package com.android.training.statepattern;

import android.app.Activity;

import android.os.Bundle;

import android.view.View;

import android.widget.Button;

public class StatePatternActivity extends Activity implements View.OnClickListener{

private Button startStatePattern;

private MyHandler handler;

/\*\* Called when the activity is first created. \*/

@Override

public void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.main);

startStatePattern = (Button)findViewById(R.id.button1);

startStatePattern.setOnClickListener(this);

handler = new MyHandler(this);

}

public void onClick(View v){

if(v.equals(startStatePattern)){

handler.sendEmptyMessage(1);

}

}

}

…………………………………………………………………….

package com.android.training.statepattern;

import android.os.Handler;

import android.os.Message;

import android.widget.Toast;

public class MyHandler extends Handler {

private StatePatternActivity MainActivity;

MyHandler(StatePatternActivity a){

this.MainActivity = a;

}

//state transition logic is strongly coupled with the different states...

public void handleMessage(Message msg){

switch(msg.what){

case 1:

////do the processing

Toast.makeText(MainActivity.getApplicationContext(),"Inside State 1", 2000).show();

this.sendEmptyMessage(2);

break;

case 2:

///do the processing

Toast.makeText(MainActivity.getApplicationContext(),"Inside State 2", 2000).show();

this.sendEmptyMessage(3);

break; case 3:

///do the processing

Toast.makeText(MainActivity.getApplicationContext(),"Inside State 3", 2000).show();

this.sendEmptyMessage(4);

break;

case 4:

break;

default:

break;

}

}

}