

ĐỀ CƯƠNG BÀI GIẢNG

BÀI 4. THỰC HÀNH XỬ LÝ SỰ KIỆN TRÊN GIAO DIỆN ỨNG DỤNG

Nội dung bài học trước khi lên lớp (trang 1 đến trang 13):

- Thực hành các kiểu lập trình sự kiện: Phiếu bài tập 4.1, 4.2, 4.3, 4.4
- Thao tác với ListView đơn giản: Phiếu bài tập: 4.5, 4.6

Nội dung bài học thực hiện lên lớp (trang 14 đến hết):

- Thực hành Tùy biến ListView: Phiếu bài tập 4.7, 4.8
- Thực hành với Spinner, TimePicker và DatePicker: Phiếu bài tập 4.9

Nội dung bài học sau khi lên lớp: Phiếu bài tập: bài 4.10 đến 4.13.

NỘI DUNG BÀI HỌC

1. Các kiểu lập trình sự kiện.....	1
1.1. Thiết kế giao diện.....	2
1.2. Đặt tiêu đề hiển thị cho cách cách lập trình giao diện	4
1.3. Kiểm tra các cài đặt trong AndroidManifest.xml	5
1.4. Viết mã lệnh với 5 kiểu lập trình sự kiện.....	5
2. Thực hành với ListView.....	11
2.1. ListView đơn giản.....	11
2.2. Tùy biến ListView.....	14
3. Spinner.....	20
4. TimePicker – DatePicker	21
5. RecyclerView.....	25

1. Các kiểu lập trình sự kiện

Cho mẫu màn hình giao diện sau:

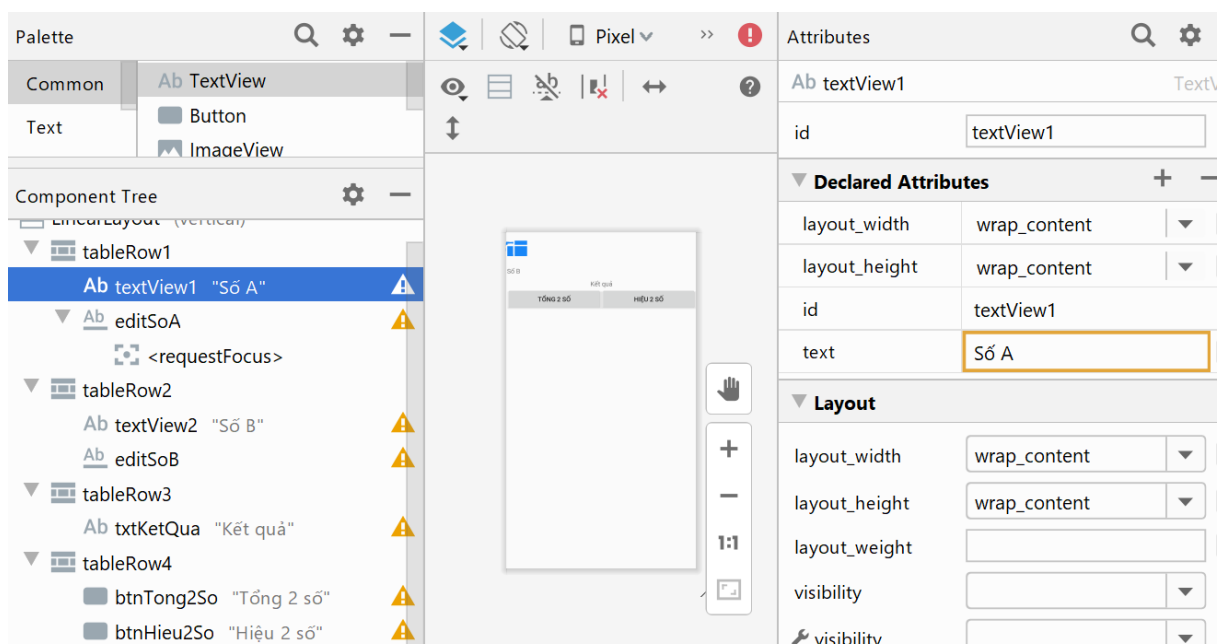


Yêu cầu: Áp dụng 5 kiểu lập trình sự kiện tính tổng, hiệu của 2 số a và số b.

1.1. Thiết kế giao diện

- Kéo thả 2 TextView trong thanh công cụ Palette, nhóm điều khiển Common. Đặt text cho 2 điều khiển: số A và Số B.
- Tương tự Kéo thả 2 EditText, Đặt tên editSoA; và editSoB. Đặt các thuộc tính hint để nhận biết
- Tương tự kéo thả 2 nút lệnh, đặt tên btnTong2So và btnHieu2So. Đặt text cho nút lệnh.
- Sử dụng layout hợp lý để quản lý các điều khiển. Gợi ý sử dụng các thuộc tính layout_weight chia tỉ lệ các không gian tạo khung nhìn cân đối.

Gợi ý thiết kế có biểu diễn như sau:



Giao diện code 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="wrap_content"
    android:orientation="vertical"
    tools:context=".MainActivity">

    <TableRow
        android:id="@+id/tableRow1"
        android:layout_width="match_parent"
        android:layout_height="wrap_content">

        <TextView
            android:id="@+id/textView1"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Số A" />

        <EditText
            android:id="@+id/editSoA"
            android:layout_width="0dp"
            android:layout_height="wrap_content"
            android:layout_weight="2"
            android:hint="so a">

            <requestFocus />
        </EditText>

    </TableRow>

    <TableRow
        android:id="@+id/tableRow2"
        android:layout_width="match_parent"
        android:layout_height="wrap_content">

        <TextView
            android:id="@+id/textView2"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Số B" />

        <EditText
            android:id="@+id/editSoB"
            android:layout_width="0dp"
            android:layout_height="wrap_content"
            android:layout_weight="2"
            android:hint="so B" />

    </TableRow>

</LinearLayout>
```

```
</TableRow>

<TableRow
    android:id="@+id/tableRow3"
    android:layout_width="match_parent"
    android:layout_height="wrap_content">

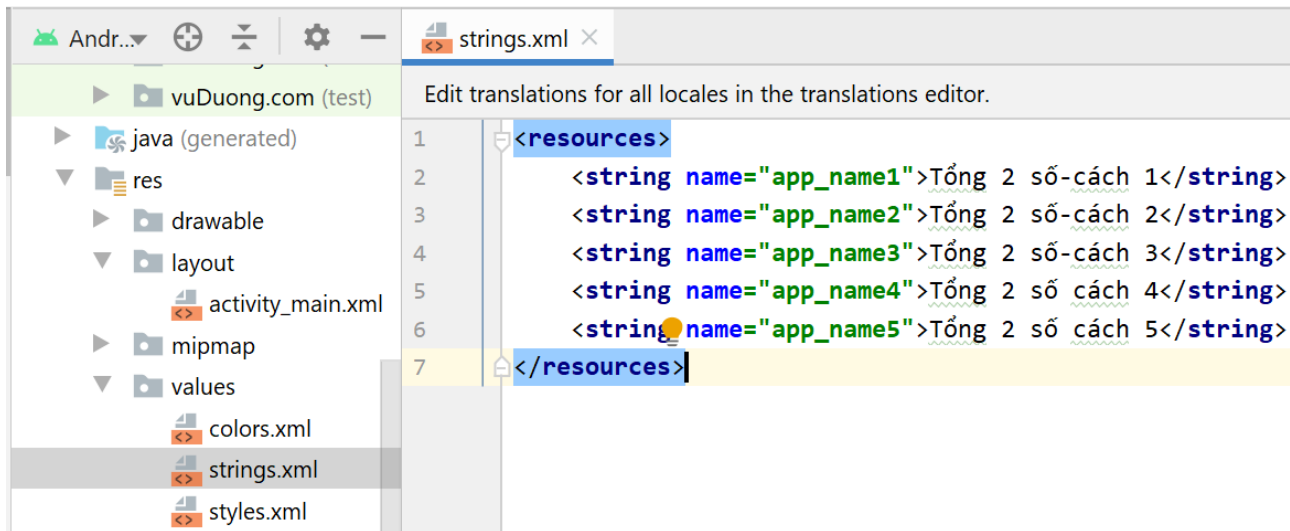
    <TextView
        android:id="@+id/txtKetQua"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="3"
        android:gravity="center"
        android:text="Kết quả" />
</TableRow>

<TableRow
    android:id="@+id/tableRow4"
    android:layout_width="match_parent"
    android:layout_height="wrap_content">

    <Button
        android:id="@+id/btnTong2So"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:text="Tổng 2 số" />
    <Button
        android:id="@+id/btnHieu2So"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:text="Hiệu 2 số" />
</TableRow>
</LinearLayout>
```

1.2. Đặt tiêu đề hiển thị cho cách cách lập trình giao diện

Mở values/string.xml đặt các chuỗi hiển thị cho màn hình



1.3. Kiểm tra các cài đặt trong AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="vuDuong.com">
    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name1"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category
                    android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
</manifest>
```

Lưu ý để sử dụng giá trị đã được đặt tên trong strings.xml đặt làm tiêu đề cho

Activity – dùng mẫu **@tên file tài nguyên/tên biến**

1.4. Viết mã lệnh với 5 kiểu lập trình sự kiện

a. Lắng nghe sự kiện với onClick XML

```
<Button
    android:id="@+id/btnTong2So"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_weight="1"
    android:onClick="tong2so"
    android:text="Tổng 2 số" />
```

Bước 1: Bổ sung thêm chuỗi gán cho thuộc tính onClick của nút lệnh.

Bước 2: thêm phương thức có tên tương ứng trong file .java

```
import androidx.appcompat.app.AppCompatActivity;
import android.view.View;
import android.os.Bundle;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;

public class MainActivity extends AppCompatActivity {
    Button btn_tong2so;
    EditText editSoA, editSoB;
    TextView txtKetQua;
    int a=0;int b=0;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        btn_tong2so=findViewById(R.id.btnTong2So);
        editSoA=(EditText)findViewById(R.id.editSoA);
        editSoB=(EditText)findViewById(R.id.editSoB);
        txtKetQua=(TextView)findViewById(R.id.txtKetQua);
    }
    //cách 1: TẠO HÀM TRÙNG VỚI CHUỖI GÁN CHO THUỘC TÍNH ONCLICK XML
    public void tong2so(View v) {
        a=Integer.parseInt(editSoA.getText()+"");
        b=Integer.parseInt(editSoB.getText()+"");
        txtKetQua.setText((a + b) + "");
    }
}
```

Thực hiện tương tự với nút lệnh Hiệu 2 số

b. Cách 2:Sử dụng sự kiện ngầm định Anomous Listener

Bước 1: Giao diện XML thực hiện như cách 1, tuy nhiên xóa bỏ thuộc tính onClick() của nút lệnh

```
<Button
    android:id="@+id/btnTong2So"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_weight="1"
    android:onClick="tong2so"
    android:text="Tổng 2 số" />
```

Bước 2 viết mã lệnh cho sự kiện ngầm định của nút lệnh

```
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
```

```
import android.widget.EditText;
import android.widget.TextView;

import androidx.appcompat.app.AppCompatActivity;
public class MainActivity extends AppCompatActivity {
    Button btn_tong2so;
    EditText editSoA,editSoB;
    TextView txtKetQua;
    int a=0;int b=0;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        getwidget();
        //cách 2- chọn sự kiện của nút lệnh
        btn_tong2so.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                a=Integer.parseInt(editSoA.getText()+"");
                b=Integer.parseInt(editSoB.getText()+"");
                txtKetQua.setText((a + b) + "");
            }
        });

        private void getwidget() {
            btn_tong2so=findViewById(R.id.btnTong2So);
            editSoA=(EditText)findViewById(R.id.editSoA);
            editSoB=(EditText)findViewById(R.id.editSoB);
            txtKetQua=(TextView)findViewById(R.id.txtKetQua);
        }
    }
}
```

c. Cụ thể hóa phương thức onClick trong giao tiếp OnClickListener

Bước 1 thực hiện như cách 2

Bước 2: Viết mã nguồn cài đặt cụ thể hóa phương thức trong interface

View.OnClickListener

```
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;

import androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity implements
View.OnClickListener {
    Button btn_tong2so,btn_Hieu2so;
    EditText editSoA,editSoB;
```

```
TextView txtKetQua;
int a=0;int b=0;
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    getwidget();
    //cách 3: bắt sự kiện cho nút lệnh
    btn_tong2so.setOnClickListener(this);
    btn_Hieu2so.setOnClickListener(this);
}
private void getwidget() {
    //gán điều khiển giao diện ứng ới đối tượng xử lý
    btn_tong2so = findViewById(R.id.btnTong2So);
    btn_Hieu2so = findViewById(R.id.btnHieu2So);
    editSoA = findViewById(R.id.editSoA);
    editSoB = findViewById(R.id.editSoB);
    txtKetQua = findViewById(R.id.txtKetQua);
}
@Override
public void onClick(View view) {
    if (view==btn_tong2so){
        a=Integer.parseInt(editSoA.getText()+"");
        b=Integer.parseInt(editSoB.getText()+"");
        txtKetQua.setText((a + b) + "");
    }
    else if (view==btn_Hieu2so){
        a=Integer.parseInt(editSoA.getText()+"");
        b=Integer.parseInt(editSoB.getText()+"");
        txtKetQua.setText((a - b) + "");
    }
}
}
```

- Các lúc lệnh gán lắng nghe sự kiện trong Activity được triển khai qua dòng lệnh: `btn_Hieu2so.setOnClickListener(this);`
- Ghi đè phương thức `onClick`. Kiểm tra view nhận sự kiện nếu phù hợp với các id của đối tượng gán sự kiện thì thực hiện hành động tương ứng

```
public void onClick(View view) {
    if (view==btn_tong2so){
        //thực hiện nội dung xử lý tương ứng với nút lệnh
    }
}
```

d. Sử dụng biến nhận sự kiện -Listener in variable

Bước 1; Thực hiện như cách 3

Bước 2 Khai báo nhận sự kiện như sau


```
package vuDuong.com;

import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;

import androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity {
    Button btn_tong2so;
    EditText editSoA, editSoB;
    TextView txtKetQua;
    int a=0;int b=0;
    //cách 4- lắng nghe sự kiện qua biến
    View.OnClickListener myListener = new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            if (view == btn_tong2so) {
                a = Integer.parseInt(editSoA.getText() + "");
                b = Integer.parseInt(editSoB.getText() + "");
                txtKetQua.setText((a + b) + "");
            }
        }
    };
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        getwidget();
        btn_tong2so.setOnClickListener( myListener);
    }

    private void getwidget() {
        btn_tong2so = findViewById(R.id.btnTong2So);
        editSoA = (EditText) findViewById(R.id.editSoA);
        editSoB = (EditText) findViewById(R.id.editSoB);
        txtKetQua = (TextView) findViewById(R.id.txtKetQua);
    }
}
```

- Trong đó biến **myListener** nhận sự kiện và xử lý các công việc trong phương thức `onClick`
- Các View nhận sự kiện thông qua biến đã lắng nghe theo cú pháp:
btn_tong2so.setOnClickListener(myListener);

e. Lớp nhận sự kiện – Explicit class Listener

Bước 1: thực hiện như cách trên

Bước 2 ;Xây dựng 1 lớp độc lập nhận sự kiện

Bước 3: Các View nhận sự kiện tương tự như biến đã thực hiện ở bước trên

```
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;

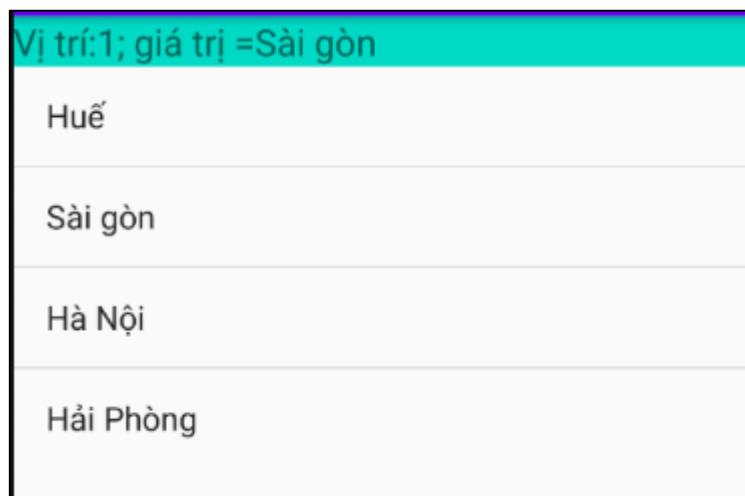
import androidx.appcompat.app.AppCompatActivity;

public class MainActivity5 extends AppCompatActivity {
    Button btn_tong2so;
    EditText editSoA, editSoB;
    TextView txtKetQua;
    int a=0;int b=0;
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        getwidget();
        //CÁCH 5- TẠO ĐỐI TƯỢNG LẮNG NGHE SỰ KIỆN GÁN CHO NÚT LỆNH
        btn_tong2so.setOnClickListener(new DoSomething());
    }
    private void getwidget() {
        btn_tong2so = findViewById(R.id.btnTong2So);
        editSoA = (EditText) findViewById(R.id.editSoA);
        editSoB = (EditText) findViewById(R.id.editSoB);
        txtKetQua = (TextView) findViewById(R.id.txtKetQua);
    }
    //CÁCH 5- TẠO LỚP LẮNG NGHE SỰ KIỆN
    protected class DoSomething implements View.OnClickListener {
        @Override
        public void onClick(View view) {
            if(view.getId()==R.id.btnTong2So){
                a=Integer.parseInt(editSoA.getText()+"");
                b=Integer.parseInt(editSoB.getText()+"");
                txtKetQua.setText((a + b) + "");
            }
        }
    }
}
```

2. Thực hành với ListView

2.1. ListView đơn giản

Giả sử có giao diện đơn giản hiển thị tên các thành phố lớn như sau:



Thực hiện chuẩn bị dữ liệu và đổ dữ liệu lên ListView như hình minh họa. Khi chọn 1 dòng trong ListView thì vị trí và giá trị hiển trở lại TextBox có nền xanh trên đỉnh màn hình.

Thiết kế giao diện:

```
<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:text="selection"
        android:textSize="18sp"
        android:background="@color/colorAccent"/>

    <ListView
        android:id="@+id/lvPerson"
        android:layout_width="match_parent"
        android:layout_height="wrap_content" >
    </ListView>
</LinearLayout>
```

a. Nguồn dữ liệu là mảng

Bước 1: Khởi tạo dữ liệu chuẩn bị đổ lên ListView

Bước 2: Lấy ListView từ giao diện

Bước 3: Tạo adapter

Bước 4: Đặt listView adapter

Bước 5: Xử lý Các thao tác lắng nghe sự kiện của ListView

Bước 6: Thực thi và trải nghiệm người dùng.

```
import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;
import android.view.View;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.ArrayAdapter;
import android.widget.ListView;
import android.widget.TextView;

public class MainActivity extends AppCompatActivity {
    ListView lv;
    TextView txt;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        //1.khởi tạo dữ liệu cho mảng arr
        String arr[]={"Huế","Sài gòn","Hà Nội","Hải Phòng"};
        //2.Lấy đối tượng listview dựa vào id
        lv=(ListView)findViewById(R.id.lvPerson);
        txt=(TextView) findViewById(R.id.txtSelection);
        //3.gán nguồn cho ArrayAdapter
        ArrayAdapter<String> adapter=new ArrayAdapter<String>(
            this,android.R.layout.simple_list_item_1,arr);
        //4.đưa datasource vào list view
        lv.setAdapter(adapter);
        //5, thiết lập sự kiện cho listView khi chọn phần tử
        lv.setOnItemClickListener(new
            AdapterView.OnItemClickListener(){
                @Override
                public void onItemClick(AdapterView<?> arg0, View arg1, int
                    arg2, long arg3) {
                    // TODO Auto-generated method stub
                    txt.setText("Vị trí:" +arg2+"; giá trị =" +arr[arg2]);
                }
            });
    }
}
```

Chú ý: Adapter và nguồn dữ liệu luôn tham chiếu tới cùng kiểu dữ liệu

```
final String arr[]={"Huế","Sài gòn","Hà Nội","Hải Phòng"};
```

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

b. Nguồn dữ liệu là ArrayList

Các bước xử lý cách làm tương tự như mảng

```
import androidx.appcompat.app.AppCompatActivity;  
import java.util.ArrayList;  
import android.os.Bundle;  
import android.view.View;  
import android.widget.AdapterView;  
import android.widget.ArrayAdapter;  
import android.widget.ListView;  
import android.widget.TextView;  
  
public class MainActivity extends AppCompatActivity {  
    //khai báo các control dùng trong chương trình  
    TextView txtSelection;  
    ListView lv;  
    //khai báo arrayList  
    ArrayList<String> arrList=null;  
    //tùy biến theo adapter mặc định  
    ArrayAdapter<String> adapter=null;  
    //tùy biến theo adapter mới  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
        //kết nối các control với Activity  
        txtSelection=(TextView)findViewById(R.id.txtSelection);  
        lv=(ListView)findViewById(R.id.lvPerson);  
        //1. tạo 1 arraylisst object  
        arrList = new ArrayList<String>();  
        arrList.add("Hà nội");  
        arrList.add("Hải phòng");  
        arrList.add("Huế");  
        arrList.add("Đà Nẵng");  
        //2. gán DataSource 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() {  
  
            @Override  
            public void onClick(View v) {
```

```
// TODO Auto-generated method stub
arrList.add(txtTen.getText()+" ");
adapter.notifyDataSetChanged();
txtTen.setText("");
    }
});
//5. Xử lý sự kiện chọn một phần tử trong listView
lv.setOnItemClickListener(new AdapterView.OnItemClickListener()
{
    @Override
    public void onItemClick(AdapterView<?> arg0, View arg1, int
arg2, long arg3) {
        txtSelection.setText("vị trí:"+arg2+" giá
trị:"+arrList.get(arg2));
    }
});
//6 Xử lý sự kiện LongClick
lv.setOnItemLongClickListener(new
AdapterView.OnItemLongClickListener() {
    @Override
    public boolean onItemLongClick(AdapterView<?> arg0, View arg1,
int arg2, long arg3) {
        arrList.remove(arg2);
        adapter.notifyDataSetChanged();
        return false;
    }
});
}
```

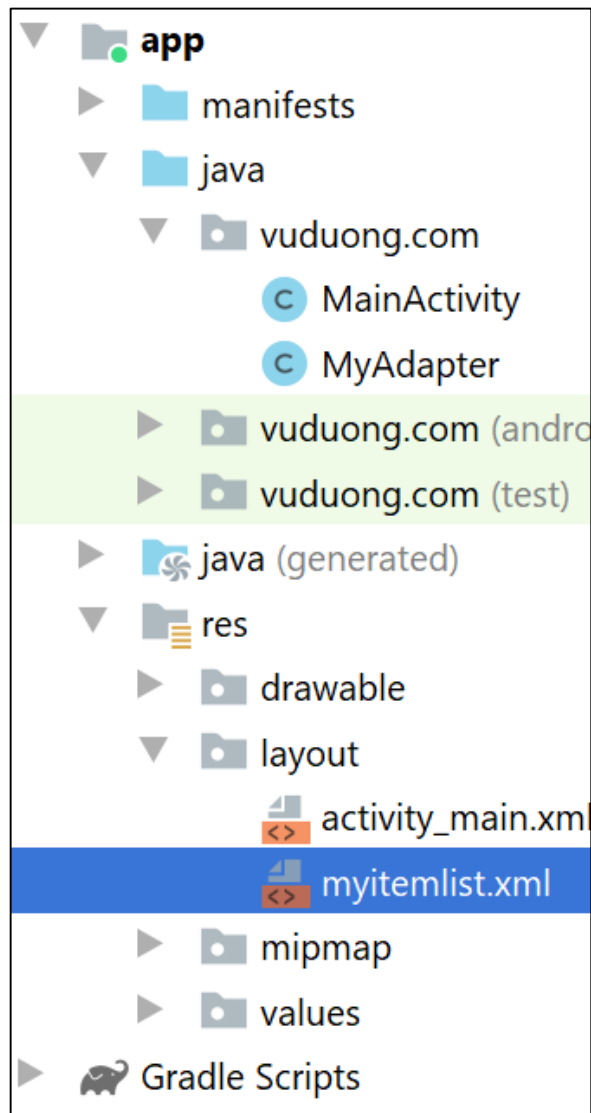
2.2. Tùy biến ListView

Giả sử tùy biến ListView đơn giản bằng cách hiển thị các dòng có thứ tự như mô tả sau

0	Hà nội
1	Hải phòng
2	Huế
3	Đà nẵng

Để thực hiện tùy biến listView ta thực hiện theo các bước trong bài 3 như sau

a. Cấu trúc Project



b. Tạo file myitemlist.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:weightSum="3"
    android:layout_height="match_parent">

    <TextView
        android:id="@+id/txtStt"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:gravity="center"
        android:text="1" />

    <TextView
        android:id="@+id/txtNoiDung"
```

```
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_weight="2"
        android:gravity="center"
        android:text="Hà nội" />
    </LinearLayout>
```

c. Activity_main.xml

<p>Component Tree</p> <ul style="list-style-type: none">LinearLayout (v)Ab textView1Ab txtTenAb txtSelectionAb btnNhaplvPerson	<pre><LinearLayout xmlns:android="http://schemas.android.com/apk/res/an droid" xmlns:tools="http://schemas.android.com/tools" 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="#deb887" android:text="Nhập tên:" /> <EditText android:id="@+id/editTen" android:layout_width="match_parent" android:layout_height="wrap_content" android:ems="10" android:inputType="text" /> <TextView android:id="@+id/txtSelection" android:layout_width="match_parent" android:layout_height="wrap_content" android:textSize="20sp" android:background="#EAF1F0" /> <Button android:id="@+id/btnNhap" android:layout_width="match_parent" android:layout_height="wrap_content" android:text="Nhập" /> <ListView android:id="@+id/lvPerson" android:layout_width="match_parent" android:layout_height="wrap_content" android:background="#cccccc"></ListView> </LinearLayout></pre>
---	---

- **editTen:** Nhập dữ liệu tên các thành phố

- **txtSelection**: hiển thị kết quả lựa chọn dòng trong ListView
- **btnNhap**: Nút lệnh thêm thông tin trong editTen vào ListView
- **lvPerson**: ListView nhận và chứa dữ liệu

Cài đặt các lớp nghiệp vụ

d. MyAdapter.java

```
class MyAdapter extends ArrayAdapter{
    Activity context;
    int layoutID;
    ArrayList<String> list=null;

    public MyAdapter(@NonNull Activity context, int resource, @NonNull
List objects) {
        super(context, resource, objects);
        this.context = context;
        this.layoutID = resource;
        this.list = (ArrayList<String>) objects;
    }

    public View getView(int position, @Nullable View convertView,
@NonNull ViewGroup parent) {
        LayoutInflater inflater = context.getLayoutInflater();
        convertView = inflater.inflate(layoutID, null);
        if ((list.size() > 0) && (position >= 0)) {
            //Laays dongf thuws i
            final TextView txtSTT =
convertView.findViewById(R.id.txtStt);
            final TextView txtNoiDung =
convertView.findViewById(R.id.txtNoiDung);
            //Laays ban ghi thu possition gans cho thanh phan tuong ung
            txtSTT.setText(position + "");
            txtNoiDung.setText(list.get(position).toString() + "");
        }
        return convertView;
    }
}
```

Hàm tạo trong lớp dùng để khởi tạo các giá trị từ MainActivity truyền vào.

- context : là Activity từ Main
- layoutId: Là layout custom do ta tạo (my_item_layout.xml)
- **list**: Danh sách nhân viên truyền từ Main

Phương thức getView dùng để custom layout đây là phương thức có sẵn trong Activity. Phương thức này cần override lại hàm này để tùy biến một số tham số. Hàm có các tham số:

- position : là vị trí của phần tử trong danh sách nhân viên
- convertView: convertView, dùng nó để xử lý Item
- parent : Danh sách nhân viên truyền từ Main
- View: trả về chính convertView

LayoutInflater

Đây là một service của android cho phép định nghĩa và hiển thị lúc runtime (render) một view đc định nghĩa từ mã XML xác định trước. View được dùng trong trường hợp cần nhiều hơn một kiểu layout trong cùng một module logic

e. MainActivity.java

```
import androidx.appcompat.app.AppCompatActivity;
import java.util.ArrayList;
import android.os.Bundle;
import android.view.View;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ListView;
import android.widget.TextView;

public class MainActivity extends AppCompatActivity {
    //khai báo các control dùng trong chương trình
    EditText txtTen;
    TextView txtSelection;
    Button btn;
    ListView lv;
    //khai báo arrayList
    ArrayList<String>arrList=null;
    //tùy biến theo adapter mặc định
    //tùy biến theo adapter mới
    MyAdapter adapter=null;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        //kết nối các control với Activity
        txtTen=(EditText)findViewById(R.id.editTen);
        txtSelection=(TextView)findViewById(R.id.txtSelection);
        lv=(ListView)findViewById(R.id.lvPerson);
        //1. tạo 1 arraylisst object
        arrList = new ArrayList<String>();
        arrList.add("Hà nội");
        arrList.add("Hải phòng");
        arrList.add("Huế");
```

```
arrList.add("Đà nẵng");
//2. gán DataSource vào ArrayAdapter
adapter=new MyAdapter(MainActivity.this,
    R.layout.myitemlist,
    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() {

    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub
        arrList.add(txtTen.getText()+" ");
        adapter.notifyDataSetChanged();
        txtTen.setText("");
    }
});
//5. Xử lý sự kiện chọn một phần tử trong listView
lv.setOnItemClickListener(new AdapterView.OnItemClickListener()
{

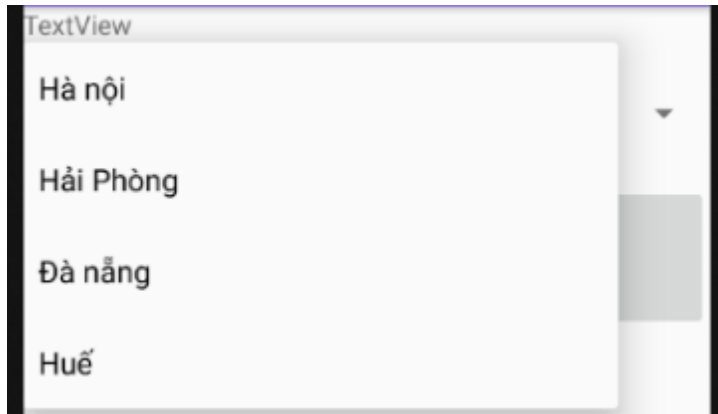
    @Override
    public void onItemClick(AdapterView<?> arg0, View arg1, int
arg2,long arg3) {
        // TODO Auto-generated method stub
        txtSelection.setText("vị trí:"+arg2+" giá
trị:"+arrList.get(arg2));
        txtTen.setText(arrList.get(arg2)+"");
    }
});
//6, Xử lý sự kiện LongClick
lv.setOnItemLongClickListener(new
AdapterView.OnItemLongClickListener() {

    @Override
    public boolean onItemLongClick(AdapterView<?> arg0, View
arg1,
                                int arg2, long arg3) {
        // TODO Auto-generated method stub
        arrList.remove(arg2);
        adapter.notifyDataSetChanged();
        return false;
    }

});
}
```

3. Spinner

Cho màn hình thực thi có giao diện như sau



Spinner nhận dữ liệu là danh sách các thành phố. Chọn 1 dòng trong Spinner thì dữ liệu dòng chọn hiển thị lên TextView. Các bước thực hiện như sau

Bước 1: Thiết kế giao diện **activity_main.xml**

Cấu trúc cây thành phần	Code XML
	<pre><?xml version="1.0" encoding="utf-8"?> <LinearLayout android:layout_width="match_parent" android:layout_height="match_parent" android:orientation="vertical" tools:context=".MainActivity"> <TextView android:id="@+id/txtHienThi" android:layout_width="match_parent" android:layout_height="wrap_content" android:text="Hiển thị lựa chọn" android:textColor="@color/colorAccent" android:textSize="28sp" /> <Spinner android:id="@+id/spinThuDo" android:layout_width="match_parent" android:layout_height="74dp" /> </LinearLayout></pre>

MainActiviy.java

```
public class MainActivity extends AppCompatActivity {
    Spinner spinThuDo;
    TextView txtHienThi;
    ArrayList<String> listThuDo=new ArrayList<String>();
    ArrayAdapter<String> adapter=null;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        //chuẩn bị dữ liệu
        listThuDo.add("Hà nội");
```

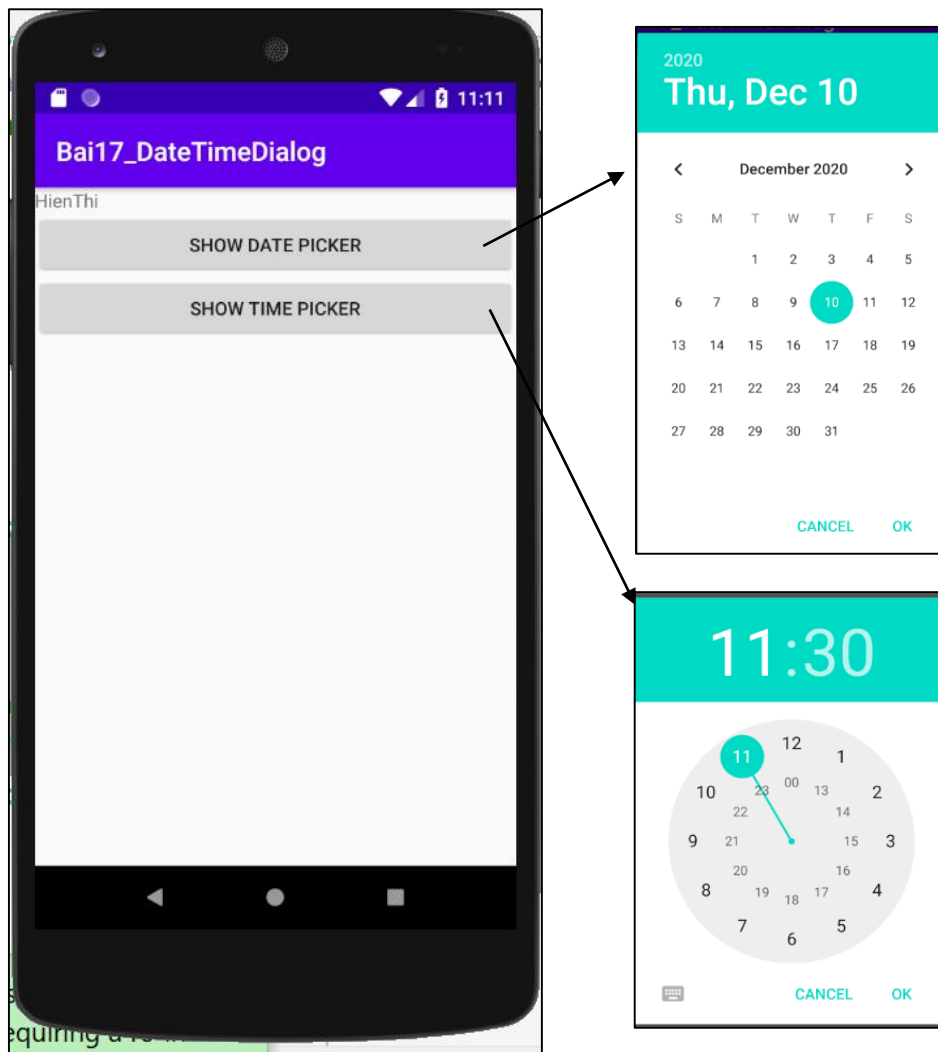
```
listThuDo.add("Hải Phòng");
listThuDo.add("Đà Nẵng");
listThuDo.add("Huế");
//lấy điều khiển
txtHienThi=findViewById(R.id.txtHienThi);
spinThuDo=findViewById(R.id.spinThuDo);
//set adapter
adapter=new ArrayAdapter<String>(
    this,
    R.layout.support_simple_spinner_dropdown_item,
    listThuDo);
adapter.setDropDownViewResource(
    R.layout.support_simple_spinner_dropdown_item);
spinThuDo.setAdapter(adapter);
//gán sự kiện cho spinner
spinThuDo.setOnItemClickListener(
    new AdapterView.OnItemClickListener() {
        @Override
        public void onItemClick(AdapterView<?> parent,
            View view, int position, long id) {
            txtHienThi.setText(listThuDo.get(position));
        }
        @Override
        public void onNothingSelected(AdapterView<?> parent)
        {}
    });
}
```

Chú ý:

- R.layout.**support_simple_spinner_dropdown_item** là layout mặc định của Android hỗ trợ spinner hiển thị và xử lý dữ liệu.
- **spinThuDo.setOnItemClickListener**: gán sự kiện cho Spinner

4. TimePicker – DatePicker

Cho giao diện màn hình chính như sau:

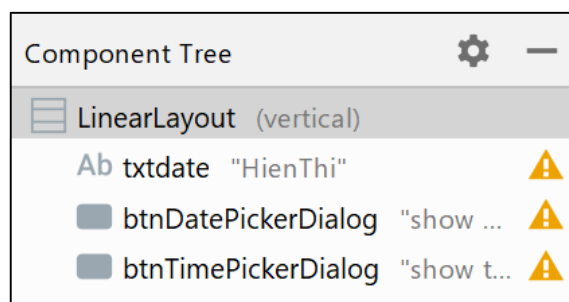


Trải nghiệm ứng dụng:

- Chọn ShowDatePicker: cửa sổ DatePicker hiển thị. Khi chọn ShowTimePicker cửa sổ TimePicker hiển thị.
- Trong khi các cửa sổ hiển thị do người dùng chọn date hay time; chọn ok. Thông tin vừa chọn hiển thị trong text minh họa.

Thực hiện

- Thiết kế giao diện



Activity_main.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">

    <TextView
        android:id="@+id/txtdate"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="HienThi" />

    <Button
        android:id="@+id/btnDatePickerDialog"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="show date picker" />

    <Button
        android:id="@+id/btnTimePickerDialog"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="show time picker" />
</LinearLayout>
```

MainActivity.java

```
package duong.com;
import androidx.appcompat.app.AppCompatActivity;
import android.app.DatePickerDialog;
import android.app.TimePickerDialog;
import android.os.Bundle;
import android.service.controls.actions.BooleanAction;
import android.view.View;
import android.widget.AnalogClock;
import android.widget.Button;
import android.widget.DatePicker;
import android.widget.LinearLayout;
import android.widget.TextView;
import android.widget.TimePicker;

public class MainActivity extends AppCompatActivity {
    Button btnDatePicker, btnTimePicker;
    TextView txtDate;
```

```
LinearLayout mylayout;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    getWidget();
    doWork();
}

private void getWidget() {
    btnDatePicker=findViewById(R.id.btnDatePickerDialog);
    btnTimePicker=findViewById(R.id.btnTimePickerDialog);
    txtDate=findViewById(R.id.txtdate);
}

//Xử lý thông tin khi gọi hiển thị xử lý thời gian
private void doWork() {
    btnDatePicker.setOnClickListener(
        new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                DatePickerDialog.OnDateSetListener callback=
                    new DatePickerDialog.OnDateSetListener() {
                        @Override
                        public void onDateSet(DatePicker datePicker,
                            int i, int i1, int i2) {
                            txtDate.setText(i2+"/"+i1+"/"+i);
                        }
                    };
                DatePickerDialog pic=new DatePickerDialog(
                    MainActivity.this,callback,
                    2020,11,10);
                pic.setTitle("my datetime picker");
                pic.show();
            }
        });

    btnTimePicker.setOnClickListener(
        new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                TimePickerDialog.OnTimeSetListener callback=
                    new TimePickerDialog.OnTimeSetListener(){
                        @Override
                        public void onTimeSet(TimePicker timePicker,
                            int i, int i1) {
                            txtDate.setText(i+"-"+i1);
                        }
                    }
            }
        });
}
```



```
        };  
        TimePickerDialog time=new TimePickerDialog(  
            MainActivity.this, callback,  
            11,30,true);  
        time.setTitle("my time picker");  
        time.show();  
    }  
});  
}  
}
```

- Interface `OnDateSetListener` có phương thức trừu tượng **`onDateSet`**, trong phương thức này ta xử lý sự thay đổi dữ liệu trong **`DatePickerDialog`**.
- Interface `OnTimeSetListener` có phương thức trừu tượng **`onTimeSet`**, trong phương thức này ta xử lý sự thay đổi giờ – phút trong **`TimePickerDialog`**.
- 2 call back listener trên rất là hữu dụng, dựa vào hai biến này sẽ kiểm tra sự thay đổi dữ liệu trong các Dialog.

5. RecyclerView.

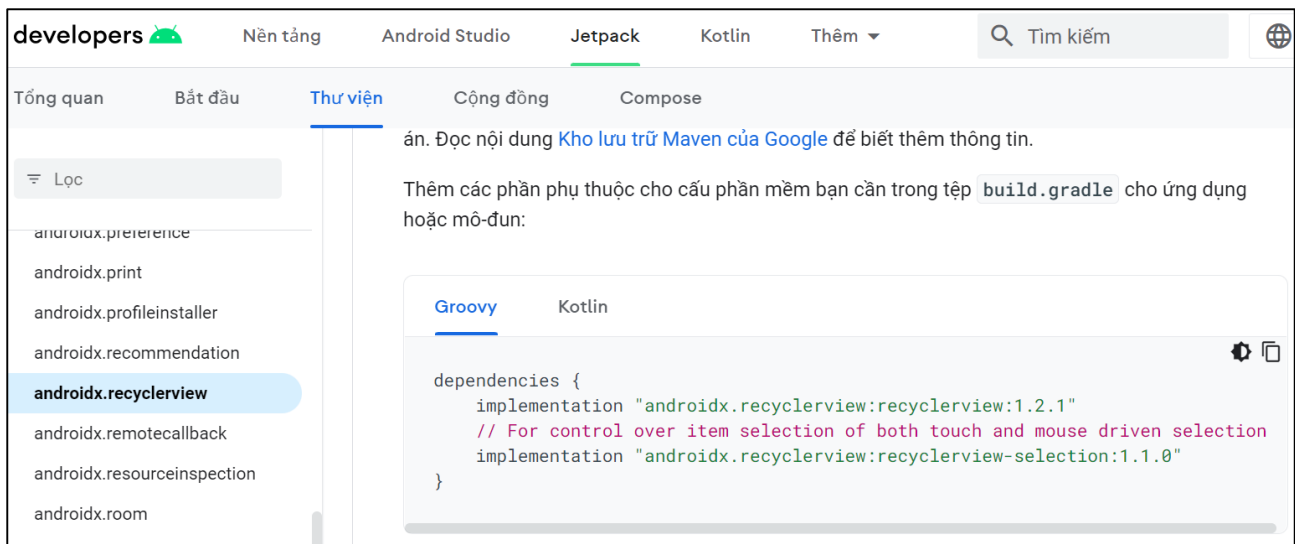
1. Thêm RecyclerView support library vào gradle build file của module
2. Định nghĩa ra model class để sử dụng data source
3. Thiết kế giao diện: Thêm RecyclerView vào *.xml mà bạn muốn hiển thị
4. Tạo một tệp XML để xác định hiển thị một item trong recyclerView
5. Tạo ra RecyclerView.Adapter và ViewHolder để gán dữ liệu cho các item
6. Kết nối adapter với data source để đưa vào RecyclerView.

Chi tiết mô tả như sau:

Bước 1: tra cứu thư viện hỗ trợ RecyclerView theo đường dẫn sau:

<https://developer.android.com/jetpack/androidx/releases/recyclerview>.

Màn hình hiển thị:



Copy dòng lệnh sau vào Buile.grade (modun):

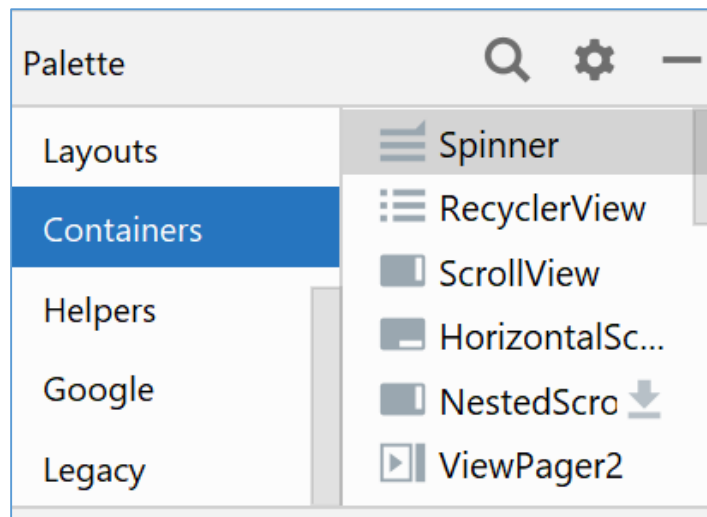
implementation "androidx.recyclerview:recyclerview:1.2.1"

đưa vào Buile.grade (modun)

```
dependencies {  
  
    implementation 'androidx.appcompat:appcompat:1.5.0'  
    implementation 'com.google.android.material:material:1.6.1'  
    implementation 'androidx.constraintlayout:constraintlayout:2.1.4'  
    testImplementation 'junit:junit:4.13.2'  
    androidTestImplementation 'androidx.test.ext:junit:1.1.3'  
    androidTestImplementation 'androidx.test.espresso:espresso-core:3.4.0'  
    //thêm thư viện hỗ trợ recyclerView  
    implementation "androidx.recyclerview:recyclerview:1.2.1"  
}
```

Bước 2: Định nghĩa ra model class để sử dụng data source. Căn cứ vào bài toán cần xử lý, định nghĩa 1 java class quản lý thông tin hiển thị.

Bước 3: Thiết kế giao diện: Thêm RecyclerView vào *.xml hiển thị. Trở lại màn hình thiết kế giao diện mục Containers\RecyclerView. Kéo và thả/bổ sung mã lệnh thiết kế điều khiển vào màn hình hiển thị



Bước 4: Tạo một tệp XML để xác định hiển thị một item trong recyclerView: thực hiện tương tự như với custom ListView.

Bước 5: Tạo ra RecyclerView.Adapter và ViewHolder để gắn dữ liệu cho các item

- Xây dựng lớp tùy biến **UserViewHolder** kế thừa **RecyclerView.ViewHolder**. Lớp chứa điều khiển hiển thị cho item của RecyclerView. Bước này nội dung làm tương tự như getView của ListView
- Xây dựng lớp ngoài tùy biến Adapter (Lớp_tùy_biến_Adapter) kế thừa lớp RecyclerView.Adapter<Lớp_tùy_biến_Adapter.UserViewHolder>. Thực hiện ghi đè 3 phương thức

Bước 6: Kết nối adapter với data source để đưa vào RecyclerView. Thực hiện tương tự như ListView.