Manisha Vyas

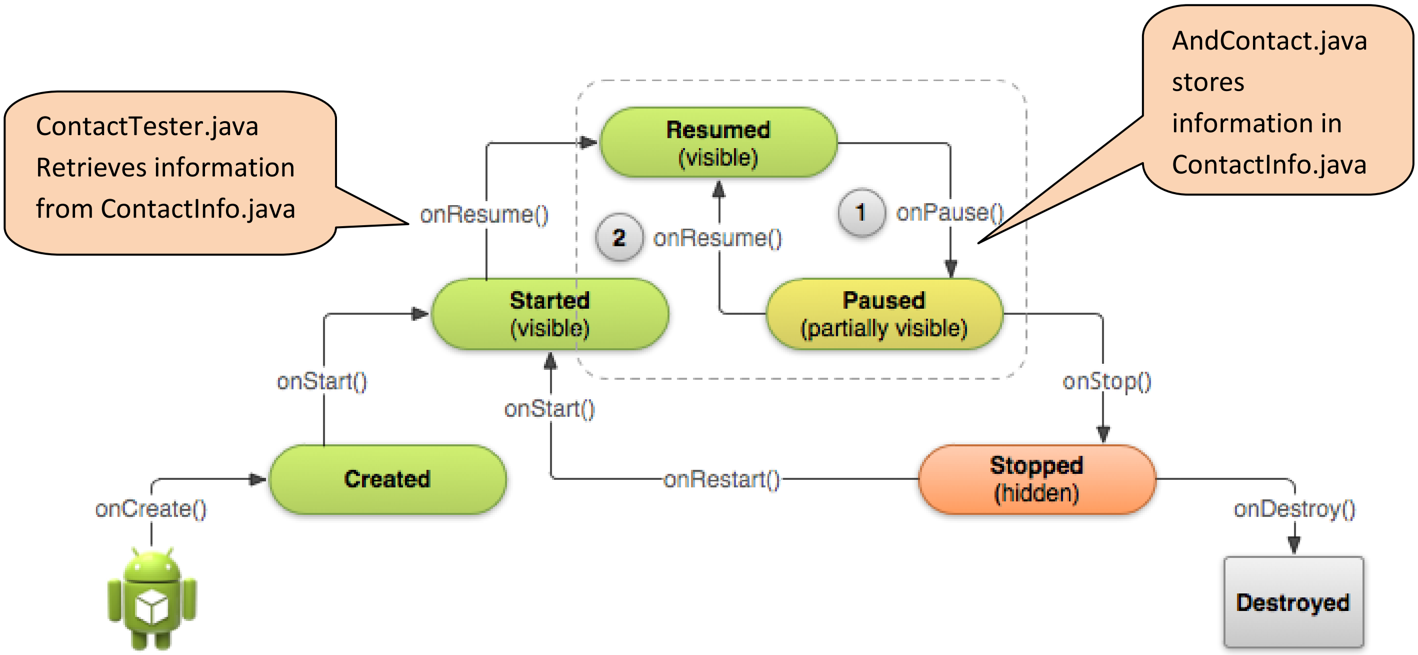
Question.

Saving Activity State of the names entered in a Tab Activity

Modify the Tab Activity project by

- The data entered in AndContacts will be saved.

  Continue Tab Activity Project



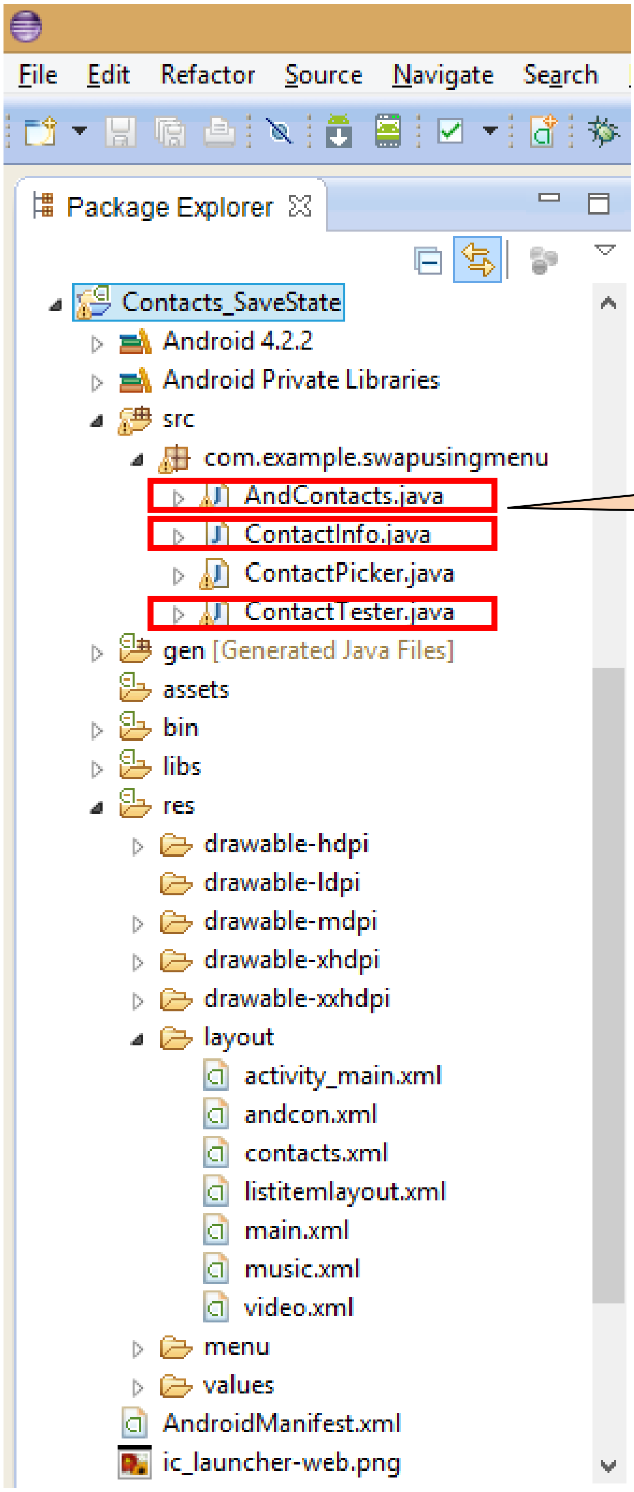
ContactTester.java Retrieves information from ContactInfo.java

AndContact.java stores information in ContactInfo.java

We need to save Activity when Pause() because ::

Think of an Example :: What if you are typing some really important message in NOTEPAD of your phone and Suddenly phone rings and after you disconnect from phone when you return to NOTEPAD screen and you find your message is Not Saved !!

That is the reason we will work on Saving State of an Activity...



```
Contacts_SaveState
    Android 4.2.2
    Android Private Libraries
    src
        com.example.swapusingmenu
            AndContacts.java
            ContactInfo.java
            ContactPicker.java
            ContactTester.java
    gen [Generated Java Files]
    assets
    bin
    libs
    res
        drawable-hdpi
        drawable-ldpi
        drawable-mdpi
        drawable-xhdpi
        drawable-xxhdpi
        layout
            activity_main.xml
            andcon.xml
            contacts.xml
            listitemlayout.xml
            main.xml
            music.xml
            video.xml
        menu
        values
    AndroidManifest.xml
    ic_launcher-web.png
```

We will be mainly modifying these three classes to save Activity State

//AndContacts.java

```
package com.example.swapusingmenu;

import java.util.ArrayList;

import android.app.Activity;
import android.app.~~TabActivity~~;
import android.content.ContentProviderOperation;
import android.content.Context;
import android.content.Intent;
import android.content.OperationApplicationException;
import android.content.SharedPreferences;
import android.os.Bundle;
import android.os.RemoteException;
import android.provider.ContactsContract;
import android.provider.ContactsContract.CommonDataKinds.*;
import android.provider.ContactsContract.*;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TabHost;

public class AndContacts extends ~~TabActivity~~ {

       Button saveb,cancelb;
       EditText txtname,txtemail,txtphone,txtpostaladd;
       TabHost mTabHost = null;
       ContactInfo dataModel;
       @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.andcon);

        mTabHost = getTabHost();

        mTabHost.addTab(mTabHost.newTabSpec("tab_test1").setIndicator("Contacts",
getResources().getDrawable(R.drawable.contact)).setContent(R.id.contactsLayout));
        mTabHost.addTab(mTabHost.newTabSpec("tab_test2").setIndicator("Music",
getResources().getDrawable(R.drawable.music)).setContent(R.id.musicLayout));
        mTabHost.addTab(mTabHost.newTabSpec("tab_test3").setIndicator("Video",
getResources().getDrawable(R.drawable.video)).setContent(R.id.videoLayout));

        dataModel = new ContactInfo();
        mTabHost.setCurrentTab(0);
       saveb = (Button) findViewById(R.id.buttonsave);
       cancelb = (Button) findViewById(R.id.buttoncancel);
       txtname = (EditText) findViewById(R.id.txtname);
       txtemail = (EditText) findViewById(R.id.txtemail);
       txtphone = (EditText) findViewById(R.id.txtphone);
       txtpostaladd = (EditText) findViewById(R.id.txtpostaladdress);

        saveb.setOnClickListener(new View.OnClickListener() {
                    @Override
```

```
                public void onClick(View v) {
                       Intent result = new Intent();
                       result.putExtra("Name",txtname.getText().toString());

                       setResult(RESULT_OK, result);

                //save current contact fields to a persistent preference file
            saveData();

   //save phone and name fields to contacts database, so database can be built up
            addcontact();
            setResult(Activity.RESULT_OK); //passing integer result code ok back
            finish(); //close the activity
                }
          });

     cancelb.setOnClickListener(new View.OnClickListener() {
                @Override
                public void onClick(View v) {

                       setResult(RESULT_CANCELED, null);
                       finish();
                }
          });
 }

   //when activity goes into background, save the data
 public void onPause() {
     super.onPause();
     saveData();
 }

//will write and store the single data set into a preference data file for sharing
 public void saveData(){

      dataModel.setName( txtname.getText().toString() );
     //set str from name field into dataModel
     dataModel.setPhone( txtphone.getText().toString() );
     dataModel.setEmail( txtemail.getText().toString() );
     dataModel.setPostalAddr( txtpostaladd.getText().toString() );

     //use this file to store, and allow access for owner of the app
     SharedPreferences prefs = getSharedPreferences("data_file.txt",
                Context.MODE_PRIVATE); //.MODE_WORLD_READABLE);
     SharedPreferences.Editor editor = prefs.edit();

     editor.putString("NAME_KEY", dataModel.getName());
    //save string data as key/value pair
     editor.putString("PHONE_KEY", dataModel.getPhone());
     editor.putString("EMAIL_KEY", dataModel.getEmail());
     editor.putString("POSTALADDR_KEY", dataModel.getPostalAddr());
     editor.commit();
 }
```

```
      protected void addcontact() {
             ArrayList ops = new
                ArrayList();
   int rawContactInsertIndex = ops.size();

   ops.add(ContentProviderOperation.newInsert(RawContacts.CONTENT_URI)
        .withValue(RawContacts.ACCOUNT_TYPE, null)
        .withValue(RawContacts.ACCOUNT_NAME,null )
      .build());

   //phone field
   ops.add(ContentProviderOperation.newInsert(ContactsContract.Data.CONTENT_URI)
           .withValueBackReference(ContactsContract.Data.RAW_CONTACT_ID,
                       rawContactInsertIndex)
           .withValue(Data.MIMETYPE, Phone.CONTENT_ITEM_TYPE)
           .withValue(Phone.NUMBER, txtphone.getText().toString())
           .build());

   //name field
   ops.add(ContentProviderOperation.newInsert(Data.CONTENT_URI)
           .withValueBackReference(Data.RAW_CONTACT_ID, rawContactInsertIndex)
           .withValue(Data.MIMETYPE, StructuredName.CONTENT_ITEM_TYPE)
           .withValue(StructuredName.DISPLAY_NAME,
                       txtname.getText().toString())
           .build());

   //email field
   ops.add(ContentProviderOperation.newInsert(ContactsContract.Data.CONTENT_URI)
           .withValueBackReference(Data.RAW_CONTACT_ID, 0)


.withValue(ContactsContract.Data.MIMETYPE,ContactsContract.CommonDataKinds.Email.CONTENT
_ITEM_TYPE)
   .withValue(ContactsContract.CommonDataKinds.Email.DATA,
txtemail.getText().toString())
   .withValue(ContactsContract.CommonDataKinds.Email.TYPE,
ContactsContract.CommonDataKinds.Email.TYPE_HOME)
   .build());


   try {
       //update data to the database, operations are run in batchmode
       getContentResolver().applyBatch(ContactsContract.AUTHORITY, ops);
   } catch (RemoteException e) {
       e.printStackTrace();
   } catch (OperationApplicationException e) {
       e.printStackTrace();
   }

      }

}
```

//Contact Tester.java

```java
package com.example.swapusingmenu;

import android.app.Activity;
import android.content.Context;
import android.content.Intent;
import android.content.SharedPreferences;
import android.content.pm.PackageManager.NameNotFoundException;
import android.database.Cursor;
import android.net.Uri;
import android.os.Bundle;
import android.provider.~~Contacts~~.~~People~~;
import android.util.Log;
import android.view.*;
import android.widget.ArrayAdapter;
import android.widget.AutoCompleteTextView;
import android.widget.Button;
import android.widget.TextView;

public class ContactTester extends Activity {
       private static final int AND_CON_ACTIVITY = 1;
       private static final int PICK_CON_ACTIVITY = 2;

       Button and_conb, pick_conb;
       TextView result_con;
       ContactInfo dataModel;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

       and_conb = (Button) findViewById(R.id.and_conb);
       result_con = (TextView) findViewById(R.id.result_text);
       pick_conb = (Button) findViewById (R.id.pick_conb);
       dataModel = new ContactInfo();
       readData();

        and_conb.setOnClickListener(new View.OnClickListener() {
                    @Override
                    public void onClick(View v) {

                          Intent intent = new
Intent(ContactTester.this,AndContacts.class);
                     startActivityForResult(intent, AND_CON_ACTIVITY);

                    }
             });
        pick_conb.setOnClickListener(new View.OnClickListener() {
                    @Override
                    public void onClick(View v) {
                          Intent intent = new
Intent(Intent.ACTION_PICK,Uri.parse("content://contacts/"));
```

```
                          startActivityForResult(intent, PICK_CON_ACTIVITY);

                    }
             });
    }
    @Override
    public void onActivityResult(int requestCode, int resultCode, Intent data)
    {
      super.onActivityResult(requestCode, resultCode, data);
      switch (requestCode)
        {
              case AND_CON_ACTIVITY:
              {
                   if(resultCode == Activity.RESULT_OK)
                   {
                          readData();
                          String new_con = data.getStringExtra("Name");
                          result_con.setText("Contact Added : "+new_con);

                   }
                   break;
              }
              case PICK_CON_ACTIVITY :
              {
                if (resultCode == Activity.RESULT_OK)
                {
                         Uri contactData = data.getData();
                         Cursor c = ~~managedQuery~~(contactData, null, null, null, null);
                         c.moveToFirst();
                         String new_con =
c.getString(c.getColumnIndexOrThrow(~~People~~.~~NAME~~));
                         result_con.setText("You have selected : "+new_con);
                }
                break;
              }
        }

    }
    @Override //when app returning back to foreground state
    public void onResume() {
        super.onResume();
        readData();  //load data
    }

    //will fetch data from persistent shared preference file
    public void readData(){
        Context context;
        String name;
        String phone;
        String email;
        String postalAddr;

        try {
            // context from pref owner data writer app's package fully qualified name
            context = createPackageContext("my.project.homework8",
Context.MODE_PRIVATE);
            SharedPreferences pref = context.getSharedPreferences("data_file.txt",
Context.MODE_MULTI_PROCESS);
            //fetch data strings off each key in preference file
            name = pref.getString("NAME_KEY", "No Value Found");
            phone = pref.getString("PHONE_KEY", "No Value Found");
            email = pref.getString("EMAIL_KEY", "No Value Found");
            postalAddr = pref.getString("POSTALADDR_KEY", "No Value Found");

            dataModel.setName(name);  dataModel.setPhone(phone); //these may be useful
later
            dataModel.setEmail(email);  dataModel.setPostalAddr(postalAddr);

          //  mainTextView.setText(dataModel.getName()); //set text for displaying name
        }
        catch (NameNotFoundException e) {
            Log.e("Exception Package Name Not Found", e.toString());
        }
    }
    @Override //Option menu/Action menu
    public boolean onCreateOptionsMenu(Menu menu)
    {
        MenuInflater menuInflater = getMenuInflater();
        menuInflater.inflate(R.menu.main, menu);
        return true;
    }
}
```

//ContactInfo.java

```
//  This class saves is Data model class to store info between Resume() and Pause()
package com.example.swapusingmenu;

public class ContactInfo {
       private String name;
       private String phone;
       private String email;
       private String postalAddr;
       public String getName() {
              return name;
       }
       public void setName(String name) {
              this.name = name;
       }
       public String getPhone() {
              return phone;
       }
       public void setPhone(String phone) {
              this.phone = phone;
       }
       public String getEmail() {
              return email;
       }
       public void setEmail(String email) {
              this.email = email;
       }
       public String getPostalAddr() {
              return postalAddr;
       }
       public void setPostalAddr(String postalAddr) {
              this.postalAddr = postalAddr;
       }
}
```

//ContactPicker.java

```
package com.example.swapusingmenu;

import android.app.Activity;
import android.content.Intent;
import android.database.Cursor;
import android.net.Uri;
import android.os.Bundle;
import android.provider.~~Contacts~~.~~People~~;
import android.view.View;
import android.widget.AdapterView;
import android.widget.ListView;
import android.widget.SimpleCursorAdapter;
import android.widget.AdapterView.OnItemClickListener;

public class ContactPicker extends Activity{
        @Override
        public void onCreate(Bundle icicle) {
          super.onCreate(icicle);
          setContentView(R.layout.main);

          Intent intent = getIntent();
          String dataPath = intent.getData().toString();

          final Uri data = Uri.parse(dataPath + "people/");
          final Cursor c = ~~managedQuery~~(data, null, null, null, null);

          String[] from = new String[] {~~People~~.~~NAME~~};
          int[]  to = new int[] { R.id.itemTextView };

          SimpleCursorAdapter adapter = new
~~SimpleCursorAdapter~~(this,R.layout.listitemlayout,c,from,to);
          ListView lv = (ListView)findViewById(R.id.contactListView);
          lv.setAdapter(adapter);

          lv.setOnItemClickListener(new OnItemClickListener() {
            public void onItemClick(AdapterView<?> parent, View view, int pos, long id)
{
              // Move the cursor to the selected item
              c.moveToPosition(pos);
              // Extract the row id.
```

```
               int rowId = c.getInt(c.getColumnIndexOrThrow("_id"));
               // Construct the result URI.
               Uri outURI = Uri.parse(data.toString() + rowId);
               Intent outData = new Intent();
               outData.setData(outURI);
               setResult(Activity.RESULT_OK, outData);
               finish();
             }
           });
         }
}
```

AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.swapusingmenu"
    android:versionCode="1"
    android:versionName="1.0" >

    <application
        android:allowBackup="true"
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/AppTheme" >

        <activity
            android:name="com.example.swapusingmenu.ContactTester"
            android:label="@string/app_name" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>

        <activity
             android:name=".AndContacts"
             android:label="@string/app_name">
        </activity>

        <activity android:name=".ContactPicker"
                  android:label="@string/app_name">
            <intent-filter>
              <action android:name="android.intent.action.PICK"></action>
              <category android:name="android.intent.category.DEFAULT"></category>
              <data android:path="contacts" android:scheme="content"></data>
            </intent-filter>
        </activity>
    </application>
    <uses-sdk android:minSdkVersion="7" />
    <uses-permission android:name="android.permission.READ_CONTACTS"/>
      <uses-permission android:name="android.permission.WRITE_CONTACTS" />
</manifest>
```