainActivity.java :

○ **Chú ý import các lớp cần thiết.**

- Phần khai báo các hằng và biến, **nằm ở đầu tệp tin MainActivity.java, trước phương thức onCreate()** (các giá trị hằng này là Request + Result code để xử lý kết quả khi chuyển giữa các Activity qua Intent)

```

//Khai báo các Request + Result code
//để xử lý kết quả khi chuyển giữa các Activity qua Intent
public static final int MO_ACTIVITY_THEM_SINH_VIEN = 1;
public static final int MO_ACTIVITY_SUA_SINH_VIEN = 2;
public static final int THEM_SINH_VIEN_THANHCONG = 3;
public static final int SUA_SINH_VIEN_THANHCONG = 4;
public static final int XEM_DS_SINH_VIEN = 5;
public static final int CAPNHAT_DS_SINH_VIEN_THANHCONG = 6;
public static final int MO_ACTIVITY_THIET_LAP_LT_LP = 7;
public static final int THIET_LAP_LT_LP_THANHCONG = 8;
public static final int MO_ACTIVITY_CHUYENLOP = 9;
public static final int CHUYENLOP_THANHCONG = 10;
//Khai báo các biến điều khiển view
private Button btnLuulophoc;
private EditText editMalophoc, editTenlophoc;
private ListView lvLophoc;
//Khai báo danh sách các lớp học
private static ArrayList<LopHoc> arrLopHoc = new ArrayList<LopHoc>();
//Khai báo LopHocAdapter
private LopHocAdapter adapterLophoc = null;
//Biên đánh dấu đối lượng lớp học được chọn trong danh sách
private LopHoc lophocSelected = null;

```

- Xây dựng phương thức khởi tạo dữ liệu ban đầu cho các lớp học. Sau này các dữ liệu khởi tạo có thể được đọc từ tệp tin XML hoặc từ Cơ sở dữ liệu SQLite


```

public void initLopHocData() {
    SinhVien sv = null;
    LopHoc lh = new LopHoc( ma: "ltmt4", ten: "Lập trình máy tính 4 K9");
    sv = new SinhVien( ma: "CD171308", ten: "Nguyễn Anh Dũng", gioitinh: true);
    sv.setChucvu(ChucVu.LopTruong);
    lh.themSv(sv);
    sv = new SinhVien( ma: "CD171750", ten: "Luu Việt Anh", gioitinh: true);
    lh.themSv(sv);
    sv = new SinhVien( ma: "CD171342", ten: "Lương Thị Minh Hằng", gioitinh: false);
    lh.themSv(sv);
    arrLopHoc.add(lh);

    lh = new LopHoc( ma: "ltmt3", ten: "Lập trình máy tính 3 K9");
    sv = new SinhVien( ma: "CD172368", ten: "Trần Văn Thứ", gioitinh: true);
    sv.setChucvu(ChucVu.LopTruong);
    lh.themSv(sv);
    sv = new SinhVien( ma: "CD171353", ten: "Nguyễn Hồng Ngân", gioitinh: false);
    lh.themSv(sv);
    arrLopHoc.add(lh);

    lh = new LopHoc( ma: "ltmt3", ten: "Lập trình máy tính 3 K9");
    arrLopHoc.add(lh);
    lh = new LopHoc( ma: "ltmt3", ten: "Lập trình máy tính 3 K9");
    arrLopHoc.add(lh);
    adapterLophoc.notifyDataSetChanged();
}

```

- Xây dựng phương thức load các điều khiển theo Id, thiết lập Adapter cho ListView lớp học và thiết lập context Menu cho ListView.

```

public void getWidgets() {
    btnLuulophoc = (Button) findViewById(R.id.btnLuulophoc);
    editMalophoc = (EditText) findViewById(R.id.editMalophoc);
    editTenlophoc = (EditText) findViewById(R.id.editTenlophoc);
    lvLophoc = (ListView) findViewById(R.id.lvLophoc);
    //khởi tạo đối tượng lớp học adapter
    //dùng layout_item_custom.xml
    adapterLophoc = new LopHocAdapter( context: this,
        R.layout.layout_item_custom,
        arrLopHoc);
    lvLophoc.setAdapter(adapterLophoc);
    //Gọi phương thức đăng ký context menu cho Listview
    registerContextMenu(lvLophoc);
}

```


- Tạo context menu với các mục chức năng :
- Nhấn chuột phải vào thư mục **res** → chọn **New** → **Android Resource Directory**



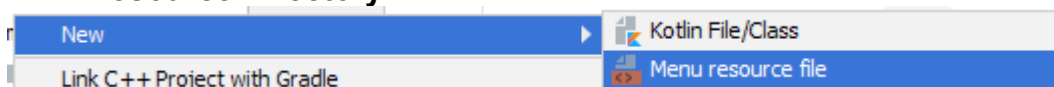
- Trong cửa sổ tạo lớp mới, nhập các thông tin như sau:

Directory name: menu

Resource type: menu

Source set: main

- Tiếp tục nhấn chuột phải vào thư mục **menu** vừa tạo → chọn **New** → **Menu Resource Directory**



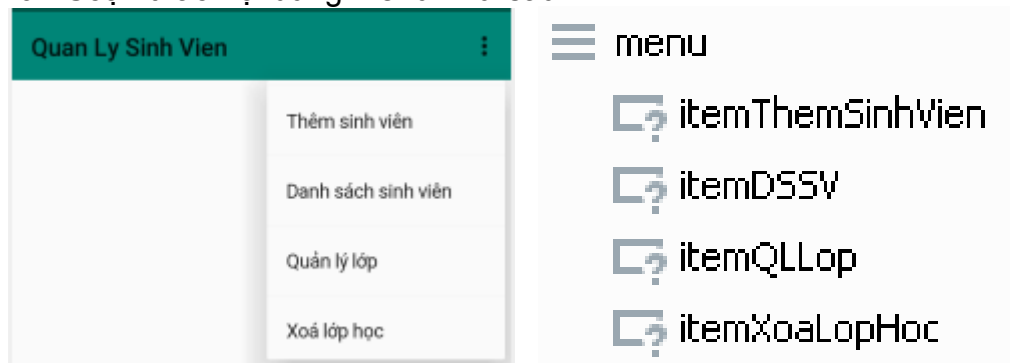
- Trong cửa sổ tạo lớp mới, nhập các thông tin như sau:

File name: contextmenu_lophoc

Source set: main

Directory name: menu

- Soạn thảo nội dung menu như sau:



- Nội dung phần XML:

```

<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android">

    <item
        android:id="@+id/itemThemSinhVien"
        android:title="Thêm sinh viên" />
    <item
        android:id="@+id/itemDSSV"
        android:title="Danh sách sinh viên" />
    <item
        android:id="@+id/itemQLLop"
        android:title="Quản lý lớp" />
    <item
        android:id="@+id/itemXoaLopHoc"
        android:title="Xóa lớp học" />

</menu>

```

- Chuyển sang phần mã nguồn của tệp tin MainActivity.java, viết phương thức gắn kết context menu với điều khiển ListView

```

public void onCreateContextMenu(ContextMenu menu, View v,
                                ContextMenu.ContextMenuInfo menuInfo) {
    super.onCreateContextMenu(menu, v, menuInfo);
    getMenuInflater().inflate(R.menu.contextmenu_lophoc, menu);
}

```

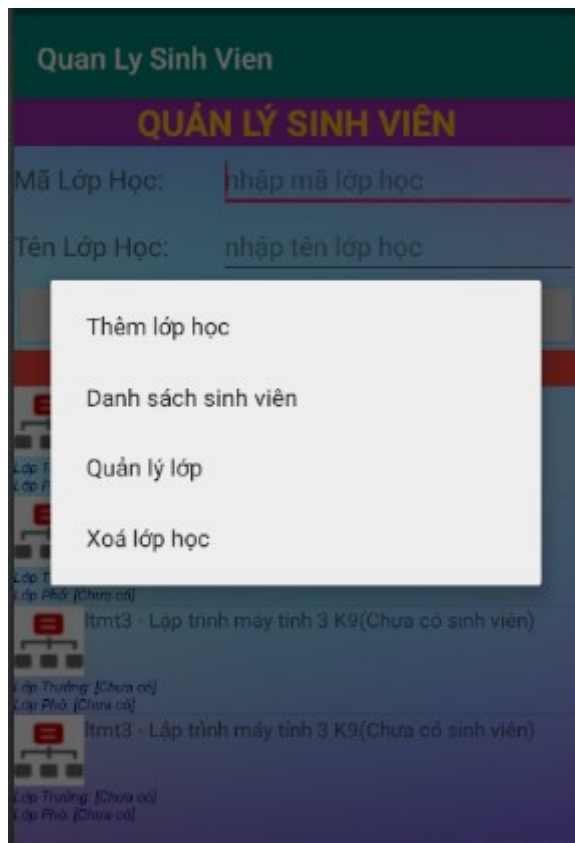
- Đặt hai lời gọi hàm vào trong phương thức onCreate() → sau đó chạy thử chương trình:

```

protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    getWidgets();
    initLopHocData();
}

```

- Kết quả nhận được, khi nhấn giữ lâu trên một phần tử trong danh sách → sẽ xuất hiện Context Menu:



➤ Tiếp theo xây dựng các phương thức xử lý sự kiện cho các điều khiển trong Activity

//Xử lý khi người dùng chọn một mục menu, dựa vào ID của item

@Override

```
public boolean onOptionsItemSelected(MenuItem item) {
    switch (item.getItemId()) {
        case R.id.itemThemSinhVien:
            ThemSinhVien();
            break;
        case R.id.itemDSSV:
            DanhSachSinhVien();
            break;
        case R.id.itemQLLop:
            ThietLapQLLop();
            break;
        case R.id.itemXoaLopHoc:
            XoaLopHoc();
            break;
    }
    return super.onOptionsItemSelected(item);
}
```

```

//Phương thức gán sự kiện cho các control: Button, ListView
public void addEvents() {
    //Bấm nút lưu để lưu lớp học
    btnLuulophoc.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View arg0) {
            // TODO Auto-generated method stub
            LuuLopHoc();
        }
    });
    //xử lý lưu biến tạm khi nhấn long - click
    //phải dùng cái này để biết được trước đó đã chọn item nào
    lvLophoc.setOnItemClickListener(new AdapterView.OnItemClickListener() {
        @Override
        public boolean onItemClick(AdapterView<?> arg0, View arg1,
                                   int arg2, long arg3) {
            //Lưu vết lại đối tượng thứ arg2 trong danh sách
            lophocSelected = arrLopHoc.get(arg2);
            return false;
        }
    });
}

```

- Tiếp theo xây dựng các phương thức xử lý các thao tác (thêm, xóa, lưu,...) trên danh sách lớp học:

```

//Phương thức xử lý lưu lớp học: đưa lớp học mới vào ArrayList
//nếu cần thận hơn thì nên kiểm tra xem mã lớp có bị trùng
//gọi adapterLopHoc.notifyDataSetChanged(); để cập nhật ListView
public void LuuLopHoc() {
    String malophoc = editMalophoc.getText().toString();
    String tenlophoc = editTenlophoc.getText().toString();
    LopHoc lophoc = new LopHoc(malophoc, tenlophoc);
    arrLopHoc.add(lophoc);
    adapterLopHoc.notifyDataSetChanged();
}
//Phương thức xử lý khi chọn menu Thêm sinh viên
//hiển thị màn hình thêm sinh viên (với dạng Dialog)
//sử dụng Intent với phương thức startActivityForResult
//lắng nghe kết quả tại phương thức onActivityResult
public void ThemSinhVien() {
    Intent i = new Intent( packageContext: this, ThemSinhVienActivity.class);
    startActivityForResult(i, MO_ACTIVITY_THEM_SINH_VIEN);
}

```

```

//Phương thức xử lý khi chọn menu Danh sách sinh viên
//mở DanhSachNhanVienActivity truyền lớp học qua
//-->toàn bộ danh sách sinh viên của lớp đó được hiển thị
//sử dụng Intent truyền dữ liệu với Bundle
public void DanhSachSinhVien() {
    Intent i = new Intent( packageContext: this, DanhSachSinhVienActivity.class);
    Bundle bundle=new Bundle();
    bundle.putSerializable("LOPHOC", lophocSelected);
    i.putExtra( name: "DATA", bundle);
    startActivityForResult(i, XEM_DS_SINH_VIEN);
}
//Phương thức xử lý khi chọn menu Quản lý lớp
//Lớp trưởng và lớp phó
//ThietLapQLLopActivity sẽ có 2 ListView
//- ListView 1 hiển thị dạng radiobutton để chỉ chọn 1 lớp trưởng
//- ListView 2 hiển thị dạng checkbox cho phép chọn nhiều lớp phó
public void ThietLapQLLop(){
    Intent i = new Intent( packageContext: this, ThietLapQLLopActivity.class);
    Bundle bundle = new Bundle();
    bundle.putSerializable("LOPHOC", lophocSelected);
    i.putExtra( name: "DATA", bundle);
    startActivityForResult(i, MO_ACTIVITY_THIET_LAP_LT_LP);
}
//Phương thức xử lý khi chọn menu Xóa lớp học
//Hiển thị AlertDialog xác nhận người dùng có xóa hay không
public void XoaLopHoc() {
    AlertDialog.Builder builder = new AlertDialog.Builder ( context: this);
    builder.setTitle("Xác nhận xóa dữ liệu");
    builder.setMessage("Bạn có chắc chắn muốn xóa [" + lophocSelected.getTen() + "]");
    builder.setIcon(android.R.drawable.ic_input_delete);
    builder.setNegativeButton( text: "Không", new DialogInterface.OnClickListener() {
        @Override
        public void onClick(DialogInterface arg0, int arg1) {
            // TODO Auto-generated method stub
            arg0.cancel();
        }
    });
    builder.setPositiveButton( text: "Đồng ý", new DialogInterface.OnClickListener() {
        @Override
        public void onClick(DialogInterface dialog, int which) {
            // TODO Auto-generated method stub
            arrLopHoc.remove(lophocSelected);
            adapterLopHoc.notifyDataSetChanged();
        }
    });
    builder.show();
}

```

```

//Phương thức onActivityResult để xử lý kết quả trả về
//sau khi gọi startActivityForResult kết thúc
protected void onActivityResult(int requestCode, int resultCode, Intent data){
    // TODO Auto-generated method stub
    super.onActivityResult(requestCode, resultCode, data);
    //màn hình thêm mới sinh viên trả kết quả về
    if(resultCode==THEM_SINH_VIEN_THANHCONG) {
        Bundle bundle= data.getBundleExtra( name: "DATA");
        SinhVien nv= (SinhVien) bundle.getSerializable( key: "SINHVIEN");
        lophocSelected.themSv(nv);
        adapterLophoc.notifyDataSetChanged();
    }
    //Màn hình thiết lập QL Lớp/ cập nhật danh sách trả kết quả về
    else if(resultCode == THIET_LAP_LT_LP_THANHCONG ||
            resultCode == CAPNHAT_DS_SINH_VIEN_THANHCONG)
    {
        Bundle bundle= data.getBundleExtra( name: "DATA");
        LopHoc pb= (LopHoc) bundle.getSerializable( key: "LOPHOC");
        //Xóa danh sách cũ
        lophocSelected.getListSinhVien().clear();
        //Cập nhật lại toàn bộ danh sách mới
        lophocSelected.getListSinhVien().addAll(pb.getListSinhVien());
        adapterLophoc.notifyDataSetChanged();
    }
}

//Thêm một phương thức để ở các Activity khác đều
//có thể truy suất được danh sách phòng ban tổng thể
public static ArrayList<LopHoc> getListLopHoc() {
    return arrLopHoc;
}

```

- Hoàn thiện phương thức onCreate(), bổ sung các lời gọi phương thức:

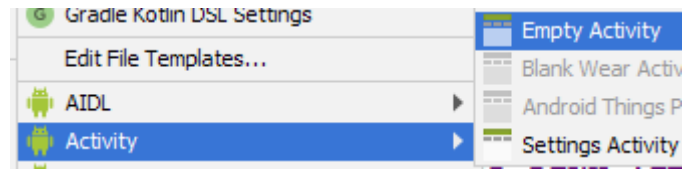
```

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    getWidgets();
    addEvents();
    initLopHocData();
}

```

Bước 5: Xây dựng Màn hình thêm sinh viên **ThemSinhVienActivity**

- Nhấn chuột phải vào thư mục **quanlysinhvien** → chọn **New** → **Activity** → **Empty Activity**



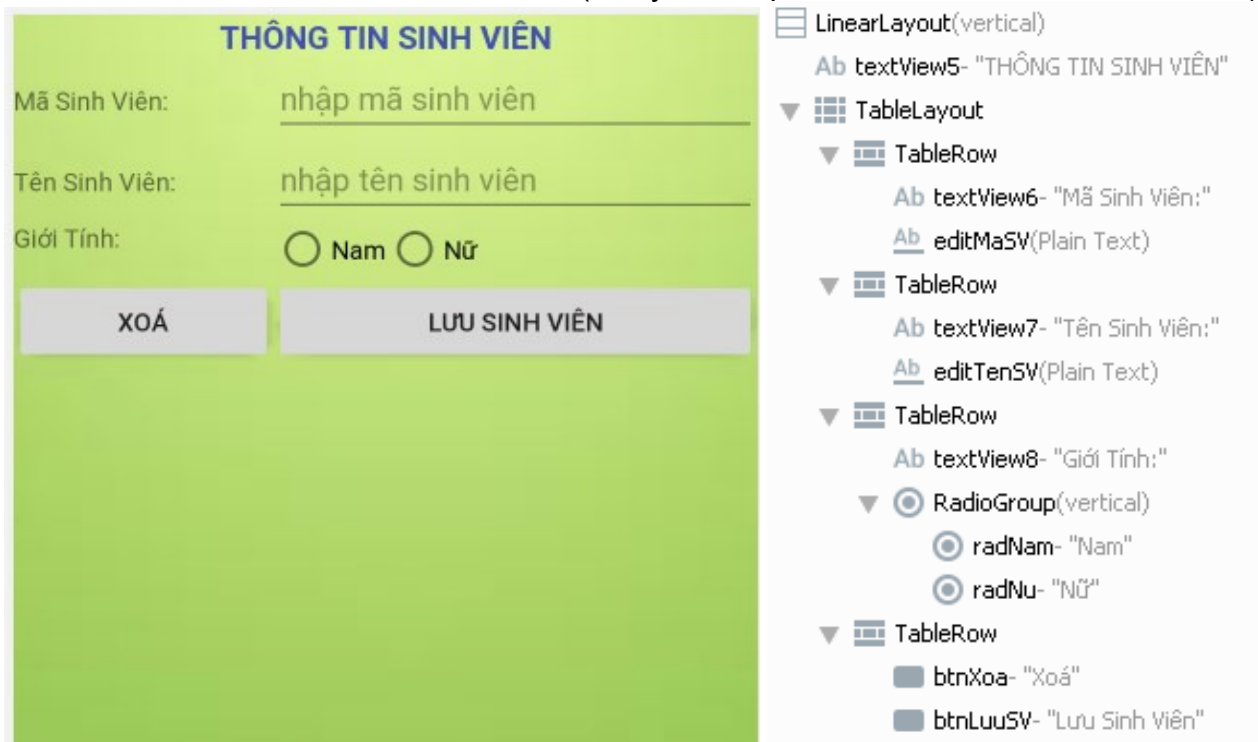
- Trong cửa sổ tạo lớp mới, nhập các thông tin như sau:

Activity Name:

☒ Generate Layout File

Layout Name:

- Màn hình thêm sinh viên như sau: (chú ý thiết lập ID của các điều khiển chính xác)



- Tiếp theo xây dựng các phương thức xử lý sự kiện cho các điều khiển trong ThemSinhVienActivity

```

//Khai báo biến
private Button btnXoa,btnLuuSV;
private EditText editMaSV,editTenSV;
private RadioButton radNam;
//Phương thức lấy các điều khiển View qua ID
public void getWidgets() {
    btnXoa = (Button) findViewById(R.id.btnXoa);
    btnLuuSV = (Button) findViewById(R.id.btnLuuSV);
    editMaSV = (EditText) findViewById(R.id.editMaSV);
    editTenSV = (EditText) findViewById(R.id.editTenSV);
    radNam=(RadioButton) findViewById(R.id.radNam);
}
//Phương thức xử lý sự kiện các button trên giao diện
public void addEvents()
{
    btnXoa.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View arg0) {
            // TODO Auto-generated method stub
            Xoa();
        }
    });
    btnLuuSV.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            // TODO Auto-generated method stub
            LuuSinhVien();
        }
    });
}

```



```

//Phương thức xoá dữ liệu trong các ô
public void Xoa() {
    editMaSV.setText("");
    editTenSV.setText("");
    editMaSV.requestFocus();
}

//Phương thức lưu thông tin sinh viên
//truyền sinh viên qua MainActivity
public void LuuSinhVien() {
    SinhVien sv=new SinhVien();
    sv.setMa(editMaSV.getText()+"");
    sv.setTen(editTenSV.getText()+"");
    sv.setChucvu(ChucVu.SinhVien);
    sv.setGioitinh(!radNam.isChecked());
    //Sử dụng Intent trả kết quả về cho MainActivity
    Intent i=getIntent();
    Bundle bundle=new Bundle();
    bundle.putSerializable("SINHVIEN", sv);
    i.putExtra( name: "DATA", bundle);
    setResult(MainActivity.THEM_SINH_VIEN_THANHCONG, i);
    finish();
}

```

- Hoàn thiện phương thức onCreate(), bổ sung các lời gọi phương thức:

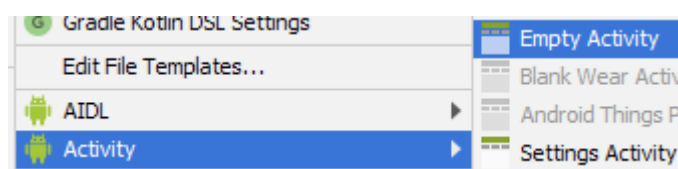
```

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_them_sinh_vien);
    getWidgets();
    addEvents();
}

```

Bước 6: Xây dựng Màn hình thêm sinh viên **DanhSachSinhVienActivity**

- Nhấn chuột phải vào thư mục **quanlysinhvien** → chọn **New** → **Activity** → **Empty Activity**



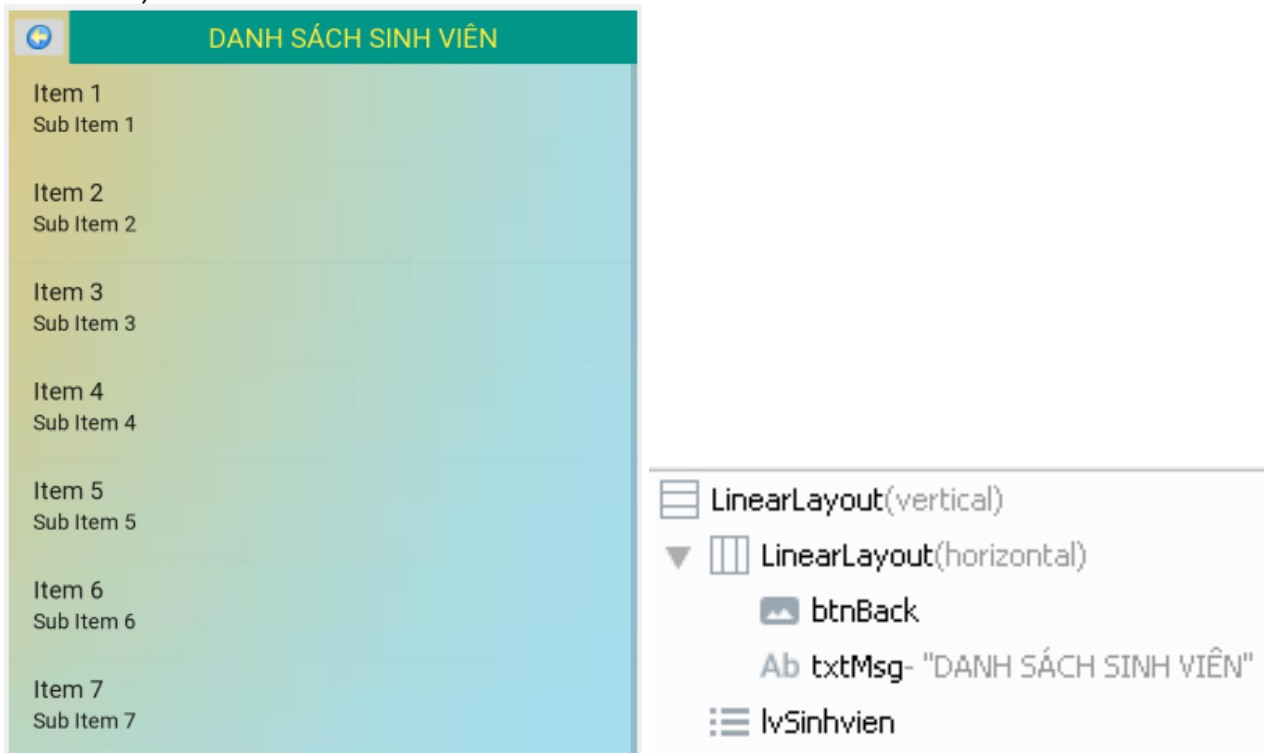
- Trong cửa sổ tạo lớp mới, nhập các thông tin như sau:

Activity Name:

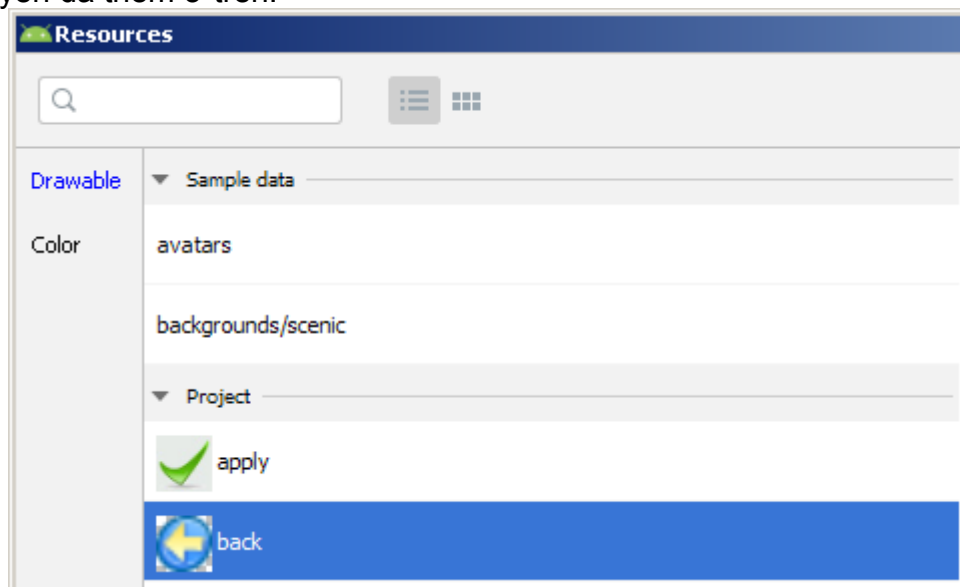
☒ Generate Layout File

Layout Name:

- Màn hình danh sách sinh viên như sau: (chú ý thiết lập ID của các điều khiển chính xác)



- Góc trên bên trái là một image button, nút này được gắn với icon back trong tài nguyên đã thêm ở trên:



- Tiếp theo xây dựng các phương thức xử lý sự kiện cho các điều khiển trong DanhSachSinhVienActivity.java

```

//Khai báo các biến
TextView txtMsg;
ImageButton btnBack;
ListView lvSinhvien;
ArrayList<SinhVien> arrSinhvien = null;
//Sinh viên Adapter để hiển thị thông tin
//chi tiết : chức vụ, giới tính
SinhVienAdapter adapter = null;
LopHoc lophoc = null;
//Biến đánh dấu sinh viên được chọn trong danh sách
private SinhVien sinhvienSelected = null;
private int position = -1;
//Phương thức lấy các điều khiển View qua ID
public void getWidgets() {
    txtMsg=(TextView) findViewById(R.id.txtMsg);
    btnBack=(ImageButton) findViewById(R.id.btnBack);
    lvSinhvien=(ListView) findViewById(R.id.lvSinhvien);
}
//Phương thức tải danh sách sinh viên
//dựa trên lớp học được chọn từ màn hình Main
//Dữ liệu được lấy về qua Bundle trong Intent
public void getDataFromMain(){
    Intent i = getIntent();
    Bundle b = i.getBundleExtra( name: "DATA");
    lophoc = (LopHoc) b.getSerializable( key: "LOPHOC");
    arrSinhvien = lophoc.getListSinhVien();
    adapter = new SinhVienAdapter( context: this,
        R.layout.layout_item_custom,
        arrSinhvien);
    lvSinhvien.setAdapter(adapter);
    txtMsg.setText("Danh sách sinh viên [" + lophoc.getTen() + "]");
}

```

```

//Phương thức gán sự kiện cho các control: Button, ListView
public void addEvents() {
    //Sự kiện khi nhấn nút Back
    btnBack.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View arg0) {
            UpdateToMain();
        }
    });
    //Sự kiện bất mục được chọn trong danh sách khi nhấn giữ lâu
    lvSinhvien.setOnClickListener(new AdapterView.OnItemClickListener() {
        @Override
        public boolean onItemClick(AdapterView<?> arg0, View arg1,
            int arg2, long arg3) {
            sinhvienSelected = arrSinhvien.get(arg2);
            position = arg2;
            return false;
        }
    });
}

//Phương thức xử lý khi nhấn nút Back
//Trả về kết quả cho Activity Main
public void UpdateToMain() {
    Intent i = getIntent();
    Bundle b = new Bundle();
    b.putSerializable("LOPHOC", lophoc);
    i.putExtra("name: DATA", b);
    setResult(MainActivity.CAPNHAT_DS_SINH_VIEN_THANHCONG, i);
    finish();
}

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_danh_sach_sinh_vien);
    getWidgets();
    getDataFromMain();
    addEvents();
}

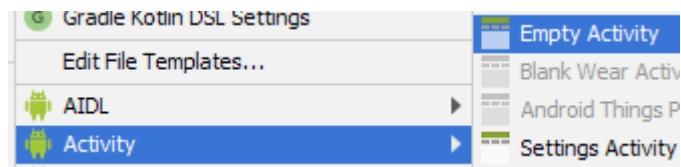
```

- Chạy thử chương trình. Khi chọn menu “Danh sách sinh viên” → sẽ xuất hiện màn hình liệt kê danh sách sinh viên:



Bước 7: Xây dựng Màn hình thêm sinh viên **ThietLapQLlopActivity**

- Nhấn chuột phải vào thư mục **quanlysinhvien** → chọn **New** → **Activity** → **Empty Activity**



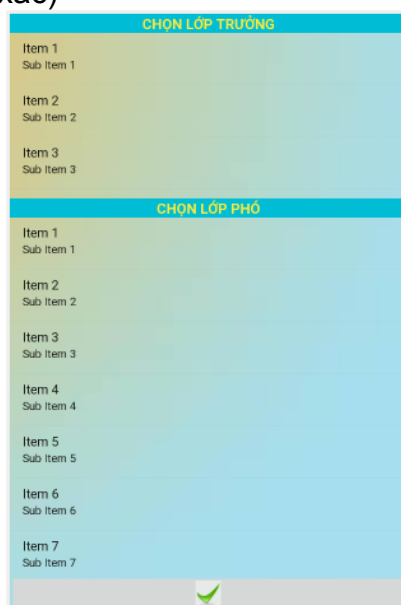
- Trong cửa sổ tạo lớp mới, nhập các thông tin như sau:

Activity Name:

☒ Generate Layout File

Layout Name:

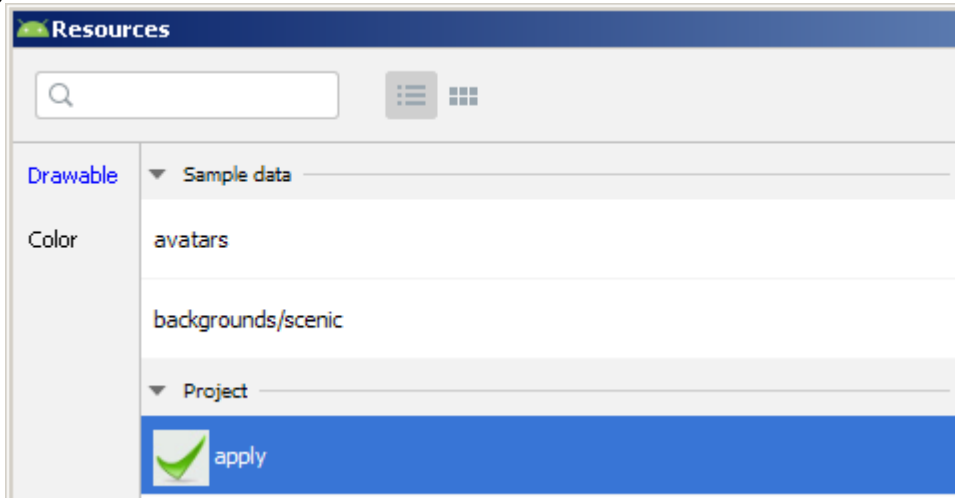
- Màn hình danh sách sinh viên như sau: (chú ý thiết lập ID của các điều khiển chính xác)



```

LinearLayout(vertical)
Ab textView9- "CHỌN LỚP TRƯỞNG"
:≡ lvLopTruong
Ab textView10- "CHỌN LỚP PHÓ"
:≡ lvLopPho
imgApply
  
```

- Màn hình sẽ hiển thị danh sách nhân viên vào 2 ListView khác nhau
 - ListView 1 dùng Radio để chọn trưởng phòng
 - ListView 2 dùng Checkbox để chọn phó phòng
- Góc dưới màn hình là một image button, nút này được gắn với icon apply trong tài nguyên đã thêm ở trên:



- Tiếp theo xây dựng các phương thức xử lý sự kiện cho các điều khiển trong ThietLapQLLopActivity.java

```
//Khai báo các biến
ListView lvLopTruong, lvLopPho;
ImageButton imgApply;
//Các ArrayList và Adapter để chứa thông tin lớp trưởng, lớp phó
ArrayList<SinhVien> arrSVLopTruong = new ArrayList<SinhVien>();
ArrayAdapter<SinhVien> adaptersVLopTruong;
ArrayList<SinhVien> arrSVLopPho = new ArrayList<SinhVien>();
ArrayAdapter<SinhVien> adaptersVLopPho;
//Biến trạng thái
int lastChecked = -1;
LopHoc lophoc = null;
```

```

//Phương thức lấy các điều khiển View qua ID
public void getWidgets() {
    imgApply = (ImageButton) findViewById(R.id.imgApply);

    lvLopTruong = (ListView) findViewById(R.id.lvLopTruong);
    lvLopTruong.setTextFilterEnabled(true);
    lvLopTruong.setChoiceMode(ListView.CHOICE_MODE_SINGLE);
    addListLopTruongEvents();

    lvLopPho = (ListView) findViewById(R.id.lvLopPho);
    addListLopPhoEvents();

    //Lấy được lớp học gửi qua từ MainActivity
    Intent i = getIntent();
    Bundle bundle = i.getBundleExtra(name: "DATA");
    lophoc = (LopHoc) bundle.getSerializable(key: "LOPHOC");
    //Điền sinh viên vào danh sách lớp trưởng
    arrSVLopTruong.clear();
    for(SinhVien sv : lophoc.getListSinhVien()){
        sv.setChucvu(ChucVu.SinhVien);
        arrSVLopTruong.add(sv);
    }
    //Điền sinh viên vào danh sách lớp phó
    arrSVLopPho.clear();
    for(SinhVien sv : lophoc.getListSinhVien()){
        sv.setChucvu(ChucVu.SinhVien);
        arrSVLopPho.add(sv);
    }

    //Thiết lập adapter hiển thị các danh sách
    adapterSVLopTruong = new ArrayAdapter<SinhVien>(context: this,
        android.R.layout.simple_list_item_single_choice,
        arrSVLopTruong);
    adapterSVLopPho = new ArrayAdapter<SinhVien>(context: this,
        android.R.layout.simple_list_item_multiple_choice,
        arrSVLopPho);
    lvLopTruong.setAdapter(adapterSVLopTruong);
    lvLopPho.setAdapter(adapterSVLopPho);

    adapterSVLopTruong.notifyDataSetChanged();
    adapterSVLopPho.notifyDataSetChanged();
}

```

```

        //Xử lý sự kiện nhấn Button Apply
        addImgApplyEvents();
    }
    //Phương thức xử lý sự kiện chọn một mục trong danh sách lớp trưởng
    public void addListLopTruongEvents() {
        lvLopTruong.setOnItemClickListener(new AdapterView.OnItemClickListener() {
            @Override
            public void onItemClick(AdapterView<?> adapterView, View view,
                                    int i, long l){
                CheckedTextView cv = (CheckedTextView) view;
                if(!cv.isChecked()) {
                    arrSVLopTruong.get(i).setChucvu(ChucVu.SinhVien);
                }
                else{
                    cv.setChecked(true);
                    arrSVLopTruong.get(i).setChucvu(ChucVu.LopTruong);
                }
                lastChecked = i;
            }
        });
    }
    //Phương thức xử lý sự kiện chọn nhiều mục trong danh sách lớp phó
    public void addListLopPhoEvents() {
        lvLopPho.setOnItemClickListener(new AdapterView.OnItemClickListener() {
            @Override
            public void onItemClick(AdapterView<?> adapterView, View view,
                                    int i, long l){
                CheckedTextView cv = (CheckedTextView) view;
                if(!cv.isChecked()){
                    cv.setChecked(true);
                    arrSVLopPho.get(i).setChucvu(ChucVu.LopPho);
                }
                else {
                    cv.setChecked(false);
                    arrSVLopPho.get(i).setChucvu(ChucVu.SinhVien);
                }
            }
        });
    }
}

```




```

//Phương thức gán sự kiện click vào nút Apply
//Gửi thông tin lại màn hình MainActivity
public void addImgApplyEvents() {
    imgApply.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            Intent i = getIntent();
            Bundle bundle = new Bundle();
            bundle.putSerializable("LOPHOC", lophoc);
            i.putExtra(name: "DATA", bundle);
            setResult(MainActivity.THUAT_LAP_LT_LP_THANHCONG, i);
            finish();
        }
    });
}
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_thiet_lap_qllop);
    getWidgets();
}

```

- **Chạy thử chương trình. Khi chọn menu “Quản lý lớp” → sẽ xuất hiện màn hình cho phép thiết lập các thông tin lớp trưởng và lớp phó:**

Quan Ly Sinh Vien	
CHỌN LỚP TRƯỞNG	
CD171308 - Nguyễn Anh Dũng	<input checked="" type="radio"/>
CD171750 - Lưu Việt Anh	<input type="radio"/>
CD171342 - Lương Thị Minh Hằng	<input type="radio"/>
CHỌN LỚP PHÓ	
CD171308 - Nguyễn Anh Dũng	<input type="checkbox"/>
CD171750 - Lưu Việt Anh	<input type="checkbox"/>
CD171342 - Lương Thị Minh Hằng	<input checked="" type="checkbox"/>
	

Bài 2. Hoàn thiện chức năng của bài 1

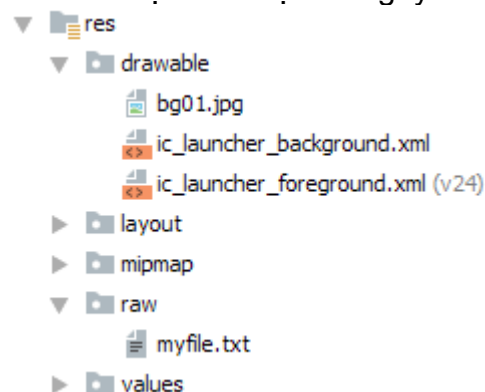
Mô tả:

- Thêm ContextMenu vào màn hình **DanhSachSinhVienActivity**
 - Menu gồm các mục chọn: Sửa thông tin sinh viên / Xóa thông tin sinh viên
 - Khi chọn "Sửa thông tin sinh viên" → chuyển sang màn hình SuaSinhVienActivity, thông tin của đối tượng sinh viên sẽ được gửi kèm theo
 - Màn hình SuaSinhVienActivity có các điều khiển giống như màn hình ThemSinhVienActivity nên **có thể tận dụng lại phần layout của màn hình ThemSinhVienActivity mà không cần tạo layout mới.**
 - Khi nhấn nút Lưu thì cập nhật thông tin sinh viên được sửa, và truyền dữ liệu qua màn hình cha để nó tự cập nhật.
 - Khi chọn "Xóa thông tin sinh viên" → hiển thị hộp thoại AlertDialog yêu cầu người dùng xác nhận có muốn xóa hay không? Nếu có thì thực hiện xóa sinh viên khỏi danh sách

Bài 3. Xử lý tập tin trong Android: Đọc Resource File

Mô tả: Xây dựng chương trình đọc nội dung tập tin trong Resource file và hiển thị lên màn hình

- Tạo Project mới
- Thêm các tài nguyên hình ảnh vào thư mục **res/drawable**: tập tin txt thì thêm vào thư mục **raw** (nếu chưa có thì tạo thư mục tài nguyên mới)



- Thiết kế giao diện đơn giản như sau:



LinearLayout(vertical)
■ btnReadFile- "ĐỌC TẬP"
Ab editFile- "nội dung tập"

➤ Nội dung phần XML:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context=".MainActivity"
    android:background="@drawable/bg01">

    <Button
        android:id="@+id/btnReadFile"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="ĐỌC TỆP" />

    <EditText
        android:id="@+id/editFile"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:ems="10"
        android:inputType="textMultiLine"
        android:text="nội dung tệp" />
</LinearLayout>
```

➤ Mã nguồn xử lý trong tệp **MainActivity.java**

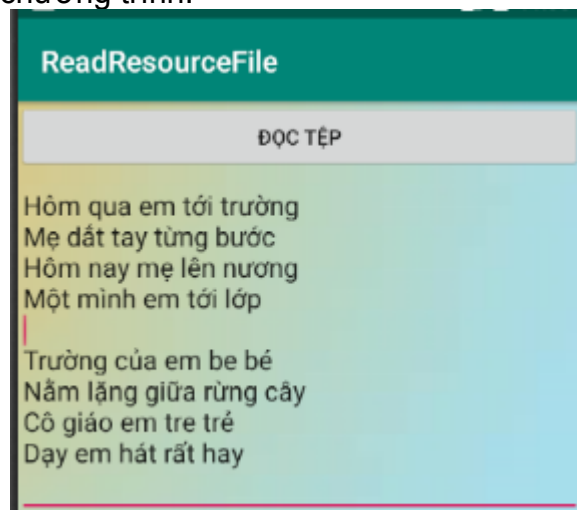
```
public class MainActivity extends AppCompatActivity {
    Button btnReadFile;
    EditText editFile;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        btnReadFile = (Button) findViewById(R.id.btnReadFile);
        editFile = (EditText) findViewById(R.id.editFile);
    }
}
```

```

//Xử lý sự kiện nhấn và nút bấm: load dữ liệu từ file
btnReadFile.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        String data;
        InputStream in = getResources().openRawResource(R.raw.myfile);
        InputStreamReader inreader =new InputStreamReader(in);
        BufferedReader bufreader =new BufferedReader(inreader);
        StringBuilder builder = new StringBuilder();
        if(in != null){
            try {
                while((data = bufreader.readLine()) != null){
                    builder.append(data);
                    builder.append("\n");
                }
                in.close();
                editFile.setText(builder.toString());
            }
            catch(IOException ex){
                Log.e( tag: "ERROR", ex.getMessage());
            }
        }
    }
});
}

```

➤ Kết quả khi chạy chương trình:



Bài 4. Hoàn thiện chương trình ghi chú

Mô tả: Mở chương trình ghi chú đã xây dựng trong bài thực hành trước. Bổ sung thêm các chức năng:

GHI CHÚ ĐƠN GIẢN	
nhập nội dung ghi chú	
THÊM GHI CHÚ	
Item 1	Sub Item 1
Item 2	Sub Item 2
Item 3	Sub Item 3
Item 4	Sub Item 4
Item 5	Sub Item 5
Item 6	Sub Item 6
Item 7	Sub Item 7
Item 8	Sub Item 8
Item 9	
LƯU GHI CHÚ	

- Các ghi chú được lưu trong file text, khi mở chương trình sẽ tải các nội dung này vào ListView (theo từng dòng)

.../...