

Mục lục

Contents

Bài tập 1: Làm quen với môi trường phát triển điện thoại di động	2
Bài tập 2: Thao tác với Android Virtual Device và DDMS	5
Bài tập 3: Cách tạo Android Project và tìm hiểu cấu trúc bên trong của một Android Project.....	14
Bài tập 4: Cách sửa một số lỗi thường gặp trong Android Project.....	22
Bài tập 5: Tìm hiểu vòng đời của một Ứng Dụng Android.....	24
Bài tập 6: Phân biệt Foreground Lifetime và Visible Lifetime	31
Bài tập 7: Làm quen với các Layout trong Android	40
Bài tập 8: Các kiểu lập trình sự kiện trong Android.....	51
Bài tập 9: Thực hành về Toast Notification và Alert Dialog.....	69
Bài tập 10: Sử dụng TextView, EditText và Button trong Android	72
Bài tập 11: Sử dụng Checkbox và RadioButton trong Android	80
Bài tập 12: Ví dụ tổng hợp TextView, EditText, CheckBox, Button và ImageButton trong Android	89
Bài tập 13: Thực hành về ListView trong Android	103
Bài tập 14: Thực hành về Custom Layout cho ListView trong Android.....	125
Bài tập 15: Thực hành về Spinner trong Android	136
Bài tập 16: Kết hợp Spinner với ListView trong Android.....	139
Bài tập 17: Thực hành về AutocompleteTextView và MultiAutocompleteTextView.....	150
Bài tập 18: Cập nhật DataSource cho AutocompleteTextView lúc Runtime.....	154
Bài tập 19: Thực hành Gridview trong Android	167
Bài tập 21: Thực hành về Tab Selector trong Android	175
Bài tập 22: Thực hành về Menu trong Android.....	184
Bài tập 23: Thực hành về Context Menu trong Android.....	193
Bài tập 24: Thực hành về Intent trong Android	196
Bài tập 25: Tiếp tục củng cố kiến thức Intent, ví dụ tổng hợp Quản Lý Nhân Viên	216
Bài tập 26: Dùng Implicit Intent để viết chương trình gọi điện thoại và nhắn tin SMS	263
Bài tập 27: Đa ngôn ngữ trong Android	274
Bài tập 28: Xử lý tập tin trong Android	279
Bài tập 29: XML Parser trong Android	293
Bài tập 30: Thực hành về Shared Preferences.....	299
Bài tập 31: Cách sử dụng SQLite trong Android	312
Bài tập 32: Tiếp tục củng cố kiến thức SQLite, ví dụ tổng hợp quản lý sách.....	320
Bài tập 33: Sử dụng ContentProvider trong Android	345
Bài tập 34: đa tiến trình trong Android (Multi-Threading)	354

Bài tập 1: Làm quen với môi trường phát triển điện thoại di động

Trong các topic sắp tới, Tôi muốn dành nhiều thời gian để hướng dẫn các sinh viên học lập trình cho thiết bị di động. Cụ thể là hệ điều hành Android. Tôi hi vọng sau khi theo dõi tất cả các Topic từ đầu tới cuối thì các sinh viên sẽ có khả năng triển khai được ứng dụng trên hệ điều hành Android.

Đây là topic đầu tiên Tôi hướng dẫn các bạn làm quen với môi trường phát triển điện thoại di động, các bạn có thể Comment vào topic, trao đổi trực tiếp với Tôi qua comment, qua email, qua điện thoại.... Đừng ngần ngại.

Nội dung kiến thức thực hành:

- Làm quen với môi trường phát triển điện thoại di động
- Sử dụng Eclipse, DDMS – cách thức Debug
- Cách cài đặt Android SDK, Update Android API, Upgrade Firmware
- Sử dụng Android Virtual Device
- Tìm hiểu các thành phần trong Android App
- Tìm hiểu các đơn vị đo lường trong Android
- Tìm hiểu vòng đời của ứng dụng
- Tìm hiểu Google Play

Trong bài tập 1 này, chúng ta phải thực hiện : **Cài đặt được Java JDK, Android SDK, ADT plugin for Eclipse, update Android SDK.**

Các bước chi tiết:

- Tải và cài đặt Java JDK: <http://java.sun.com/javase/downloads/index.jsp> (cài đặt cái này trước, nên chọn bản mới nhất)
- Tải và cài đặt Android SDK: <http://developer.android.com/sdk/index.html> , tải gói **ADT Bundle for Windows**, gói này sẽ chứa các thành phần:
 - Eclipse + ADT plugin

- Android SDK Tools
- Android Platform tools
- Android API mới nhất
- Android system image mới nhất cho Emulator

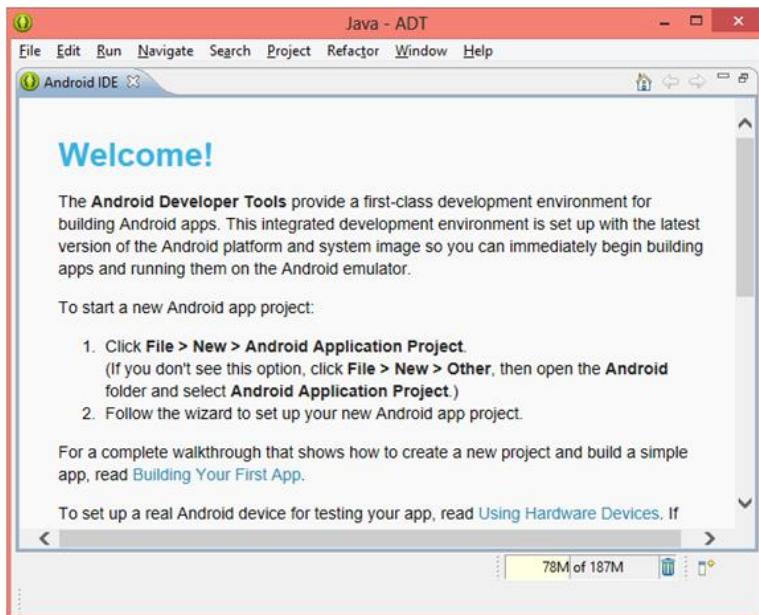
- Cập nhật ADT plugin mới nhất cho Eclipse:

 1. Vào menu Help / chọn Install New Software
 2. Nhập đường dẫn: <https://dl-ssl.google.com/android/eclipse/> và bấm nút Add ở góc bên phải
 3. Chọn các thông số thích hợp để tiến hành cập nhật.

- Tiến hành Cập nhật Android SDK (nếu cần thiết)

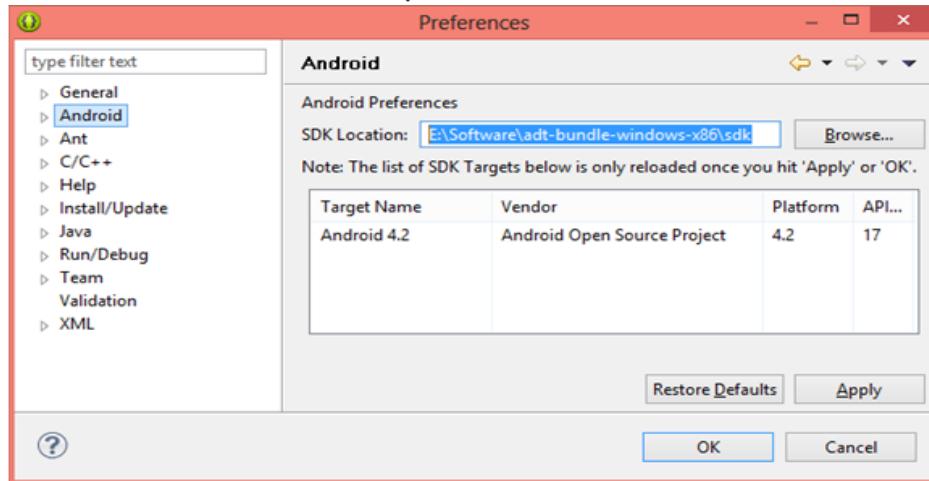
- Tham chiếu Android SDK (bắt buộc, nếu cài gói ADT Bundle for Windows trên trang developer.android.com thì nó đã tự tham chiếu cho chúng ta rồi).

Sau khi cài đặt xong đầy đủ (JDK, Android SDK ...), ta khởi động Elipse trong gói **ADT Bundle for windows**, ta có giao diện như hình bên dưới:



- Đóng màn hình Welcome bằng cách click vào dấu chéo kẽ bên chữ **Android IDE**

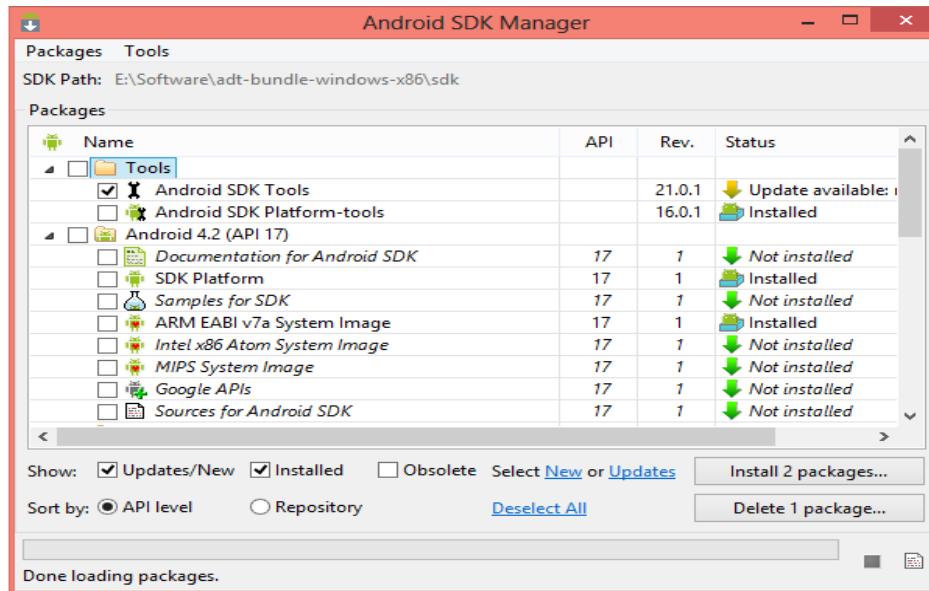
- Để tham chiếu tới Android SDK, ta vào **menu Windows/ Preferences/ chọn Android** và **Browse** tới SDK location mà bạn lưu trữ:



Các bạn nhìn vào hình trên, thấy có Android 4.2, API 17 hiển thị ra. Đó chính là API mới nhất của Android, tùy vào danh sách API được tải về máy nó sẽ hiển thị ra đây. Hiện tại trong máy của Tôi chỉ có 1 API 17, nếu máy bạn có nhiều API khác thì nó sẽ liệt kê ra đây hết.

- Nếu như máy tính của bạn có kết nối internet, thì bạn có thể cập nhật API theo cách sau:

Vào menu **Windows/ Android SDK Manager** (hoặc bấm vào biểu tượng trên thanh toolbar), màn hình Android SDK Manager sẽ hiện thị ra như bên dưới:



Bạn quan sát trong màn hình bên trên, Những gói nào chưa được cài đặt thì có dòng chữ “Not installed”, bạn checked vào nó và nhấn nút **Install packages..** ở góc phải dưới cùng.

Tới đây là bạn đã biết cài đặt JDK, cài đặt Android SDK, cách chạy Eclipse và cập nhật API.

Bài tập 2: Thao tác với Android Virtual Device và DDMS

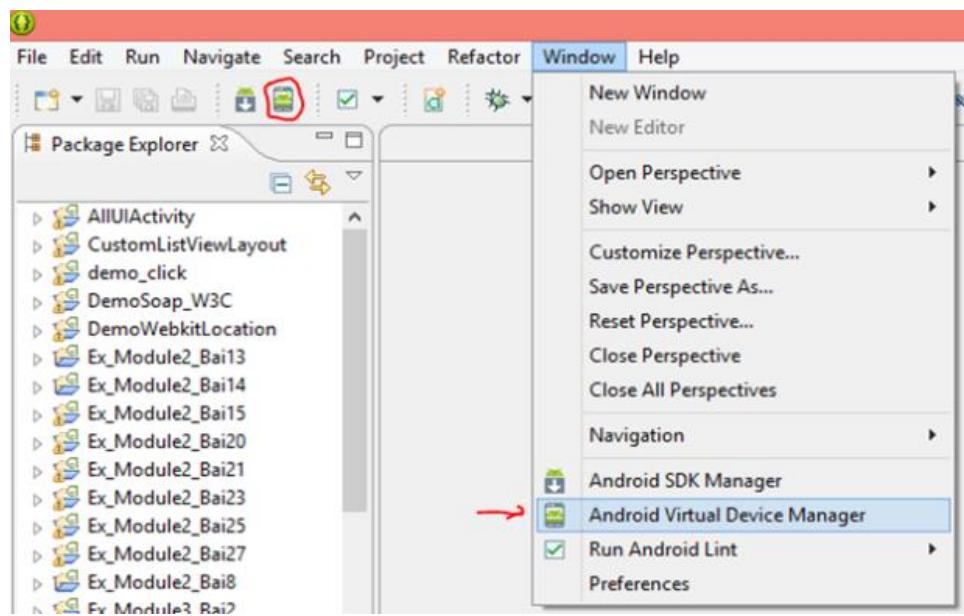
Tiếp nối [bài tập số 1](#), Mục đích của bài tập này sẽ giúp các bạn thao tác được với AVD (tạo, sửa, xóa, cấu hình...) và cách sử dụng DDMS cho việc nhắn tin, gọi điện thoại, sử dụng File Explorer (Đưa tập tin từ máy tính vào máy ảo, và lấy tập tin từ máy ảo ra máy tính....). Biết được cách kết nối một số thiết bị Android thiết với DDMS, cũng như cách thức debug.

- Tại sao sử dụng AVD?

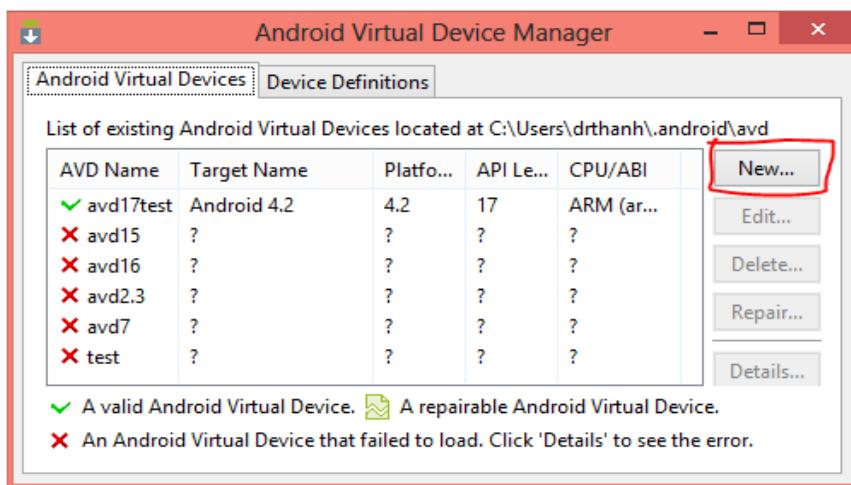
- + Có nhiều lý do để sử dụng AVD, còn theo Tôi thì đơn giản là nó thay thế cho thiết bị thật (Android Emulator) nên cho dù bạn không có chiếc Alô Android nào thì vẫn lập trình àm àm như thường. Mọi sự tương tác trên Emulator cũng chính là tương tác trên thiết bị thật
- + AVD – DDMS hỗ trợ giả lập location base service với định dạng KML nên dễ dàng cho việc lập trình LBS
- + Chỉ có một vài hạn chế của Emulator : Khó hỗ trợ Camera, không hỗ trợ thiết bị cảm ứng. Nhưng bạn yên tâm là trên mạng cũng [đã hỗ trợ thư viện tương tác cảm ứng trong Emulator](#) (hỗ trợ accelerometer, compass, và temperature sensors)

1) Cách tạo Android emulator:

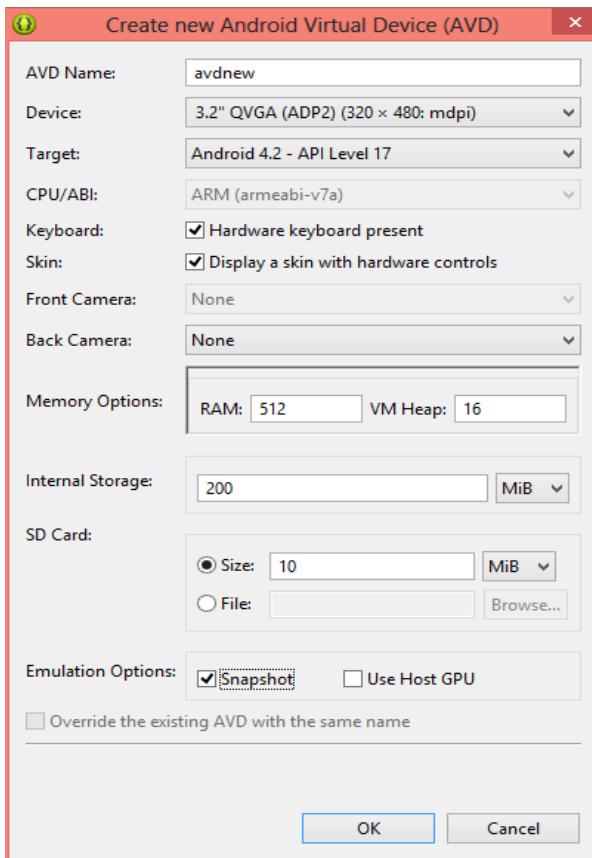
- Vào menu Windows/ chọn Android Virtual Device Manager hoặc click vào biểu tượng máy ảo trên thanh Toolbar:



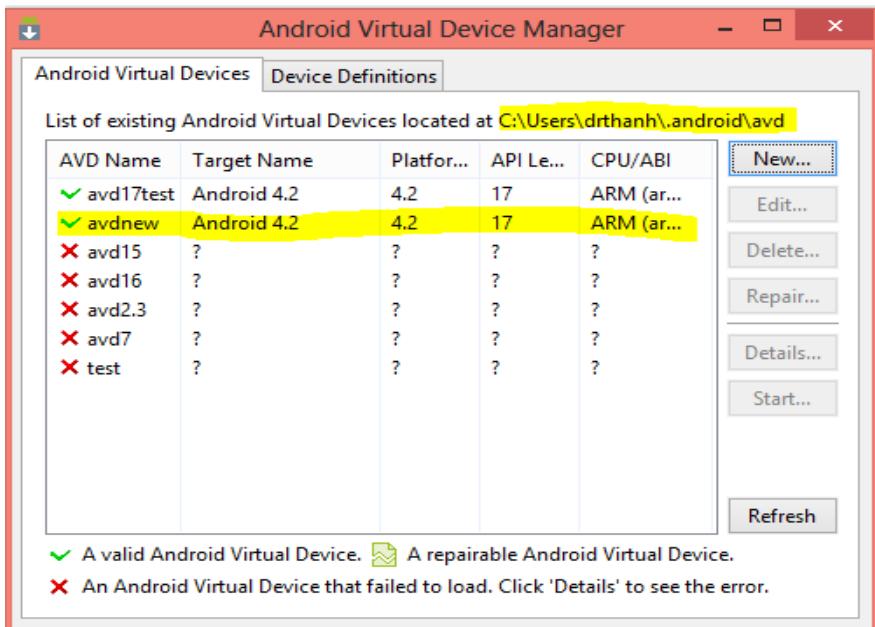
- Cửa sổ Android Virtual Device Manager sẽ hiển thị ra như bên dưới:



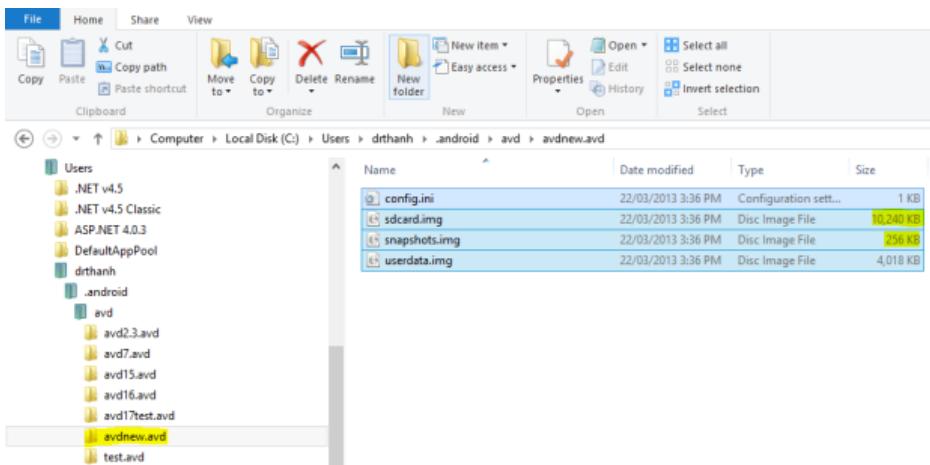
- Để tạo một Android Emulator : click nút New (khoanh vùng màu đỏ):



- Ở màn hình bên trên, ta thiết lập các thông số: Tên máy ảo, độ phân giải, API, bộ nhớ trong, bộ nhớ ngoài (SD Card) rồi nhấn nút OK. Muốn lần tiếp theo khởi động máy ảo nhanh thì ta checked vào mục **Snapshot**



- Như bạn thấy máy ảo **avdnew** đã được tạo ra, bạn chú ý là khi một máy ảo được tạo ra thì nó sẽ được lưu trữ vào user của máy tính đang hoạt động, ở hình trên thì lưu trong “**c:\Users\drthanh\.android\avd**“. Máy của bạn cài đặt với user nào thì bạn vào đúng chỗ mà xem cấu trúc bên trong:

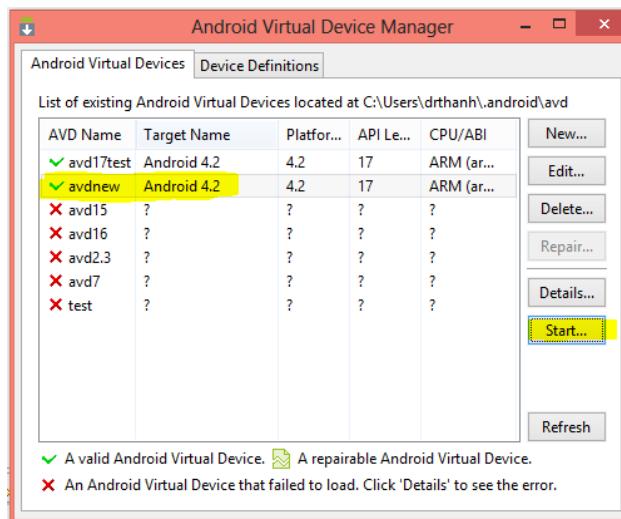


- Ở đây bạn chú ý là nếu bạn cấp dung lượng cho SD Card tại màn hình tạo máy ảo bao nhiêu thì tập tin **sdcard.img** sẽ có dung lượng bấy nhiêu trong này (do đó bạn phải cân trọng khi tạo SD Card đối với máy tính có dung lượng ổ cứng khiêm tốn)

- tương tự bạn nhìn thấy tập tin **snapshots.img**, hiện giờ bạn chỉ thấy có 250kb, nhưng nếu như bạn khởi động lần đầu tiên và vào đây quan sát thì dung lượng của nó sẽ tăng lên rất lớn, mục đích để lưu trữ lại toàn bộ thông số để lần khởi chạy thứ 2 cho lẹ

2) Chạy Android emulator:

- Từ màn hình Android Virtual Device Manager, ta chọn tên máy ảo cần chạy rồi nhấn nút Start:



- Bạn chờ khoảng một thời gian vài phút (hoặc lâu hơn tùy vào khả năng của máy). Tùy vào cấu hình mà bạn chọn (API, độ phân giải...), Android Emulator sẽ xuất hiện như bên dưới:



- Khi máy ảo ở trên khởi động xong, bạn vào thư mục trong ổ C: nơi mà lưu trữ máy ảo, bạn quan sát xem có những thư mục và tập tin mới nào được tạo ra. Sau đó bạn tắt máy ảo này đi, tiếp tục quan sát bạn sẽ thấy tập tin **Snapshots.img** mà Tôi đề cập ở trên sẽ có dung lượng rất lớn:

(C:) > Users > drthanh > .android > avd > avdnew.avd			
Name	Date modified	Type	Size
cache.img	22/03/2013 3:54 PM	Disc Image File	9 KB
config.ini	22/03/2013 3:36 PM	Configuration sett...	1 KB
emulator-user.ini	22/03/2013 3:54 PM	Configuration sett...	1 KB
hardware-qemu.ini	22/03/2013 3:50 PM	Configuration sett...	2 KB
sdcard.img	22/03/2013 3:54 PM	Disc Image File	10,240 KB
snapshots.img	22/03/2013 3:54 PM	Disc Image File	426,561 KB
snapshots.img.default-boot.ini	22/03/2013 3:54 PM	Configuration sett...	2 KB
userdata.img	22/03/2013 3:36 PM	Disc Image File	4,018 KB
userdata-qemu.img	22/03/2013 3:54 PM	Disc Image File	14,360 KB

3) cách sử dụng Android Emulator:

- Bạn để ý là mỗi một máy ảo được khởi động nó sẽ đi với một port nào đó, ví dụ trong trường hợp ở trên là port **5554** (xem ở tiêu đề của cửa sổ). Bạn chú ý là port này chính là đại diện cho số điện thoại của máy ảo đó. Ta sẽ dựa vào port này để thao tác : Gửi tin nhắn, gọi điện thoại.... trong cửa sổ DDMS.

- Các phím chức năng : (không quan trọng lám nên ở đây Tôi không liệt kê ra, các bạn tự mò). Ở đây Tôi chỉ nói một phím chức năng mà Tôi cho là đáng lưu tâm, đó là phím chức năng cho phép quay ngang màn hình điện thoại. Bởi vì người sử dụng điện thoại Smart Phone thường có thói quen quay ngang, quay dọc điện thoại để thao tác cái gì đó. Ta phải lập trình để hỗ trợ cho xem đứng và ngang đối với một ứng dụng cụ thể nào đó.

+ Bạn nhấn tổ hợp phím **Ctrl + F11** để xoay ngang (bấm lại để trở về như cũ):

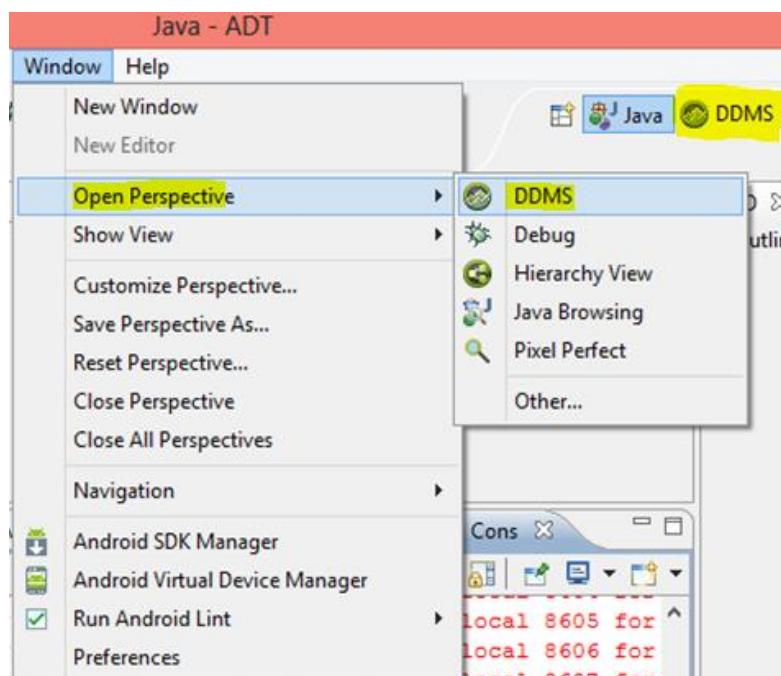


- Cách gửi tin nhắn – gọi điện thoại trong máy ảo:

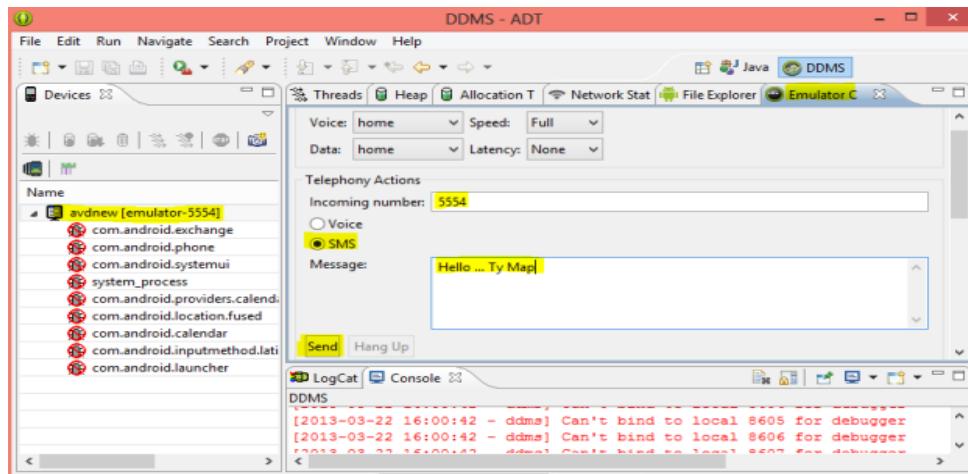
+ **Cách 1:** Bạn tạo thêm một máy ảo Android khác, khi một máy thứ 2 được khởi chạy thì thông thường port của nó sẽ là **5556**, Như bên trên Tôi đã nói Port này là đại diện cho số điện thoại. Máy tính của bạn hiện tại sẽ có 2 máy ảo Android (Bạn tưởng tượng đó là 2 cái điện thoại thật mà bạn đang cầm trên tay đi). Do đó cách mà bạn thao tác với điện thoại thật như thế nào thì với Emulator nó sẽ y xì như vậy.

+ **Cách 2:** Trên thanh công cụ bạn quan sát thấy nút **DDMS**, nhấp vào nó (chú ý là bạn phải ít nhất đang chạy 1 máy ảo android)

Thông thường thì trên thanh công cụ sẽ có DDMS, nhưng nếu như không thấy thì bạn vào menu **Windows/** chọn **Open Perspective/ DDMS**, xem hình dưới:



- sau khi chọn DDMS, bạn vào tab Emulator Control như hình bên dưới:



- Ở màn hình trên:

+ Mục Incoming Number Tôi nhập là 5554 (port này chính là số điện thoại của máy ảo)

+ Tôi chọn SMS và nhập vào “Hello ... Ty Map”

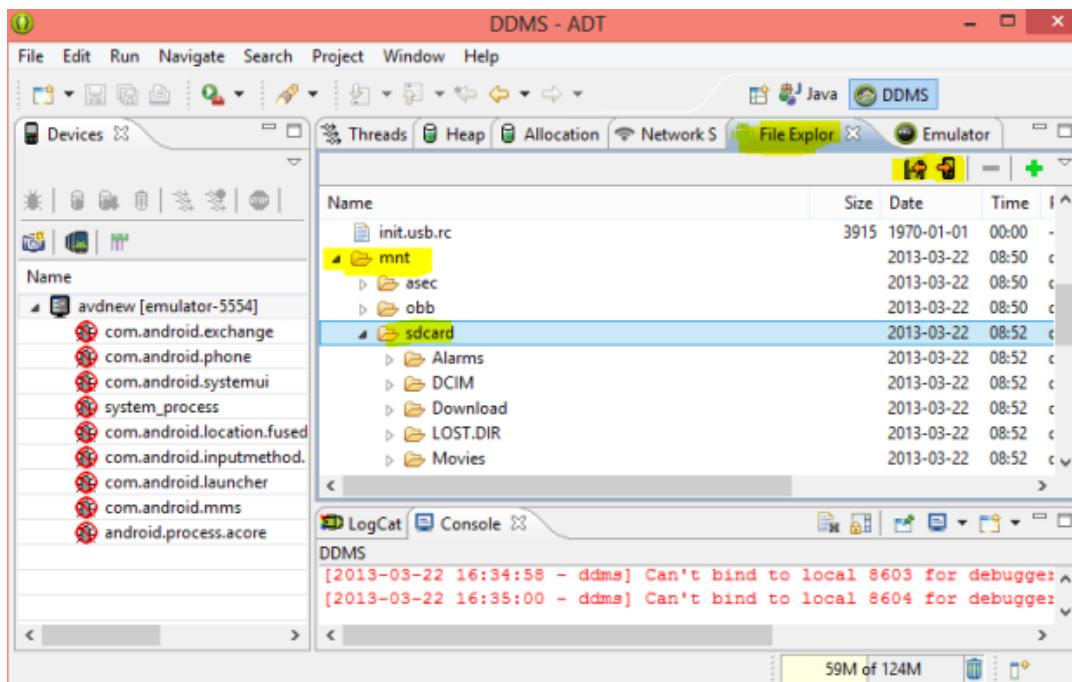
+ Nhấn nút Send

→Bạn quan sát máy Ảo có port 5554 sẽ nhận được tin nhắn này:



- **Cách sử dụng Profile Explorer:**

+ Trong màn hình DDMS, bạn chọn tab Profile Explorer: Trong bài tập này bạn chỉ cần tập cách thả 1 tập tin từ máy tính vào SD Card, và kéo 1 tập tin từ SD Card ra máy tính (Chúng ta còn phải làm việc với nó rất nhiều lần ở các bài tập kế tiếp):



- bạn quan sát thư mục **mnt/ sdcard** . Chú ý là có một vài nơi cùng tên sdcard, nhưng bạn phải vào **mnt**.

+ Để lấy 1 tập tin từ SD Card ra thì bạn chọn tập tin đó rồi nhấn vào biểu tượng ở đĩa mềm ở bên trên (Tôi tô màu vàng)

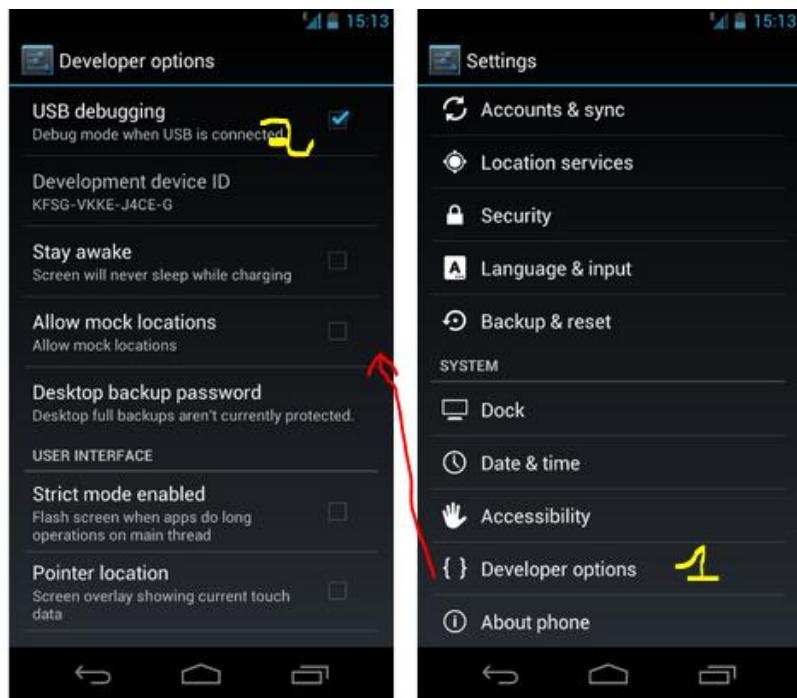
+ Để đưa một tập tin từ Máy tính vào SD card bạn nhấn vào biểu tượng điện thoại (kép bên biểu tượng đĩa mềm), chọn tập tin cần đưa, Hoặc bạn kéo thả trực tiếp một tập tin nào đó vào màn hình này luôn. cả 2 cách đều như nhau.

4) Cách kết nối với Thiết Bị Android Thật

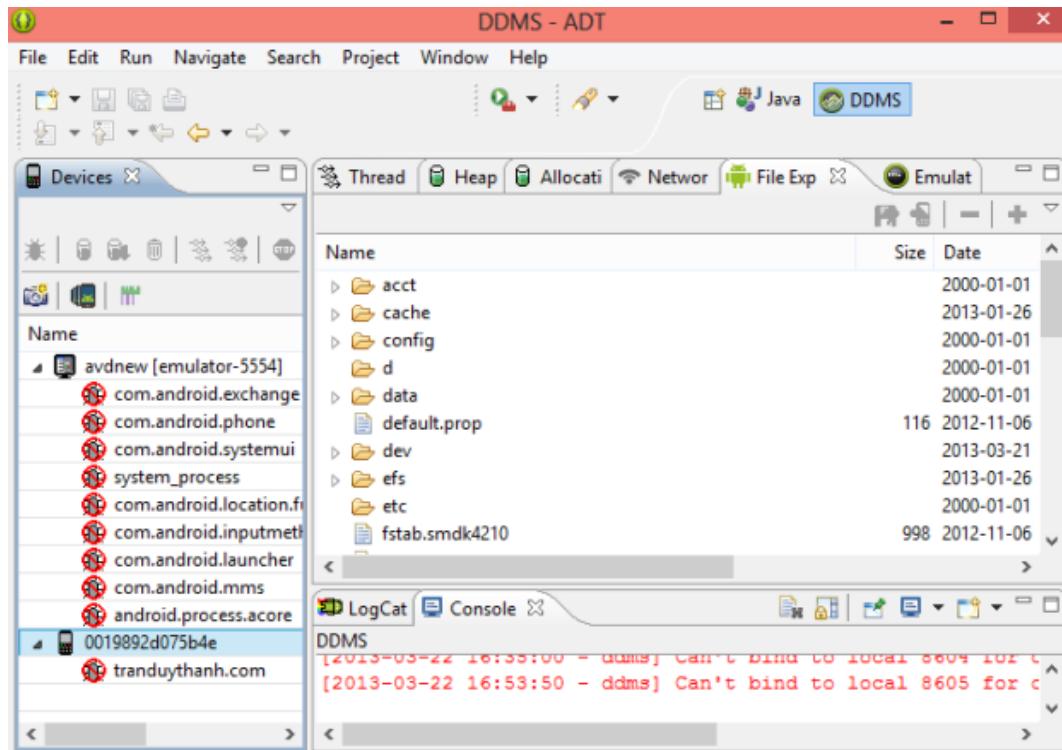
- Để làm được điều này thì bạn phải am hiểu về Driver cho mỗi Model điện thoại. Sau khi đã cài Driver đầy đủ:

Đối với Điện thoại Android sử dụng API 4.0 trở lên:

- + Vào Settings
- + Chọn Developer Options
- + Checked vào USB debugging.



- Bạn xem hình bên dưới (ở đây Tôi kết nối điện thoại SAM SUNG SII của Tôi tới DDMS):



- Bạn thao tác bình thường: Thêm tập tin vào Điện thoại thật, lấy tập tin ra.... Khi chạy chương trình Android ta sẽ chọn Điện thoại Thật nếu như bạn muốn chạy trên thiết bị thật. Eclipse cho phép cấu hình lựa chọn các loại Máy ảo, máy thật khi chạy ứng dụng.

**** Như vậy Tôi đã hướng dẫn các bạn xong phần thao tác với AVD, trong bài tập kế tiếp Tôi muốn bạn hiểu được cách thức mà một ứng dụng Android khi triển khai nó đi theo đường nào:

- Ví dụ:

+ Dalvik Virtual Machine là gì? nó có tác dụng gì trong ứng dụng?

+ Biên dịch như thế nào?

+ Biên dịch xong thì làm gì?

+ Cách thức ứng dụng upload vào thiết bị

+ Cách thực ứng dụng được cài đặt vào thiết bị

+ Chạy ứng dụng như thế nào trong thiết bị

+ Làm sao tháo gỡ ứng dụng ra khỏi thiết bị

+ Làm sao kiểm tra có bao nhiêu ứng dụng đang chạy trong điện thoại.

Bài tập này Tôi hi vọng các bạn phải thao tác thật nhuần nhuyễn để khi vào sâu bên trong để lập trình thì nó giúp ích cho các bạn rất nhiều.

Bài tập 3: Cách tạo Android Project và tìm hiểu cấu trúc bên trong của một Android Project

Ở [bài tập số 2](#), bạn đã biết cách tạo và sử dụng máy ảo Android cũng như DDMS.

- Trong bài tập này các bạn sẽ thực hành cách tạo Android Project và tìm hiểu các thành phần bên trong của nó:

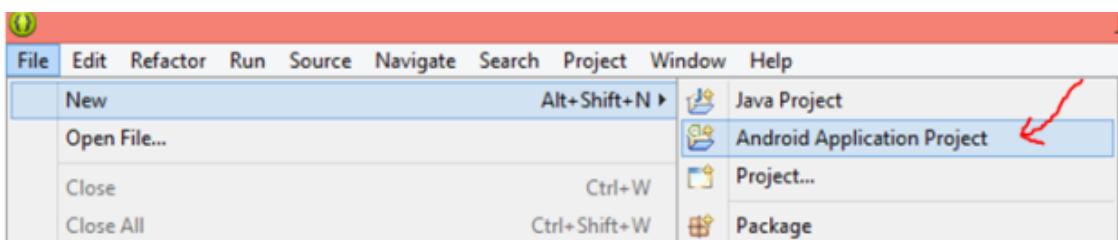
- Cách tạo Android Project
- Activity, Intent, View
- Auto gen
- Android libs
- Resource
- Layout, menu, values
- Manifest XML

- Cũng như cách chạy chương trình Android, hiểu được cơ chế vận hành của nó.

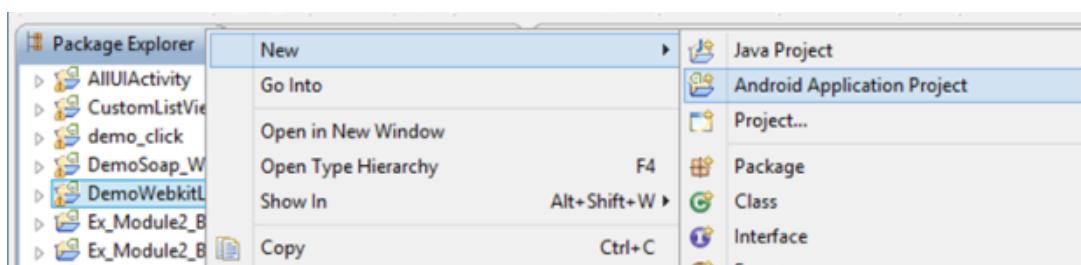
1) Cách tạo một Android Project:

Ở đây Tôi cung cấp 2 cách tạo Android Project:

Cách 1: Bạn vào Menu File/ Chọn New/ chọn Android Project (xem hình bên dưới):



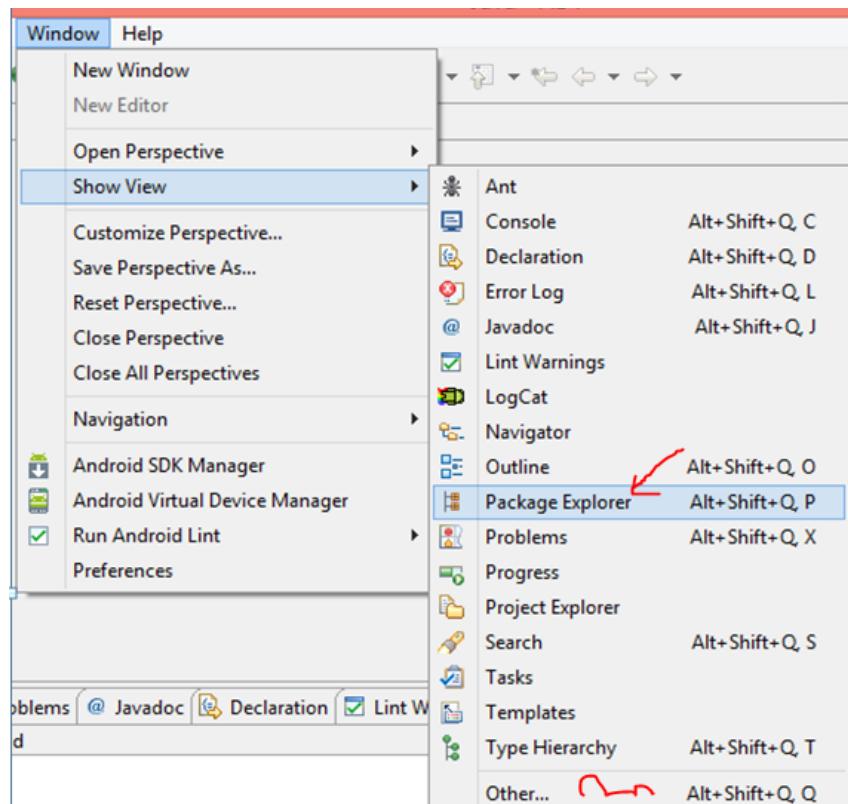
Cách 2: Bấm chuột phải vào vị trí bất kỳ trong Package Explorer / Chọn New / Chọn Android Project (xem hình bên dưới):



Các bạn chú ý là khi một Android Project được tạo ra thì nó sẽ được lưu trữ trong Package Explorer. Một số trường hợp bạn không thể thấy được Package Explorer (do bạn lỡ tay đóng nó đi, hoặc một nguyên nhân nào đó), nếu như chưa biết cách lôi nó ra thì các bạn làm như sau:

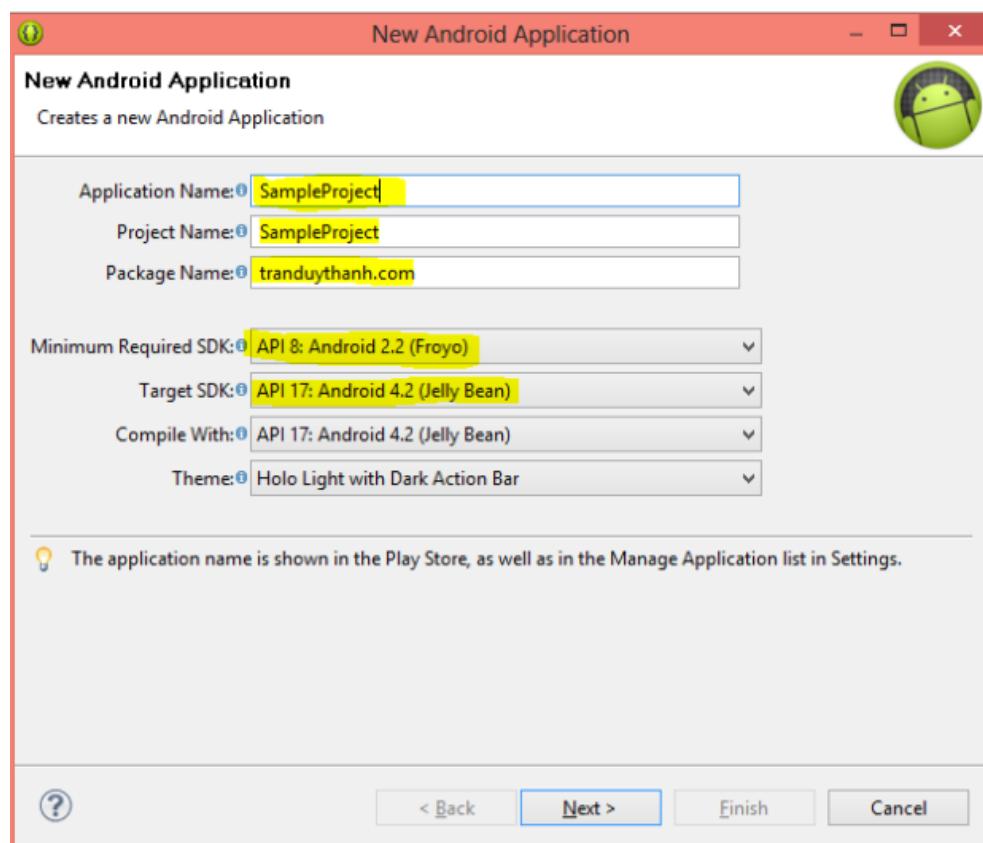
- Vào menu Windows/ chọn Show View/ click chọn **Package Explorer**

- Nếu như trong Show View mà không thấy Package Explorer thì bạn nhìn mục dưới cùng có nhãn “Other...”, click vào nó thì chắc chắn bên trong sẽ có Package Explorer:



2) Nhập thông số cho một Android Project mới:

Khi bạn chọn New Android Project thì một màn hình sẽ hiển thị ra như bên dưới:



Mục Application Name: bạn đặt tên cho ứng dụng mà bạn mong muốn, trong ví dụ này là Tôi đặt “SampleProject”

Mục Project Name : thông thường khi bạn đặt tên cho Application Name thì mục Project Name sẽ tự động cập nhật giống như vậy

Mục Package Name: bạn nên viết thường hết và phải ít nhất có 1 dấu chấm ngăn cách, ví dụ bạn có thể đặt “tranduythanh.com” nhưng không thể đặt “tranduythanh”.

Mục Minimum Required SDK : Chọn giới hạn API thấp nhất mà ứng dụng có thể cài đặt (Điện thoại đó có API phải \geq Minimum Require)

Mục Target SDK : chọn API mà ứng dụng mong muốn tốt nhất có thể thực thi trên API này

Bạn cần phân biệt giữa Minimum và Target:

- Tôi muốn giải thích sơ qua chỗ này, ở trên bạn thấy Tôi chọn Minimum là API 8, và Target là API 17. Điều này nghĩa là:

+ Nếu như ứng dụng của Tôi mà được cài đặt trên điện thoại có API 17 tức là cái đích Tôi mong muốn và nó sẽ được hỗ trợ tối đa cho các đặc tính bên trong ứng dụng của Tôi (tức là cái tốt nhất)

+ Nếu như ứng dụng của Tôi cài trên điện thoại có API là 7 thì chắc là không thể vì ở đây Tôi yêu cầu Minimum là API 8

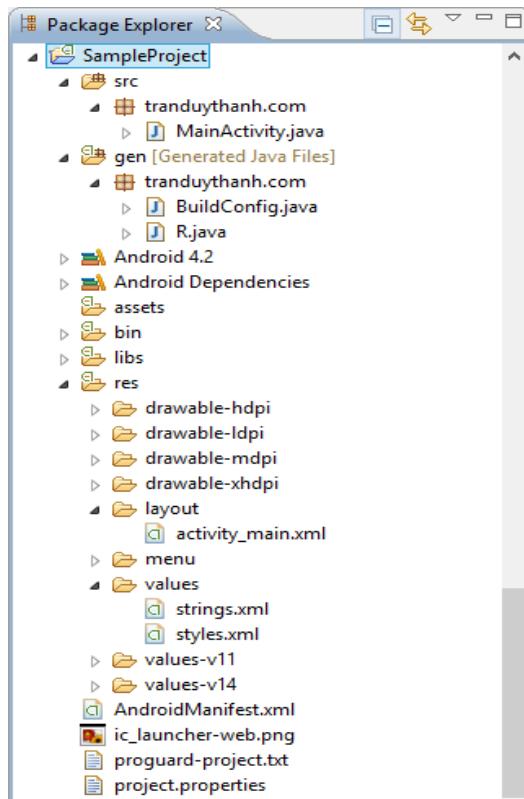
+ Nếu như ứng dụng của Tôi cài trên điện thoại có API >17 chặng hạn, thì thông thường là sẽ được vì Version mới đa phần hỗ trợ version cũ, nhưng dĩ nhiên có thể không tốt bằng đúng cái điện thoại sử dụng API 17

- Lý do cho phần này là bởi vì đối với từng API luôn có một lỗi nào đó, các API mới ra đời để sửa vá lỗi hoặc là nâng cấp thêm một số đặc tính mới. Ví dụ như để sử dụng Nhận Diện Khuôn Mặt có sẵn của Android thì bắt buộc phải chọn Minimum API =14 vì từ Android 4.0 mới hỗ trợ chức năng Nhận Diện Khuôn Mặt

- Sau khi chọn các thông số xong, bạn nhấn **Next** cho tới khi nút **Finish** xuất hiện (Tôi khuyên là các bạn cứ bấm Next, đừng có chỉnh sửa gì cả vì các bạn mới bắt đầu làm quen với nó. Khi nào quen rồi thì Ta sẽ quay lại chi tiết sau):

- Khi bấm Finish thì bạn quan sát Package Explorer để xem cấu trúc bên trong của ứng dụng Android:

3) Tìm hiểu cấu trúc bên trong của Ứng dụng Android:

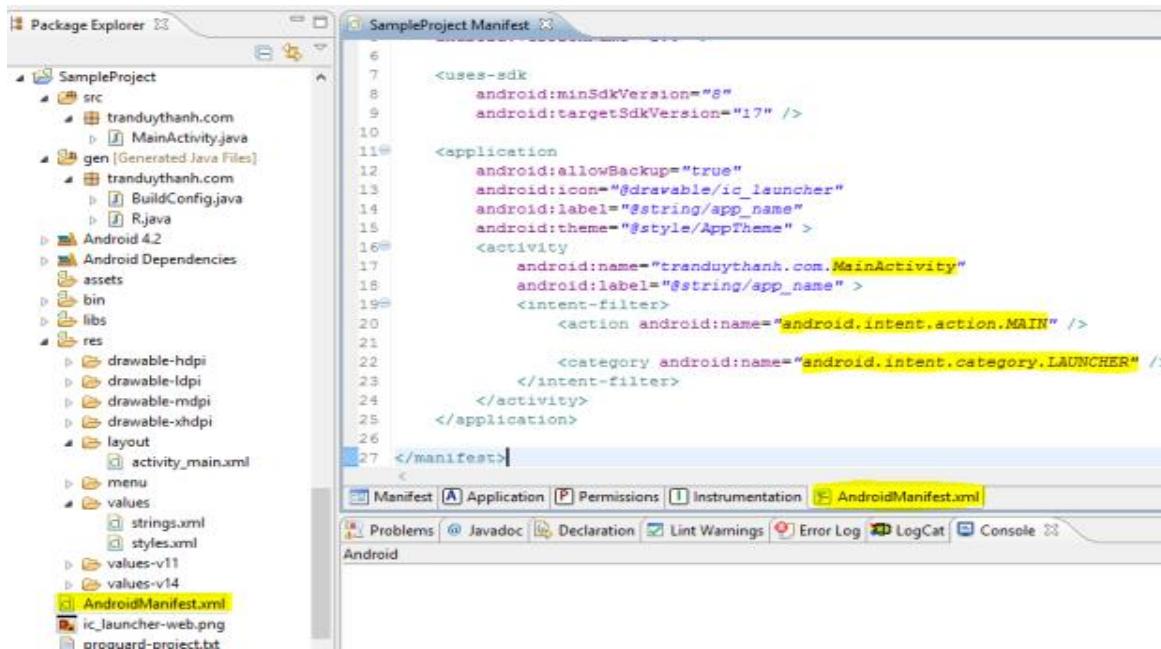


- Bạn cần hiểu được cấu trúc cây trong Ứng Dụng Android ở trên:

Hãy quan sát **MainActivity.java và **activity_main.xml****. Khi một Ứng dụng được tạo ra thì thông thường sẽ có một Activity để khởi chạy ứng dụng. Ở đây bạn hiểu rằng **MainActivity.java** chính là class chứa toàn bộ source code, còn **activity_main.xml** chính là phần giao diện. Đối với Android khi một Activity tạo ra thì thường nó đi kèm với một Layout giao diện nào đó (Tức là nó luôn được tách thành 2 phần: phần source code riêng và phần giao diện riêng). Bạn hiểu Activity giống như là các màn hình (cửa sổ) tương tự như là C#, mỗi Activity là một màn hình tương tác cụ thể nào đó.

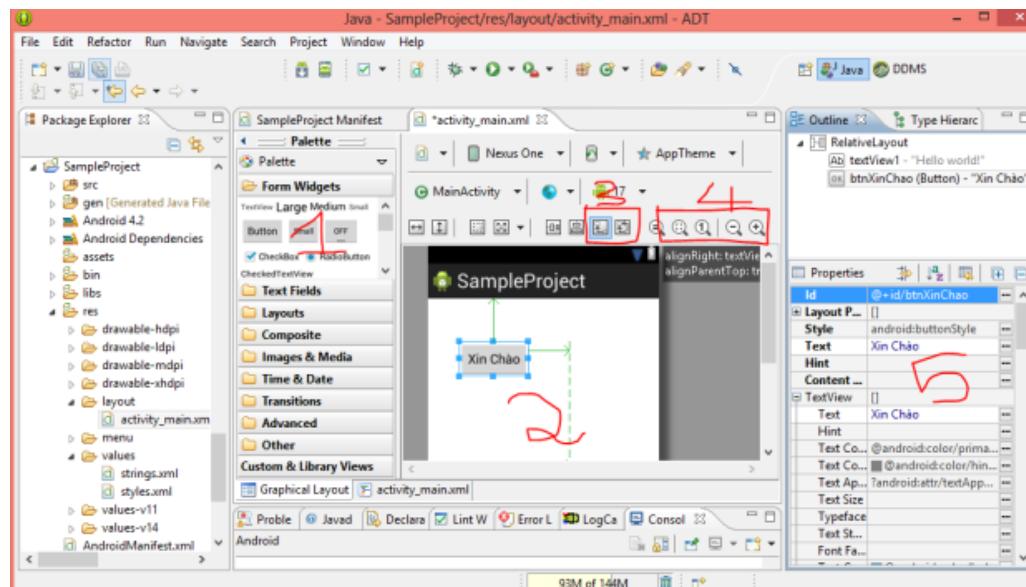
- Bạn cũng cần nhớ rằng bất kỳ một Activity nào muốn được triệu gọi thành công trong Android Project thì bắt buộc nó phải được khai báo trong tập tin **AndroidManifest.xml**. Nếu như bạn gọi một Activity mà Activity này không được khai báo trong Manifest thì chắc chắn chương trình sẽ bị lỗi và tắt luôn.

- Hãy xem cấu trúc bên trong của **AndroidManifest.xml** (double click vào nó và chọn tab như hình bên dưới):



- Như hình bên trên bạn thấy đó, MainActivity muốn được triệu gọi thì nó phải được khai báo trong này. Và đặc biệt nếu như muốn nó là màn hình đầu tiên được thực thi khi chạy ứng dụng thì bạn phải khai báo giống như tag **<intent-filter>** ở trên. Như vậy nếu bạn muốn một Activity bất kỳ nào đó được gọi đầu tiên khi chạy ứng dụng thì bạn chỉ khai báo y chang như vậy, còn các Activity khác bạn không cần khai báo **<intent-filter>** như trên (tức là hoàn toàn không có tag này)

- Tiếp theo bạn double – click vào **activity_main.xml** :



- Hãy quan sát màn hình ở trên (nhìn kỹ có 2 phần **Graphical Layout** và **activity_main.xml**. Bạn nên làm quen trong phần Graphical Layout trước):

+ **Vùng số 1** : chính là nơi chứa các control, layout, component... bạn muốn sử dụng cái nào thì kéo thả nó vào Vùng số 2. Như bạn thấy tôi vừa kéo một Button vào và đặt Id nó là “**btnXinChao**”,

+ **Vùng số 2**: là giao diện, nơi mà bạn cần thiết kế

+ **Vùng số 3**: cho phép thiết kế theo chiều đứng hay chiều ngang

- + **Vùng số 4:** chức năng Zoom in – zoom out để dễ thiết kế
- + **Vùng số 5:** chính là nơi thiết lập các thuộc tính cho các control được kéo thả vào giao diện

- Bây giờ bạn vào thư mục gen trong Package Explorer và mở tập tin R.java lên:

```

Java - SampleProject/gen/tranduythanh/com/R.java - ADT
File Edit Refactor Run Source Navigate Search Project Window Help
R.java
8 package tranduythanh.com;
9
10 ic final class R {
11     public static final class attr {
12     }
13     public static final class drawable {
14         public static final int ic_launcher=0x7f020000;
15     }
16     public static final class id {
17         public static final int btnXinChao=0x7f070001;
18         public static final int menu_settings=0x7f070002;
19         public static final int textView1=0x7f070000;
20     }
21     public static final class layout {
22         public static final int activity_main=0x7f030000
23     }

```

Thư mục **gen**, là thực mục cho Android tự động tạo ra, cho dù bạn có xóa nó thì nó cũng lại tự tạo ra. Nội dung bên trong bạn đừng có chỉnh sửa nó. Tất cả những gì bạn kéo thả vào giao diện, hay thiết lập string.xml, menu ... tất tần tật liên quan tới resource thì nó sẽ được sinh ra bên trong R.java. Dựa vào đây để ta có thể truy suất các đối tượng trong coding.

Ví dụ:

- Ở trên bạn thấy class Id có chứa Id của Button mà lúc nãy Tôi kéo vào giao diện đặt tên là **btnXinChao**. Ta dựa vào Id này để tương tác với control.

- Hay layout **activity_main** cũng sẽ tự động lưu trữ trong này

- Tiếp tục double – click vào MainActivity.java:

```

Java - SampleProject/src/tranduythanh/com/MainActivity.java - ADT
File Edit Refactor Run Source Navigate Project Window Help
R.java MainActivity.java
7
8 public class MainActivity extends Activity {
9
10     @Override
11     protected void onCreate(Bundle savedInstanceState) {
12         super.onCreate(savedInstanceState);
13         setContentView(R.layout.activity_main);
14         Button btnChao=(Button)
15             findViewById(R.id.btnXinChao);
16     }
17 }
18

```

- Trong màn hình trên, bạn thấy bên trong hàm onCreate có lệnh:

+ `setContentView(R.Layout.activity_main)` ==> thiết lập giao diện cho Activity.
Với **activity_main** lấy từ R.java

+ `findViewById(R.id.btnXinChao)` ==> truy suất control là Button trên giao diện.
với `btnXinChao` lấy từ R.java

* Bạn sẽ hiểu hơn về các hàm này vào các phần sau. Ở đây mục đích là Tôi đang giải thích chức năng của từng phần trong Android

- Tiếp tục bạn quan sát các thư mục: **Drawable-hdpi, Drawable-ldpi, Drawable-mdpi, Drawable-xdpi:**

Bạn có thể tự tạo thêm một thư mục cùng cấp tên là **Drawable**, các tập tin bạn kéo thả trực tiếp vào trong này (tạo Resource). Khi chương trình load các ReSource sẽ tự động vào đây lấy. Còn -hdpi, -ldpi, -xipi là tùy thuộc vào độ phân giải màn hình mà chương trình tự động vào lấy đúng dữ liệu ở bên trong.

- **Tiến hành thực thi chương trình:**

Bấm chuột phải vào ứng dụng / chọn Run As/ chọn Android Application , xem kết quả:



- Tôi gom lại thành 5 bước thực hiện của một ứng dụng Android như sau:

bước 1: Android Project sẽ được tự động biên dịch và chuyển qua Android Executables (.dex)

bước 2: Đóng gói thành tập tin .apk

bước 3: Upload .apk vào thiết bị android

bước 4: Tiến hành cài đặt .apk đó

bước 5: Khi cài đặt thành công, chương trình sẽ được thực hiện Activity được thiết lập : `android.intent.action.MAIN`

Mỗi một ứng dụng Android sẽ được thực thi trên một máy ảo Dalvik (không phải máy ảo Java). Theo như giải thích của developer Android thì mục đích là chạy đa tiến trình, giúp tối ưu bộ nhớ.

Như vậy đến đây là bạn đã biết cách tạo một ứng dụng Android và khởi động nó như thế nào, đồng thời cũng biết được một số thành phần bên trong ứng dụng và công năng của chúng

Bạn cần hiểu rõ bài tập này, Trong bài tập tiếp theo Tôi sẽ nêu ra các tình huống LỖI của chương trình và cách xử lý chúng như thế nào. Vì Tôi chắc chắn rằng trong quá trình học lập Trình Android bạn sẽ gặp các tình huống đó mà không hiểu vì sao nó bị lỗi, bạn sẽ bê tắc trong việc giải quyết lỗi.

Bài tập 4: Cách sửa một số lỗi thường gặp trong Android Project

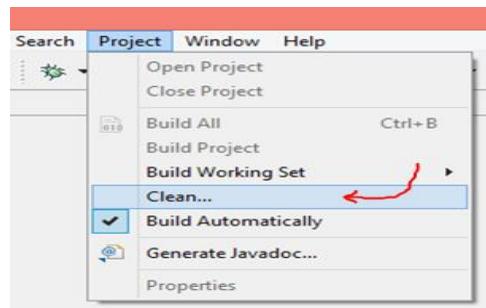
Ở bài tập 3 các bạn đã biết cách tạo và thực thi một ứng dụng Android, trong bài tập này Tôi sẽ nói một số cách sửa lỗi thường gặp trong quá trình thực thi một ứng dụng Android:

- Bạn nhớ là những lỗi này thường không phải do bạn gây ra mà do Eclipse hoặc một số nguyên nhân nào đó (ta gọi tạm thời là Chương trình bị tung tung). Bạn cũng không hiểu vì sao nó lại bị lỗi, nó không cho phép bạn biên dịch và thực thi ứng dụng mặc dù bạn chả làm gì cả?

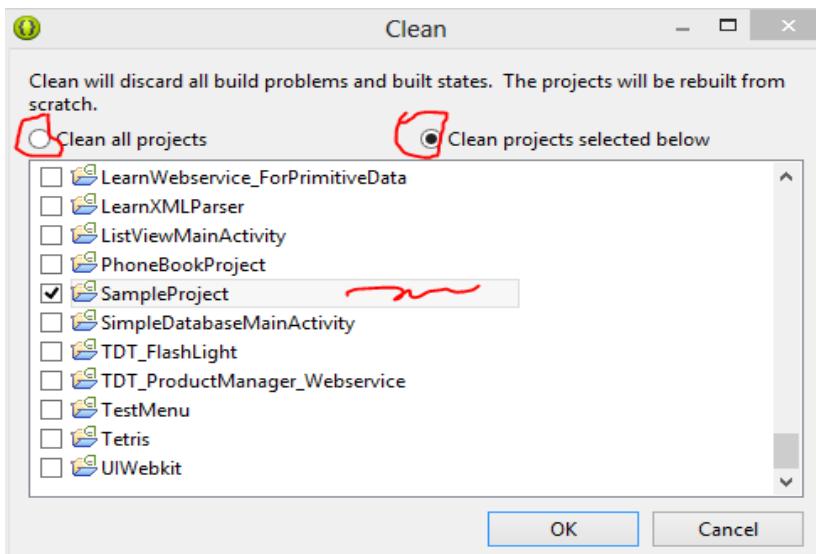
- Khi bạn gặp những trường hợp khó đỡ như vậy thì hãy làm theo một số cách dưới đây (nếu mà vẫn không được thì bạn có thể sử dụng phần mềm TeamView, cung cấp Id và Password Tôi sẽ login vào máy bạn sửa giúp nếu được):

Giải pháp 1:

- b1) Vào menu Project / chọn Clean :



- b2) Màn hình Clean hiển thị lên, bạn chọn Project bị báo lỗi và bấm Ok:



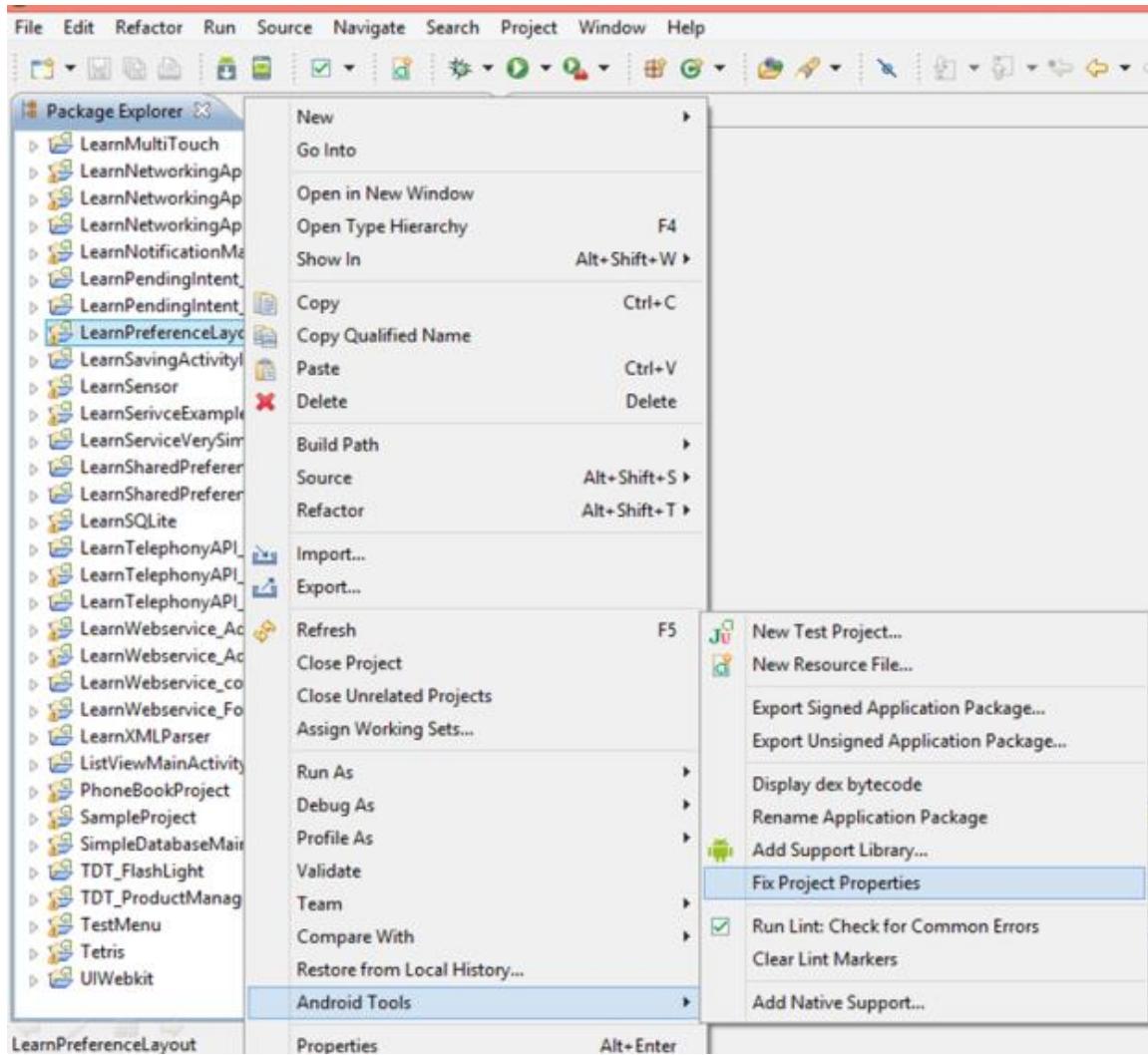
- Ở hình trên, nếu bạn chọn “Clean All Project ”, eclipse sẽ clean toàn bộ Project trong Package Explorer.

- Nếu bạn chọn “**Clean Projects selected below**“, eclipse sẽ clean những Project mà bạn Checked ở bên dưới.

b3) Bạn Unchecked “**Build Automatically**” và Checked lại “**Build Automatically**” trong menu Project

Giải pháp 2:

- Bấm chuột phải vào Project / chọn **Android Tools** / chọn **Fix Project Properties**:



Giải pháp 3:

- Khai bạn kéo thả một số tập tin, hình ảnh vào Android Project nó sẽ báo lỗi và không thể tạo ra gen. Bạn hãy kiểm tra xem có phải tên các tập tin đó bị Việt Hoa ký tự đầu hay không, hay tên tập tin bị đê khoảng trắng hoặc là có các ký tự không hợp lệ. Bạn chỉ cần sửa lại là hết lỗi

Giải pháp 4:

- Đơn giản là tắt và khởi động lại Eclipse.

Trong bài tập tiếp theo, Tôi sẽ hướng dẫn các bạn hiểu được Vòng Đời của một Ứng dụng Android và đưa ra một số lời khuyên về việc lưu trạng thái cũng như phục hồi trạng thái của ứng dụng.

Bài tập 5: Tìm hiểu vòng đời của một Ứng Dụng Android

Trong [bài tập 3](#) và [bài tập 4](#) bạn đã biết cách tạo một Android Project cũng như giải quyết một số vấn đề gặp phải khi thực hiện ứng dụng.

Trong bài tập 5 bạn cần hiểu các khái niệm sau:

- 1) Applications là gì?
 - 2) Activities là gì?
 - 3) Activity Stack là gì?
 - 4) Tasks là gì?
 - 5) Life Cycle States là gì?
-

1) Applications là gì?

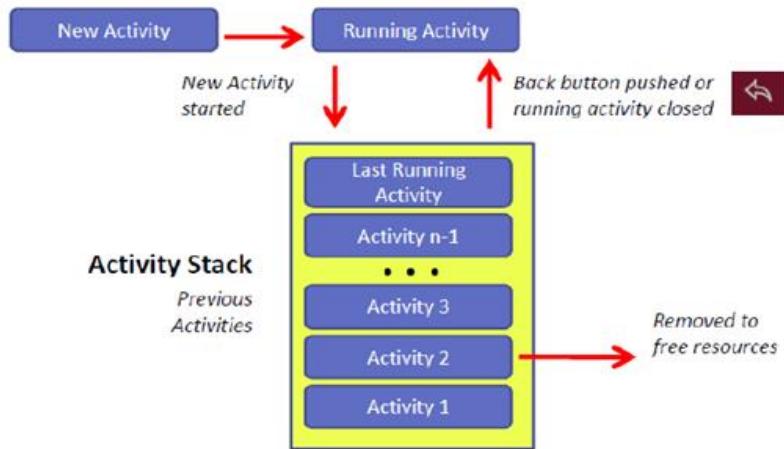
- Bạn hiểu nôm na như sau: Mỗi một Android Project khi bạn biên dịch thành công thì sẽ được đóng gói thành tập tin **.apk**, tập tin **.apk** được gọi là một Application (Ứng dụng cụ thể nào đó – Ví dụ như ứng dụng chống tin nhắn rác, ứng dụng tìm đường đi ngắn nhất, ứng dụng đăng ký học phần bằng sms)

2) Activities là gì?

- Thông thường trong một ứng dụng (Application) sẽ có một hoặc nhiều Activity (Bạn hiểu đại khái là các màn hình tương tác giống như Form trong .Net).
- Mỗi một Activity này sẽ có một vòng đời riêng độc lập hoàn toàn với các Activity khác, bạn sẽ hiểu rõ hơn về vòng đời trong phần Life Cycle States. Việc hiểu rõ vòng đời của Activity là rất quan trọng trong việc xử lý thông tin.
- Mỗi một Activity muốn được triệu gọi trong ứng dụng thì bắt buộc nó phải được khai báo trong Manifest

3) Activity Stack là gì?

- Tương tự như các ngôn ngữ lập trình khác, Activity Stack hoạt động theo cơ chế LIFO (LAST IN FIRST OUT)
- Mỗi một Activity mới được mở lên nó sẽ ở bên trên Activity cũ, để trở về Activity thì bạn chỉ cần nhấn nút “Back” để trở về hoặc viết lệnh. Tuy nhiên nếu bạn nhấn nút Home rồi thì sẽ không thể dùng nút “Back” để quay lại màn hình cũ được.



- Ở đây bạn chú ý là có 2 kiểu mở Activity mới :

a) Mở Activity mới lên làm che khuất toàn bộ Activity cũ (không nhìn thấy Activity cũ): xảy ra sự kiện **onPause** rồi **onStop** đối với Activity cũ

b) Mở Activity mới lên làm che khuất một phần Activity cũ (vẫn nhìn thấy Activity cũ): Sảy ra sự kiện **onPause** với Activity cũ.

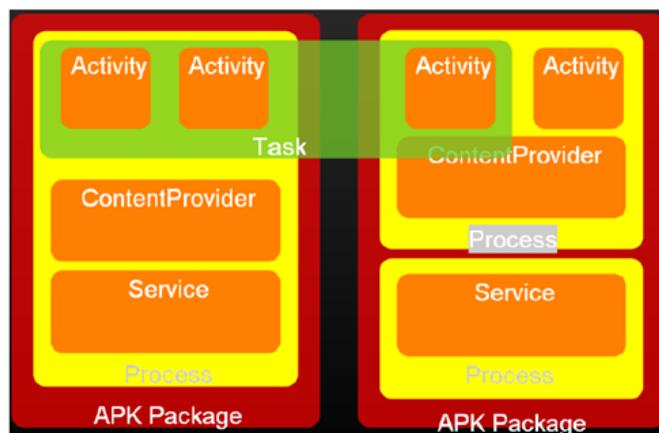
- Khi quay trở về Activity cũ thì sau khi thực hiện xong các hàm cần thiết, chắc chắn nó phải gọi hàm **onResume** để phục hồi lại trạng thái ứng dụng

- Như vậy ta thường lưu lại trạng thái của ứng dụng trong sự kiện **onPause** và đọc lại trạng thái ứng dụng trong sự kiện **onResume**

4) Tasks là gì?

- Bạn hiểu đại khái Task là khả năng thực hiện một công việc nào đó giữa các Ứng dụng với nhau, cụ thể là các Activity

- Ví dụ bạn đang mở chương trình quản lý BlackList, trong chương trình này cho phép mở danh bạ để đưa vào danh sách đen. Lúc đó chương trình bạn sẽ gọi Activity của ứng dụng danh bạ, sau khi lấy xong lại quay trở về ứng dụng của bạn. Nhớ là 2 ứng dụng này hoàn toàn không liên quan gì tới nhau cả.



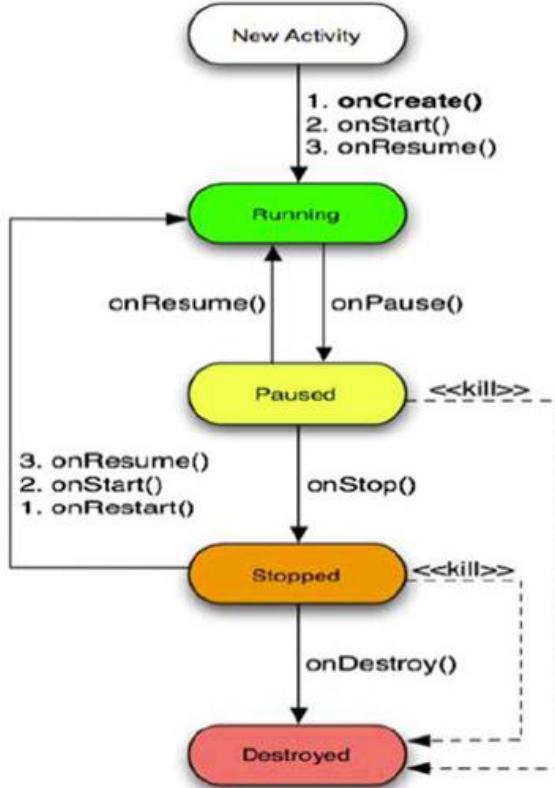
5) Life Cycle States là gì?

Với mỗi Activity thường vòng đời có 3 trạng thái sau:

1- **Running** (đang kích hoạt)

2- **Paused** (tạm dừng)

3- **Stopped** (dừng – không phải Destroyed)

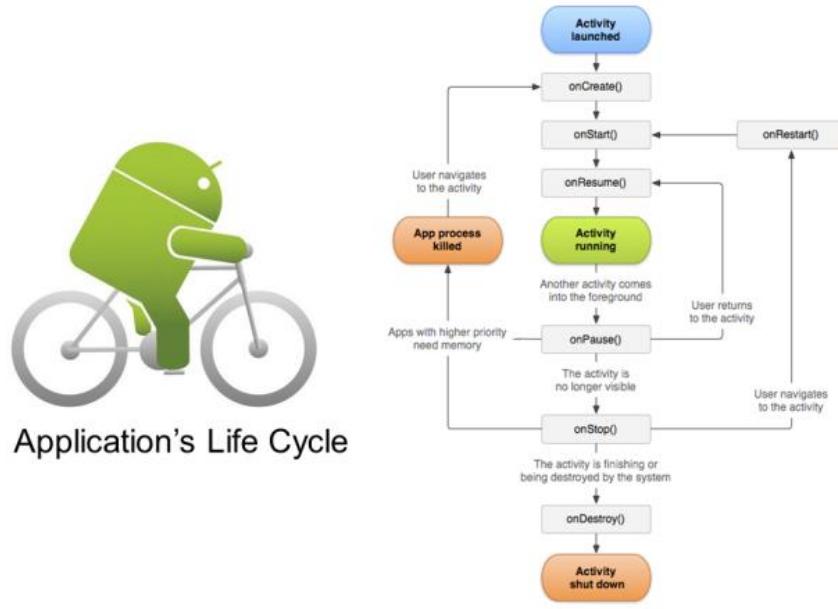


1- **Running** (đang kích hoạt): Khi màn hình là Foreground (Activity nằm trên cùng ứng dụng và cho phép người sử dụng tương tác)

2- **Paused** (tạm dừng) : Activity bị mất focus nhưng mà vẫn nhìn thấy được Activity này (Ví dụ bạn mở một Activity mới lên dưới dạng Dialog). Trường hợp này nó vẫn có khả năng bị hệ thống tự động “XỬ” trong tình huống bộ nhớ quá ít.

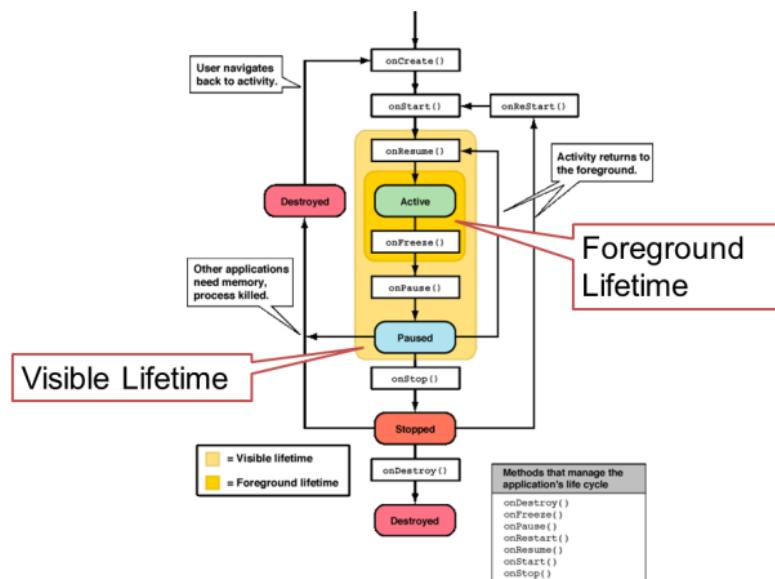
3- **Stopped** (dừng – không phải Destroyed): Activity mất focus và không nhìn thấy được (ví dụ bạn mở một Activity mới lên mà Full màn hình chẳng hạn). Trong trường hợp này nó có thể bị hệ thống “XỬ” trong bất kỳ tình huống nào.

*** Như vậy cả **Paused** hay **Stopped** đều có khả năng bị **Destroyed** (hủy) khi bộ nhớ cần cho việc khác ưu tiên hơn.



- Trong vòng đời của ứng dụng Android bạn cần phân biệt 2 loại sau:

- Visible Lifetime và Foreground Lifetime



- Visible Lifetime:

+ xảy ra từ sau khi gọi onStart → cho tới lúc gọi onStop : trong trường hợp này TA vẫn có thể thấy màn hình Activity (có thể tương tác khi nó là foreground, không tương tác được khi nó không phải foreground như đã giải thích ở trên)

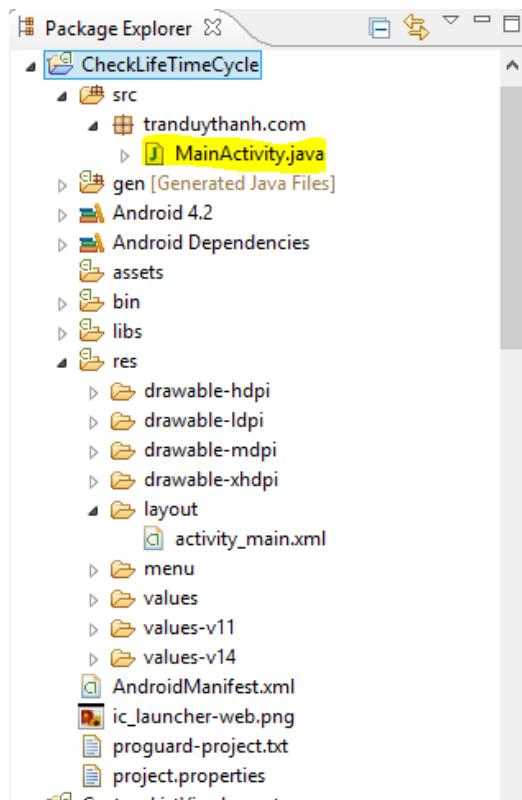
- Foreground Lifetime:

+ Sảy ra từ khi gọi onResume → cho tới lúc gọi onPause : trong suốt thời gian này Activity luôn nằm ở trên cùng và Ta có thể tương tác được với nó

Như vậy bạn đã hiểu được vòng đời của một ứng dụng Android diễn ra như thế nào.

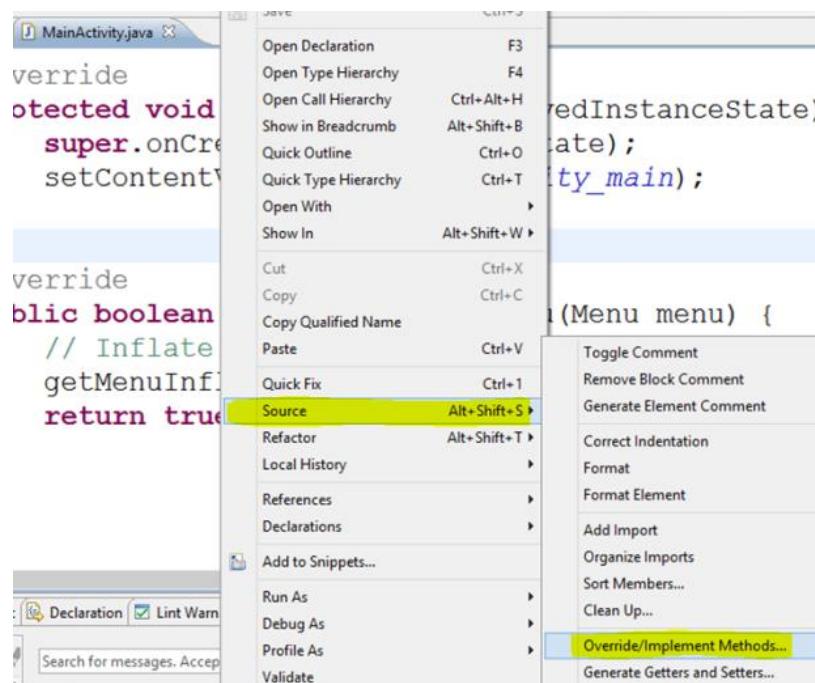
Bây giờ Tôi sẽ Demo ứng dụng Android để kiểm tra vòng đời của nó:

- Bạn tạo một ứng dụng tên là : **CheckLifeTimeCycle** với cấu trúc như hình dưới đây:

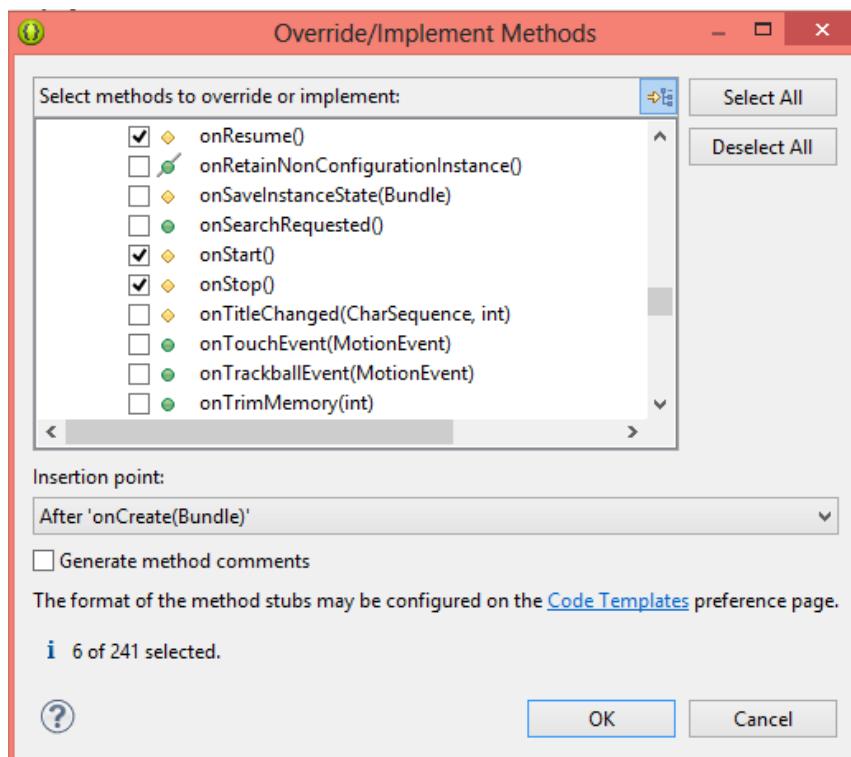


- Double Click vào **MainActivity.java**:

- Sau đó bấm chuột phải vào màn hình Coding/ chọn **Source/** chọn **Override / Implement Methods...** :



- Màn hình Override / Impement Methods sẽ hiển thị ra như bên dưới, bạn chọn các hàm: **onStart, onRestart, onResume, onPause, onStop, onDestroy...**: rồi bấm **OK**



Bạn xem coding bên trong:

```
MainActivity.java
import android.os.Bundle;

public class MainActivity extends Activity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }
    protected void onDestroy() {}
    protected void onPause() {}
    protected void onStop() {}
    @Override
    protected void onResume() {
        Toast.makeText(this,"onResume", Toast.LENGTH_SHORT)
            .show();
        super.onResume();
    }
    protected void onRestart() {}
    protected void onStart() {}
}
```

Ở trên bạn thấy dòng lệnh trong hàm onResume:

protected void onResume() {

 Toast.makeText(this,"onResume", Toast.LENGTH_SHORT)

 .show();

 super.onResume();

}

Đơn giản là Tôi chỉ hiển thị lên xem hàm nào sẽ được triệu gọi ứng với Life time cycle của Activity

- Bây giờ bạn chạy ứng dụng vào Máy ảo Android và thực hiện một số thao tác : Mở một ứng dụng khác, mở Menu, nhấn nút Back, nhấn nút Home ... quan sát hiện tượng bạn sẽ hiểu được cách vận hành các hành này.

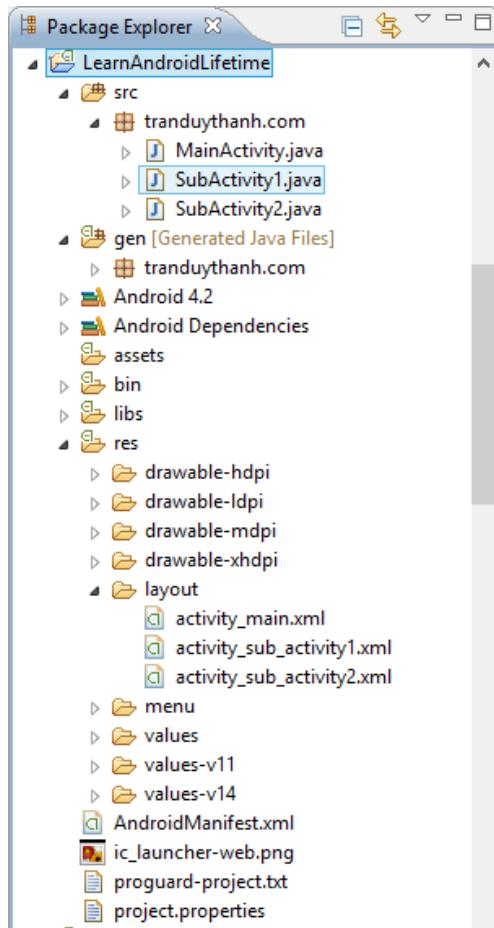
- Trong bài tập tiếp theo Tôi sẽ làm thêm một ví dụ về Life time cycle để bạn hiểu rõ hơn về nó, đặc biệt là tận mắt chứng kiến đâu là Visible Life time, đâu là Foreground Life time

- Bạn cần phải hiểu rõ về Life time cycle để giúp ích cho việc quản lý ứng dụng

Bài tập 6: Phân biệt Foreground Lifetime và Visible Lifetime

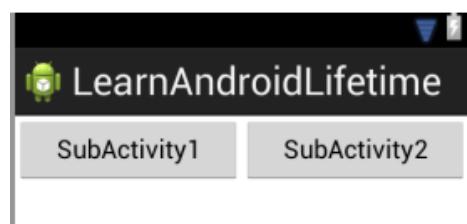
Để củng cố thêm sự hiểu biết về Lifetime Cycle trong [bài tập 5](#), bài tập này Tôi sẽ Demo cho các bạn một ứng dụng để bạn có thể hiểu rõ hơn về Lifetime Cycle, đặc biệt là nhận biết được Foreground Lifetime và Visible Lifetime:

Bạn tạo một ứng dụng tên là “**LearnAndroidLifetime**“, với cấu trúc như hình dưới:



- Nhìn vào hình bên trên thì ứng dụng này sẽ có tổng cộng 3 Activities. Chú ý là ta sẽ cấu hình **SubActivity1** để hiển thị dưới dạng Dialog (khi **SubActivity1** kích hoạt thì nó sẽ nằm phía trên **MainActivity**, nhưng mà vẫn nhìn thấy màn hình **MainActivity** – tức là đồng thời nhìn thấy 2 Activity). Còn khi **SubActivity2** hiển thị thì nó sẽ chiếm toàn bộ màn hình, không thể thấy được **MainActivity**.

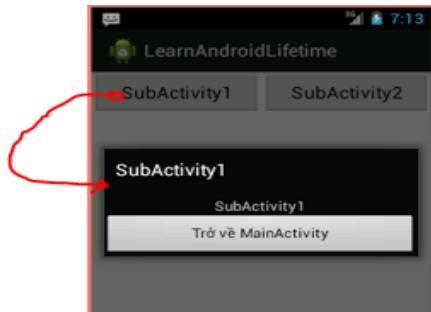
- Giao diện và xml layout của **MainActivity** sẽ như bên dưới:



đây là **activity_main.xml** của MainActivity

```
<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="horizontal"  
    tools:context=".MainActivity" >  
  
<Button  
    android:id="@+id/button1"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_weight="1"  
    android:text="SubActivity1" />  
  
<Button  
    android:id="@+id/button2"  
    android:layout_weight="1"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="SubActivity2" />  
  
</LinearLayout>
```

- Tại MainActivity: Khi chọn nút **SubActivity1**, chương trình sẽ mở SubActivity với dạng Dialog như hình bên dưới:



- Khi màn hình SubActivity1 hiển thị lên như trên, nếu chọn “Trở về MainActivity”, ứng dụng sẽ đóng SubActivity1.

- Bạn xem cấu hình **AndroidManifest.xml** để cho phép **SubActivity1** hiển thị dưới dạng Dialog:

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

    <uses-sdk
        android:minSdkVersion="14"
        android:targetSdkVersion="17" />

    <application
        android:allowBackup="true"
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/AppTheme" >
        <activity
            android:name="tranduythanh.com.MainActivity"
            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="tranduythanh.com.SubActivity1"
            android:label="@string/title_activity_sub_activity1"
            android:theme="@android:style/Theme.Dialog">
        </activity>
        <activity
            android:name="tranduythanh.com.SubActivity2"
            android:label="@string/title_activity_sub_activity2" >
        </activity>
    </application>

</manifest>
```

- Bạn xem code xử lý nút “**SubActivity1**” và “**SubActivity2**“ trong class **MainActivity**:

```
package tranduythanh.com;

import android.os.Bundle;

import android.app.Activity;

import android.content.Intent;

import android.view.View;

import android.widget.Button;

import android.widget.Toast;

public class MainActivity extends Activity {

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity_main);

Button btn1=(Button) findViewById(R.id.button1);

btn1.setOnClickListener(new View.OnClickListener() {

public void onClick(View arg0) {

Intent intent=new Intent(MainActivity.this,

SubActivity1.class);

startActivity(intent);

}

});

Button btn2=(Button) findViewById(R.id.button2);

btn2.setOnClickListener(new View.OnClickListener() {

public void onClick(View arg0) {

Intent intent=new Intent(MainActivity.this,
```

```

SubActivity2.class);

startActivity(intent);

}

});

}

}

```

- Phương thức **findViewById** để trả về View theo đúng Id truyền vào, mọi Control trong Android đều kế thừa từ View, ở đây ta ép kiểu về cho đúng kiểu Button.
- Dùng phương thức **startActivity** để mở một Activity, chú ý là Activity này được đưa vào một Intent với cách khai báo như trên.
- Cách chạy **SubActivity1** và **SubActivity2** là hoàn toàn giống nhau. Chỉ khác nhau ở lúc hiển thị lên màn hình điện thoại (như đã giải thích ở trên).
- Ta sẽ học rõ hơn về Intent trong phần sau, tạm thời phần này chưa cần phải hiểu sâu về nó. Chỉ cần biết rằng để chạy 1 Activity nào đó thì phải tiến hành như vậy.

*** Tiếp tục làm việc với class **MainActivity**:

Bạn tạo Override thêm một số hàm : onResume, onPause, onStop, onDestroy...:

```

@Override

protected void onResume() {

super.onResume();

Toast.makeText

(this, "onResume", Toast.LENGTH_LONG)

.show();

}

@Override

protected void onPause() {

super.onPause();

```

```

Toast.makeText
(this, "onPause", Toast.LENGTH_LONG)

.show();

}

@Override

protected void onStop() {

super.onStop();

Toast.makeText
(this, "onStop", Toast.LENGTH_LONG)

.show();

}

@Override

protected void onDestroy() {

super.onDestroy();

Toast.makeText
(this, "onDestroy", Toast.LENGTH_LONG)

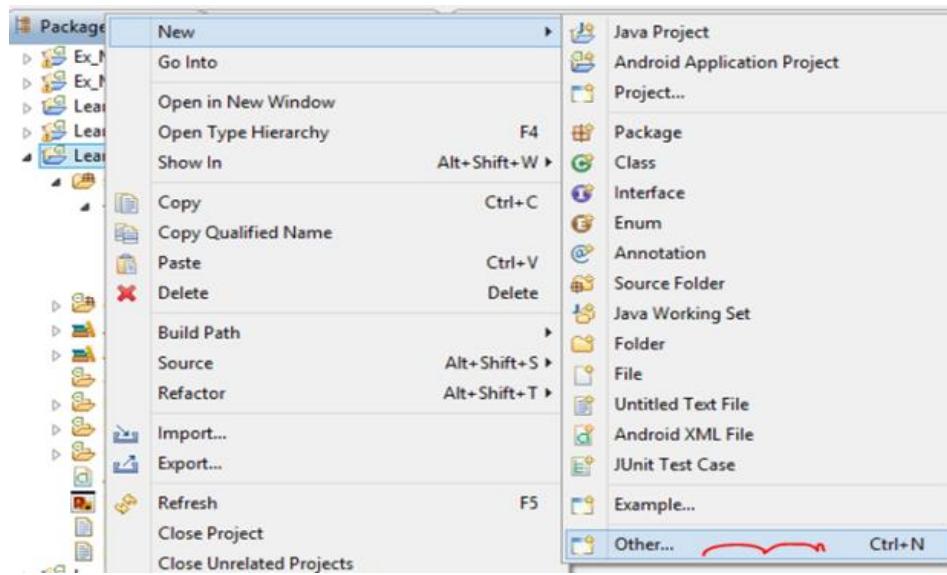
.show();

}

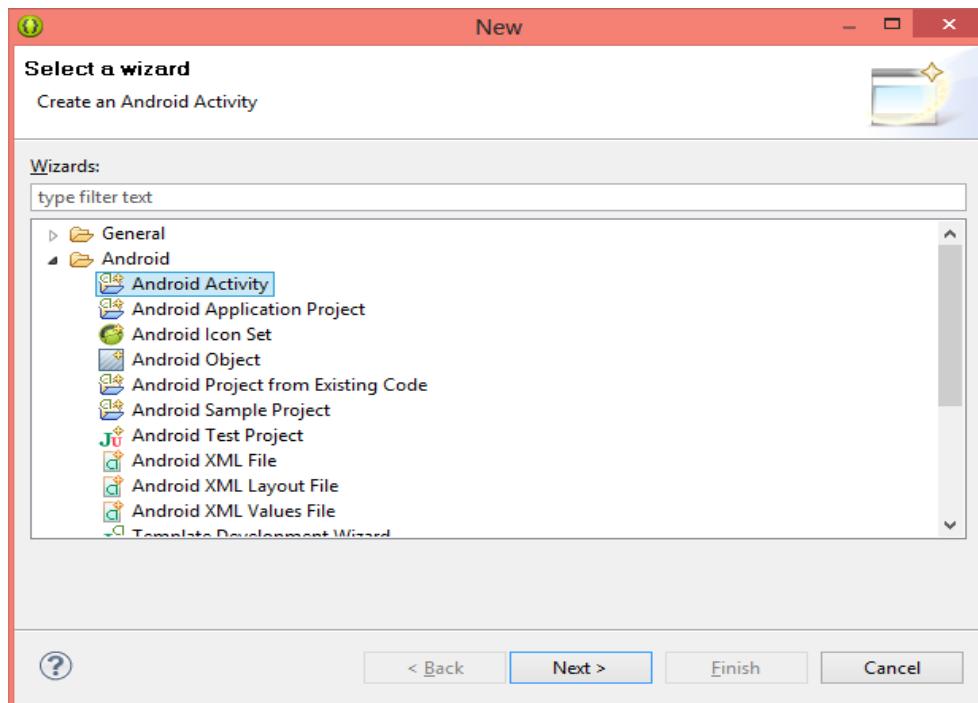
```

- Bạn thiết kế SubActivity1 sao cho giống như hình . Bạn cần lưu ý là mỗi khi chúng ta tạo mới 1 Activity thì nó sẽ tự động tạo ra 1 Layout cho nó luôn:

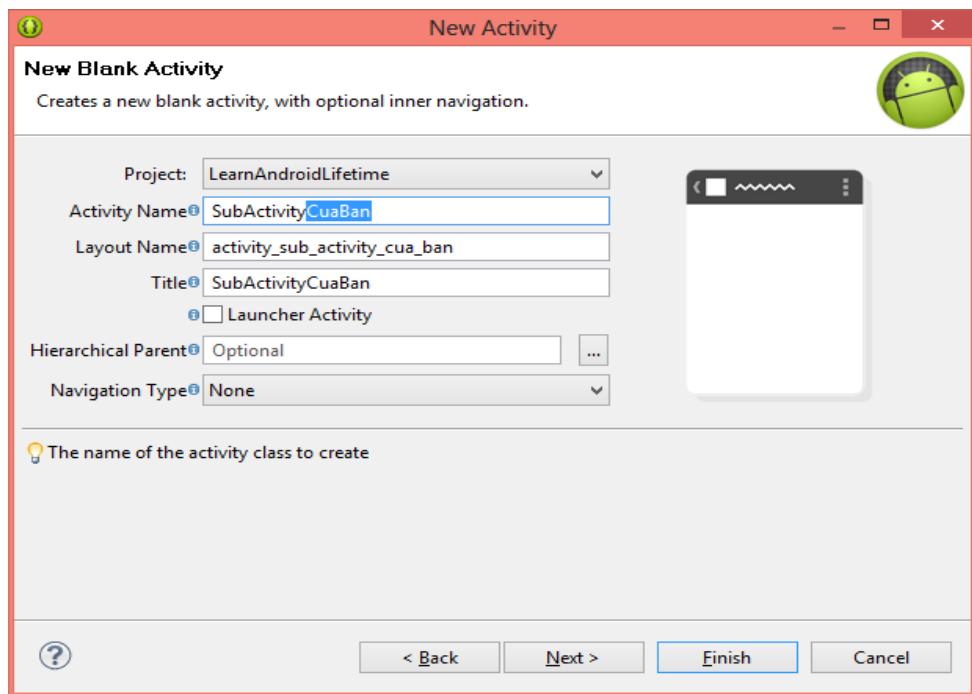
+ 1 – Bấm chuột phải vào **Project** / chọn **New** / chọn **Other** ...



+ 2 – sau khi bấm Other thì màn hình New được hiển thị lên, tại màn hình này bạn chọn AndroidActivity rồi bấm next:



+3 – tiếp tục bấm Next cho tới màn hình bên dưới, trong màn hình này bạn đặt tên Activity mà bạn mong muốn, để ý là bên dưới tên Activity sẽ cho phép bạn đặt tên Layout cho Activity này, thông thường tên Layout sẽ đi kèm với tên Activity do đó bạn không cần thiết phải đổi tên Layout:



+ 4- bấm Finish, và quan sát ứng dụng sẽ xuất hiện Activity và Layout cho bạn. Bây giờ bạn tha hồ thiết kế.

Trong ví dụ của Tôi thì bạn chỉ cần kéo 1 Button vào Layout của SubActivity1

Bạn đặt tên Button trong SubActivity1 là **btntrove**, để cho giống với Code trở về MainActivity dưới này:

```
package tranduythanh.com;

import android.os.Bundle;

import android.app.Activity;

import android.view.View;

import android.widget.Button;

public class SubActivity1 extends Activity {

    @Override

    protected void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity_sub_activity1);

        Button btn=(Button) findViewById(R.id.btntrove);
```

```
btn.setOnClickListener(new View.OnClickListener() {  
    @Override  
  
    public void onClick(View arg0) {  
  
        finish();  
  
    }  
  
});  
  
}  
  
}
```

- Bạn thấy trong SubActivity1 Tôi gọi hàm **finish()** khi chọn “Trở về MainActivity”. Công dụng của hàm **finish()** là tắt màn hình hiện tại→ như vậy nó tự động mở MainActivity vì theo cơ chế Activities Stack.

- Nay giờ bạn thực thi chương trình và làm các thao tác sau:

- 1) Chọn nút SubActivity1 để mở Activity này lên, quan sát hiện tượng xảy ra (chú ý những hàm nào được gọi)
- 2) Khi chọn nút “Trở về MainActivity”, quan sát hiện tượng xảy ra (chú ý những hàm nào được gọi)
- 3) Chọn nút SubActivity2 để mở Activity này lên, quan sát hiện tượng xảy ra (chú ý những hàm nào được gọi)
- 4) Nhấn nút “Back” của điện thoại, quan sát hiện tượng (chú ý những hàm nào được gọi)

*** Tới đây Tôi sẽ không giải thích thứ tự các hàm xảy ra như thế nào, mà để dành cho các bạn tự giải thích. Bạn sẽ thấy rõ sự khác biệt trong 2 trường hợp này. Và dĩ nhiên bạn sẽ tự động hiểu được Foreground Lifetime và Visible Lifetime

Các bạn có thể tải coding mẫu ở

đây: <http://www.mediafire.com/download/7ra8h7vyttkjq6u/LearnAndroidLifetime.rar>

Trong bài tập kế tiếp Tôi sẽ hướng dẫn các bạn cách sử dụng XML Layout và một số control căn bản thường dùng cũng như các kiểu lập trình sự kiện.

Bài tập 7: Làm quen với các Layout trong Android

Để thiết kế giao diện tốt cho Android thì trước tiên các bạn phải biết sử dụng một số Layout cơ bản trong Android:

1) Cách tạo Layout và kết nối Layout vào Activity

2) Cách sử dụng HierarchyViewer

3) Các Layout cơ bản:

- **FrameLayout**
- **LinearLayout**
- **TableLayout**
- **RelativeLayout**
- **AbsoluteLayout**

*** Đồng thời bạn phải biết kết hợp các Layout này lại với nhau để cho ra được giao diện phức tạp như ý muốn.

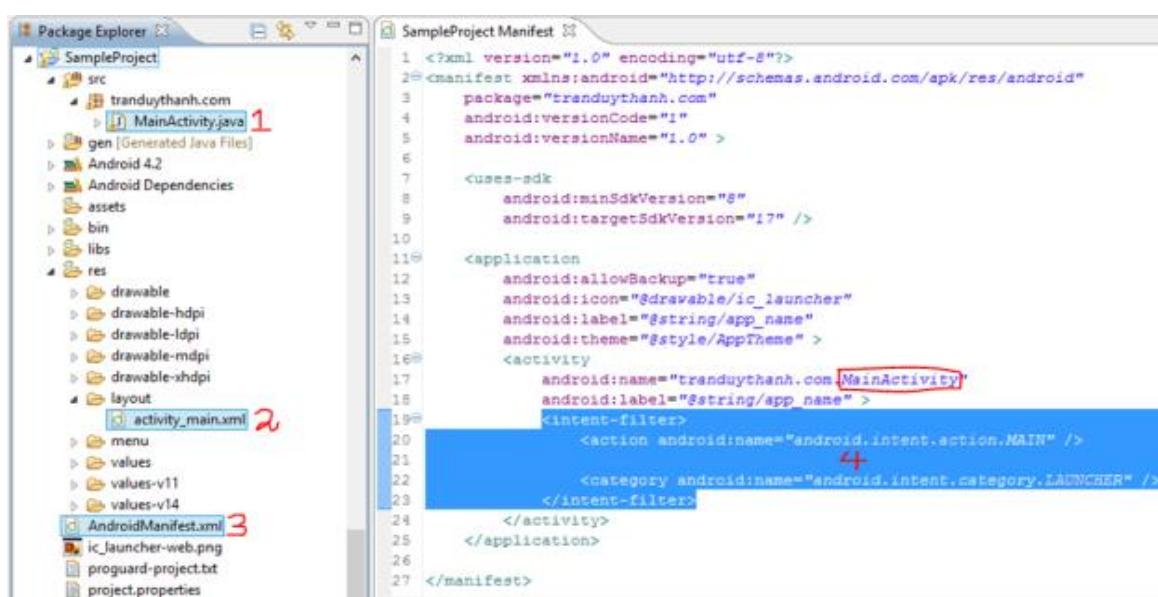
- Tôi chỉ nêu ra một số đặc điểm, không thể nói hết ở đây. Các bạn sẽ tự cập nhật sau khi đã quen với Layout.

- Trước tiên bạn cần biết cách tạo một Layout mới và cách kết nối nó vào Activity như thế nào:

1) Cách tạo Layout và kết nối Layout vào Activity:

a) Layout mặc định đầu tiên khi bạn tạo một Android Project:

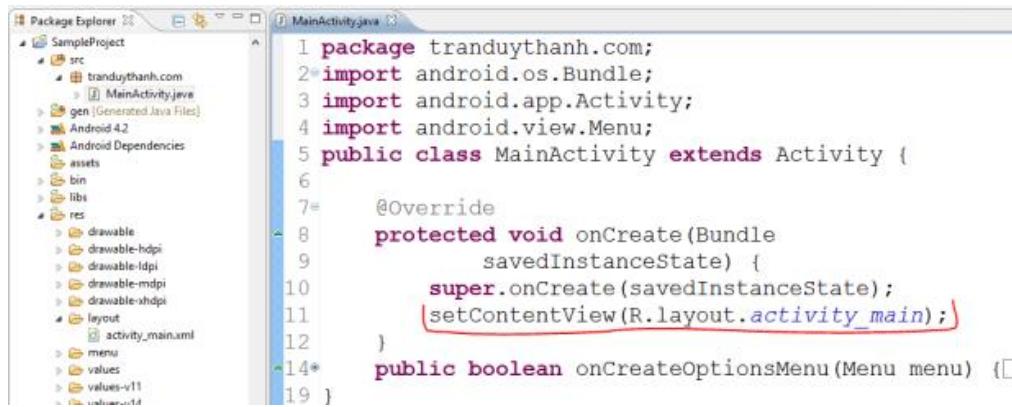
- Khi bạn tạo một Android Project thì mặc nhiên sẽ có một Activity được chỉ định chạy đầu tiên khi bạn thực thi ứng dụng:



- Bạn quan sát hình trên: Khi bạn tạo một ứng dụng Android thì mặc nhiên sẽ tạo luôn một Activity để thực thi đầu tiên khi bạn chạy ứng dụng. Ứng với một Activity thì nó sẽ có 1 Layout đi kèm. Trong hình trên thì **MainActivity.java** (số 1) sẽ có layout đi kèm là **activity_main.xml** (số 2 - và bạn nhớ luôn là activity_main này sẽ được tự động tạo ra trong thư mục **gen** của Android, dựa vào đây để ta kết nối Layout vào Activity).

- Như đã nói ở những bài tập trước: Mọi Activity muốn được triết gọi thành công trong ứng dụng thì bắt buộc nó phải được khai báo trong **AndroidManifest.xml** (số 3). Bạn double click vào Manifest và nhìn vào vùng số 4, **MainActivity** được khai báo trong này đồng thời đăng ký là Activity sẽ khởi động lúc ứng dụng được thực thi (xem vùng Tôi bôi màu xanh phần **intent-filter**).

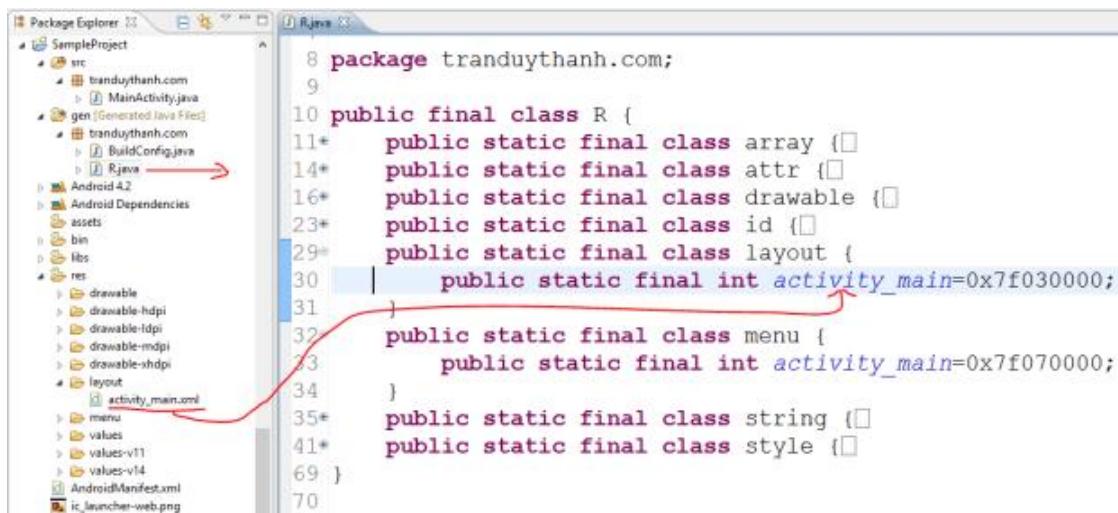
- Tiếp theo bạn Double – click vào tập tin **MainActivity.java**:



```

1 package tranduythanh.com;
2 import android.os.Bundle;
3 import android.app.Activity;
4 import android.view.Menu;
5 public class MainActivity extends Activity {
6
7     @Override
8     protected void onCreate(Bundle savedInstanceState) {
9         super.onCreate(savedInstanceState);
10        setContentView(R.layout.activity_main);
11    }
12
13    public boolean onCreateOptionsMenu(Menu menu) { }
14
15 }
```

- Nhìn vào hàm **onCreate**, bạn thấy dòng lệnh : **setContentView(R.layout.activity_main)**; chính là dòng lệnh dùng để kết nối Layout vào Activity. Bạn nhớ là activity_main phải được tự động tạo ra trong **gen** như hình bên dưới:

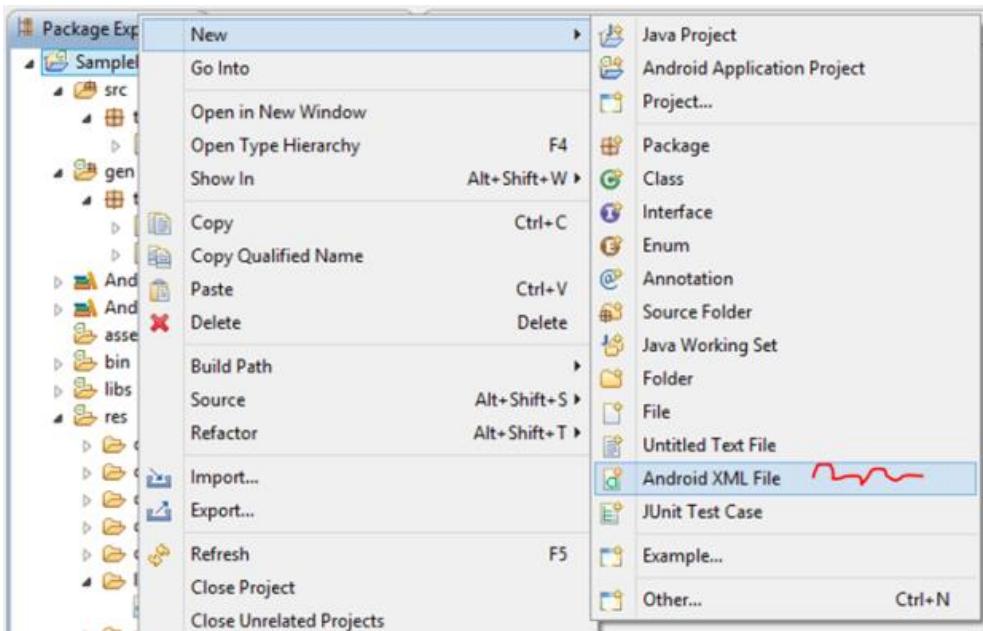


```

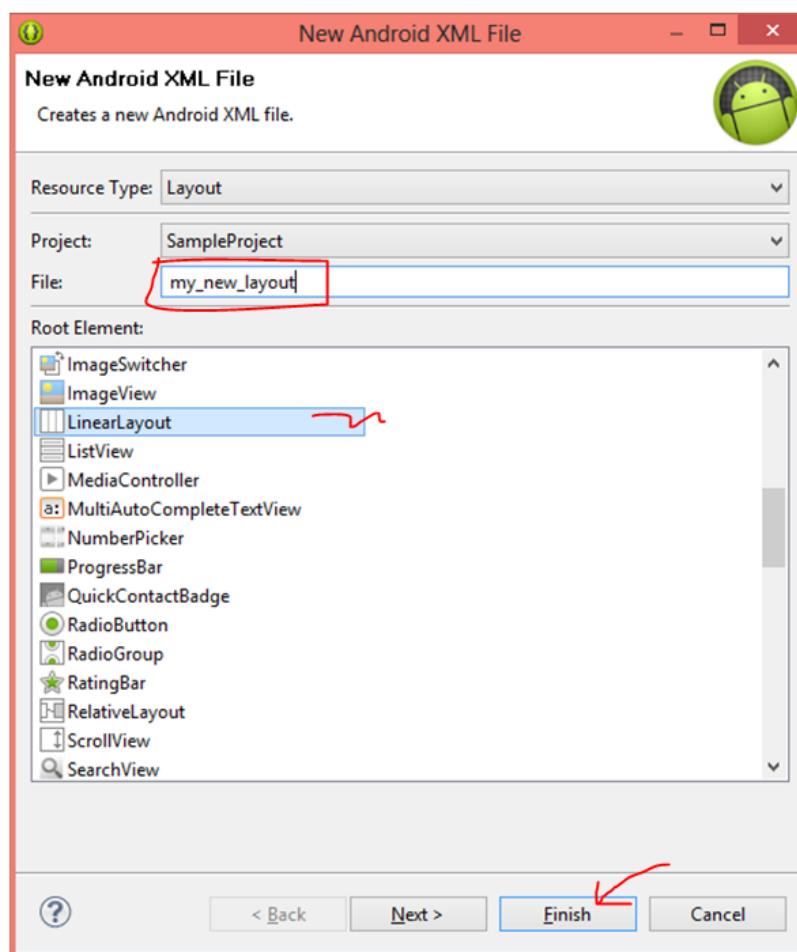
8 package tranduythanh.com;
9
10 public final class R {
11     public static final class array {}
12     public static final class attr {}
13     public static final class drawable {}
14     public static final class id {}
15     public static final class layout {
16         public static final int activity_main=0x7f030000;
17     }
18     public static final class menu {
19         public static final int activity_main=0x7f070000;
20     }
21     public static final class string {}
22     public static final class style {}
23 }
```

b) Đổi Layout mặc định bằng một Layout khác bất kỳ:

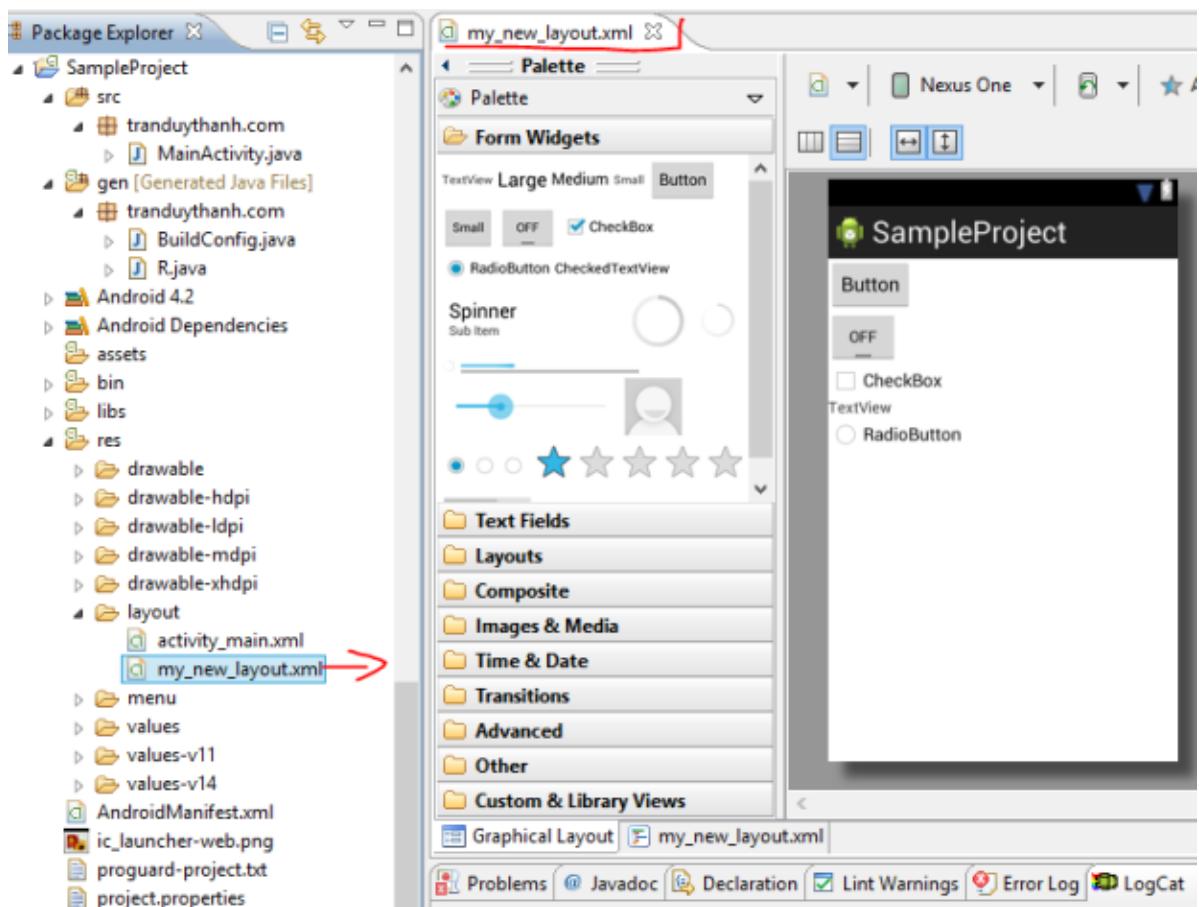
- Để tạo một Layout mới, bạn click chuột phải vào **Project**/ chọn **New**/ chọn **Android XML File**:



- Khi bạn chọn Android XML File thì màn hình bên dưới xuất hiện: Đặt tên cho Layout, chọn kiểu Layout rồi nhấn nút Finish:



- Ở trên Tôi đặt tên là : **my_new_layout**, sau khi nhấn nút Finish bạn quan sát Package Explorer:

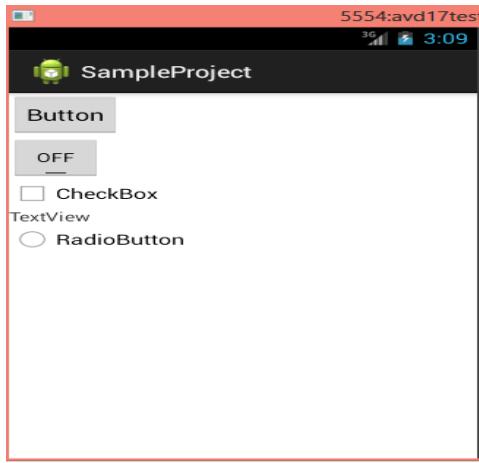


- Bạn thấy Tôi double – click vào layout : **my_new_layout** và kéo thả một số control vào giao diện như hình bên trên (bạn nhớ là **my_new_layout** phải được tự động xuất hiện trong **gen** – **bạn tự kiểm tra**).

- Nay bạn vào lại MainActivity.java. Sửa lại dòng lệnh setContentView thành:

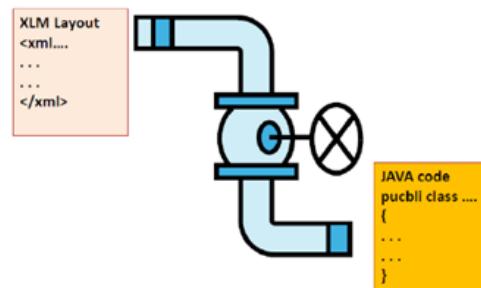
```
protected void onCreate(Bundle
    savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.my_new_layout);
}
```

- Khởi động chương trình và bạn thấy ứng dụng sẽ chạy cái Layout mới này chứ không phải Layout cũ nữa:



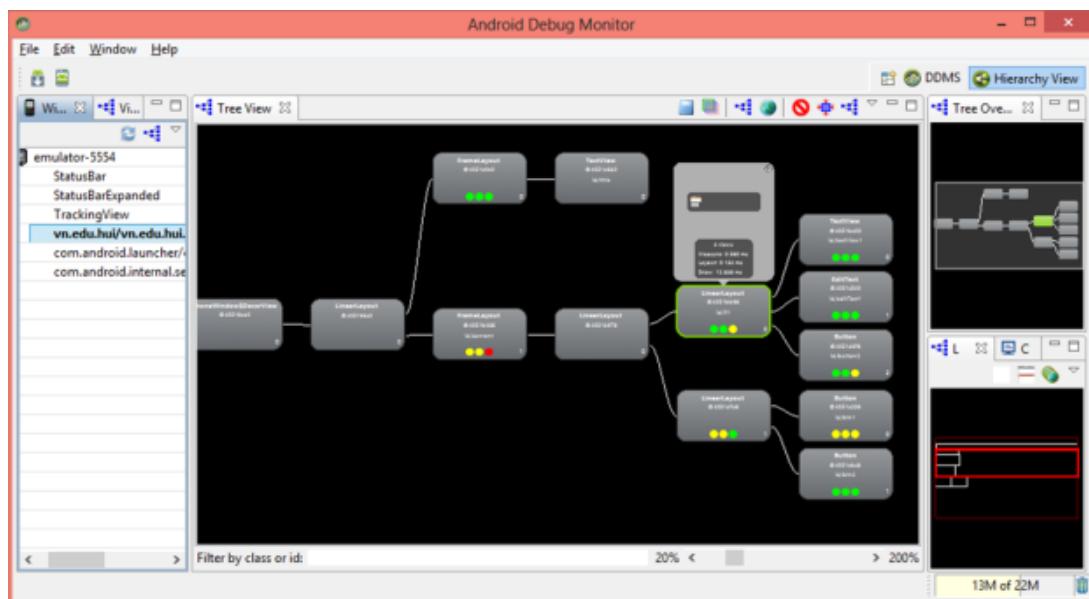
2) Cách sử dụng HierarchyViewer:

- Như bạn đã biết, một Layout phải được kết nối vào Activity nào đó thông qua hàm **setContentView**, Android sẽ có cơ chế dịch XML thành Java code:



- Ta có thể dùng **HierarchyViewer** để hiển thị cấu trúc UI của màn hình hiện tại trên emulator hoặc thiết bị thật:

Bạn vào thư mục **SDK** của bạn / vào thư mục **Tools**/ rồi chạy tập tin **monitor.bat** :



3) Các Layout cơ bản:

a) FrameLayout:

- Là loại Layout cơ bản nhất, đặc điểm của nó là khi gắn các control lên giao diện thì các control này sẽ luôn được “Neo” ở góc trái trên cùng màn hình, nó không cho phép chúng ta thay đổi vị trí của các control theo một Location nào đó.

- Các control đưa vào sau nó sẽ đè lên trên và che khuất control trước đó (trừ khi ta thiết lập transparent cho control sau):

- Bạn xem đoạn cấu trúc XML dưới này:

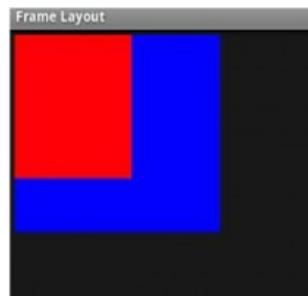
```
<?xml version="1.0" encoding="utf-8"?>

<FrameLayout android:id="@+id/mainlayout" android:layout_height="fill_parent"
    android:layout_width="fill_parent" android:orientation="vertical"
    xmlns:android="http://schemas.android.com/apk/res/android">

    <ImageView android:layout_height="wrap_content" android:layout_width="wrap_content"
        android:padding="5px" android:src="@drawable/blue"/>

    <ImageView android:layout_height="wrap_content" android:layout_width="wrap_content"
        android:padding="5px" android:src="@drawable/red"/>

</FrameLayout>
```



- Bạn thấy đó, hình màu đỏ và màu xanh luôn được “neo” ở góc trái màn hình. Hình màu đỏ đưa vào sau sẽ đè lên trên hình màu xanh.

Chú ý 2 dòng lệnh bên dưới:

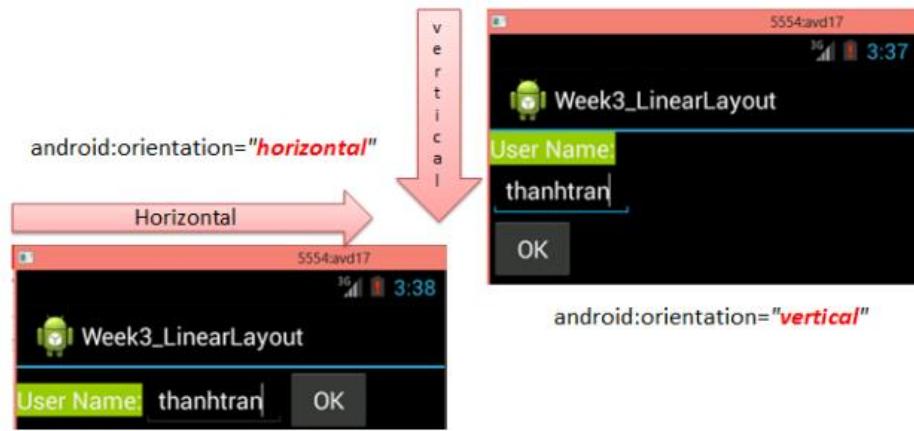
- android:src="@drawable/blue"

- android:src="@drawable/red"

Là do ta kéo 2 cái hình tên là **blue** và **red** vào thư mục drawable của ứng dụng

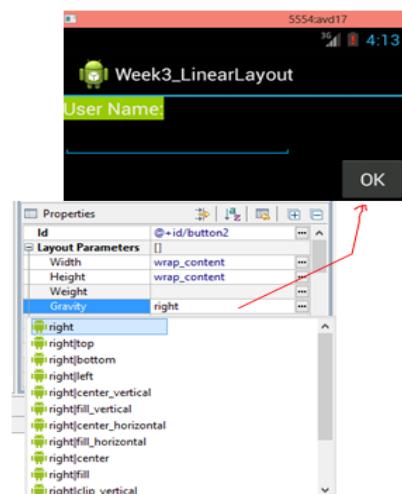
b) LinearLayout

- Layout này cho phép sắp xếp các control theo 2 hướng trên giao diện: Hướng từ trái qua phải và hướng từ trên xuống dưới.



- Bạn có thể dùng margin, gravity, weight để hỗ trợ cho việc thiết kế. Ở đây Tôi không có thời gian nhiều nên chỉ nói đặc điểm chính của LinearLayout, các bạn tự tìm hiểu thêm.

- Ta có thể dùng Properties hỗ trợ sẵn trong Eclipse để thiết lập các thuộc tính cho control:
- Ví dụ như để căn lề các control trên giao diện ta dùng **layout_gravity**:

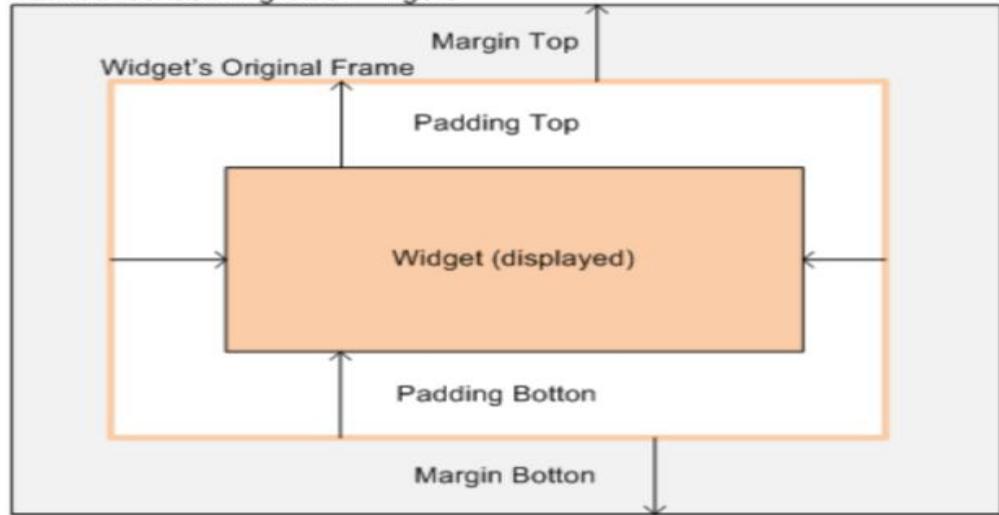


- hay để căn lề nội dung bên trong của control dùng **gravity**:

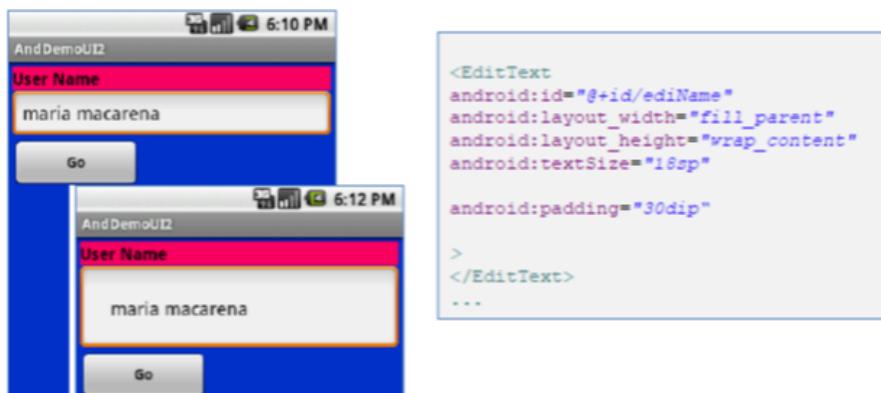


- Bạn cũng phải biết so sánh sự khác biệt giữa **Padding** và **Margin**:

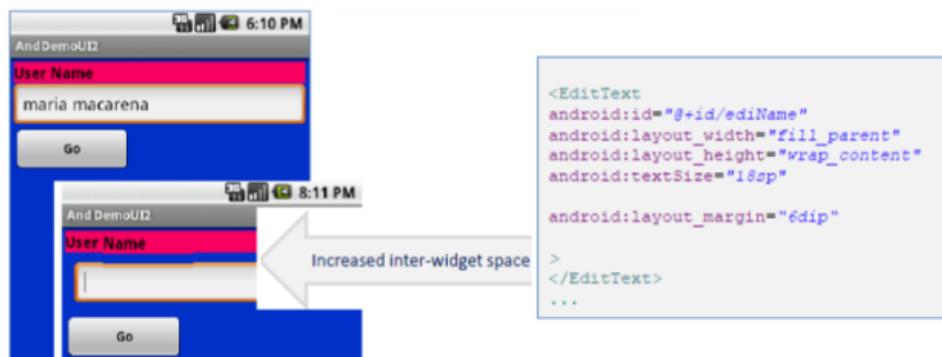
Boundaries touching other widgets



- Ví dụ thay đổi Padding (internal spacing – khoảng cách giữa nội dung bên trong so với đường viền của control):



- Ví dụ về đổi Margin (external spacing – khoảng cách giữa control này với control khác):



c) TableLayout

- Cho phép sắp các control theo dạng lưới (dòng và cột)

- TableLayout sẽ xem dòng nào có số lượng control nhiều nhất để xác định rằng nó có bao nhiêu cột (lấy dòng có số lượng control nhiều nhất làm số cột chuẩn).

0		1	
0		1	2
0	1	2	3

- Như vậy theo hình trên thì bạn phải nói là TableLayout này có 4 cột, 3 dòng.

- Dùng **layout_span** để trộn các cột:

```
<TableRow>
    <TextView android:text="URL:" />
    <EditText
        android:id="@+id/entry"
        android:layout_span="3" />
</TableRow>
```

- Dùng **layout_column** để di chuyển vị trí của control đến một cột nào đó trên 1 dòng:

Label (URL)	EditText	EditText-span	EditText-span
Column 0	Column 1	Column 2 Button Cancel	Column 3 Button OK

- Dùng **stretchColumns** để dãn đều các control, các cell (ta thường dùng dấu “*”):

```
<TableLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:stretchColumns="*"
    >
```

d) **RelativeLayout:**

- RelativeLayout cho phép sắp xếp các control theo vị trí tương đối giữa các control khác trên giao diện (kể cả control chưa nó). Thường nó dựa vào Id của các control khác để sắp xếp theo vị trí tương đối. Do đó khi làm RelativeLayout bạn phải chú ý là đặt Id control cho chuẩn xác, nếu sau khi Layout xong mà bạn lại đổi Id của các control thì giao diện sẽ bị xáo trộn (do đó nếu đổi ID thì phải đổi luôn các tham chiếu khác sao cho khớp với Id bạn mới đổi).

- Dưới đây là ví dụ về cách sử dụng RelativeLayout (Bạn để ý những dòng tô đậm):

```

<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:background="#ff0000ff"
    android:padding="10px" >

    <TextView android:id="@+id/label"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:background="#ffff0077"
        android:text="Type here:" />

    <EditText android:id="@+id/entry"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_below="@+id/label" />

```

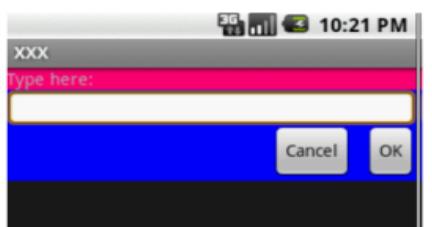
```

<Button
    android:id="@+id/ok"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@+id/entry"
    android:layout_alignParentRight="true"
    android:layout_marginLeft="10px"
    android:text="OK" />

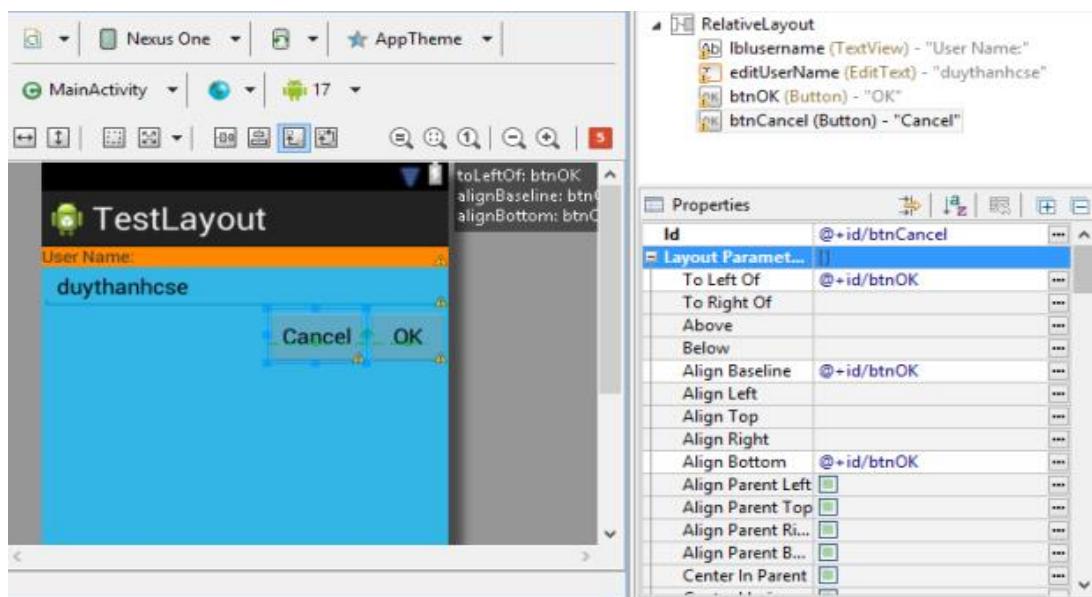
<Button
    android:text="Cancel"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_toLeftOf="@+id/ok"
    android:layout_alignTop="@+id/ok" />

```

</RelativeLayout>



- Ta có thể sử dụng công cụ trong Eclipse để thiết kế :



e) AbsoluteLayout:

- Cho phép thiết lập các control giao diện theo vị trí tùy thích:

```

<?xml version="1.0" encoding="utf-8"?>
<AbsoluteLayout
    android:id="@+id/myAbsoluteLayout"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    xmlns:android="http://schemas.android.com/apk/res/android">

    <Button
        android:id="@+id/myButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Button"
        android:layout_x="120px"
        android:layout_y="32px"
        >

```



- Như vậy các Tôi đã giới thiệu sơ qua cách tạo Layout và cách sử dụng một số Layout cụ thể, tùy từng trường hợp mà các bạn ứng dụng vào. Thông thường ứng với mỗi một ứng dụng cụ thể thì bạn phải kết hợp nhiều loại Layout lại với nhau.
- Bạn cần hiểu rõ các loại Layout để giúp ích cho quá trình thiết kế giao diện
- Bài tập sau Tôi sẽ hướng dẫn các bạn cách sử dụng một số control cơ bản và cách viết sự kiện cho Button như thế nào.

Bài tập 8: Các kiểu lập trình sự kiện trong Android

Bài tập này Tôi sẽ trình bày 6 kiểu lập trình sự kiện trong Android. Tôi sẽ sử dụng một số Control căn bản để Demo, đặc biệt là **Button** dùng để tạo sự kiện.

1. **Onclick in XML**
2. **Inline anonymous listener**
3. **Activity is listener**
4. **Listener in variable**
5. **Explicit listener class**
6. **View Subclassing**

- Tôi sẽ lần lượt đưa ra 6 ví dụ khác nhau cho 6 kiểu lập trình sự kiện ở trên, các bạn hãy cố gắng theo dõi, nó rất quan trọng để làm các bài tập tiếp theo.

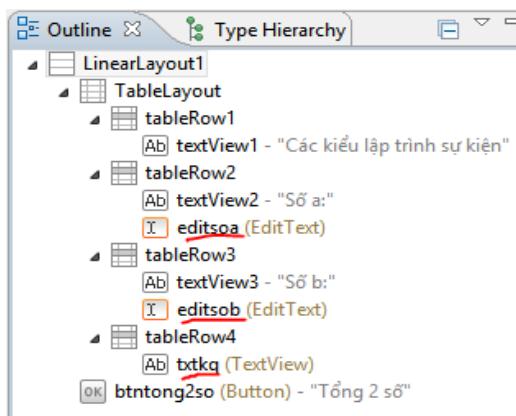
1. Onclick in XML:

- Ví dụ 1: Đơn giản chỉ là cộng 2 số, bạn thiết kế giao diện như bên dưới:



- Khi nhấn vào nút “Tổng 2 số”, chương trình sẽ xuất kết quả như hình bên trên : $80+33=113$

- Bạn xem Layout Outline để dễ thiết kế (chú ý là bạn có thể bỏ LinearLayout1 đi):



- Chú ý là ta sử dụng Onclick in XML:

```

<Button
    android:id="@+id/btn_tong2so"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:onClick="btn_tong2so"
    android:text="Tổng 2 số" />

```

- Trong đoạn lệnh ở trên thì ta sử dụng **android:onClick="btn_tong2so"**, tức là ta đã gán một sự kiện click cho Button này, sự kiện này tên là **btn_tong2so**. Ta cần khai báo một hàm **btn_tong2so** ở trong Activity class như hình bên dưới:

```

10 public class MainActivity extends Activity {
11     protected void onCreate(Bundle savedInstanceState) { }
12     public void btn_tong2so(View v)
13     {
14         EditText edita=(EditText)
15             findViewById(R.id.editsoa);
16         int a=Integer.parseInt(edita.getText()+"");
17
18         EditText editb=(EditText) findViewById(R.id.editsob);
19
20         int b=Integer.parseInt(editb.getText()+"");
21
22         TextView txtkq=(TextView) findViewById(R.id.txtkq);
23         txtkq.setText((a+b)+"");
24     }
25     public boolean onCreateOptionsMenu(Menu menu) { }
26 }

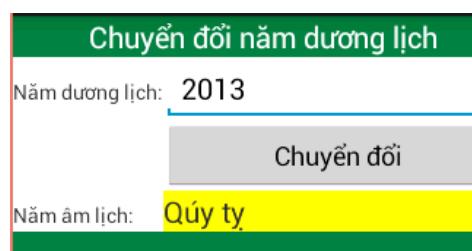
```

- Khi chạy ứng dụng bạn sẽ được kết quả như bên dưới:



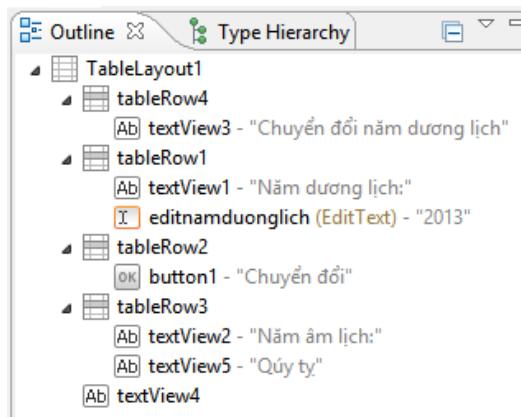
2. Inline anonymous listener

- Ví dụ viết chương trình chuyển đổi năm dương lịch qua năm âm lịch như hình bên dưới:



- Khi người sử dụng nhập vào EditText giá trị là 1 năm Dương Lịch bất kỳ nào đó rồi nhấn nút “Chuyển đổi”, chương trình sẽ chuyển năm dương lịch thành năm âm lịch. Trong ví dụ trên nếu người sử dụng nhập 2013 thì sẽ ra năm âm lịch là “Quý Tỵ”.

- Chú ý là ta tạo một anonymous listener, trước tiên bạn hãy xem Outline XML để cho dễ bè thiết kế:



- Để chuyển từ năm dương lịch sang năm âm lịch bạn cần biết một số thông tin sau:

Năm Âm = Can + Chi;
Can = Năm dương % 10 :

0	1	2	3	4	5	6	7	8	9
Canh	Tân	Nhâm	Quý	Giáp	Ất	Bính	Đinh	Mậu	Kỷ

0	1	2	3	4	5	6	7	8	9	10	11
Thân	Dậu	Tuất	Hợi	Tý	Sửu	Dần	Mão	Thìn	Tỵ	Ngo	Mùi

Chi = Năm dương % 12 :

```
String can,chi;
switch(namduong%10){
    //lấy can trong này dựa vào bảng số 1
}
switch(namduong%12){
    //lấy chi trong này dựa vào bảng số 2
}
Ghép can và chi ta được năm âm lịch
```

- Nay giờ ta tiến hành gán sự kiện cho nút “Chuyển đổi” (ở đây id Tôi để là **button1**), mở Activity class lên vào sửa lệnh như bên dưới:

```

public class MainActivity extends Activity {
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        Button btnDoi=(Button) findViewById(R.id.button1);
        btnDoi.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View arg0) {
                //Bước 1. Lấy giá trị nhập từ EditText
                //Bước 2. ép giá trị nhập vào về số nguyên
                //Bước 3. có số nguyên là đại diện cho năm dương
                //         áp dụng hướng dân tính Can + Chi ở trên
                //         để tìm ra năm âm lịch
            }
        });
    }
    public boolean onCreateOptionsMenu(Menu menu) {□
}
}

```

- Bạn tự đưa lệnh vào Bước 1, Bước 2, bước 3 ở trên. Cách lấy dữ liệu nhập vào từ EditText đã hướng dẫn ở phần **Onclick in XML**, làm theo cái này để lấy được giá trị là năm dương lịch ra, sau đó lấy năm này xử lý theo bảng **Can** và **Chi** như hướng dẫn thì Ta sẽ ra được năm Âm lịch tương ứng.

3. Activity is listener:

- Trong cách viết sự kiện này thì **Activity sẽ implements interface** có kiểu sự kiện (rất nhiều loại interface). Ở đây Tôi chỉ ví dụ trường hợp cho Button các trường hợp khác các bạn tự tìm hiểu và suy luận ra.

- Tôi sẽ có một bài ví dụ nhỏ sau: Hãy xây dựng ứng dụng tính **Chỉ số khối cơ thể** -chữ viết tắt **BMI (Body Mass Index)** - được dùng để đánh giá mức độ gầy hay béo của một người. Chỉ số này có thể giúp xác định một người bị bệnh **béo phì** hay bị bệnh **suy dinh dưỡng**.

+Cách tính như sau:

Gọi **W** là khối lượng của một người (tính bằng **kg**) và **H** là chiều cao của người đó (tính bằng **m**), chỉ số khối cơ thể được tính theo công thức:

$$BMI = \frac{W}{(H)^2}$$

Phân loại để đánh giá như sau:

- BMI < 18: người gầy
- BMI = 18 – 24,9: người bình thường
- BMI = 25 – 29,9: người béo phì độ I
- BMI = 30 – 34,9: người béo phì độ II
- BMI > 35: người béo phì độ III

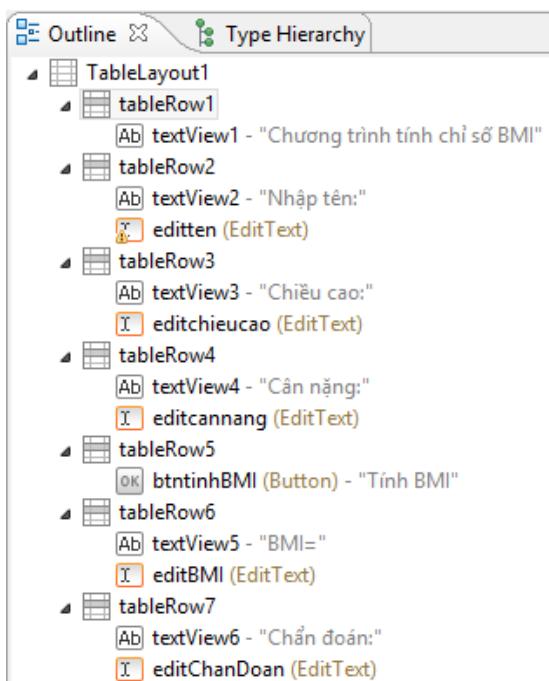
Tôi sẽ thiết kế giao diện như hình bên dưới và cung cấp Outline, các bạn hãy thiết kế lại để nâng cao kinh nghiệm:



-Bạn thấy đó: Thông số bên chính là của chính Tôi, Tên Tôi là Thanh, chiều cao 1.68 mét, cân nặng 58 kg. Khi Tôi chọn “Tính BMI” thì chương trình sẽ tính ra được BMI của Tôi là 20.5 và chẩn đoán là “Bình thường”, tức là Tôi có sức khỏe tốt, dáng người chuẩn (có thể là niềm ước ao của một số các bạn trong lớp).

- Để các bạn đỡ căng thẳng thì Tôi xin nói lý do vì sao Tôi lại viết phần mềm này làm ví dụ minh họa. Bởi vì bạn gái của Tôi luôn luôn chê Tôi gầy (hay gọi là Xí Quách gì đó). Tôi đã chứng minh là Tôi có dáng người chuẩn không cần chỉnh bằng cách viết phần mềm này (công thức theo chuẩn Quốc Tế nên nó là một bằng chứng không thể chối cãi). Tôi nghĩ các bạn cũng có thể dùng nó để chứng minh rằng mình không bị Béo Phì Cấp độ 3.

- Quay trở lại ví dụ ,Bạn xem Outline của giao diện này dưới đây (Tôi cố tình không cung cấp source XML nhằm mục đích kích thích các bạn tự tìm tòi):



- Đây là nội dung Coding trong Activity:

```
import java.text.DecimalFormat;
```

```
import android.os.Bundle;  
  
import android.app.Activity;  
  
import android.view.View;  
  
import android.view.View.OnClickListener;  
  
import android.widget.Button;  
  
import android.widget.EditText;  
  
public class MainActivity extends Activity  
  
implements OnClickListener{  
  
Button btnChandoan;  
  
EditText editTen,editChieucao,  
  
editCannang,editBMI,editChandoan;  
  
@Override  
  
protected void onCreate(Bundle savedInstanceState) {  
  
super.onCreate(savedInstanceState);  
  
setContentView(R.layout.activity_main);  
  
btnChandoan=(Button) findViewById(R.id.btntinhBMI);  
  
btnChandoan.setOnClickListener(this);  
  
editTen=(EditText) findViewById(R.id.editten);  
  
editChieucao=(EditText) findViewById(R.id.editchieucao);  
  
editCannang=(EditText) findViewById(R.id.editcannang);  
  
editBMI=(EditText) findViewById(R.id.editBMI);  
  
editChandoan=(EditText) findViewById(R.id.editChanDoan);  
  
}  
  
@Override
```

```
public void onClick(View arg0) {  
  
    double H=Double.parseDouble(editChieucao.getText().toString());  
  
    double W=Double.parseDouble(editCannang.getText().toString());  
  
    double BMI=W/Math.pow(H, 2);  
  
    String chandoan="";  
  
    if(BMI<18)  
    {  
  
        chandoan="Bạn gầy";  
  
    }  
  
    else if(BMI<=24.9)  
    {  
  
        chandoan="Bạn bình thường";  
  
    }  
  
    else if(BMI<=29.9)  
    {  
  
        chandoan="Bạn béo phì độ 1";  
  
    }  
  
    else if(BMI<=34.9)  
    {  
  
        chandoan="Bạn béo phì độ 2";  
  
    }  
  
    else  
    {  
  
        chandoan="Bạn béo phì độ 3";  
  
    }  
}
```

```
}

DecimalFormat dcf=new DecimalFormat("#.0");

editBMI.setText(dcf.format(BMI));

editChandoan.setText(chandoan);

}

}
```

- a) Bạn quan sát Activity này sẽ implements interface **OnClickListener**
 - b) Bản thân interface **OnClickListener** có một Abstract Method là **onClick(View arg0)** nên ta phải Override nó.
 - c) Để Button có thể hiểu được sự kiện thì Ta phải gọi dòng lệnh: **btnChandoan.setOnClickListener(this);**; bản thân Activity là một sự kiện nên ta dùng **this** để truyền vào hàm.
 - d) Bạn quan sát thêm Tôi có sử dụng : **DecimalFormat dcf=new DecimalFormat("#.0");** Mục đích là Tôi định dạng 1 số lẻ thập phân, bạn muốn 2 số lẻ thập phân thì ghi “#.00” hay muốn 3 thì “#.000” ...vân vân.
- Bạn có thể load đầy đủ coding mẫu ở đây: <http://www.mediafire.com/?71khpisxa3wmvq5>
- 4) Listener in variable:**
- Tương tự như Activity Is listener, nhưng khác ở chỗ thay vì implement interface cho Activity thì nó lại được lưu trữ vào một biến có kiểu Listener trong activity. Làm cách này thì ta có thể chia sẻ chung một biến sự kiện cho các control khác nhau.
 - Ví dụ bạn xem hình bên dưới:

```

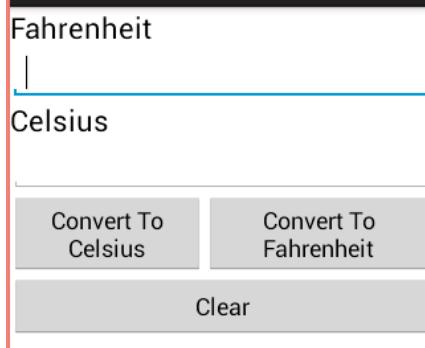
public class LoginExampleVariableImplementation extends Activity {
    ...
    OnClickListener myClickListener = new OnClickListener() {
        @Override
        public void onClick(View v) {
            if(v==btnLogin) {
                // Check Login
            } else if(v==btnCancel) {
                // Close the application
                finish();
            }
        }
    };
    ...
    /**
     * Called when the activity is first created.
     */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        ...
        // Set Click Listener
        btnLogin.setOnClickListener(myClickListener);
        btnCancel.setOnClickListener(myClickListener);
    }
}

```

- Bạn thấy đó, ở trên khai báo 1 biến có kiểu listener là interface **OnItemClickListener**.

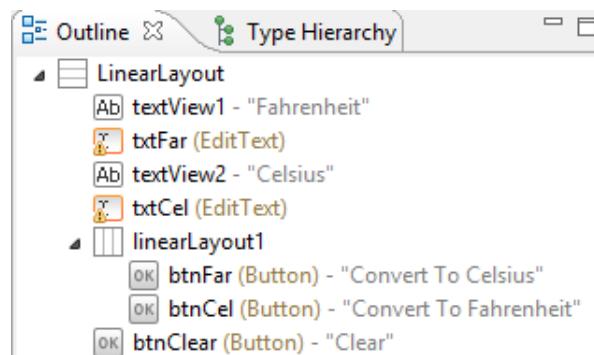
- Và biến này sẽ được chia sẻ cho 2 Button Login và Cancel.

- Nay tôi sẽ cung cấp ví dụ Chuyển đổi độ F qua C và ngược lại. Bạn xem giao diện bên dưới:



- Giao diện bên trên sẽ có 3 button. Ta sẽ tạo một biến sự kiện và chia sẻ cho 3 Button ở trên.

- Bạn xem Outline để thiết kế:



- Đây là công thức chuyển đổi:

- Công thức:

$$\begin{aligned} {}^{\circ}\text{C} \times 9/5 + 32 &= {}^{\circ}\text{F} \\ ({}^{\circ}\text{F} - 32) \times 5/9 &= {}^{\circ}\text{C} \end{aligned}$$

Ví dụ:

Chuyển từ 37°C tới F : $37^{\circ}\text{C} \times 9/5 + 32 = 98.6^{\circ}\text{F}$

Chuyển từ 98.6°F tới C : $(98.6^{\circ}\text{F} - 32) \times 5/9 = 37^{\circ}\text{C}$

-Bạn xem coding Activity:

```
import android.os.Bundle;

import android.app.Activity;

import android.view.View;

import android.view.View.OnClickListener;

import android.widget.Button;

import android.widget.EditText;

public class MainActivity extends Activity {

    private Button btnFar,btnCel,btnClear;

    private EditText txtFar,txtCel;

    private OnClickListener myVarListener=new OnClickListener() {

        @Override

        public void onClick(View arg0) {

            //Lấy F và C từ control ở đây

            if(arg0==btnFar)

            {

                //Bạn xử lý chuyển đổi F->C theo công thức

            }

            else if(arg0==btnCel)
```

```
{  
//Bạn xử lý chuyển đổi C->F theo công thức  
}  
  
else if(arg0==btnClear)  
  
{  
  
txtFar.setText("");  
  
txtCel.setText("");  
  
txtFar.requestFocus();  
  
}  
  
}  
  
};  
  
@Override  
  
protected void onCreate(Bundle savedInstanceState) {  
  
super.onCreate(savedInstanceState);  
  
setContentView(R.layout.activity_main);  
  
btnFar = (Button)findViewById(R.id.btnFar);  
  
btnCel = (Button)findViewById(R.id.btnCel);  
  
btnClear = (Button)findViewById(R.id.btnClear);  
  
txtFar = (EditText)findViewById(R.id.txtFar);  
  
txtCel = (EditText)findViewById(R.id.txtCel);  
  
btnFar.setOnClickListener(myVarListener);  
  
btnCel.setOnClickListener(myVarListener);  
  
btnClear.setOnClickListener(myVarListener);  
}
```

}

Bạn tự viết lệnh cho 2 nút chuyển đổi : cách lấy dữ liệu đã hướng dẫn ở những kiểu lập trình sự kiện trước, bắt buộc bạn phải lấy được. Sau khi lấy được thì chỉ cần ráp vào công thức là xong.

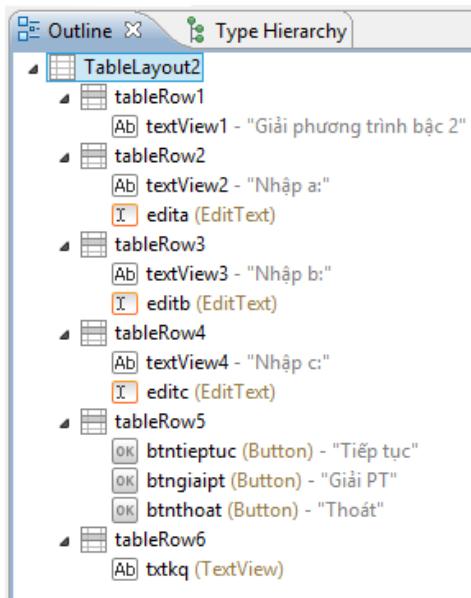
5) Explicit listener class:

- Trường hợp này ta tách riêng một class đóng vai trò là class sự kiện riêng.
- Khi nào lượng coding trong ứng dụng không lồ và phức tạp thì ta nên tách class sự kiện riêng để dễ quản lý.

Tôi ví dụ giải phương trình bậc 2, bạn xem giao diện bên dưới:



- Khi chọn “Tiếp tục”, chương trình sẽ xóa trống toàn bộ dữ liệu trên màn hình đồng thời focus tới ô Nhập a.
- Khi chọn “Giải PT”, chương trình sẽ tiến hành lấy thông số a,b,c và tiến hành giải phương trình bậc 2 và cho ra kết quả như hình trên.
- Khi chọn “Thoát”, chương trình sẽ được đóng lại.
- Bạn xem Outline dưới đây:



- Tiến hành coding, bạn mở Activity class và coding như bên dưới:

```

import java.text.DecimalFormat;

import android.os.Bundle;

import android.app.Activity;

import android.view.View;

import android.view.View.OnClickListener;

import android.widget.Button;

import android.widget.EditText;

import android.widget.TextView;

public class MainActivity extends Activity {

    Button btnTieptuc,btnGiai,btnThoat;

    EditText edita,editb,editc;

    TextView txtkq;

    @Override

    protected void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);
    }
}

```

```
setContentView(R.layout.activity_main);

btnTieptuc=(Button) findViewById(R.id.btn tieptuc);

btnGiai=(Button) findViewById(R.id.btn giai);

btnThoat=(Button) findViewById(R.id.btn thoat);

btnTieptuc.setOnClickListener(new MyEvent());

btnGiai.setOnClickListener(new MyEvent());

btnThoat.setOnClickListener(new MyEvent());

edita=(EditText) findViewById(R.id.edita);

editb=(EditText) findViewById(R.id.editb);

editc=(EditText) findViewById(R.id.editc);

txtkq=(TextView) findViewById(R.id.txtkq);

}

public void giaiPtb2()

{

String sa=edita.getText()+"';

String sb=editb.getText()+"';

String sc=editc.getText()+"';

int a=Integer.parseInt(sa);

int b=Integer.parseInt(sb);

int c=Integer.parseInt(sc);

String kq="";

DecimalFormat dcf=new DecimalFormat("#.00");

if(a==0)

{
```

```

if(b==0)

{
    if(c==0)

        kq="PT vô số nghiệm";

    else

        kq="PT vô nghiệm";

    }

else

{
    kq="Pt có 1 No, x=" + dcf.format(-c/b);

}
}

else

{
    double delta=b*b-4*a*c;

    if(delta<0)

    {
        kq="PT vô nghiệm";
    }

    else if(delta==0)

    {
        kq="Pt có No kép x1=x2=" + dcf.format(-b/(2*a));
    }

    else

```

```

{

kq="Pt có 2 No: x1='"+dcf.format((-b-Math.sqrt(delta))/(2*a))+

"; x2='"+dcf.format((-b-Math.sqrt(delta))/(2*a));

}

}

txtkq.setText(kq);

}

private class MyEvent implements OnClickListener

{

@Override

public void onClick(View arg0) {

if(arg0==btnTieptuc)

{

edita.setText("");

editb.setText("");

editc.setText("");

edita.requestFocus();

}

else if(arg0.getId()==R.id.btngiaipt)

{

giaiPtB2();

}

else if(arg0.getId()==R.id.btnthoat)

{

```

```
finish();
```

```
}
```

```
}
```

```
}
```

```
}
```

- Bạn quan sát coding ở bên trên. Tôi tạo một lớp sự kiện tên là **MyEvent**, control nào muốn được gán sự kiện chỉ cần gọi lệnh giống như Tôi gán cho Button Giải phương trình:

```
btnGiai.setOnClickListener(new MyEvent());
```

- Trong hàm **public void onClick(View arg0)** . Bạn để ý là lúc thì Tôi so sánh theo Object, lúc thì Tôi lại lấy Id ra để so sánh. Đây là cố ý của Tôi, Tôi muốn nói rằng các bạn có thể kiểm tra xem Button nào sẽ được chọn trên giao diện bằng cách so sánh đối tượng hoặc lấy Id ra để so sánh (tùy ý đồ lập trình của mỗi người)

- Ở đây Tôi có 1 lời khuyên cho các bạn là khi phải xử lý quá nhiều dòng lệnh (lệnh phức tạp) thì bạn nên viết thành từng hàm riêng, và trong hàm xử lý sự kiện bạn chỉ cần gọi tên hàm mà thôi. Cụ thể là đối với Button “**Giai PT**” thì Tôi lại viết và gọi riêng hàm **giaiPtB2()**, còn đối với Button Tiếp tục và Thoát thì Tôi lại không cần thiết viết hàm vì xử lý quá đơn giản.

6. View Subclassing

- Kỹ thuật này không được phổ biến cho lắm. Bạn chỉ sài khi thêm Control động (lúc runtime) vào màn hình. Ta có thể dùng bất kỳ kỹ thuật nào (6 cách Tôi vừa nêu) để thêm sự kiện động cho một Button động.

- Ở cách cuối cùng này thì bạn phải override phương trình **performClick** của chính Button control:

```
public class FsdActivity extends Activity {
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);

        Button But9= new Button(this) {
            @Override
            public boolean performClick() {
                //your code here
                return false;
            }
        };
        But9.setText("But9");
        LinearLayout layout = (LinearLayout)findViewById(R.id.lll);
        layout.addView(But9);
    }
}
```

- Như vậy các bạn đã được thực hành về các kiểu lập trình sự kiện trong Android, biết được cách lấy dữ liệu từ EditText, biết xử lý định dạng dữ liệu, cũng có thể thêm được Layout.

- Trong các bài tập sắp tới các bạn sẽ được thực hành về Toast & Alert Dialog, và rất nhiều các control cơ bản cũng như nâng cao trong Android .

- Bạn phải hiểu rõ các kỹ thuật lập trình này để tùy vào từng trường hợp cụ thể mà bạn nên quyết định kỹ thuật nào cho phù hợp.

Bài tập 9: Thực hành về Toast Notification và Alert Dialog

Vì trong tất cả các ứng dụng Tôi thấy rằng chúng ta hay sử dụng Toast và Alert Dialog để kiểm tra một điều gì đó, hay đơn giản chỉ là xuất thông báo. Vì vậy bài tập này Tôi sẽ hướng dẫn các bạn làm quen với Toast & Alert Dialog, sau đó các bạn sẽ được thực hành với các control cơ bản và nâng cao.

- Cả Toast và Alert Dialog khi hiển thị lên thì các tiến trình (hay các lệnh) khác **vẫn cứ tiếp tục làm việc.**

1) Toast:

- Toast có thể được tạo và hiển thị trong Activity hoặc trong Service.
- Không cho phép người sử dụng tương tác
- Khi hiển thị sau khoảng thời gian nào đó sẽ tự đóng lại
- Có 2 giá trị mặc định (ta nên sử dụng 2 giá trị này, không nên gõ con số cụ thể vào): hằng số **Toast.LENGTH_SHORT** hiển thị trong **2** giây, **Toast.LENGTH_LONG** hiển thị trong **3.5** giây.

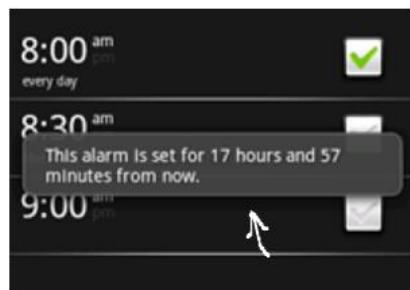
Cách tạo Toast:

```
Toast toast=Toast.makeText(YourActivity.this, "Hiển thị gì thì ghi ở đây", Toast.LENGTH_SHORT);  
toast.show();
```

- Khi nào bạn nên sử dụng Toast?

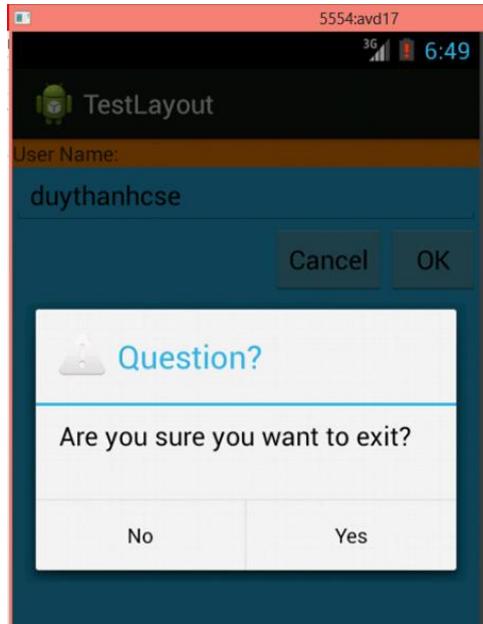
Theo Tôi thì tùy bạn, bạn có thể sử dụng trong trường hợp hiển thông báo trong các mục thiết lập thông số cấu hình, hay đơn giản chỉ là hiển thị lên để xem thông tin tạm thời nào đó (giống như để kiểm tra một vấn đề xảy ra chẳng hạn).

- Hình dưới đây cho bạn biết 1 Toast đang hiển thị:



2) Alert Dialog:

- Hiển thị và cho phép người dùng tương tác, ví dụ bạn nhìn hình Tôi chụp bên dưới, khi nhấn nút “Cancel”, chương trình sẽ hiển thị Alert Dialog hỏi xem có chắc chắn muốn xóa hay không? Bấm No thì không, bấm Yes thì tắt chương trình.



- Cách tạo Alert Dialog:

```
AlertDialog.Builder b=new AlertDialog.Builder(YourActivity.this);  
  
b.setTitle("Question");  
b.setMessage("Are you sure you want to exit?");  
b.setPositiveButton("Yes", new DialogInterface.OnClickListener() {  
    @Override  
    public void onClick(DialogInterface dialog, int which)  
    {  
        finish();  
    } });  
  
b.setNegativeButton("No", new DialogInterface.OnClickListener() {  
  
    @Override  
  
    public void onClick(DialogInterface dialog, int which)  
    {  
        dialog.cancel();  
    }  
});  
  
b.create().show();
```

Ý nghĩa của các hàm :

- **setTitle** : thiết lập tiêu đề cho Dialog
- **setMessage**: Thiết lập nội dung cho Dialog
- setIcon** : để thiết lập Icon
- setPositiveButton**, **setNegativeButton** thiết lập hiển thị Nút chọn cho Dialog (ở đây bạn không quan tâm Nút chấp nhận hay Nút hủy nó thuộc PositiveButton hay NegativeButton vì đó là tùy thuộc bạn chọn. Chú ý là ở đối số thứ 2 của các hàm này sẽ là **DialogInterface.OnClickListener** chứ không phải **View.OnClickListener**)
- **create()** để tạo Dialog
- **show()** để hiển thị Dialog.

Bài tập 10: Sử dụng TextView, EditText và Button trong Android

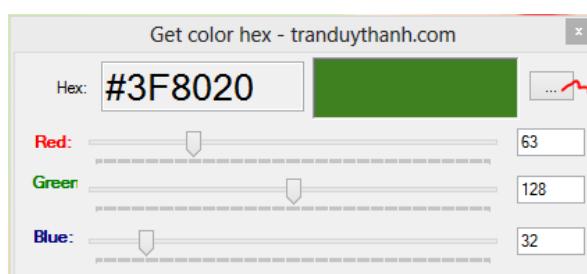
- TextView, EditText, Button là 3 control cơ bản nhất của Android. Trong các bài tập trước bạn đã được làm quen với 3 control này rồi. Bài tập này Tôi sẽ làm lại mục đích giúp các bạn ôn tập lại đồng thời giúp hiểu thêm được một số thuộc tính mới của nó (trong Android người ta thường gọi một số các control là **Form Widgets**).

1) TextView:

- Bạn chỉ muốn hiển thị thông tin mà không cho phép người dùng chỉnh sửa thì nên sử dụng control này.
- TextView tương tự như JLabel bên Java, và như Label bên C#
- Dưới này Tôi chụp một số thuộc tính của TextView mà chúng ta thường xuyên sử dụng nhất:

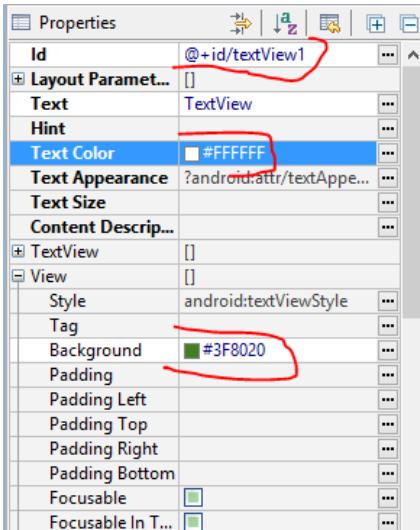
```
<TextView  
    android:id="@+id/textView1"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:background="#ff0000ff"  
    android:textStyle="bold"  
    android:textSize="25sp"  
    android:padding="5dp"  
    android:fontFamily="tahoma"  
    android:textColor="@android:color/holo_red_dark"  
    android:text="Nhập tên:" />
```

- Ta nên thiết lập **id** cho control để tiện bề xử lý.
- layout_width, layout_height nên thiết lập cho control(bắt buộc)
- Để thay đổi màu nền dùng background, thay đổi màu chữ dùng textColor...
- Để trợ giúp các bạn thay đổi màu nền và màu chữ của các control trong Android (dùng hex color), Tôi đã viết chương trình lấy màu theo mã hex color, các bạn có thể tải về tại : <http://www.mediafire.com/?ujj2pyppdwemx69>. Giao diện như sau:



Chương trình này bạn có thể chọn button “...” để chọn custom color. Ứng với mỗi bảng màu khác nhau thì sẽ có mã Hex color khác nhau. Bạn copy mã này dán vào Ứng dụng Android của bạn thì sẽ có màu như ý muốn.

Thí dụ, ở màn hình trên Hex là “#3F8020“, bạn copy hex này và dán vào background của TextView (xem hình):



- Như bên trên Tôi nói là nên đặt tên Id cho control, mục đích để xử lý một số công việc theo yêu cầu. Dựa vào Id ta sẽ lấy được control theo đúng Id này, xem code bên dưới để biết cách lấy control theo Id:

TextView txt1= (TextView) findViewById(R.id.textView1);

- Mọi control đều kế thừa từ View, và hàm **findViewById** cũng trả về 1 View theo đúng Id truyền vào, đó là lý do ta ép kiểu về cho đúng với TextView (cách làm nhanh: ngay dòng lệnh này nhấn tổ hợp phím **Ctrl +1** là nó sẽ tự ép kiểu nhanh cho bạn)

- Để hiển thị thông tin lên control TextView ta dùng lệnh dưới đây:

txt1.setText("Hello tèo");

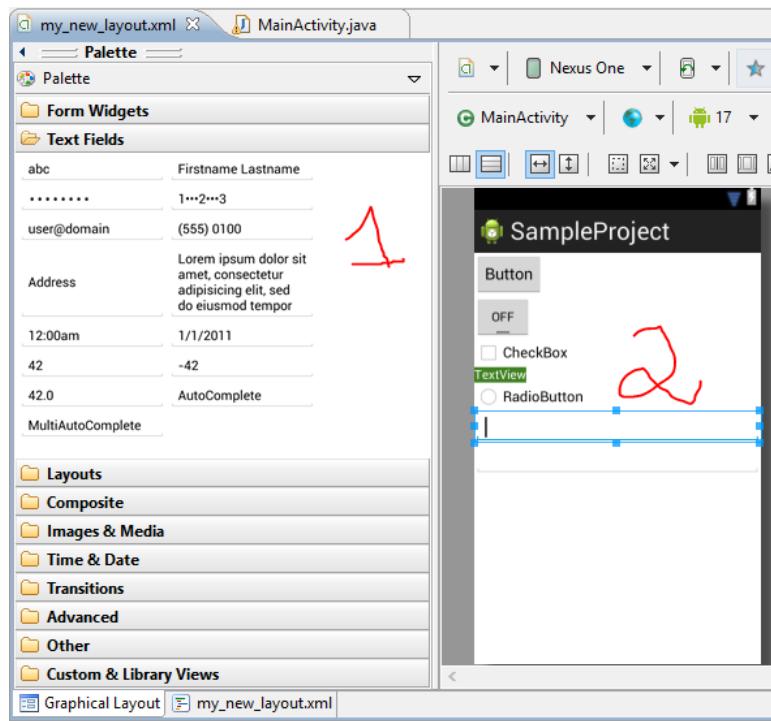
- Để lấy thông tin bên trong control TextView ta dùng lệnh dưới đây:

String msg=**txt1.getText().toString();**

2) EditText:

- Control này kế thừa từ TextView và cho phép chỉnh sửa dữ liệu (dù nhiên bạn có thể cầm chỉnh sửa dữ liệu bằng coding hay trong xml)

- Để sử dụng EditText rất đơn giản, bạn chỉ việc kéo thả control này vào giao diện và tiến hành thiết lập một số thuộc tính:



- Như hình bên trên thì bạn chỉ cần kéo loại EditText mà bạn cần (vùng **số 1**) rồi thả vào giao diện (vùng **số 2**)

- Bạn xem Tôi chụp một số thuộc tính của EditText trong hình dưới này:

```
<EditText
    android:id="@+id/editText2"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:inputType="text|textAutoCorrect|textCapWords"
    android:hint="Nhập tài khoản"
    android:textSize="18sp"
    android:ems="10" />
```

- Tương tự như TextView bạn cần thiết lập **Id**, các layout_width, layout_height

- Thuộc tính **hint** : để hiển thị thông tin gợi ý trong vùng nhập dữ liệu khi bạn chưa nhập bất kỳ dữ liệu nào vào, chỉ cần có dữ liệu là phần **hint** sẽ tự động mất đi.

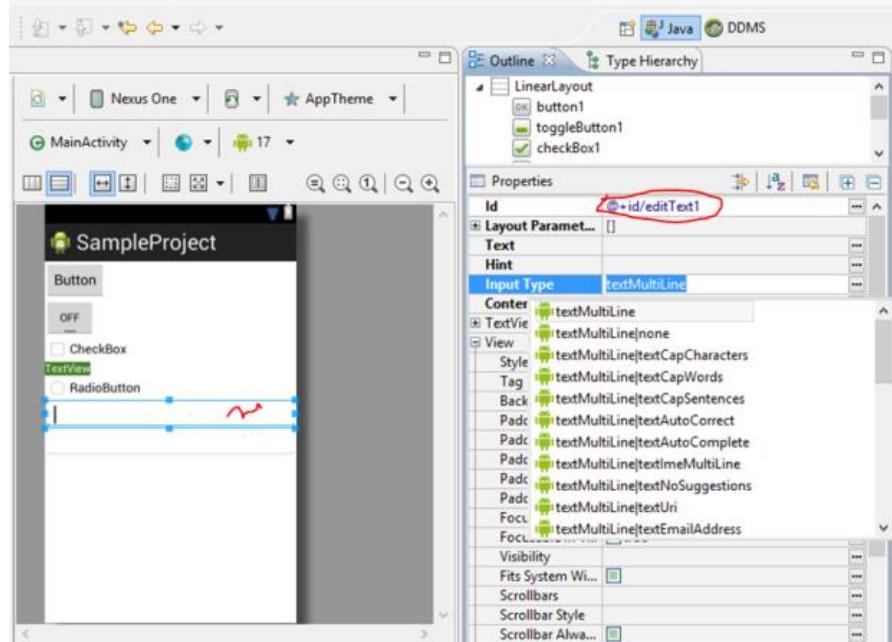
-textSize để thiết lập kích cỡ font chữ cho EditText

- Trong **inputType** bạn thấy Tôi kết hợp nhiều giá trị lại với nhau bằng cách dùng toán tử “|”, tức là EditText này sẽ có đầy đủ các đặc tính ở bên vẽ phải mà ta truyền vào, ví dụ:

+**textAutoCorrect** : Tự động sửa đúng chính tả, giả sử bạn nhập “**teh**” thì nó sẽ tự động sửa thành “**the**”

+ vân vân... bạn tự tìm hiểu thêm trên mạng

- Ta cũng có thể dùng cửa sổ Properties để thiết lập thuộc tính cho dễ dàng hơn (click chuột vào EditText muốn đổi thuộc tính):



- Màn hình trên cho phép ta thay đổi thuộc tính của control một cách dễ dàng.

- Tương tự như TextView, ta cũng phải lấy được control thông qua Id, thao tác với dữ liệu bên trong EditText:

+ Lấy control theo Id:

```
EditText txtbox=(EditText) findViewById(R.id.editText1);
```

+ Thiết lập giá trị cho EditText

```
txtBox.setText("nhập bất cứ cái gì vào đây xem sao")
```

+ Lấy dữ liệu bên trong EditText:

```
String msg=txtBox.getText().toString()
```

3) Button:

- Dùng để thiết lập sự kiện khi người dùng chọn lựa.

- Cũng kế thừa từ TextView

- Có 2 sự kiện mà người sử dụng thường xuyên thao tác:

```

Button btnok=(Button) findViewById(R.id.btnOk);
btnok.setOnClickListener(new View.OnClickListener() {
    public void onClick(View arg0) {
        //perform click here
    }
});
btnok.setOnLongClickListener(new View.OnLongClickListener() {
    public boolean onLongClick(View arg0) {
        //perform long click here
        return false;
    }
});

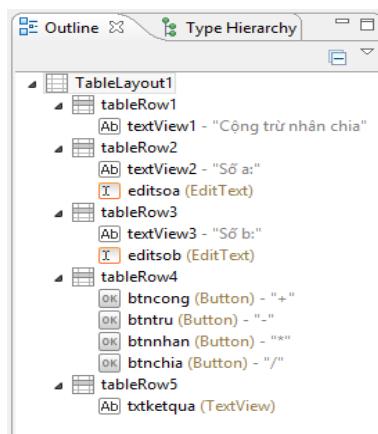
```

- Bây giờ Tôi sẽ làm một ví dụ về cách sử dụng 3 control này ([ban có thể xem lại bài tập các kiểu lập trình sự kiện trong Android](#)):

- Ví dụ đơn giản là tính cộng trừ nhân chia, giao diện như bên dưới (nhấn nút nào thì thực hiện phép toán cho nút đó):



- Bạn xem Layout để dễ thiết kế:



- Coding mẫu:

```

package tranduythanh.com;

import android.os.Bundle;

import android.app.Activity;

import android.view.View;

```

```
import android.view.View.OnClickListener;

import android.widget.Button;

import android.widget.EditText;

import android.widget.TextView;

public class MainActivity extends Activity {

    Button btncong,btntru,btnnhan,btnchia;

    EditText editsoa,editsob;

    TextView txtkq;

    OnClickListener myclick=new OnClickListener() {

        @Override

        public void onClick(View arg0) {

            switch(arg0.getId()) {

                case R.id.btncong:

                    String sa=editsoa.getText().toString();

                    String sb=editsob.getText().toString();

                    int a=Integer.parseInt(sa);

                    int b=Integer.parseInt(sb);

                    txtkq.setText(a+" + "+b+" = "+(a+b));

                    break;

                case R.id.btntru:

                    break;

                case R.id.btnnhan:

                    break;
            }
        }
    };
}
```

```

case R.id.btnchia:
break;

}

}

};

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity_main);

btncong=(Button) findViewById(R.id.btncong);

btntru=(Button) findViewById(R.id.btntru);

btnnhan=(Button) findViewById(R.id.btnnhan);

btnchia=(Button) findViewById(R.id.btnchia);

editsoa=(EditText) findViewById(R.id.editsoa);

editsob=(EditText) findViewById(R.id.editsob);

txtkq=(TextView) findViewById(R.id.txtketqua);

btncong.setOnClickListener(myclick);

btntru.setOnClickListener(myclick);

btnnhan.setOnClickListener(myclick);

btnchia.setOnClickListener(myclick);

}
}
}

```

Bạn hãy trả lời cho Tôi: Hiện giờ đoạn code trên Tôi sử dụng loại Sự Kiện nào?

- Như vậy bạn đã hiểu được cách sử dụng TextView, EditText, Button.

- Các bài tập sau Tôi sẽ nói về CheckBox, RadioButton.

- Bạn phải hiểu thật tốt về 3 Control Tôi vừa nêu ở trên, nó rất là cơ bản và quan trọng, đa phần ứng dụng Android nào cũng sử dụng tối thiểu 3 Control này.

Bài tập 11: Sử dụng Checkbox và RadioButton trong Android

Ở bài tập 10 bạn đã biết cách sử dụng TextView, EditText và Button. Trong bài tập này Tôi sẽ hướng dẫn các bạn cách sử dụng CheckBox và RadioButton. 2 Control này cũng rất thường xuyên được sử dụng trong các chương trình.

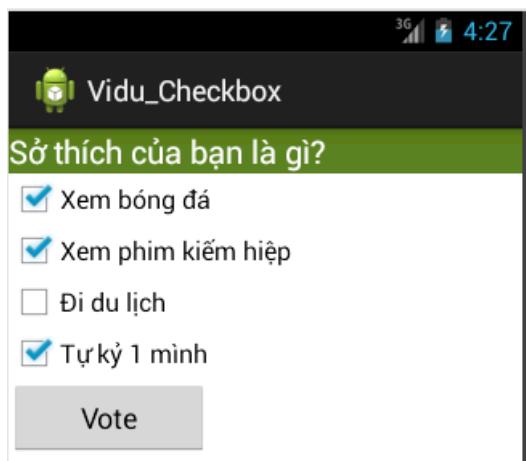
- CheckBox và RadioButton đều sử dụng chung 2 phương thức :

1) phương thức setChecked, dùng để thiết lập checked. Nếu ta gọi setChecked(true) tức là cho phép control được checked, còn gọi setChecked(false) thì control sẽ bị unchecked.

2) phương thức isChecked, kiểm tra xem control có được checked hay không. Nếu có checked thì trả về true ngược lại trả về false

- Checkbox cho phép ta checked nhiều đối tượng, còn RadioButton thì tại một thời điểm nó chỉ cho ta checked 1 đối tượng trong cùng một group mà thôi.

- Nếu bạn muốn người sử dụng có thể chọn nhiều lựa chọn thì bạn nên sử dụng Checkbox, ví dụ xem hình bên dưới:



- Ta có thẻ thiết lập cho Checkbox bất kỳ được checked mặc định trong XML:

```
<CheckBox  
    android:id="@+id/checkBox1"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:checked="true"  
    android:text="Xem bóng đá" />
```

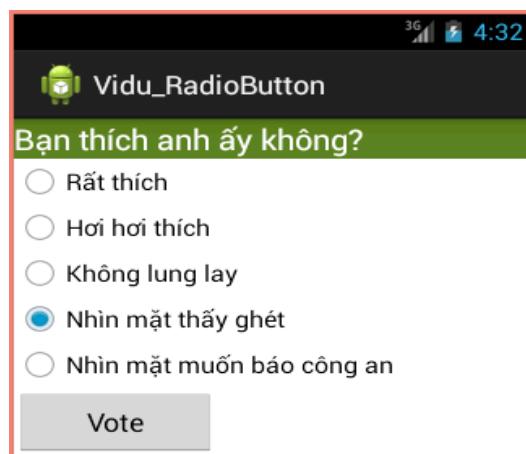
- Trong coding để kiểm tra xem Checkbox đó có được checked hay không thì làm như sau:

```

CheckBox chk=(CheckBox) findViewById(R.id.checkBox1);
if(chk.isChecked())
{
    //xử lý checked
}
else
{
    //xử lý unchecked
}
//Muốn thiết lập checked:
chk.setChecked(true);
//Muốn clear checked:
chk.setChecked(false);

```

- Nếu bạn muốn người sử dụng chỉ được chọn 1 lựa chọn trong nhiều chọn lựa bạn đưa ra thì nên sử dụng RadioButton, ví dụ xem hình bên dưới:



- Tương tự như Checkbox, ta cũng có thể thiết lập checked cho RadioButton bất kỳ trong XML:

```

<RadioGroup
    android:id="@+id/radioGroup1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content" >

    <RadioButton
        android:id="@+id/radio0"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Rất thích" />

    <RadioButton
        android:id="@+id/radio1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:checked="true" ✓
        android:text="Hơi hơi thích" />
</RadioGroup>

```

- Nhìn vào hình trên bạn thấy là ta phải sử dụng RadioGroup để gom nhóm các RadioButton lại cùng một nhóm nào đó, những RadioButton mà cùng một nhóm thì tại 1 thời điểm chỉ có 1 RadioButton được checked mà thôi. Trong một màn hình ta có thể tạo nhiều nhóm RadioGroup khác nhau.

- Tôi cung cấp 2 cách xử lý RadioButton nào được checked như sau:

Cách 1: Dựa vào RadioGroup để biết chính xác Id của RadioButton nào được checked. Dựa vào Id này ta sẽ xử lý đúng nghiệp vụ:

```
RadioGroup group=(RadioGroup) findViewById(R.id.radioGroup1);
int idChecked=group.getCheckedRadioButtonId();
switch(idChecked)
{
case R.id.radrathailong:
    break;
case R.id.radhailong:
    break;
case R.id.radtamchapnhan:
    break;
case R.id.radthayghe:
    break;
}
```

- Như hình trên, bạn thấy hàm **getCheckedRadioButtonId()** : hàm này trả về Id của RadioButton nằm trong RadioGroup 1 được checked. Dựa vào Id này bạn so sánh để biết được trên giao diện người sử dụng đang checked lựa chọn nào.

Cách 2: Kiểm tra trực tiếp RadioButton đó có được checked hay không?

```
RadioButton rad=(RadioButton) findViewById(R.id.radrathailong);
if(rad.isChecked())
{
}
```

Cả 2 cách trên đều có cùng chung một mục đích chỉ là kỹ thuật xử lý khác nhau, tương tự để xóa bỏ checked trong group, ta dùng lệnh:

group.clearChecked();

với **group** là đối tượng RadioGroup.

- Nay tôi sẽ demo một ví dụ kết hợp giữa RadioButton và CheckBox để bạn hiểu sâu hơn về 2 control này:

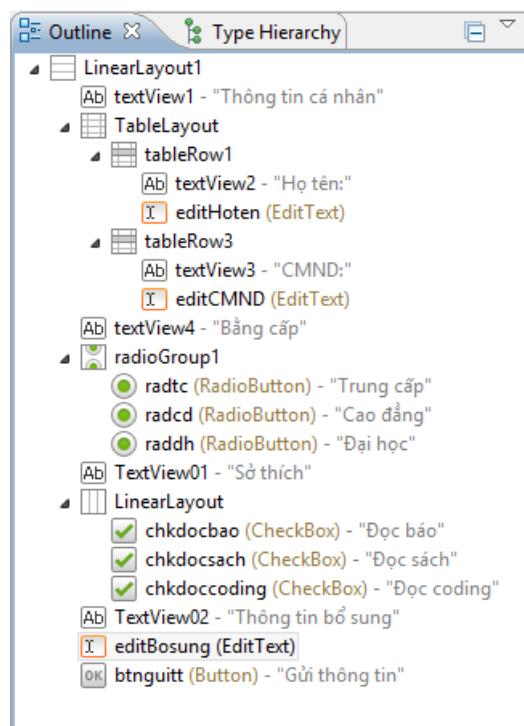


- Mô tả:

- Tên người không được để trống và phải có ít nhất 3 ký tự
- Chứng minh nhân dân chỉ được nhập kiểu số và phải có đúng 9 chữ số
- Bằng cấp mặc định sẽ chọn là Đại học
- Sở thích phải chọn ít nhất 1 chọn lựa
- Thông tin bổ sung có thể để trống
- Khi bấm gửi thông tin, chương trình sẽ hiển thị toàn bộ thông tin cá nhân cho người sử dụng biết (dùng Alert Dialog):



- Bạn xem Outline của MainActivity để dễ thiết kế:



Bạn xem coding trong MainActivity:

```
package tranduythanh.com;

import android.os.Bundle;

import android.app.Activity;

import android.app.AlertDialog;

import android.content.DialogInterface;

import android.view.Menu;

import android.view.View;

import android.widget.Button;

import android.widget.CheckBox;

import android.widget.EditText;

import android.widget.RadioButton;

import android.widget.RadioGroup;

import android.widget.Toast;

public class MainActivity extends Activity {

    EditText editTen,editCMND,editBosung;

    CheckBox chkdocbao,chkdocsach,chkdoccode;

    @Override

    protected void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity_main);

        editTen=(EditText) findViewById(R.id.editHoten);

        editCMND=(EditText) findViewById(R.id.editCMND);

        editBosung=(EditText) findViewById(R.id.editBosung);

        chkdocbao=(CheckBox) findViewById(R.id.chkdocbao);
```

```
chkdoccode=(CheckBox) findViewById(R.id.chkdoccoding);

chkdocsach=(CheckBox) findViewById(R.id.chkdocsach);

Button btn=(Button) findViewById(R.id.bnnguitt);

btn.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View arg0) {

// TODO Auto-generated method stub

doShowInformation();

}

});

}

public void doShowInformation()

{

//Kiểm tra tên hợp lệ

String ten=editTen.getText()+"';

ten=ten.trim();

if(ten.length()<3)

{

editTen.requestFocus();

editTen.selectAll();

Toast.makeText(this, "Tên phải >= 3 ký tự", Toast.LENGTH_LONG).show();

return;

}

//kiểm tra CMND hợp lệ
```

```

String cmnd=editCMND.getText()+"";
cmnd=cmnd.trim();

if(cmnd.length()!=9)

{
    editCMND.requestFocus();
    editCMND.selectAll();

    Toast.makeText(this, “CMND phải đúng 9 ký tự”, Toast.LENGTH_LONG).show();

return;
}

//Kiểm tra bằng cấp

String bang="";

RadioGroup group=(RadioGroup) findViewById(R.id.radioGroup1);

int id=group.getCheckedRadioButtonId();

if(id== -1)

{
    Toast.makeText(this, “Phải chọn bằng cấp”, Toast.LENGTH_LONG).show();

return;
}

RadioButton rad=(RadioButton) findViewById(id);

bang=rad.getText()+"";
//Kiểm tra sở thích

String sothich="";
if(chkdocbao.isChecked())
    sothich+=chkdocbao.getText()+"\n";

```

```

if(chkdocsach.isChecked())
sothich+=chkdocsach.getText()+"\n";
if(chkdoccode.isChecked())
sothich+=chkdoccode.getText()+"\n";
String bosung=editBosung.getText()+"";
AlertDialog.Builder builder=new AlertDialog.Builder(this);
builder.setTitle("Thông tin cá nhân");
builder.setPositiveButton("Đóng", new DialogInterface.OnClickListener() {
@Override
public void onClick(DialogInterface dialog, int which) {
// TODO Auto-generated method stub
dialog.cancel();
}
});
//tao noi dung
String msg=ten+"\n";
msg+= cmnd+"\n";
msg+=bang+"\n";
msg+=sothich;
msg+="—————\n";
msg+="Thông tin bổ sung:\n";
msg+=bosung+"\n";
msg+="—————";
builder.setMessage(msg); //thiet lap noi dung

```

```
builder.create().show(); //hiển thi Dialog  
}  
  
@Override  
  
public boolean onCreateOptionsMenu(Menu menu) {  
  
    // Inflate the menu; this adds items to the action bar if it is present.  
  
    getMenuInflater().inflate(R.menu.activity_main, menu);  
  
    return true;  
  
}  
  
}
```

chú ý các cách dùng sự kiện trong đoạn code ở trên.

Bạn có thể tải coding mẫu đầy đủ ở đây: <http://www.mediafire.com/?wqih1ln20rilsv8>

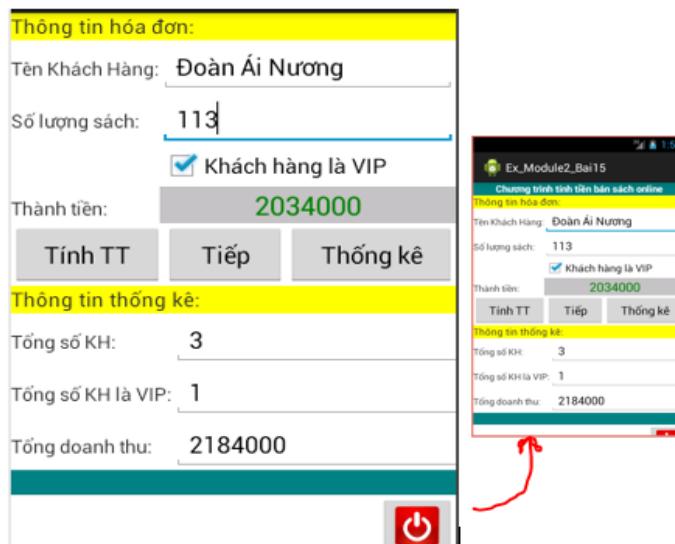
Tới đây bạn đã hiểu được phần nào CheckBox và RadioButton

- Bài tập kế tiếp Tôi sẽ làm một ví dụ tổng hợp về EditText, CheckBox, Button ... cùng ArrayList để bạn hiểu thêm về các control cơ bản này.

Bài tập 12: Ví dụ tổng hợp TextView, EditText, CheckBox, Button và ImageButton trong Android

Trong bài tập 10 và bài tập 11 bạn đã làm quen được với các control này. Trong bài tập 12 Tôi sẽ làm một ví dụ tổng hợp + kết hợp với ArrayList để bạn củng cố thêm kiến thức về các control đã học.

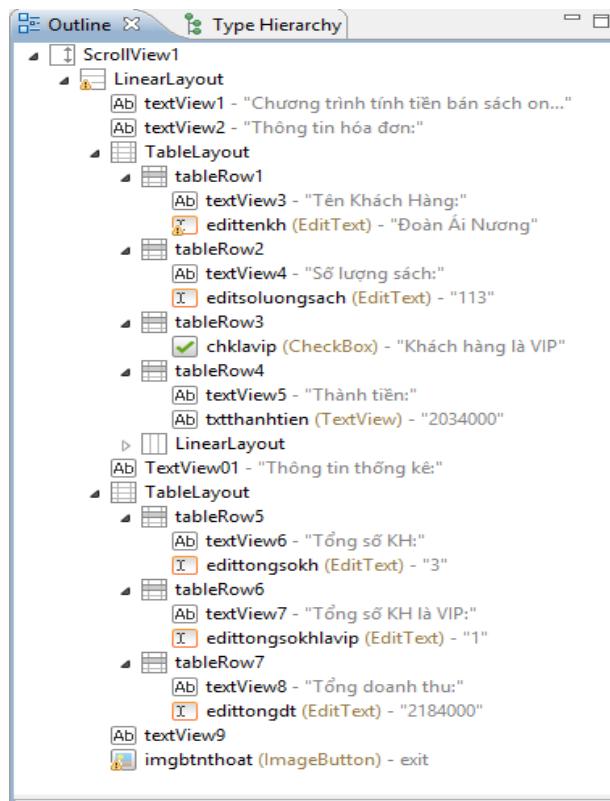
- Viết chương trình thanh toán tiền bán sách (Bạn đừng quan tâm là chương trình này nó có được ứng dụng thực tế hay không, hãy quan tâm cách viết code để hiểu thêm về các control):



- Mô tả yêu cầu:

- Khi bấm nút Tính Thành Tiền (Tính TT) chương trình sẽ tính thành tiền biết rằng mỗi cuốn sách có đơn giá là 20000, nếu là khách hàng VIP thì giảm 10%
- Khi bấm nút Tiếp, chương trình sẽ lưu thông tin hóa đơn vừa tính Thành Tiền vào danh sách, đồng thời xóa trống dữ liệu trong hóa đơn và cho focus tới EditText Tên khách hàng
- Khi bấm nút Thống kê, chương trình sẽ hiển thị thông tin vào mục Thông tin thống kê: tổng số KH, tổng số KH VIP và tổng doanh thu
- Khi bấm vào nút thoát (dùng ImageButton): hiển thị AlertDialog hỏi xem người sử dụng có chắc chắn muốn thoát hay không?
- Dùng ScrollView để chương trình có thể làm việc tốt hơn khi sử dụng các thiết bị có màn hình nhỏ.

- Bạn tham khảo Outline để cho dễ thiết kế:



- Vì có thể bạn sẽ “bực bội” khi nhìn thấy Outline ở trên vì vậy Tôi cung cấp luôn XML của Outline này, bạn chỉ việc copy paste vào nếu “làm biếng”:

```

1 <ScrollView
2     xmlns:android="http://schemas.android.com/apk/res/android"
3     xmlns:tools="http://schemas.android.com/tools"
4     android:id="@+id/ScrollView1"
5     android:layout_width="match_parent"
6     android:layout_height="match_parent"
7     tools:context=".MainActivity" >
8
9     <LinearLayout
10        android:layout_width="match_parent"
11        android:layout_height="match_parent"
12        android:orientation="vertical" >
13
14         <TextView
15             android:id="@+id/textView1"
16             android:layout_width="match_parent"
17             android:layout_height="wrap_content"
18             android:background="#008080"
19             android:gravity="center"
20             android:text="Chương trình tính tiền bán sách online"
21             android:textColor="#FFFFFF"
22             android:textSize="15sp"
23             android:textStyle="bold" />
24
25         <TextView
26             android:id="@+id/textView2"
27             android:layout_width="match_parent"
28             android:layout_height="wrap_content"
29             android:background="#FFFF00" />
30
31         <TableLayout
32             android:id="@+id/tableLayout1"
33             android:layout_width="match_parent"
34             android:layout_height="wrap_content" >
35
36             <TableRow
37                 android:id="@+id/tableRow1"
38                 android:layout_width="match_parent"
39                 android:layout_height="wrap_content" >
40
41                 <TextView
42                     android:id="@+id/textView3"
43                     android:layout_width="wrap_content"
44                     android:layout_height="wrap_content" />
45                 <EditText
46                     android:id="@+id/editText1"
47                     android:layout_width="wrap_content"
48                     android:layout_height="wrap_content" />
49             </TableRow>
50
51             <TableRow
52                 android:id="@+id/tableRow2"
53                 android:layout_width="match_parent"
54                 android:layout_height="wrap_content" >
55
56                 <TextView
57                     android:id="@+id/textView4"
58                     android:layout_width="wrap_content"
59                     android:layout_height="wrap_content" />
60                 <EditText
61                     android:id="@+id/editText2"
62                     android:layout_width="wrap_content"
63                     android:layout_height="wrap_content" />
64             </TableRow>
65
66             <TableRow
67                 android:id="@+id/tableRow3"
68                 android:layout_width="match_parent"
69                 android:layout_height="wrap_content" >
70
71                 <TextView
72                     android:id="@+id/textView5"
73                     android:layout_width="wrap_content"
74                     android:layout_height="wrap_content" />
75                 <TextView
76                     android:id="@+id/textView6"
77                     android:layout_width="wrap_content"
78                     android:layout_height="wrap_content" />
79             </TableRow>
80
81             <TableRow
82                 android:id="@+id/tableRow4"
83                 android:layout_width="match_parent"
84                 android:layout_height="wrap_content" >
85
86                 <TextView
87                     android:id="@+id/textView7"
88                     android:layout_width="wrap_content"
89                     android:layout_height="wrap_content" />
90                 <EditText
91                     android:id="@+id/editText3"
92                     android:layout_width="wrap_content"
93                     android:layout_height="wrap_content" />
94             </TableRow>
95
96             <TableRow
97                 android:id="@+id/tableRow5"
98                 android:layout_width="match_parent"
99                 android:layout_height="wrap_content" >
100
101                 <TextView
102                     android:id="@+id/textView8"
103                     android:layout_width="wrap_content"
104                     android:layout_height="wrap_content" />
105                 <EditText
106                     android:id="@+id/editText4"
107                     android:layout_width="wrap_content"
108                     android:layout_height="wrap_content" />
109             </TableRow>
110
111         <LinearLayout
112             android:layout_width="match_parent"
113             android:layout_height="wrap_content" >
114
115             <TextView
116                 android:id="@+id/textView9"
117                 android:layout_width="wrap_content"
118                 android:layout_height="wrap_content" />
119             <ImageButton
120                 android:id="@+id/imgBtnThoat"
121                 android:layout_width="wrap_content"
122                 android:layout_height="wrap_content" />
123         </LinearLayout>
124     </TableLayout>
125 
```

```
29     android:text="Thông tin hóa đơn:"  
30     android:textSize="15sp" />  
31  
32 <TableLayout  
33     android:layout_width="match_parent"  
34     android:layout_height="wrap_content"  
35     android:stretchColumns="1" >  
36  
37 <TableRow  
38     android:id="@+id/tableRow1"  
39     android:layout_width="wrap_content"  
40     android:layout_height="wrap_content" >  
41  
42 <TextView  
43     android:id="@+id/textView3"  
44     android:layout_width="wrap_content"  
45     android:layout_height="wrap_content"  
46     android:text="Tên Khách Hàng:" />  
47  
48 <EditText  
49     android:id="@+id/edittenkh"  
50     android:layout_width="wrap_content"  
51     android:layout_height="wrap_content"  
52     android:ems="10"  
53     android:text="Đoàn Ái Nương" >  
54  
55 <requestFocus />  
56 </EditText>  
57 </TableRow>  
58  
59 <TableRow  
60     android:id="@+id/tableRow2"  
61     android:layout_width="wrap_content"  
62     android:layout_height="wrap_content" >  
63  
64 <TextView  
65     android:id="@+id/textView4"  
66     android:layout_width="wrap_content"  
67     android:layout_height="wrap_content"  
68     android:text="Số lượng sách:" />  
69  
70 <EditText  
71     android:id="@+id/editluongsach"  
72     android:layout_width="wrap_content"  
73     android:layout_height="wrap_content"  
74     android:ems="10"  
75     android:inputType="number"  
76     android:text="113" />  
77 </TableRow>  
78  
79 <TableRow  
80     android:id="@+id/tableRow3"  
81     android:layout_width="wrap_content"  
82     android:layout_height="wrap_content" >
```

```
83 <CheckBox
84     android:id="@+id/chklavip"
85     android:layout_width="wrap_content"
86     android:layout_height="wrap_content"
87     android:layout_column="1"
88     android:checked="true"
89     android:text="Khách hàng là VIP" />
90 </TableRow>
91
92 <TableRow
93     android:id="@+id/tableRow4"
94     android:layout_width="wrap_content"
95     android:layout_height="wrap_content" >
96
97 <TextView
98     android:id="@+id/textView5"
99     android:layout_width="wrap_content"
100    android:layout_height="wrap_content"
101    android:text="Thành tiền:" />
102
103 <TextView
104     android:id="@+id/txtthanhtien"
105    android:layout_width="wrap_content"
106    android:layout_height="wrap_content"
107    android:background="#C0C0C0"
108    android:gravity="center"
109    android:text="2034000"
110    android:textColor="#008000"
111    android:textSize="20sp" />
112 </TableRow>
113
114 <LinearLayout
115     android:layout_width="wrap_content"
116     android:layout_height="wrap_content" >
117
118 <Button
119     android:id="@+id/btntinhht"
120     android:layout_width="wrap_content"
121     android:layout_height="wrap_content"
122     android:layout_weight="1"
123     android:text="Tính TT" />
124
125 <Button
126     android:id="@+id/btntiep"
127     android:layout_width="wrap_content"
128     android:layout_height="wrap_content"
129     android:layout_weight="1"
130     android:text="Tiếp" />
131
132 <Button
133     android:id="@+id/btnthongke"
134     android:layout_width="wrap_content"
135     android:layout_height="wrap_content"
136     android:layout_weight="1"
137     android:text="Thống kê" />
```

```
137    </LinearLayout>
138    </TableLayout>
139
140    <TextView
141        android:id="@+id/TextView01"
142        android:layout_width="match_parent"
143        android:layout_height="wrap_content"
144        android:background="#FFFF00"
145        android:text="Thông tin thống kê:"
146        android:textSize="15sp" />
147
148    <TableLayout
149        android:layout_width="match_parent"
150        android:layout_height="wrap_content"
151        android:stretchColumns="1" >
152
153        <TableRow
154            android:id="@+id/tableRow5"
155            android:layout_width="wrap_content"
156            android:layout_height="wrap_content" >
157
158            <TextView
159                android:id="@+id/textView6"
160                android:layout_width="wrap_content"
161                android:layout_height="wrap_content"
162                android:text="Tổng số KH:" />
163
164            <EditText
165                android:id="@+id/edittongsokh"
166                android:layout_width="wrap_content"
167                android:layout_height="wrap_content"
168                android:ems="10"
169                android:inputType="number"
170                android:text="3" />
171        </TableRow>
172
173        <TableRow
174            android:id="@+id/tableRow6"
175            android:layout_width="wrap_content"
176            android:layout_height="wrap_content" >
177
178            <TextView
179                android:id="@+id/textView7"
180                android:layout_width="wrap_content"
181                android:layout_height="wrap_content"
182                android:text="Tổng số KH là VIP:" />
183
184            <EditText
185                android:id="@+id/edittongsokhvip"
186                android:layout_width="wrap_content"
187                android:layout_height="wrap_content"
188                android:ems="10"
189                android:inputType="number"
190                android:text="1" />
191        </TableRow>
```

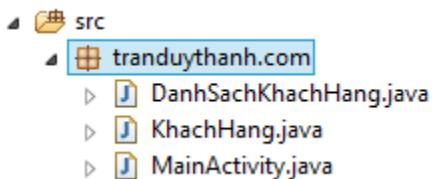
```

191
192 <TableRow
193     android:id="@+id/tableRow7"
194     android:layout_width="wrap_content"
195     android:layout_height="wrap_content" >
196
197     <TextView
198         android:id="@+id/textView8"
199         android:layout_width="wrap_content"
200         android:layout_height="wrap_content"
201         android:text="Tổng doanh thu:" />
202
203     <EditText
204         android:id="@+id/edittongdt"
205         android:layout_width="wrap_content"
206         android:layout_height="wrap_content"
207         android:ems="10"
208         android:inputType="numberDecimal"
209         android:text="2184000" />
210     </TableRow>
211 </TableLayout>
212
213 <TextView
214     android:id="@+id/textView9"
215     android:layout_width="match_parent"
216     android:layout_height="wrap_content"
217     android:background="#008080" />
218
219 <ImageButton
220     android:id="@+id/imgbtnt thoat"
221     android:layout_width="wrap_content"
222     android:layout_height="wrap_content"
223     android:layout_gravity="right"
224     android:src="@drawable/exit" />
225 </LinearLayout>
226
227 </ScrollView>
228
229
230

```

Bạn cần chú ý là ở cuối XML có ImageButton, nhìn vào thuộc tính **android:src="“@drawable/exit”**.
Ở đây bạn tạo một hình có tên exit.png rồi kéo vào thư mục drawable (bạn có thể tự tạo 1 thư mục mới tên là drawable vào trong ứng dụng của bạn).

- Bạn xem cấu trúc tập tin src của ứng dụng này:



-Ở trên có MainActivity.java (là màn hình chính của ứng dụng mà bạn thấy ở phần giới thiệu)

- class KhachHang dùng để lưu thông tin của khách hàng: Tên khách hàng, số lượng mua, thành tiền, là VIP hay không

- class DanhSachKhachHang dùng để lưu trữ các khách hàng mua sách, đồng thời cung cấp một số hàm như: tính tổng tiền, tính tổng số khách hàng, tính tổng số khách hàng VIP....

Dưới đây là chi tiết cho từng class:

class KhachHang:

```
1  package tranduythanh.com;
2
3  public class KhachHang {
4
5      private String tenKh;
6
7      private int slmua;
8
9      private boolean isVip;
10
11     public static final int GIA=20000;
12
13     public KhachHang()
14     {
15
16     }
17
18
19     public KhachHang(String tenKh, int slmua, boolean isVip)
20
21     {
22
23         this.tenKh=tenKh;
24
25         this.slmua=slmua;
26
27         this.isVip=isVip;
28
29     }
30
31     public String getTenKh() {
32
33         return tenKh;
34
35     }
36
37     public void setTenKh(String tenKh) {
38
39         this.tenKh = tenKh;
40
41     }
```

```

42
43     public int getSlmua() {
44
45         return slmua;
46
47     }
48
49     public void setSlmua(int slmua) {
50
51         this.slmua = slmua;
52
53     }
54
55     public boolean isVip() {
56
57         return isVip;
58
59     }
60
61     public void setVip(boolean isVip) {
62
63         this.isVip = isVip;
64
65     }
66
67     public double tinhThanhTien()
68
69     {
70
71         return (!isVip?slmua*GIA:slmua*GIA*0.9);
72
73     }
74
75 }
```

Class DanhSachKhachHang:

```

1   package tranduythanh.com;
2
3   import java.util.ArrayList;
4
5   public class DanhSachKhachHang {
6
7       ArrayList<KhachHang>listKH=new ArrayList<KhachHang>();
8
9       public void addKhachHang(KhachHang kh)
10
11    {
12
13        listKH.add(kh);
14
15    }
16
17    public double tongDoanhThu()
```

```

18
19     {
20
21     double tien=0.0;
22
23     for (KhachHang kh:listKH)
24     {
25
26         tien+=kh.tinhThanhTien();
27
28     }
29
30     return tien;
31
32 }
33
34
35     public int tongKhachHang()
36     {
37
38         return listKH.size();
39
40     }
41
42
43     public int tongKhachHangVip()
44     {
45
46         int s=0;
47
48         for (KhachHang kh:listKH)
49         {
50
51             if(kh.isVip())
52                 s++;
53
54         }
55
56         return s;
57
58     }
59
60 }
61
62 }
63

```

class MainActivity:

- Dùng để triệu gọi 2 class trên và thực thi các nghiệp vụ:

```

1     package tranduythanh.com;
2

```

```
3 import android.os.Bundle;
4
5 import android.app.Activity;
6
7 import android.app.AlertDialog;
8
9 import android.content.DialogInterface;
10
11 import android.view.View;
12
13 import android.view.View.OnClickListener;
14
15 import android.widget.Button;
16
17 import android.widget.CheckBox;
18
19 import android.widget.EditText;
20
21 import android.widget.ImageButton;
22
23 import android.widget.TextView;
24
25 public class MainActivity extends Activity {
26
27     Button btnTT,btnTiep,btnTK;
28
29     ImageButton btnThoat;
30
31     EditText editTen,editSl,editTongKh,editTongKhVip,ediTongTT;
32
33     TextView txtTT;
34
35     CheckBox chkVip;
36
37     DanhSachKhachHang danhsach=new DanhSachKhachHang();
38
39     @Override
40
41     protected void onCreate(Bundle savedInstanceState) {
42
43         super.onCreate(savedInstanceState);
44
45         setContentView(R.layout.activity_main);
46
47         getControls();
48
49         addEvents();
50
51     }
52
53     private void getControls()
54     {
55
56         btnTT=(Button) findViewById(R.id.btntintht);
```

```

57
58     btnTiep=(Button) findViewById(R.id.btntiep);
59
60     btnTK=(Button) findViewById(R.id.btnthongke);
61
62     btnThoat=(ImageButton) findViewById(R.id.imgbtnthoat);
63
64     editTen=(EditText) findViewById(R.id.edittenkh);
65
66     editSl=(EditText) findViewById(R.id.editsoluongsach);
67
68     editTongKh=(EditText) findViewById(R.id.edittongsokh);
69
70     editTongKhVip=(EditText)
71     findViewById(R.id.edittongsokhlavip);
72
73     ediTongTT=(EditText) findViewById(R.id.edittongdt);
74
75     txtTT=(TextView) findViewById(R.id.txtthanhtien);
76
77     chkVip =(CheckBox) findViewById(R.id.chklavip);
78
79 }
80
81     private void addEvents()
82 {
83
84     btnTT.setOnClickListener(new ProcessMyEvent());
85
86     btnTiep.setOnClickListener(new ProcessMyEvent());
87
88     btnTK.setOnClickListener(new ProcessMyEvent());
89
90     btnThoat.setOnClickListener(new ProcessMyEvent());
91
92 }
93
94     private void doTinhTien()
95 {
96
97
98     KhachHang kh=new KhachHang();
99
100    kh.setTenKh(editTen.getText()+"");
101
102    kh.setSlmu(Integer.parseInt(editSl.getText()+""));
103
104    kh.setVip(chkVip.isChecked());
105
106    txtTT.setText(kh.tinhThanhTien()+"");
107
108    danhsach.addKhachHang(kh);
109
110 }

```

```

111
112     private void doTiep()
113     {
114
115         editTen.setText("");
116
117         editSl.setText("");
118
119         txtTT.setText("");
120
121         editTen.requestFocus();
122
123     }
124
125
126     private void doThongKe()
127     {
128
129         editTongKh.setText(danh sach.tongKhachHang() + "");
130
131         editTongKhVip.setText(danh sach.tongKhachHangVip() + "");
132
133         ediTongTT.setText(danh sach.tongDoanhThu() + "");
134
135     }
136
137
138     private void doThoat()
139     {
140
141         AlertDialog.Builder builder=new AlertDialog.Builder(this);
142
143         builder.setTitle("hỏi thoát chương trình");
144
145         builder.setMessage("Muốn thoát chương trình này hả?");
146
147         builder.setNegativeButton("Không", new
148             DialogInterface.OnClickListener() {
149
150             @Override
151
152             public void onClick(DialogInterface dialog, int which) {
153
154                 dialog.cancel();
155
156             }
157
158         });
159
160         builder.setPositiveButton("Có", new
161             DialogInterface.OnClickListener() {
162
163             @Override
164

```

```
165     public void onClick(DialogInterface dialog, int which) {
166         finish();
167     }
168 }
169 }
170 }
171 }
172 builder.create().show();
173 }
174 }
175 }
176
177 private class ProcessMyEvent implements OnClickListener
178 {
179     @Override
180     public void onClick(View arg0) {
181         switch(arg0.getId())
182         {
183             case R.id.btntinhht:
184                 doTinhTien();
185                 break;
186             case R.id.btntiep:
187                 doTiep();
188                 break;
189             case R.id.btnthongke:
190                 doThongKe();
191                 break;
192             case R.id.imgbtnthoat:
193                 doThoat();
194                 break;
195         }
196     }
197 }
198 }
199 }
200 }
201 }
202 }
```

- Trong MainActivity Tôi có tình tách các nghiệp vụ ra thành các hàm riêng biệt như vậy để bạn dễ dàng xử lý. Chúng ta nên tập viết như vậy để Coding được sạch sẽ, khi có lỗi xảy ra cũng giúp các bạn dễ Fixed bug, sai hàm nào thì sửa hàm đó mà nó không bị ảnh hưởng tới các nghiệp vụ khác.

- Bạn có thể load đầy đủ coding mẫu trong link này: <http://www.mediafire.com/?lgq5g2l28xb222c>

- Bạn phải hiểu cách làm bài tập này để ứng dụng cho các phần sau.

- Hãy thực hành bài này nhiều lần cho tới khi thực sự hiểu logic của nó.

- Bài tập tiếp theo bạn sẽ được học về các control nâng cao trong Android, và ta sẽ thực hành thật kỹ từng control cụ thể, đầu tiên là ListView (control rất thường xuyên được sử dụng trong một chương trình Android nào đó)

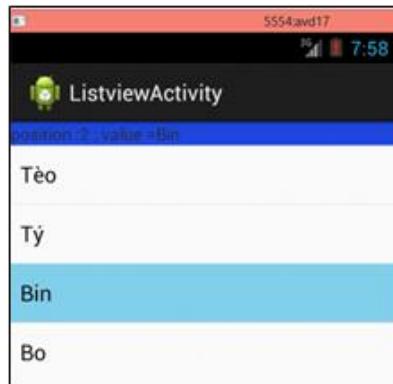
Bài tập 13: Thực hành về ListView trong Android

Trong các bài tập trước các bạn đã được làm quen với nhiều control cơ bản, bài tập này bạn sẽ được làm quen với control nâng cao, cụ thể là ListView. Trong ứng dụng cần lưu trữ và hiển thị danh sách các thông tin đa phần chúng ta sài control ListView. Hiện tại bạn chỉ cần biết sử dụng ListView có sẵn của Android là được rồi, trong các bài tập tiếp theo Tôi sẽ hướng dẫn các bạn Custom Layout lại ListView (tự làm mới ListView theo ý mình).

- Bài tập này Tôi sẽ cung cấp nhiều cách hành xử với ListView, ứng với mỗi cách là có các ví dụ mẫu khác nhau, vì vậy các bạn nên cố gắng theo dõi và thực hành lại những ví dụ.
- Bạn hãy thực hành tốt **5 trường hợp** Tôi trình bày dưới đây:

1) Trường hợp 1: Sử dụng ListView control với mảng dữ liệu định sẵn.

- Trường hợp này Tôi đưa ra một ví dụ đơn giản là cho phép hiển thị mảng dữ liệu lên trên ListView, bạn xem hình minh họa:

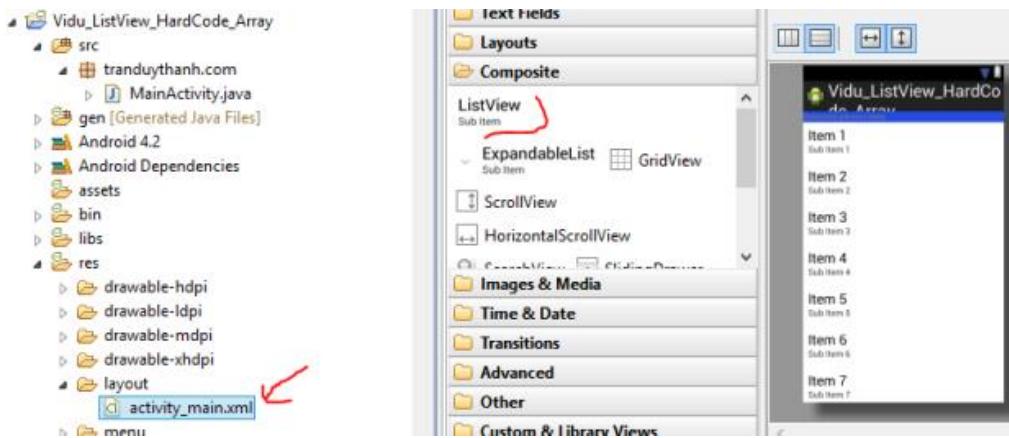


- Giao diện trên có 2 control:

+**ListView** : dùng để hiển thị mảng dữ liệu

+**TextView** có màu xanh lục: Dùng để hiển thị vị trí và giá trị của phần tử được chọn trong ListView

- Bạn tạo một Android Project tên là : **Vidu.ListView_HardCode_Array**, chọn layout phù hợp và kéo thả các control vào giao diện:



- Dưới đây là nội dung của **activity_main.xml**:

```

1 <LinearLayout
2   xmlns:android="http://schemas.android.com/apk/res/android"
3   xmlns:tools="http://schemas.android.com/tools"
4   android:id="@+id/LinearLayout1"
5   android:layout_width="match_parent"
6   android:layout_height="match_parent"
7   android:orientation="vertical"
8   tools:context=".MainActivity" >
9   <TextView
10  android:id="@+id/txtselection"
11  android:layout_width="match_parent"
12  android:layout_height="wrap_content"
13  android:background="#dd0230dd"
14  android:hint="Selected person here" />
15  <ListView
16  android:id="@+id/lvperson"
17  android:layout_width="match_parent"
18  android:layout_height="wrap_content" >
19  </ListView>
</LinearLayout>
```

-Đặt id cho Listview là **lvperson** (**nhìn dòng lệnh 15 ở trên**), bạn có thể định dạng thêm một số đặc tính khác nhưng trong bài tập này thì chưa cần thiết, chỉ cần hiển thị được dữ liệu lên giao diện là đã đạt yêu cầu.

- Bây giờ bạn mở **MainActivity.java** lên để viết code:

```

1 package tranduythanh.com;
2
3 import android.os.Bundle;
4 import android.app.Activity;
5 import android.view.View;
6 import android.widget.AdapterView;
7 import android.widget.ArrayAdapter;
8 import android.widget.ListView;
9 import android.widget.TextView;
10
11 public class MainActivity extends Activity {
```

```

13     protected void onCreate(Bundle savedInstanceState) {
14         super.onCreate(savedInstanceState);
15         setContentView(R.layout.activity_main);
16         //1. Khởi tạo dữ liệu cho mảng arr (còn gọi là data source)
17         final String arr[]={ "Teo", "Ty", "Bin", "Bo"};
18         //2. Lấy đối tượng Listview dựa vào id
19         ListView lv=(ListView) findViewById(R.id.lvperson);
20         //3. Gán Data source vào ArrayAdapter
21         ArrayAdapter<String>adapter=new ArrayAdapter<String>
22             (this, android.R.layout.simple_list_item_1, arr);
23         //4. Đưa Data source vào ListView
24         lv.setAdapter(adapter);
25         final TextView txt=(TextView) findViewById(R.id.txtselection);
26         //5. Thiết lập sự kiện cho Listview, khi chọn phần tử nào thì
27         hiển thị lên TextView
28         lv.setOnItemClickListener(
29             new AdapterView.OnItemClickListener() {
30                 public void onItemClick(AdapterView<?> arg0,
31                     View arg1,
32                     int arg2,
33                     long arg3) {
34                     //đối số arg2 là vị trí phần tử trong Data Source (arr)
35                     txt.setText("position :" +arg2+ " ; value =" +arr[arg2]);
36                 }
37             });
38         }

```

- Tôi đã giải thích từng dòng lệnh ở bên trong code, giờ Tôi giải thích thêm về ArrayAdapter, bạn nhìn vào **dòng lệnh 21**.

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

- Dữ liệu từ Data source (arr) sẽ được gắn vào ArrayAdapter, ArrayAdapter sẽ gắn vào ListView.

- Bạn nhìn vào đối số đầu tiên của constructor ArrayAdapter : **this**, chính là context của Activity hiện tại, bạn có thể viết **MainActivity.this** (nếu bạn viết như thế này thì ở bất kỳ vị trí nào nó cũng hiểu là context của MainActivity, do đó các bạn nên viết như thế này để bạn có thể copy paste nó tới bất kỳ vị trí nào thì nó cũng hiểu)

- Đối số thứ 2 **android.R.layout.simple_list_item_1** : bạn để ý android Tôi tô màu xanh, đây chính là layout Listview mà được Android xây dựng sẵn, các bài tập kế tiếp ta sẽ tự xây dựng mà không sử dụng cái có sẵn này. Như vậy thì **simple_list_item_1** lưu ở đâu? và bên trong nó như thế nào?. Nó được lưu trong **SDK/platforms/android-api (x)/data/res/layout/simple_list_item_1.xml**. Bạn có thể xem nội dung và vị trí của layout này một cách nhanh chóng bằng đè phím Ctrl + click chuột vào dòng lệnh này, bạn sẽ thấy như bên dưới:

```

<?xml version="1.0" encoding="utf-8"?>
<!-- Copyright (C) 2006 The Android Open Source Project
     Licensed under the Apache License, Version 2.0 (the "License");
     you may not use this file except in compliance with the License.
     You may obtain a copy of the License at
     http://www.apache.org/licenses/LICENSE-2.0
     Unless required by applicable law or agreed to in writing, software
     distributed under the License is distributed on an "AS IS" BASIS,
     WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
     See the License for the specific language governing permissions and
     limitations under the License.
-->
<TextView xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/text1"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:textAppearance="?android:attr/textAppearanceListItemSmall"
    android:gravity="center_vertical"
    android:paddingStart="?android:attr/listPreferredItemPaddingStart"
    android:paddingEnd="?android:attr/listPreferredItemPaddingEnd"
    android:minHeight="?android:attr/listPreferredItemHeightSmall"/>

```

- Đối số thứ 3: chính là arr (data source), bạn có thể truyền vào ArrayList.

- Nhìn vào dòng lệnh 27 chõ gán sự kiện cho ListView (bạn nhớ là chỉ cần gõ một vài ký tự đầu rồi nhấn Ctrl+ Space Bar thì các lệnh đằng sau sẽ tự động xuất hiện ra cho bạn):

+ Ta có interface **AdapterView.OnItemClickListener**, nó dùng để thiết lập sự kiện cho ListView, interface này có 1 phương thức trừu tượng là **onItemClick** nên ta override nó về xử lý trong này. Bạn cũng nhớ là chõ này không có gõ bằng tay mà chỉ cần nhấn tổ hợp phím **Ctrl + 1** chọn add unimplement method là nó tự xuất hiện. Ngoài ra nó còn nhiều sự kiện khác các bạn tự tìm hiểu thêm.

Bạn có thể tải code đầy đủ ở đây: <http://www.mediafire.com/?uwy4lp0e1jt0mik>

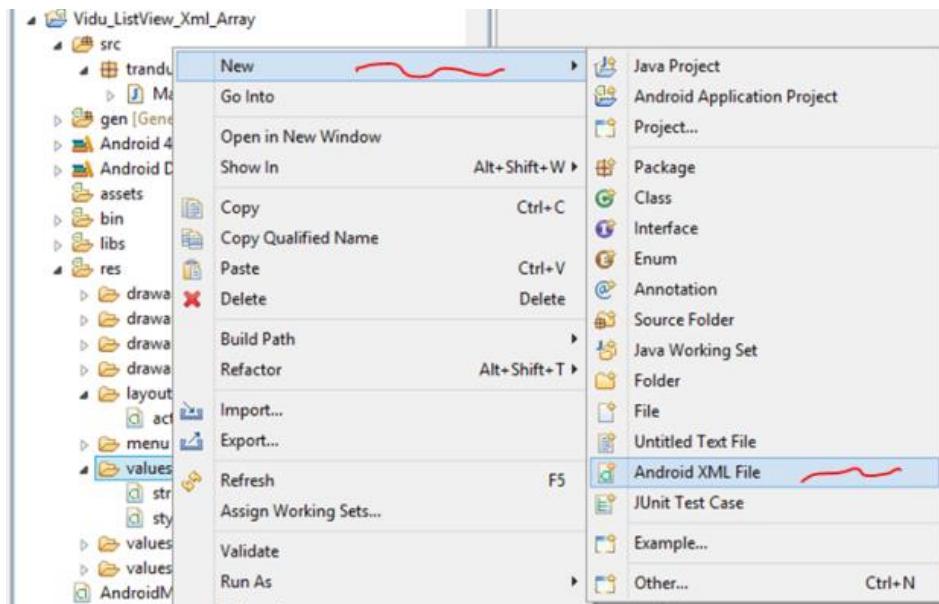
2) Trường hợp 2: Sử dụng ListView với mảng dữ liệu được lưu trong Xml:

- Giao diện và xử lý sự kiện trong trường hợp này là y xì trường hợp 1. Chỉ khác ở chõ là dữ liệu sẽ được load từ XML, nên Tôi chỉ hướng dẫn **cách tạo String – Array** trong XML và **cách load String-Array** trong coding như thế nào.

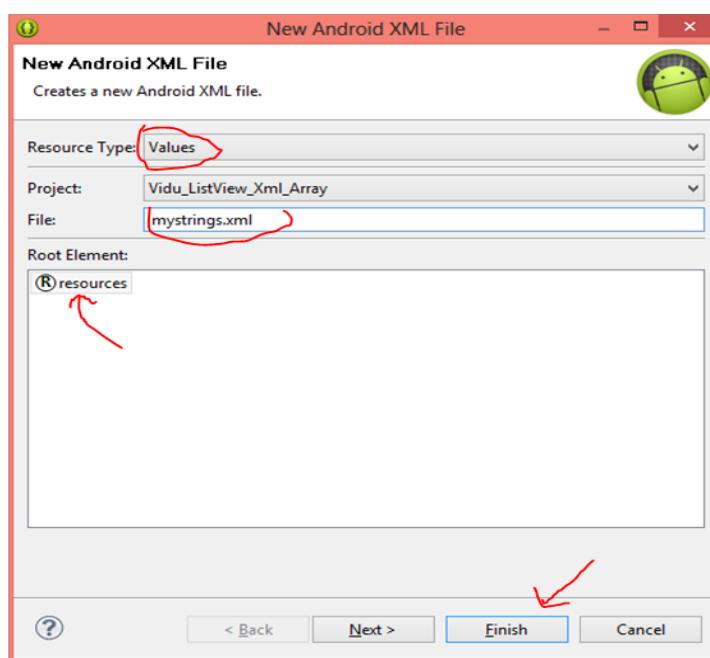
- Bạn tạo một Android Project tên là: **Vidu_ListView_Xml_Array**

- **Để tạo String – Array trong XML** bạn làm như sau:

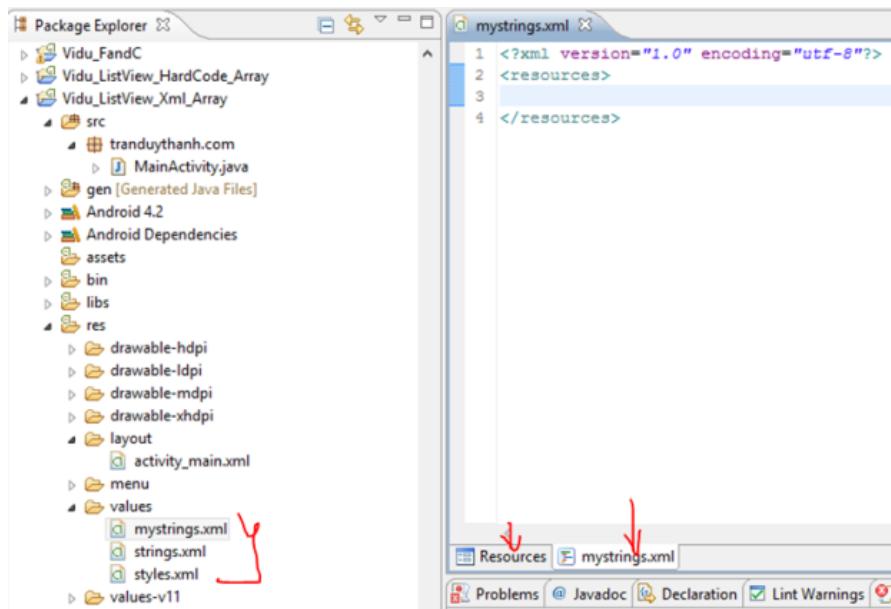
Bước 1: Bấm chuột phải vào thư mục **values** của Project/ chọn New/Android XML File:



- **Bước 2:** Màn hình New Android XML hiển thị lên, bạn chọn thông số giống như hình bên dưới, đặt tên tập tin là **mystrings.xml** rồi nhấn nút **Finish**:

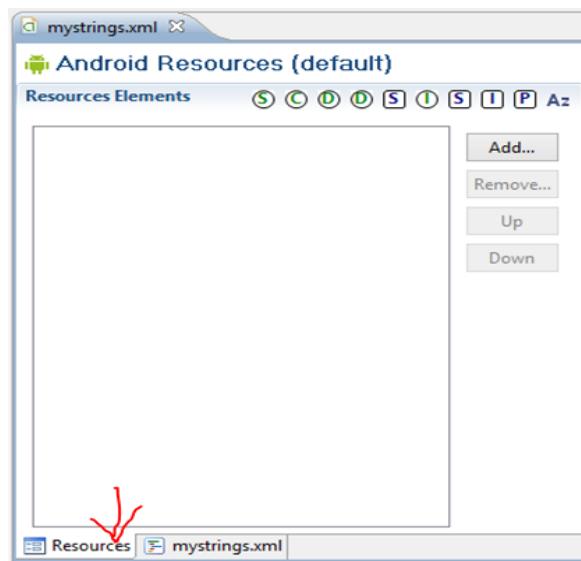


Xem kết quả khi bấm nút **Finish** và quan sát cho nhận xét:

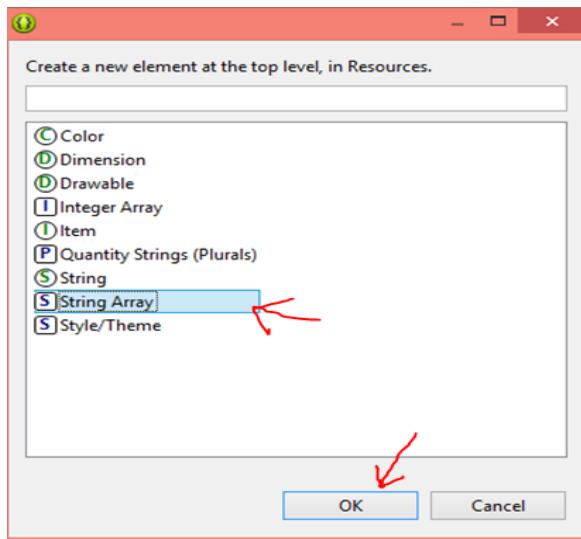


Chú ý là ta cũng có thể thêm dữ liệu vào tập tin **strings.xml**, nhưng đây là ý đồ của Tôi, Tôi muốn hướng dẫn các bạn cách tạo 1 tập tin XML mới và thêm dữ liệu vào tập tin mới này luôn.

Bước 3: chọn tab Resource ở hình trên:

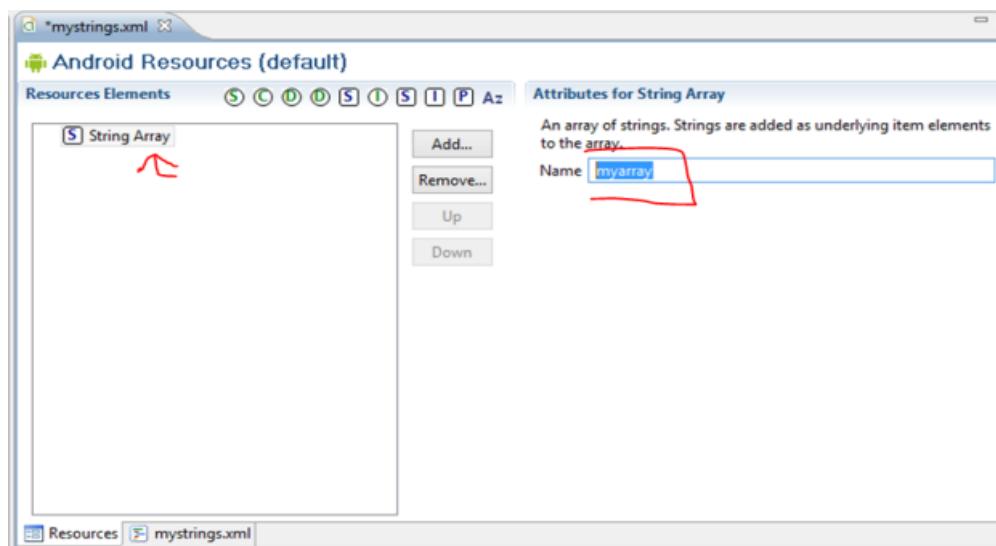


- **Bước 4:** Bấm nút “Add...” ở màn hình trên để thêm **String-Array**:

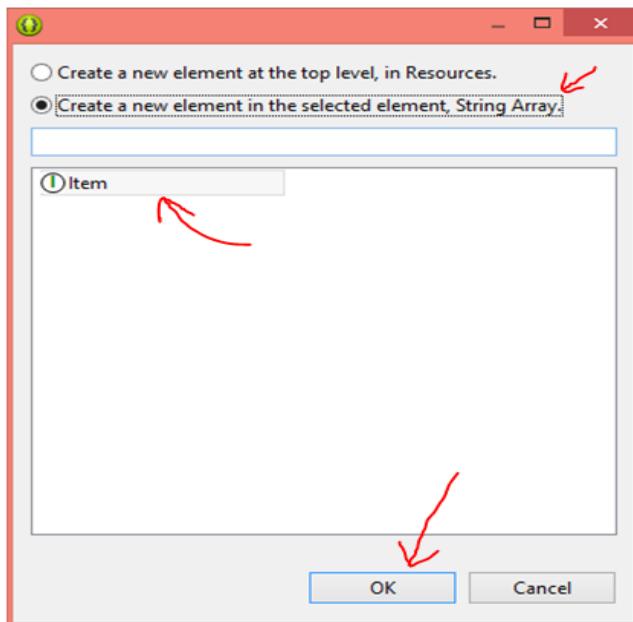


Ở màn hình trên bạn chọn String Array rồi nhấn OK:

- **Bước 5:** Đặt tên cho String Array, sau khi nhấn OK thì màn hình bên dưới hiển thị ra, bạn đặt tên String Array này là **myarray** rồi nhấn **ctrl+s** để lưu:



- **Bước 6:** Thêm các phần tử vào String Array, tiếp tục bấm nút **Add** ở màn hình bên trên:



- Sau khi nhấn OK thì nó sẽ cho phép bạn nhập giá trị cho phần tử.

- Bạn cứ lặp liên tục thao tác ở bước 6 này, thêm bao nhiêu phần tử thì click từng đó lần Add, ở đây là Tôi thêm 3 phần tử, bạn xem hình:



- Xem nội dung XML:

```

1 <?xml version="1.0" encoding="utf-8"?>
2 <resources>
3   <string-array name="myarray">
4     <item>Trần Văn Tèo</item>
5     <item>Nguyễn Thị Tết</item>
6     <item>Hồ Văn Hiến</item>
7   </string-array>
8 </resources>

```

- Như vậy bạn đã biết cách tạo 1 tập tin XML và biết cách tạo String Array cũng như tạo các phần tử nằm bên trong nó, Bây giờ trong Coding ta sẽ dựa vào myarray để đọc toàn bộ dữ liệu từ XML ra. Bạn cũng chú ý là myarray khi bạn tạo ra ở trên thì nó cũng được tạo ra trong gen:

```

1 */* AUTO-GENERATED FILE. DO NOT MODIFY. */
2
3 package tranduythanh.com;
4
5 public final class R {
6     public static final class array {
7         public static final int myarray=0x7f040000;
8     }
9     public static final class attr {
10 }
11 }
12 }
13 }
14 }
15 }

```

- Như bạn thấy đó, trong coding ta sẽ dựa vào đây để lấy ra: **R.array.myarray**.

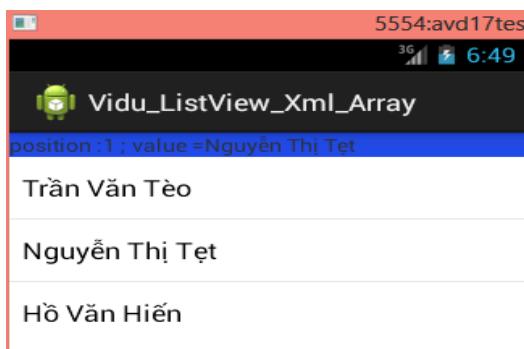
* Mở MainActivity.java lên:

- Như Tôi đã nói là mọi thứ nó y xì như trường hợp số 1, dó đó trong trường hợp này Tôi chỉ viết dòng lệnh đọc dữ liệu từ XML đổ về một Mảng (thay vì trường hợp 1 Tôi hardcoded mấy phần tử khi khai báo mảng), còn các phần khác bạn tự làm giống trường hợp 1:

```
final String arr[]=getResources().getStringArray(R.array.myarray);
```

Tức là bạn thay thế dòng lệnh thứ 17 trong trường hợp 1 của MainActivity.java bằng dòng lệnh trên.

Thực thi ứng dụng bạn sẽ được kết quả như hình bên dưới:

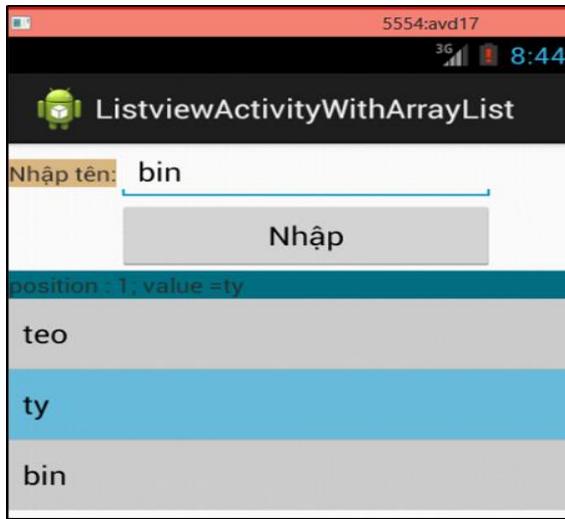


- Tôi nghĩ tới đây bạn đã thật sự hiểu trường hợp 2.

- Bạn có thể tải toàn bộ coding mẫu ở đây: <http://www.mediafire.com/?168jsno9iq85qsc>

3) Trường hợp 3: Sử dụng ArrayList và Listview control:

- Trường hợp này Tôi muốn hướng dẫn các bạn cách sử dụng ArrayList để lưu trữ dữ liệu và đổ lên ListView như thế nào, bạn xem giao diện của chương trình:



- Mô tả:

- + Nhập dữ liệu và nhấn nút “Nhập” thì sẽ đưa vào ArrayList và hiển thị lên ListView
 - + Nhấn vào phần tử nào thì hiển thị vị trí và giá trị của phần tử đó lên TextView
 - + Nhấn thật lâu (long click) vào phần tử nào đó trên ListView thì sẽ xóa phần tử đó.
- * Tạo Android Project tên: **Vidu_ListView_ArrayList**,

Xem Layout XML của ứng dụng (**activity_main.xml**):

```

1   <RelativeLayout
2     xmlns:android="http://schemas.android.com/apk/res/android"
3       xmlns:tools="http://schemas.android.com/tools"
4       android:layout_width="match_parent"
5       android:layout_height="match_parent"
6       tools:context=".MainActivity" >
7       <EditText
8         android:id="@+id/txtTen"
9         android:layout_width="wrap_content"
10        android:layout_height="wrap_content"
11        android:layout_alignParentTop="true"
12        android:inputType="text"
13        android:layout_toRightOf="@+id/textView1"
14        android:ems="10" />
15       <TextView
16         android:id="@+id/textView1"
17         android:layout_width="wrap_content"
18         android:layout_height="wrap_content"
19         android:layout_alignBaseline="@+id/txtTen"
20         android:layout_alignBottom="@+id/txtTen"
21         android:layout_alignParentLeft="true"
22         android:background="#deb887"
23         android:text="Nhập tên:" />
24       <Button
25         android:id="@+id/btnNhap"
26         android:layout_width="wrap_content"
```

```

26    android:layout_height="wrap_content"
27    android:layout_alignRight="@+id/txtTen"
28    android:layout_below="@+id/txtTen"
29    android:layout_toRightOf="@+id/textView1"
30    android:textAlignment="center"
31    android:text="Nhập" />
32
33 <TextView
34     android:id="@+id/txtselection"
35     android:layout_width="match_parent"
36     android:layout_height="wrap_content"
37     android:layout_alignParentLeft="true"
38     android:layout_below="@+id/btnNhap"
39     android:background="#007380" />
40
41 <ListView
42     android:id="@+id/lvperson"
43     android:layout_width="match_parent"
44     android:layout_height="wrap_content"
45     android:layout_alignParentLeft="true"
46     android:layout_below="@+id/txtselection"
47     android:background="#cccccc" >
48 </ListView>
49 </RelativeLayout>

```

Xem MainActivity.java:

```

1  package tranduythanh.com;
2
3  import java.util.ArrayList;
4  import android.os.Bundle;
5  import android.app.Activity;
6  import android.view.View;
7  import android.widget.AdapterView;
8  import android.widget.ArrayAdapter;
9  import android.widget.Button;
10 import android.widget.EditText;
11 import android.widget.ListView;
12 import android.widget.TextView;
13
14 public class MainActivity extends Activity {
15     EditText txtten;
16     TextView txtchon;
17     Button btn;
18     ListView lv;
19     ArrayList<String>arrList=null;
20     ArrayAdapter<String> adapter=null;
21     @Override
22     protected void onCreate(Bundle savedInstanceState) {
23         super.onCreate(savedInstanceState);
24         setContentView(R.layout.activity_main);
25
26         txtten=(EditText) findViewById(R.id.txtTen);
27         txtchon=(TextView) findViewById(R.id.txtselection);

```

```

28
29     lv=(ListView) findViewById(R.id.lvperson);
30     //1. Tạo ArrayList object
31     arrList=new ArrayList<String>();
32     //2. Gán Data Source (ArrayList object) vào ArrayAdapter
33     adapter=new ArrayAdapter<String>
34     (this,
35     android.R.layout.simple_list_item_1,
36     arrList);
37     //3. gán Adapter vào ListView
38     lv.setAdapter(adapter);
39
40     btn=(Button) findViewById(R.id.btnNhap);
41     //4. Xử lý sự kiện nhấn nút Nhập
42     btn.setOnClickListener(new View.OnClickListener() {
43         public void onClick(View arg0) {
44             arrList.add(txtten.getText()+"");
45             adapter.notifyDataSetChanged();
46         }
47     });
48     //5. Xử lý sự kiện chọn một phần tử trong ListView
49     lv.setOnItemClickListener(new AdapterView
50         .OnItemClickListener() {
51         public void onItemClick(
52             AdapterView<?> arg0,View arg1,
53             int arg2,long arg3) {
54
55             txtchon.setText("position : "+ arg2+
56             "; value ="+arrList.get(arg2));
57         }
58     });
59     //6. xử lý sự kiện Long click
60     lv.setOnItemLongClickListener(new AdapterView
61         .OnItemLongClickListener() {
62         @Override
63         public boolean onItemLongClick(AdapterView<?> arg0, View
64             arg1,
65             int arg2, long arg3) {
66             arrList.remove(arg2);//xóa phần tử thứ arg2
67             adapter.notifyDataSetChanged();
68             return false;
69         }
70     });
71 }

```

Tôi giải thích thêm về coding:

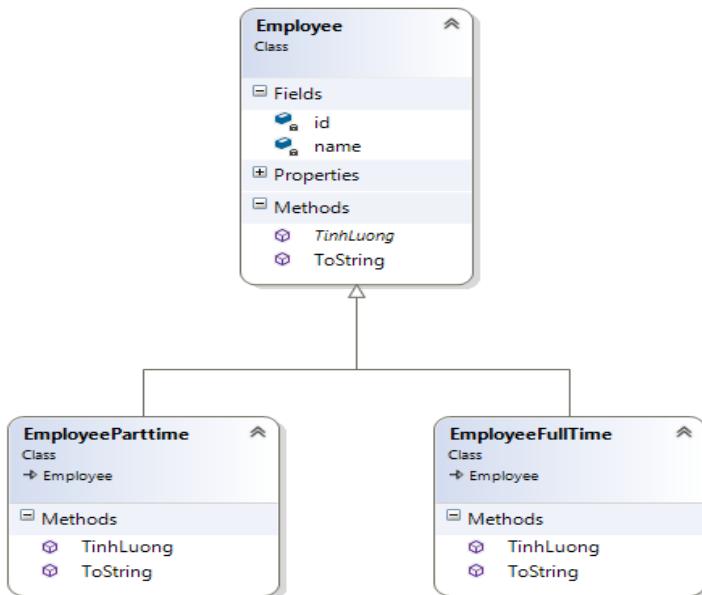
ArrayList bạn đã được học trong môn Java 1 rồi. Ở đây Tôi nói hàm **adapter.notifyDataSetChanged()**; Bạn chú ý là ArrayList được gán vào adapter nên mọi sự thay đổi trong ArrayList thì adapter đều nhận biết được. Khi có sự thay đổi trong ArrayList bạn chỉ cần gọi **notifyDataSetChanged** thì ListView sẽ được cập nhật (bởi vì Adapter được gắn vào ListView).

- Sự kiện **setOnItemLongClickListener**, được gắn cho ListView Item, khi nhấn lâu từ 2.5 tới 3 giây thì sự kiện này sẽ xảy ra. Tương tự như **setOnItemClickListener**, đối số có tên **arg2** được dùng để xác định được vị trí của phần tử nằm trong ArrayList.

- Bạn có thể vào đây để tải coding mẫu: <http://www.mediafire.com/?64k77e8yiazar8i>

4) Trường hợp 4: Sử dụng ArrayList và ListView nhưng từng phần tử trong ArrayList là các Object bất kỳ:

- Tôi có một ví dụ về hiển thị danh sách nhân viên theo mô hình sau:



- Có 2 loại nhân viên : Nhân viên chính thức (EmployeeFullTime) và nhân viên thời vụ (EmployeeParttime).

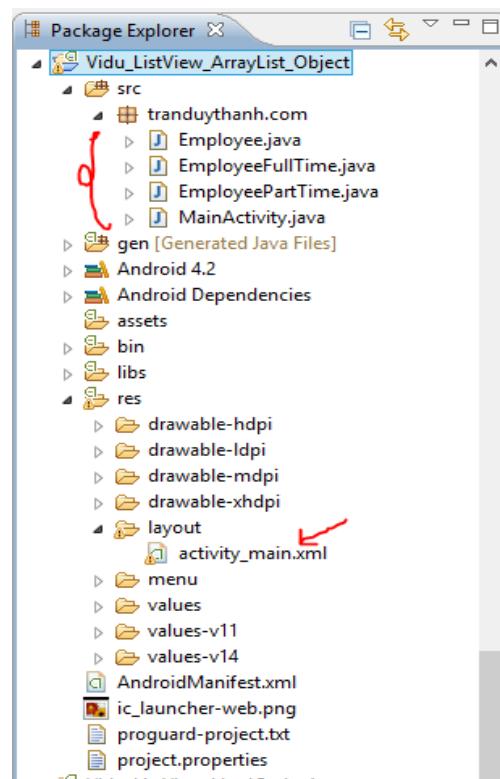
- Mỗi nhân viên sẽ có cách tính lương khác nhau (tên phương thức tính lương giống nhau)

- Mỗi nhân viên có phương thức `toString` để xuất thông tin, Nội dung xuất khác nhau. Thêm FullTime đằng sau Id và Name đối với nhân viên chính thức. Thêm Partime đằng sau Id và Name đối với nhân viên thời vụ.

- Xem giao diện chương trình:



-Tạo một Android Project tên: **Vidu_ListView_ArrayList_Object**, cấu trúc nhu bên dưới:



- Layout XML (**activity_main.xml**):

```

1 <LinearLayout
2     xmlns:android="http://schemas.android.com/apk/res/android"
3     xmlns:tools="http://schemas.android.com/tools"
4     android:id="@+id/LinearLayout1"
5     android:layout_width="match_parent"
6     android:layout_height="match_parent"
7     android:orientation="vertical"
8     tools:context=".MainActivity" >
9
10    <TextView

```

```
11     android:id="@+id/textView1"
12     android:layout_width="match_parent"
13     android:layout_height="wrap_content"
14     android:background="#008000"
15     android:gravity="center"
16     android:text="Quản lý nhân viên"
17     android:textColor="#FFFFFF"
18     android:textSize="20sp" />
19
20 <TableLayout
21     android:layout_width="match_parent"
22     android:stretchColumns="*"
23     android:layout_height="wrap_content" >
24
25 <TableRow
26     android:id="@+id/tableRow1"
27     android:layout_width="wrap_content"
28     android:layout_height="wrap_content" >
29
30 <TextView
31     android:id="@+id/textView2"
32     android:layout_width="wrap_content"
33     android:layout_height="wrap_content"
34     android:text="Mã NV:" />
35
36 <EditText
37     android:id="@+id/editMa"
38     android:layout_width="wrap_content"
39     android:layout_height="wrap_content"
40     android:layout_span="2"
41     android:ems="10" >
42
43     <requestFocus />
44 </EditText>
45
46 </TableRow>
47
48 <TableRow
49     android:id="@+id/tableRow2"
50     android:layout_width="wrap_content"
51     android:layout_height="wrap_content" >
52
53 <TextView
54     android:id="@+id/textView3"
55     android:layout_width="wrap_content"
56     android:layout_height="wrap_content"
57     android:text="Tên NV:" />
58
59 <EditText
60     android:id="@+id/editTen"
61     android:layout_width="wrap_content"
62     android:layout_height="wrap_content"
63     android:layout_span="2"
64     android:ems="10" />
```

```
65    </TableRow>
66
67    <TableRow
68        android:id="@+id/tableRow3"
69        android:layout_width="wrap_content"
70        android:layout_height="wrap_content" >
71
72        <TextView
73            android:id="@+id/textView4"
74            android:layout_width="wrap_content"
75            android:layout_height="wrap_content"
76            android:text="Loại NV:" />
77
78        <RadioGroup
79            android:id="@+id/radiogroud1"
80            android:orientation="horizontal" >
81
82            <RadioButton
83                android:id="@+id/radChinhthuc"
84                android:layout_width="wrap_content"
85                android:layout_height="wrap_content"
86                android:checked="true"
87                android:text="Chính thức" />
88
89            <RadioButton
90                android:id="@+id/radThoivu"
91                android:layout_width="wrap_content"
92                android:layout_height="wrap_content"
93                android:text="Thời vụ" />
94        </RadioGroup>
95
96    </TableRow>
97
98    <TableRow
99        android:id="@+id/tableRow4"
100       android:layout_width="wrap_content"
101       android:layout_height="wrap_content" >
102
103        <Button
104            android:id="@+id/btnnhap"
105            android:layout_width="wrap_content"
106            android:layout_height="wrap_content"
107            android:layout_column="1"
108            android:text="Nhập NV" />
109
110    </TableRow>
111    </TableLayout>
112
113    <TextView
114        android:id="@+id/textView5"
115        android:layout_width="match_parent"
116        android:layout_height="wrap_content"
117        android:background="#008000" />
118
119    <ListView
```

```

119     android:id="@+id/lvnhanvien"
120     android:layout_width="match_parent"
121     android:layout_height="wrap_content" >
122   </ListView>
123
124 </LinearLayout>
125

```

- Xem nội dung từng class:

- Abstract class Employee:

```

1  package tranduythanh.com;
2
3  public abstract class Employee {
4      private String id;
5      private String name;
6
7      public String getId() {
8          return id;
9      }
10
11     public void setId(String id) {
12         this.id = id;
13     }
14
15     public String getName() {
16         return name;
17     }
18
19     public void setName(String name) {
20         this.name = name;
21     }
22
23     public abstract double TinhLuong();
24     @Override
25     public String toString() {
26         // TODO Auto-generated method stub
27         return this.id+" - "+this.name;
28     }
29 }

```

- class EmployeeFullTime:

```

1  package tranduythanh.com;
2
3  public class EmployeeFullTime extends Employee {
4
5      @Override
6      public double TinhLuong() {
7          return 500;
8      }
9      @Override

```

```

10     public String toString() {
11         // TODO Auto-generated method stub
12         return super.toString() +" -->FullTime="+TinhLuong();
13     }
14 }
```

- Class EmployeePartTime:

```

1     package tranduythanh.com;
2
3     public class EmployeePartTime extends Employee {
4
5         @Override
6         public double TinhLuong() {
7             // TODO Auto-generated method stub
8             return 150;
9         }
10        @Override
11        public String toString() {
12            // TODO Auto-generated method stub
13            return super.toString() +" -->PartTime="+TinhLuong();
14        }
15    }
```

- Bạn thấy là 2 class dẫn xuất từ Employee Tôi làm đơn giản là cách tính lương khác nhau. Đối với FullTime thì lương 500, Partime lương 150. và hàm Xuất `toString()` cũng khác nhau 1 xí.

- Ở đây ta sẽ áp dụng tính đa hình thông qua thừa kế, chỉ cần dùng một biến có kiểu Employee, nhưng nó có thể hiểu FullTime hoặc Partime và xuất ra thông tin đúng như mình mong đợi.

- Xem class MainActivity.java:

```

1     package tranduythanh.com;
2
3     import java.util.ArrayList;
4     import android.os.Bundle;
5     import android.app.Activity;
6     import android.view.View;
7     import android.view.View.OnClickListener;
8     import android.widget.ArrayAdapter;
9     import android.widget.Button;
10    import android.widget.EditText;
11    import android.widget.ListView;
12    import android.widget.RadioGroup;
13
14    public class MainActivity extends Activity {
15
16        EditText editId,editName;
17        Button btnNhaph;
18        RadioGroup radgroup;
19        ListView lvNhanvien;
20        ArrayList<Employee>arrEmployee=new ArrayList<Employee>();
21        ArrayAdapter<Employee>adapter=null;
```

```

22 //Khai báo 1 employee object
23 Employee employee=null;
24
25 @Override
26 protected void onCreate(Bundle savedInstanceState) {
27     super.onCreate(savedInstanceState);
28     setContentView(R.layout.activity_main);
29     editId=(EditText) findViewById(R.id.editMa);
30     editName=(EditText) findViewById(R.id.editTen);
31     btnNhap=(Button) findViewById(R.id.btnnhap);
32     radgroup=(RadioGroup) findViewById(R.id.radiogroud1);
33     lvNhanvien=(ListView) findViewById(R.id.lvnhanvien);
34     //đưa Data Source là các employee vào Adapter
35     adapter=new ArrayAdapter<Employee>(this,
36         android.R.layout.simple_list_item_1,
37         arrEmployee);
38     //đưa adapter vào ListView
39     lvNhanvien.setAdapter(adapter);
40
41     btnNhap.setOnClickListener(new OnClickListener() {
42
43         @Override
44         public void onClick(View arg0) {
45             // TODO Auto-generated method stub
46             processNhap();
47         }
48     });
49 }
50 //Xử lý sự kiện nhập
51 public void processNhap()
52 {
53     //Lấy ra đúng id của Radio Button được checked
54     int radId=radgroup.getCheckedRadioButtonId();
55     String id=editId.getText()+"";
56     String name=editName.getText()+"";
57     if(radId==R.id.radChinhthuc)
58     {
59         //tạo instance là FullTime
60         employee=new EmployeeFullTime();
61     }
62     else
63     {
64         //Tạo instance là Partime
65         employee=new EmployeePartTime();
66     }
67     //FullTime hay Partime thì cũng là Employee
68     //nên có các hàm này là hiển nhiên
69     employee.setId(id);
70     employee.setName(name);
71     //Đưa employee vào ArrayList
72     arrEmployee.add(employee);
73     //Cập nhập giao diện
74     adapter.notifyDataSetChanged();
75 }

```

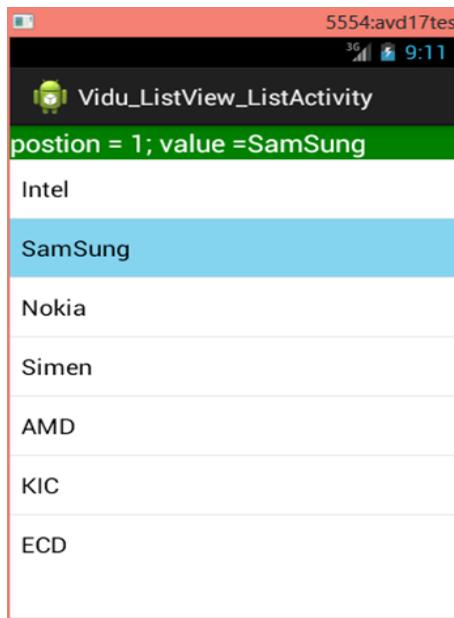
- Chương trình sẽ tự động hiển thị đúng loại Employee mà ta chọn lựa trên giao diện, ở đây bạn lại được ôn tập thêm về tính đa hình trong java. Ứng với mỗi loại Employee nó sẽ tự động gọi hàm `toString` của loại đó.

Ví dụ: Nếu Employee đó là FullTime thì nó gọi `toString` của FullTime và ngược lại với PartTime cũng vậy nó sẽ lấy `toString` của PartTime.

- Đến đây bạn đã hiểu tương đối sâu về ListView, bạn có thể tải coding mẫu ở đây: <http://www.mediafire.com/?o1we6e0mw8cpbba>

5) Trường hợp 5: Sử dụng ListView nhưng dưới dạng ListActivity:

- Thay vì kế thừa từ **Activity**, ta sẽ cho kế thừa từ **ListActivity**.
- Và dĩ nhiên cách đặt Id cho ListView cũng có sự khác biệt.
- Bạn xem giao diện bên dưới:



- Xem cách làm XML layout (**activity_main.xml**):

```

1 <LinearLayout
2   xmlns:android="http://schemas.android.com/apk/res/android"
3   xmlns:tools="http://schemas.android.com/tools"
4   android:id="@+id/LinearLayout1"
5   android:layout_width="match_parent"
6   android:layout_height="match_parent"
7   android:orientation="vertical"
8   tools:context=".MainActivity" >
9
10 <TextView
11   android:id="@+id/selection"
12   android:layout_width="match_parent"
```

```

13     android:layout_height="wrap_content"
14     android:background="#008000"
15     android:textColor="#FFFFFF"
16     android:textSize="20sp" />
17
18 <ListView
19     android:id="@+id/list"
20     android:layout_width="match_parent"
21     android:layout_height="wrap_content" >
22 </ListView>
23
24 <TextView
25     android:id="@+id/empty"
26     android:layout_width="wrap_content"
27     android:layout_height="wrap_content"
28     android:text="Không có gì cả" />
29
</LinearLayout>
```

- Bạn nhìn vào dòng lệnh thứ 18:

android:id=”@+id/list“, bạn viết y xì như thế này.

Bởi vì nó là id được định nghĩa sẵn bên trong Android.

- Tiếp tục nhìn vào dòng lệnh 24:

android:id=”@+id/empty“, cũng là có sẵn của Android. Nó có tác dụng tự động thông báo khi ListView của bạn không có bất kỳ một phần tử nào cả.

- Xem class MainActivity.java:

```

1  package tranduythanh.com;
2
3  import android.os.Bundle;
4  import android.app.ListActivity;
5  import android.view.View;
6  import android.widget.ArrayAdapter;
7  import android.widget.ListView;
8  import android.widget.TextView;
9
10 public class MainActivity extends ListActivity {
11
12     TextView selection;
13     String arr[]={ "Intel", "SamSung",
14     "Nokia", "Simen", "AMD",
15     "KIC", "ECD" };
16     ArrayAdapter<String >adapter=null;
17     @Override
18     protected void onCreate(Bundle savedInstanceState) {
19         super.onCreate(savedInstanceState);
20         setContentView(R.layout.activity_main);
21         //Thiết lập Data Source cho Adapter
22         adapter=new ArrayAdapter<String>
```

```

23     (this,
24     android.R.layout.simple_list_item_1,
25     arr);
26     //Gán Adapter vào ListView
27     //Nhớ là phải đặt id cho ListView theo đúng quy tắc
28     setListAdapter(adapter);
29
30     selection=(TextView) findViewById(R.id.selection);
31   }
32   @Override
33   protected void onListItemClick(ListView l, View v, int
34   position, long id) {
35     // TODO Auto-generated method stub
36     super.onListItemClick(l, v, position, id);
37     String txt="postion = "+position +" ; value =" +arr[position];
38     selection.setText(txt);
39   }
}

```

- Bạn nhìn vào dòng lệnh số 10: bạn thấy đây, Tôi cho kế thừa từ **ListActivity** chứ không phải**Activity**. (Android đã viết class ListActivity kế thừa từ Activity rồi). Tức là ListActivity cũng chính là Activity.

- Bạn nhìn vào dòng lệnh số 28:

setListAdapter(adapter);

Ở đây ta hoàn toàn không “Lôi” control ListView ra. Mà ta chỉ cần gọi hàm **setListAdapter** thì ListView cũng tự động cập nhập dữ liệu. Bạn phải làm chính xác 2 nơi thì mới được như vậy:

1. Kết thừa từ **ListActivity**,

2. đặt id cho ListView theo đúng quy tắc **android:id=”@+id/list”**

- Bạn load coding đầy đủ ở đây: <http://www.mediafire.com/?mjxgs1yv0mvujuz>

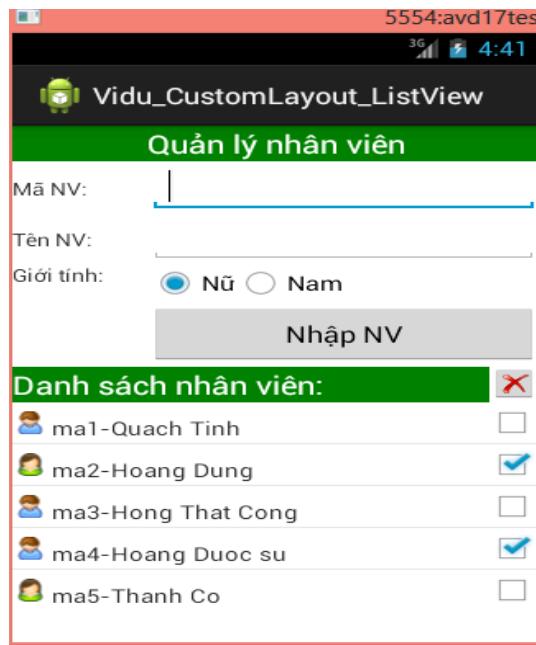
*** Như vậy bạn đã thực hành xong 5 trường hợp mà Tôi đã nêu. Tôi hi vọng các bạn hãy làm lại nhiều lần để có thể hiểu sâu hơn về nó.

Bài tập kế tiếp bạn sẽ được thực hành về cách Custom layout lại ListView theo ý thích của bạn.

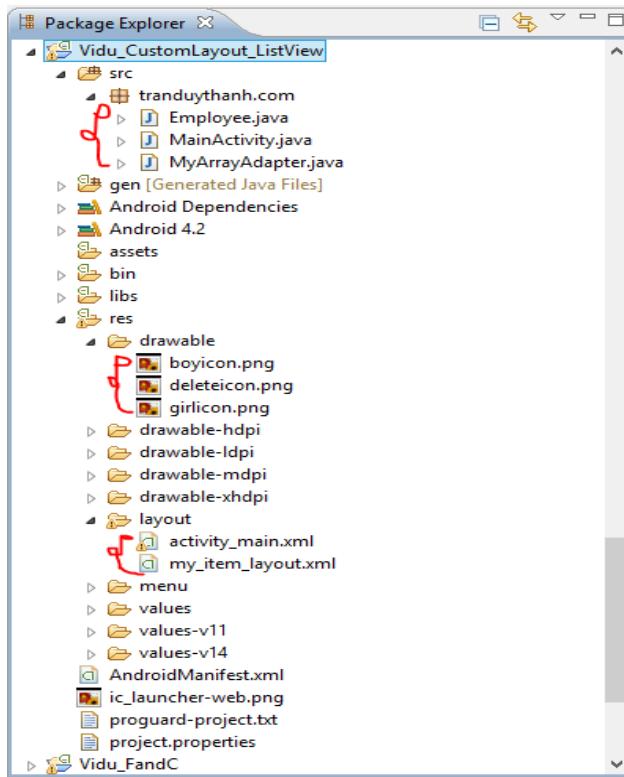
Bài tập 14: Thực hành về Custom Layout cho ListView trong Android

Ở bài tập 13 bạn đã được thực hành với ListView control. Trong bài tập này bạn sẽ học cách Custom lại layout cho ListView trong ứng dụng Android của bạn. Tôi nghĩ bài tập này nó rất quan trọng và thực tế bởi vì trong các ứng dụng Android có liên quan tới ListView thì đa phần chúng ta phải custom lại cho đúng với yêu cầu của khách hàng.

Tôi có làm một ví dụ về quản lý nhân viên với giao diện bên dưới đây:



- Bạn quan sát là phần danh sách nhân viên bên dưới là Custom Layout.
- Mỗi dòng trong ListView sẽ **có 3 đối tượng: ImageView, TextView và Checkbox**.
- Khi nhập nhân viên nếu người sử dụng chọn Nữ thì sẽ hiển thị hình là con gái, nếu chọn Nam thì hiển thị hình là con trai (bạn nhìn danh sách hình nhỏ nhở 16×16 ở ListView).
- Mã và tên của nhân viên sẽ được hiển thị vào TextView
- Checkbox cho phép người sử dụng checked (nhằm đánh dấu những nhân viên muốn xóa, ở đây cho phép xóa nhiều nhân viên)
- Bạn để ý Tôi có thêm 1 ImageButton có hình màu Chéo đỏ, nó dùng để xóa tất cả các nhân viên được Checked trong ListView, sau khi xóa thành công thì phải cập nhật lại ListView.
- Để làm được điều trên thì ta sẽ kế thừa từ **ArrayAdapter** và **override phương thức getView**, cụ thể:
 - Bạn xem Cấu trúc chương trình quản lý nhân viên:



- Tôi tạo thêm thư mục **drawable** và kéo thả 3 icon mà Tôi sử dụng vào (bạn cũng tạo thư mục tên y xì vậy). Nhớ là tên hình phải viết liền và chữ thường đầu tiên.

- Trong thư mục layout: Tôi tạo thêm **my_item_layout.xml** dùng để Custom lại ListView, dưới đây là cấu trúc XML của nó:

```

1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout
3     xmlns:android="http://schemas.android.com/apk/res/android"
4         android:id="@+id/LinearLayout1"
5         android:layout_width="match_parent"
6         android:layout_height="match_parent"
7         android:orientation="horizontal" >
8
9     <ImageView
10        android:id="@+id/imgitem"
11        android:layout_width="22dip"
12        android:layout_height="22dip"
13        android:paddingLeft="2dp"
14        android:paddingRight="2dp"
15        android:paddingTop="2dp"
16        android:layout_marginTop="4dp"
17        android:contentDescription="here"
18        android:src="@drawable/ic_launcher" />
19
20    <TextView
21        android:id="@+id/txtitem"
22        android:layout_height="wrap_content"
23        android:layout_width="0dip"
24        android:layout_weight="2"
25        android:layout_marginTop="4dp"
26        android:paddingLeft="2dp"
```

```

26     android:paddingRight="2dp"
27     android:paddingTop="2dp"
28     android:textSize="15sp" />
29
30 <CheckBox
31     android:id="@+id/chkitem"
32     android:layout_width="wrap_content"
33     android:layout_height="wrap_content" />
34
35 </LinearLayout>

```

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

- Bạn xem **activity_main.xml**:

```

1   <LinearLayout
2     xmlns:android="http://schemas.android.com/apk/res/android"
3     xmlns:tools="http://schemas.android.com/tools"
4     android:id="@+id/LinearLayout1"
5     android:layout_width="match_parent"
6     android:layout_height="match_parent"
7     android:orientation="vertical"
8     tools:context=".MainActivity" >
9
10  <TextView
11    android:id="@+id/textView2"
12    android:layout_width="match_parent"
13    android:layout_height="wrap_content"
14    android:background="#008000"
15    android:gravity="center"
16    android:text="Quản lý nhân viên"
17    android:textColor="#FFFFFF"
18    android:textSize="20sp" />
19
20  <TableLayout
21    android:layout_width="match_parent"
22    android:layout_height="wrap_content"
23    android:stretchColumns="*" >
24
25  <TableRow
26    android:id="@+id/tableRow1"
27    android:layout_width="wrap_content"
28    android:layout_height="wrap_content" >
29
30  <TextView
31    android:id="@+id/textView3"
32    android:layout_width="wrap_content"
33    android:layout_height="wrap_content"
34    android:text="Mã NV:" />
35
36  <EditText

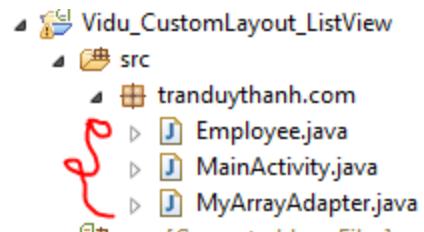
```

```
37     android:id="@+id/editMa"
38     android:layout_width="wrap_content"
39     android:layout_height="wrap_content"
40     android:ems="10" >
41
42     <requestFocus />
43   </EditText>
44
45 </TableRow>
46
47 <TableRow
48   android:id="@+id/tableRow2"
49   android:layout_width="wrap_content"
50   android:layout_height="wrap_content" >
51
52   <TextView
53     android:id="@+id/textView4"
54     android:layout_width="wrap_content"
55     android:layout_height="wrap_content"
56     android:text="Tên NV:" />
57
58   <EditText
59     android:id="@+id/editTen"
60     android:layout_width="wrap_content"
61     android:layout_height="wrap_content"
62     android:ems="10" />
63
64 </TableRow>
65
66 <TableRow
67   android:id="@+id/tableRow3"
68   android:layout_width="wrap_content"
69   android:layout_height="wrap_content" >
70
71   <TextView
72     android:id="@+id/textView5"
73     android:layout_width="wrap_content"
74     android:layout_height="wrap_content"
75     android:text="Giới tính:" />
76
77   <RadioGroup
78     android:id="@+id/radioGroup1"
79     android:layout_width="wrap_content"
80     android:layout_height="wrap_content"
81     android:orientation="horizontal" >
82
83     <RadioButton
84       android:id="@+id/radNu"
85       android:layout_width="wrap_content"
86       android:layout_height="wrap_content"
87       android:checked="true"
88       android:text="Nữ" />
89
90     <RadioButton
91       android:id="@+id/radNam"
```

```
91      android:layout_width="wrap_content"
92      android:layout_height="wrap_content"
93      android:text="Nam" />
94  </RadioGroup>
95
96  </TableRow>
97
98  <TableRow
99      android:id="@+id/tableRow4"
100     android:layout_width="wrap_content"
101     android:layout_height="wrap_content" >
102
103 <Button
104     android:id="@+id/btnNhap"
105     android:layout_width="wrap_content"
106     android:layout_height="wrap_content"
107     android:layout_column="1"
108     android:text="Nhập NV" />
109
110 </TableRow>
111 </TableLayout>
112
113 <LinearLayout
114     android:layout_width="match_parent"
115     android:layout_height="wrap_content" >
116
117 <TextView
118     android:id="@+id/textView1"
119     android:layout_width="wrap_content"
120     android:layout_height="wrap_content"
121     android:layout_weight="9"
122     android:background="#008000"
123     android:layout_marginTop="2dp"
124     android:text="Danh sách nhân viên:"
125     android:textColor="#FFFFFF"
126     android:textSize="20sp" />
127
128 <ImageButton
129     android:id="@+id/btnDelete"
130     android:layout_width="30dip"
131     android:layout_height="30dip"
132     android:src="@drawable/deleteicon" />
133
134 </LinearLayout>
135
136 <ListView
137     android:id="@+id/lvnhanvien"
138     android:layout_width="match_parent"
139     android:layout_height="wrap_content" >
140
141 </ListView>
142
143
```

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

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



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

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

- Cuối cùng **MainActivity**.

- Nay ta vào chi tiết từng class:

1) class Employee:

```
1  package tranduythanh.com;
2
3  public class Employee {
4      private String id;
5      private String name;
6      private boolean gender;
7      public String getId() {
8          return id;
9      }
10     public void setId(String id) {
11         this.id = id;
12     }
13     public String getName() {
14         return name;
15     }
16     public void setName(String name) {
17         this.name = name;
18     }
19     public boolean isGender() {
20         return gender;
21     }
22     public void setGender(boolean gender) {
23         this.gender = gender;
24     }
25     @Override
26     public String toString() {
27         return this.id+"-"+this.name;
28     }
29 }
```

2) class MyArrayAdapter:

```
1  package tranduythanh.com;
2
3  import java.util.ArrayList;
4  import android.app.Activity;
5  import android.view.LayoutInflater;
6  import android.view.View;
7  import android.view.ViewGroup;
8  import android.widget.ArrayAdapter;
9  import android.widget.ImageView;
10 import android.widget.TextView;
11
12 public class MyArrayAdapter extends
13     ArrayAdapter<Employee>
14 {
15     Activity context=null;
16     ArrayList<Employee>myArray=null;
17     int layoutId;
18     /**
19      * Constructor này dùng để khởi tạo các giá trị
20      * từ MainActivity truyền vào
21      * @param context : là Activity từ Main
22      * @param layoutId: Là layout custom do ta tạo
23      * (my_item_layout.xml)
24      * @param arr : Danh sách nhân viên truyền từ Main
25      */
26     public MyArrayAdapter(Activity context,
27             int layoutId,
28             ArrayList<Employee>arr) {
29         super(context, layoutId, arr);
30         this.context=context;
31         this.layoutId=layoutId;
32         this.myArray=arr;
33     }
34     /**
35      * hàm dùng để custom layout, ta phải override lại hàm này
36      * từ MainActivity truyền vào
37      * @param position : là vị trí của phần tử trong danh sách
38      * nhân viên
39      * @param convertView: convertView, dùng nó để xử lý Item
40      * @param parent : Danh sách nhân viên truyền từ Main
41      * @return View: trả về chính convertView
42      */
43     public View getView(int position, View convertView,
44             ViewGroup parent) {
45         /**
46          * bạn chú ý là ở đây Tôi không làm:
47          * if(convertView==null)
48          * {
49          * LayoutInflater inflater=
50          * context.getLayoutInflater();
51          * convertView=inflater.inflate(layoutId, null);
52          * }
53          * Lý do là ta phải xử lý xóa phần tử Checked, nếu dùng If
```

```

53     thì
54         * nó lại checked cho các phần tử khác sau khi xóa vì
55     convertView
56         * lưu lại trạng thái trước đó
57         */
58     LayoutInflater inflater=
59     context.getLayoutInflater();
60     convertView=inflater.inflate(layoutId, null);
61     //chỉ là test thôi, bạn có thể bỏ If đi
62     if(myArray.size()>0 && position>=0)
63     {
64         //dòng lệnh lấy TextView ra để hiển thị Mã và tên lên
65         final TextView txtdisplay=(TextView)
66         convertView.findViewById(R.id.txtitem);
67         //lấy ra nhân viên thứ position
68         final Employee emp=myArray.get(position);
69         //đưa thông tin lên TextView
70         //emp.toString() sẽ trả về Id và Name
71         txtdisplay.setText(emp.toString());
72         //lấy ImageView ra để thiết lập hình ảnh cho đúng
73         final ImageView imgitem=(ImageView)
74         convertView.findViewById(R.id.imgitem);
75         //nếu là Nữ thì lấy hình con gái
76         if(emp.isGender())
77             imgitem.setImageResource(R.drawable.girlicon);
78         else//nếu là Nam thì lấy hình con trai
79             imgitem.setImageResource(R.drawable.boyicon );
80     }
81     //Vì View là Object là dạng tham chiếu đối tượng, nên
82     //mọi sự thay đổi của các object bên trong convertView
83     //thì nó cũng biết sự thay đổi đó
84     return convertView;//trả về View này, tức là trả luôn
85     //về các thông số mới mà ta vừa thay đổi
86     }
87 }

```

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

3) class MainActivity:

```

1  package tranduythanh.com;
2
3  import java.util.ArrayList;
4
5  import android.os.Bundle;
6  import android.app.Activity;
7  import android.view.View;
8  import android.view.View.OnClickListener;
9  import android.widget.Button;
10 import android.widget.CheckBox;
11 import android.widget.EditText;
12 import android.widget.ImageButton;
13 import android.widget.ListView;
14 import android.widget.RadioGroup;

```

```

15
16 public class MainActivity extends Activity {
17
18     ArrayList<Employee> arrEmployee=new ArrayList<Employee>();
19     //Sử dụng MyArrayAdapter thay thi ArrayAdapter
20     MyArrayAdapter adapter=null;
21     ListView lvNhanvien=null;
22
23     Button btnNhaph;
24     ImageButton btnRemoveAll;
25     EditText editMa,editTen;
26     RadioGroup genderGroup;
27
28     @Override
29     protected void onCreate(Bundle savedInstanceState) {
30         super.onCreate(savedInstanceState);
31         setContentView(R.layout.activity_main);
32
33         btnNhaph=(Button) findViewById(R.id.btnNhaph);
34         btnRemoveAll=(ImageButton) findViewById(R.id.btndelete);
35         editMa=(EditText) findViewById(R.id.editMa);
36         editTen=(EditText) findViewById(R.id.editTen);
37         genderGroup=(RadioGroup) findViewById(R.id.radioGroup1);
38
39         lvNhanvien=(ListView) findViewById(R.id.lvnhanvien);
40         arrEmployee=new ArrayList<Employee>();
41         //Khởi tạo đối tượng adapter và gán Data source
42         adapter=new MyArrayAdapter(
43             this,
44             R.layout.my_item_layout,// lấy custom layout
45             arrEmployee/*thiết lập data source*/);
46         lvNhanvien.setAdapter(adapter); //gán Adapter vào Lisview
47
48         btnNhaph.setOnClickListener(new OnClickListener() {
49
50             @Override
51             public void onClick(View arg0) {
52                 // TODO Auto-generated method stub
53                 xulyNhaph();
54             }
55         });
56         btnRemoveAll.setOnClickListener(new OnClickListener() {
57
58             @Override
59             public void onClick(View arg0) {
60                 xulyXoa();
61             }
62         });
63         //gọi hàm xử lý nhập thông tin nhân viên
64         public void xulyNhaph()
65         {
66             String ma=editMa.getText()+"";
67             String ten=editTen.getText()+"";
68             boolean gioitinh=false;//Nam =false

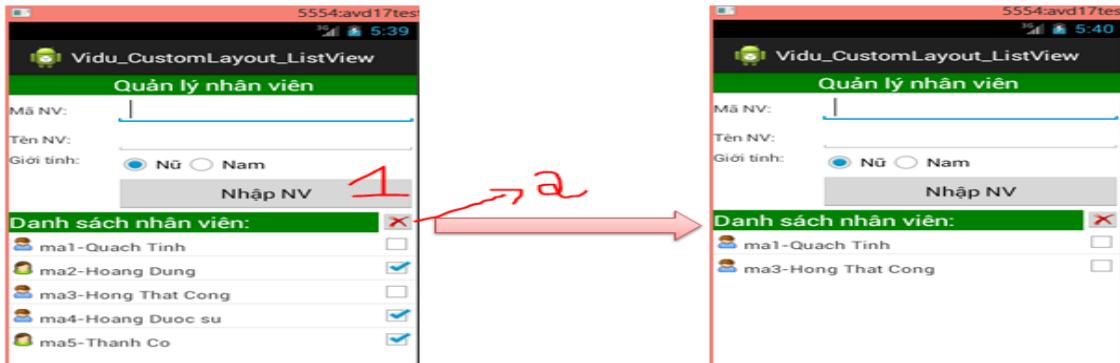
```

```

69     if(genderGroup.getCheckedRadioButtonId()==R.id.radNu)
70     gioitinh=true;
71     //Tạo một employee
72     Employee emp=new Employee();
73     emp.setId(ma);
74     emp.setName(ten);
75     emp.setGender(gioitinh);
76     //Đưa vào danh sách
77     arrEmployee.add(emp);
78     //gọi hàm cập nhật giao diện
79     adapter.notifyDataSetChanged();
80     //Sau khi update thì xóa trắng dữ liệu và cho editma
81     focus
82     editMa.setText("");
83     editTen.setText("");
84     editMa.requestFocus();
85     }
86     //hàm xử lý xóa
87     public void xulyXoa()
88     {
89     //ta nên đi ngược danh sách, kiểm tra phần tử nào checked
90     //thì xóa đúng vị trí đó ra khỏi arrEmployee
91     for(int i=lvNhanvien.getChildCount()-1;i>=0;i--)
92     {
93     //lấy ra dòng thứ i trong ListView
94     //Dòng thứ i sẽ có 3 phần tử: ImageView, TextView,
95     Checkbox
96     View v=lvNhanvien.getChildAt(i);
97     //Ta chỉ lấy CheckBox ra kiểm tra
98     CheckBox chk=(CheckBox) v.findViewById(R.id.chkitem);
99     //Nếu nó Checked thì xóa ra khỏi arrEmployee
100    if(chk.isChecked())
101    {
102    //xóa phần tử thứ i ra khỏi danh sách
103    arrEmployee.remove(i);
104    }
105    }
106    //Sau khi xóa xong thì gọi update giao diện
107    adapter.notifyDataSetChanged();
108  }

```

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



- Bạn hãy tìm hiểu thêm trên mạng về cách xử lý các phần tử khi custom, ở đây Tôi chưa xử lý xong: Ví dụ bạn chưa thể chọn được vào từng phần tử trong ListView (cho dù bạn có nghiên răng ngoáy mạnh ngón tay vào thì nó cũng không lung lay, Nếu bạn bỏ CheckBox đi thì lại được)... Nên bạn tìm hiểu thêm phần xử lý này (how to selected item in custom ListView).

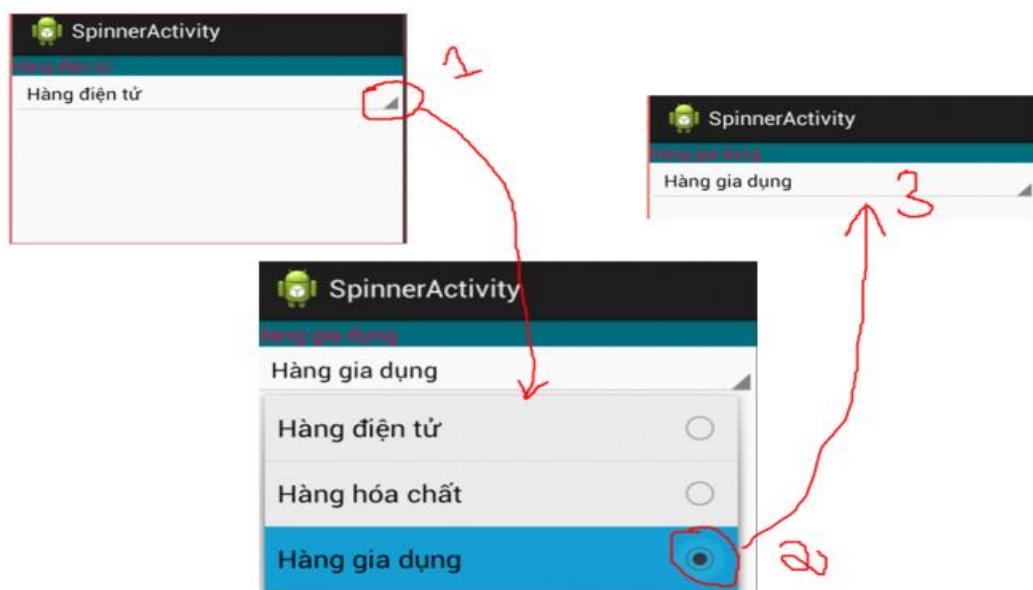
- Bạn có thể tải toàn bộ coding mẫu ở đây:<http://www.mediafire.com/?16tw6xf55nivr8k>

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

- Trong các bài tập tiếp theo bạn sẽ được thực hành về Spinner, GridView, kết hợp Spinner với ListView, kết hợp Spinner với GridView.

Bài tập 15: Thực hành về Spinner trong Android

- Trong bài tập này các bạn sẽ thực hành cách sử dụng **Spinner**.
- **Spinner** tương tự như ComboBox trong C#, tương tự như JComboBox trong Java.
- Nếu bạn đã hiểu về ListView thì việc hiểu Spinner là chuyện thường.
- Cách đỗ dữ liệu lên **Spinner** là y xì như đỗ lên ListView, nó chỉ khác một chỗ duy nhất trong ArrayAdapter đó là ta phải gọi **setDropDownViewResource**.
- Bạn xem hình ví dụ dưới đây về Spinner:



- Ví dụ trên làm rất đơn giản, bạn chỉ việc kéo 2 control: TextView và Spinner vào ứng dụng (xem [activity_spinner.xml](#)):

```
1 <LinearLayout  
2   xmlns:android="http://schemas.android.com/apk/res/android"  
3   xmlns:tools="http://schemas.android.com/tools"  
4   android:id="@+id/LinearLayout1"  
5   android:layout_width="match_parent"  
6   android:layout_height="match_parent"  
7   android:orientation="vertical"  
8   tools:context=".SpinnerActivity" >  
9     <TextView  
10    android:id="@+id/selection"  
11    android:layout_width="match_parent"  
12    android:layout_height="wrap_content"  
13    android:background="#007380"  
14    android:hint="selected here"  
15    android:textColor="#ff003c" />  
16    <Spinner
```

```

17     android:id="@+id/spinner1"
18     android:layout_width="match_parent"
19     android:layout_height="wrap_content" />
</LinearLayout>
```

- Ở đây Tôi đặt Id cho spinner là **spinner1** (nhìn dòng lệnh 16).

- Coding SpinnerActivity.java:

```

1  package tranduythanh.com;
2  import android.os.Bundle;
3  import android.app.Activity;
4  import android.view.View;
5  import android.widget.AdapterView;
6  import android.widget.AdapterView.OnItemSelectedListener;
7  import android.widget.ArrayAdapter;
8  import android.widget.Spinner;
9  import android.widget.TextView;
10
11 public class SpinnerActivity extends Activity {
12
13 //Tạo một mảng dữ liệu giả
14 String arr[]={
15     "Hàng điện tử",
16     "Hàng hóa chất",
17     "Hàng gia dụng"};
18 TextView selection;
19 @Override
20 protected void onCreate(Bundle savedInstanceState) {
21     super.onCreate(savedInstanceState);
22     setContentView(R.layout.activity_spinner);
23     selection =(TextView) findViewById(R.id.selection);
24     //Lấy đối tượng Spinner ra
25     Spinner spin=(Spinner) findViewById(R.id.spinner1);
26     //Gán Data source (arr) vào Adapter
27     ArrayAdapter<String> adapter=new ArrayAdapter<String>
28     (
29         this,
30         android.R.layout.simple_spinner_item,
31         arr
32     );
33     //phải gọi lệnh này để hiển thị danh sách cho Spinner
34     adapter.setDropDownViewResource
35     (android.R.layout.simple_list_item_single_choice);
36     //Thiết lập adapter cho Spinner
37     spin.setAdapter(adapter);
38     //thiết lập sự kiện chọn phần tử cho Spinner
39     spin.setOnItemSelectedListener(new MyProcessEvent());
40 }
41 //Class tạo sự kiện
42 private class MyProcessEvent implements
43     OnItemSelectedListener
44 {
45     //Khi có chọn lựa thì vào hàm này
```

```

46     public void onItemSelected(AdapterView<?> arg0,
47         View arg1,
48         int arg2,
49         long arg3) {
50         //arg2 là phần tử được chọn trong data source
51         selection.setText(arr[arg2]);
52     }
53     //Nếu không chọn gì cả
54     public void onNothingSelected(AdapterView<?> arg0) {
55         selection.setText("");
56     }
57 }
58 }
```

- Bạn xem Tôi giải thích dưới này:



```

protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_spinner);
    selection =(TextView) findViewById(R.id.selection);
    Spinner spin=(Spinner) findViewById(R.id.spinner1);
    ArrayAdapter<String> adapter=new ArrayAdapter<String>
    (
        this,
        android.R.layout.simple_spinner_item,
        arr
    );
    adapter.setDropDownViewResource
        (android.R.layout.simple_list_item_single_choice);
    spin.setAdapter(adapter);
    spin.setOnItemSelectedListener(new MyProcessEvent());
}
```

- Bạn thấy đó **android.R.layout.simple_spinner_item** dùng để hiển thị phần tử bạn chọn lên spinner. Tương tự như trong ListView bạn có thể custom lại

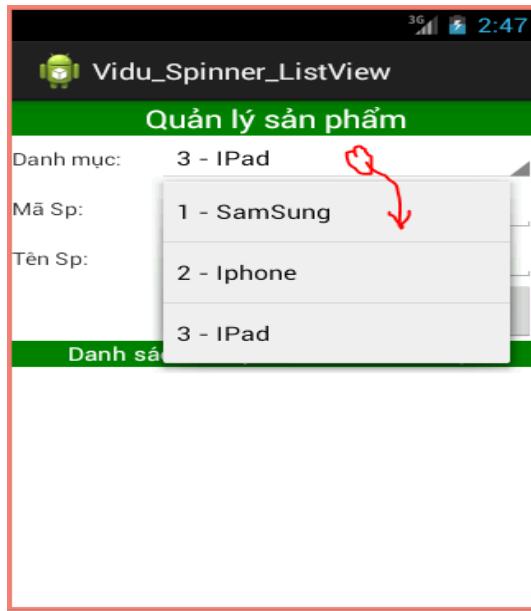
- Dòng lệnh dưới: **android.R.layout.simple_list_item_single_choice** để hiển thị danh sách các phần tử trong Spinner khi bạn nhấn vào xem Spinner. Bạn phải gọi hàm **setDropDownViewResource** nếu không nó sẽ lỗi chương trình khi bạn nhấn vào xem. Bạn có thể dùng layout khác nhau, chẳng hạn bạn có thể thay thế bằng : **android.R.layout.simple_spinner_dropdown_item**

- Như vậy bạn đã làm quen được với Spinner, bạn có thể load code mẫu ở đây: <http://www.mediafire.com/?1pmikmscb30po3s>

- Ban đầu Tôi tính kết hợp ListView và Spinner trong bài tập này luôn, nhưng vì thấy nó hơi phức tạp nên Tôi đã tách ra một bài tập thực hành riêng, trong bài tập tới bạn sẽ học phần này.

Bài tập 16: Kết hợp Spinner với ListView trong Android

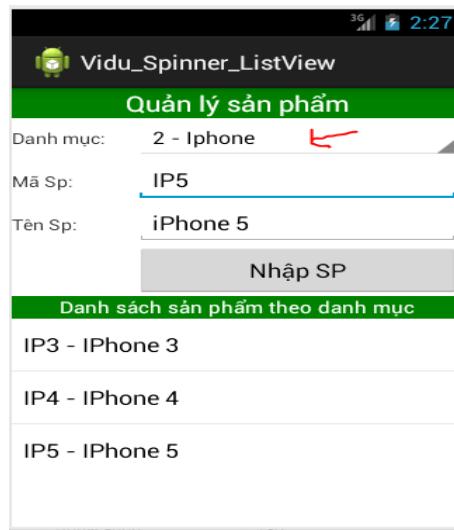
Ở bài tập 15 bạn đã làm quen được với **Spinner**, trong bài tập này bạn sẽ làm một ví dụ về cách kết hợp giữa **Spinner** với **ListView**. Thường thì 2 control này đi với nhau sẽ tạo thành cặp bài trùng, **Spinner** dùng để lưu trữ danh mục còn **ListView** lưu trữ danh sách của từng danh mục. Ở đây Tôi có làm một ví dụ về quản lý sản phẩm, bạn xem hình:



- Bạn quan sát hình Tôi chụp ứng dụng, phần trên cùng là Danh mục các sản phẩm được lưu vào **Spinner**. Khi bạn chọn vào nó thì sẽ xổ ra danh sách như bên dưới. khi bạn chọn danh mục nào thì nó sẽ load các sản phẩm thuộc danh mục đó.
- Ví dụ bây giờ bạn chọn số 1 là SamSung, nó sẽ load toàn bộ sản phẩm là SamSung vào ListView bên dưới:



- Nếu bạn chọn 2- Iphone thì nó sẽ load toàn bộ sản phẩm là IPhone vào ListView bên dưới:



- Chương trình cung cấp nút “Nhập SP“, khi người sử dụng nhập thông tin cho sản phẩm và nhấn nút này thì chương trình sẽ lưu sản phẩm vào đúng với danh mục được chọn trong **Spinner** đồng thời cập nhật vào **ListView** bên dưới.

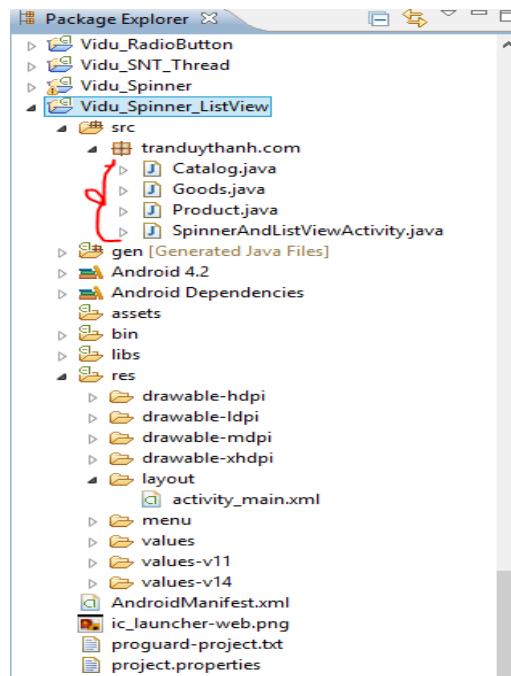
- Bạn cần có **ArrayList + ArrayAdapter** cho Spinner

- Và cần có **ArrayList + ArrayAdapter** cho ListView

-> Tức là bạn phải có 2 cặp (4 đối tượng trên)

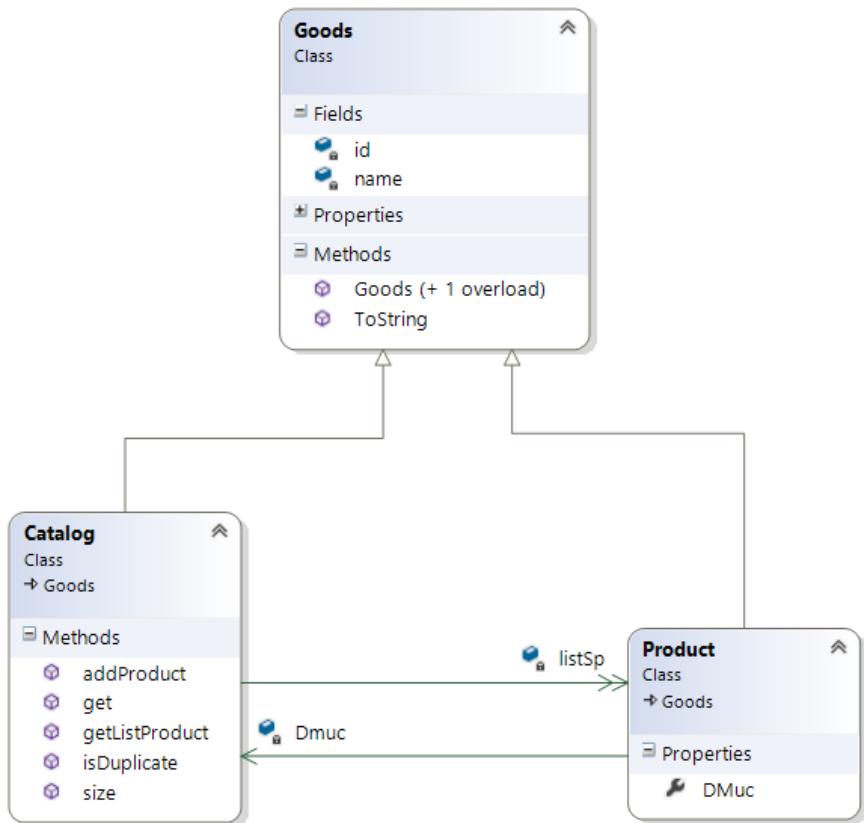
- Ví dụ này Tôi viết thuần hướng đối tượng, và có hơi phá cách một chút so với quy tắc hướng đối tượng thông thường, đó là trong lớp Sản phẩm Tôi cho phép nó tham chiếu trực tiếp tới đối tượng Danh mục chứa nó. Như vậy thì đứng tại danh mục nào cũng có thể lấy được toàn bộ danh sách sản phẩm của nó, và đứng tại một sản phẩm bất kỳ nào cũng biết được nó thuộc danh mục nào.

- Bạn xem cấu trúc chương trình:



- Ở trên bạn thấy có 3 class: **Goods**, **Product**, **Catalog**: Product và Catalog sẽ kế thừa từ **Goods**, Goods sẽ có Id và Name. Sản phẩm và danh mục cũng phải có Id và Name nên nó kế thừa từ Goods là đều hợp lý.

- Bạn xem mô hình lớp:



-Bạn xem cấu trúc XML cho phần giao diện (**activity_main.xml**):

```
1 <LinearLayout
2     xmlns:android="http://schemas.android.com/apk/res/android"
3     xmlns:tools="http://schemas.android.com/tools"
4     android:id="@+id/LinearLayout1"
5     android:layout_width="match_parent"
6     android:layout_height="match_parent"
7     android:orientation="vertical"
8     tools:context=".MainActivity" >
9
10    <TextView
11        android:id="@+id/textView1"
12        android:layout_width="match_parent"
13        android:layout_height="wrap_content"
14        android:background="#008000"
15        android:gravity="center"
16        android:text="Quản lý sản phẩm"
17        android:textColor="#FFFFFF"
18        android:textSize="20sp" />
19
20    <TableLayout
21        android:layout_width="match_parent"
22        android:layout_height="wrap_content"
```

```
23     android:stretchColumns="*"
24   >
25
26   <TableRow
27     android:id="@+id/tableRow1"
28     android:layout_width="wrap_content"
29     android:layout_height="wrap_content" >
30
31   <TextView
32     android:id="@+id/textView2"
33     android:layout_width="wrap_content"
34     android:layout_height="wrap_content"
35     android:text="Danh mục:" />
36
37   <Spinner
38     android:id="@+id/spDanhmuc"
39     android:layout_width="wrap_content"
40     android:layout_height="wrap_content" />
41
42 </TableRow>
43
44   <TableRow
45     android:id="@+id/tableRow2"
46     android:layout_width="wrap_content"
47     android:layout_height="wrap_content" >
48
49   <TextView
50     android:id="@+id/textView3"
51     android:layout_width="wrap_content"
52     android:layout_height="wrap_content"
53     android:text="Mã Sp:" />
54
55   <EditText
56     android:id="@+id/editId"
57     android:layout_width="wrap_content"
58     android:layout_height="wrap_content"
59     android:ems="10" >
60
61   <requestFocus />
62   </EditText>
63
64 </TableRow>
65
66   <TableRow
67     android:id="@+id/tableRow3"
68     android:layout_width="wrap_content"
69     android:layout_height="wrap_content" >
70
71   <TextView
72     android:id="@+id/textView4"
73     android:layout_width="wrap_content"
74     android:layout_height="wrap_content"
75     android:text="Tên Sp:" />
76
77   <EditText
```

```

77     android:id="@+id/editName"
78     android:layout_width="wrap_content"
79     android:layout_height="wrap_content"
80     android:ems="10" />
81
82 </TableRow>
83
84 <TableRow
85     android:id="@+id/tableRow4"
86     android:layout_width="wrap_content"
87     android:layout_height="wrap_content" >
88
89 <Button
90     android:id="@+id	btnInput"
91     android:layout_width="wrap_content"
92     android:layout_height="wrap_content"
93     android:layout_column="1"
94     android:text="Nhập SP" />
95
96 </TableRow>
97 </TableLayout>
98
99 <TextView
100    android:id="@+id/textView5"
101    android:layout_width="match_parent"
102    android:layout_height="wrap_content"
103    android:background="#008000"
104    android:gravity="center"
105    android:text="Danh sách sản phẩm theo danh mục"
106    android:textColor="#FFFFFF"
107    android:textSize="15sp" />
108
109 <ListView
110    android:id="@+id/lvsanpham"
111    android:layout_width="match_parent"
112    android:layout_height="wrap_content" >
113
114 </LinearLayout>
115

```

- Tương tự như các bài tập trước, chúng ta phải đặt id cho các control. Bài tập này bạn tự nhìn vào để xem Tôi đặt Id như thế nào, Tôi không nhắc lại nữa.

- Chúng ta lần lượt xem nội dung coding của các class trong phần xử lý nghiệp vụ:

- 1) Class Goods:

```

1 package tranduythanh.com;
2 /**
3  * Class này là class cha của Product và Catalog
4  * vì Product và Catalog đều có Id và Name
5  * nên Tôi tạo class này để sử dụng lại code
6  * @author drthanh

```

```

7   *
8   */
9  public class Goods {
10    //Id để lưu mã
11    //Name để lưu tên
12    private String id;
13    private String name;
14    public String getId() {
15      return id;
16    }
17    public void setId(String id) {
18      this.id = id;
19    }
20    public String getName() {
21      return name;
22    }
23    public void setName(String name) {
24      this.name = name;
25    }
26    public Goods(String id, String name) {
27      super();
28      this.id = id;
29      this.name = name;
30    }
31    public Goods() {
32      super();
33    }
34    public String toString() {
35      return this.id + " - " + this.name;
36    }
37 }

```

- 2 – class Product:

```

1  package tranduythanh.com;
2 /**
3  * Class này để lưu thông tin sản phẩm
4  * nó kế thừa từ Goods để lấy mã và tên
5  * Tôi cho nó tham chiếu tới Catalog
6  * để nó có thể biết được nó thuộc danh mục nào
7  * @author drthanh
8  *
9  */
10 public class Product extends Goods{
11   //Lấy tham chiếu để lập trình cho lẹ
12   private Catalog Dmuc;
13
14   public Catalog getDmuc() {
15     return Dmuc;
16   }
17
18   public void setDmuc(Catalog dmuc) {
19     Dmuc = dmuc;
20   }

```

```

21
22 public Product(String ma, String name, Catalog dmuc) {
23     super(ma, name);
24     Dmuc = dmuc;
25 }
26
27 public Product(String ma, String name) {
28     super(ma, name);
29 }
30
31 public Product() {
32     super();
33 }
34
35 }
```

- 3 – class Catalog:

```

1 package tranduythanh.com;
2
3 import java.util.ArrayList;
4 /**
5  * Class dùng để lưu trữ thông tin danh mục
6  * và danh sách các sản phẩm thuộc danh mục
7  * @author drthanh
8  *
9 */
10 public class Catalog extends Goods {
11     private ArrayList<Product>listSp=null;
12     public Catalog(String ma, String name) {
13         super(ma, name);
14         this.listSp=new ArrayList<Product>();
15     }
16
17     public Catalog() {
18         super();
19         this.listSp=new ArrayList<Product>();
20     }
21     /**
22      * kiểm tra sản phẩm đã tồn tại trong danh mục hay chưa
23      * @param p
24      * @return true nếu tồn tại
25      */
26     public boolean isDuplicate(Product p)
27     {
28         for(Product p1: listSp)
29         {
30             if(p1.getid().trim().equalsIgnoreCase(p.getid().trim()))
31                 return true;
32         }
33         return false;
34     }
35     /**
36      * thêm 1 sản phẩm vào danh mục
```

```

37     * thêm thành công =true
38     * @param p
39     * @return
40     */
41    public boolean addProduct(Product p)
42    {
43        boolean isDup=isDuplicate(p);
44        if(!isDup)
45        {
46            p.setDmuc(this);
47            return listSp.add(p);
48        }
49        return !isDup;
50    }
51    public ArrayList<Product>getListProduct()
52    {
53        return this.listSp;
54    }
55    public int size()
56    {
57        return listSp.size();
58    }
59    public Product get(int i)
60    {
61        return listSp.get(i);
62    }
63}

```

- 4 – class SpinnerAndListViewActivity – xử lý nghiệp vụ trong Activity:

```

1 package tranduythanh.com;
2
3 import java.util.ArrayList;
4
5 import android.os.Bundle;
6 import android.app.Activity;
7 import android.view.View;
8 import android.view.View.OnClickListener;
9 import android.widget.AdapterView;
10 import android.widget.AdapterView.OnItemSelectedListener;
11 import android.widget.ArrayAdapter;
12 import android.widget.Button;
13 import android.widget.EditText;
14 import android.widget.ListView;
15 import android.widget.Spinner;
16
17 public class SpinnerAndListViewActivity extends Activity {
18
19     Spinner spinDm;
20     EditText editma,editten;
21     Button btnNhap;
22     ListView lvSp;
23     //cặp đôi tương dùng cho Spinner
24     ArrayList<Catalog> arraySpinner=new ArrayList<Catalog>();

```

```

25     ArrayAdapter<Catalog>adapterSpinner=null;
26     //Cặp đôi tương dùng cho ListView
27     ArrayList<Product>arrayListview=new ArrayList<Product>();
28     ArrayAdapter<Product>adapterListview=null;
29
30     @Override
31     protected void onCreate(Bundle savedInstanceState) {
32         super.onCreate(savedInstanceState);
33         setContentView(R.layout.activity_main);
34         getWidgetsControl();
35         fakeDataCatalog();
36         addEventsForFormWidgets();
37     }
38     /**
39      * Hàm lấy các control theo Id
40      */
41     private void getWidgetsControl()
42     {
43         spinDm=(Spinner) findViewById(R.id.spDanhmuc);
44         editma=(EditText) findViewById(R.id.editId);
45         editten=(EditText) findViewById(R.id.editName);
46         btnNhap=(Button) findViewById(R.id.btnInput);
47         lvSp=(ListView) findViewById(R.id.lvsanpham);
48
49         //Cấu hình cho Spinner
50         adapterSpinner=new ArrayAdapter<Catalog>(this,
51             android.R.layout.simple_spinner_item,
52             arraySpinner);
53
54         adapterSpinner.setDropDownViewResource
55             (android.R.layout.simple_spinner_dropdown_item);
56         spinDm.setAdapter(adapterSpinner);
57
58         //Cấu hình cho ListView
59         adapterListview=new ArrayAdapter<Product>(this,
60             android.R.layout.simple_list_item_1,
61             arrayListview);
62         lvSp.setAdapter(adapterListview);
63     }
64     /**
65      * Hàm giả dữ liệu, tạo 3 danh mục mặc định cho Spinner
66      */
67     private void fakeDataCatalog()
68     {
69         Catalog cat1=new Catalog("1", "SamSung");
70         Catalog cat2=new Catalog("2", "Iphone");
71         Catalog cat3=new Catalog("3", "IPad");
72         arraySpinner.add(cat1);
73         arraySpinner.add(cat2);
74         arraySpinner.add(cat3);
75         adapterSpinner.notifyDataSetChanged();
76     }
77     /**
78      * Hàm gán sự kiện cho Button và Spinner
79      */

```

```

79     private void addEventsForFormWidgets()
80     {
81         btnNhaph.setOnTouchListener(new OnTouchListener() {
82
83             @Override
84             public void onClick(View arg0) {
85                 addProductForCatalog();
86             }
87         });
88         spinDm.setOnItemSelectedListener(new OnItemSelectedListener()
89         {
90
91             @Override
92             public void onItemSelected(AdapterView<?> arg0, View arg1,
93             int arg2, long arg3) {
94                 //mỗi lần chọn danh mục trong Spinner thì cập nhập ListView
95                 loadListProductByCatalog(arraySpinner.get(arg2));
96             }
97
98             @Override
99             public void onNothingSelected(AdapterView<?> arg0) {
100                 // TODO Auto-generated method stub
101             }
102         }
103     );
104     }
105     /**
106      * Hàm thêm một sản phẩm vào cho danh mục được chọn trong
107      * Spinner
108      */
109     private void addProductForCatalog()
110     {
111         Product p=new Product();
112         p.setid(editma.getText()+"");
113         p.setName(editten.getText());
114         Catalog c= (Catalog) spinDm.getSelectedItem();
115         c.addProduct(p);
116         //Mỗi lần thêm xong thì cập nhập lại ListView
117         loadListProductByCatalog(c);
118     }
119     /**
120      * Lọc danh sách sản phẩm theo danh mục và update lại ListView
121      * @param c
122      */
123     private void loadListProductByCatalog(Catalog c)
124     {
125         //xóa danh sách cũ
126         arrayListview.clear();
127         //lấy danh sách mới từ Catalog chọn trong Spinner
128         arrayListview.addAll(c.getListProduct());
129         //cập nhật lại ListView
130         adapterListview.notifyDataSetChanged();
131     }
132 }

```

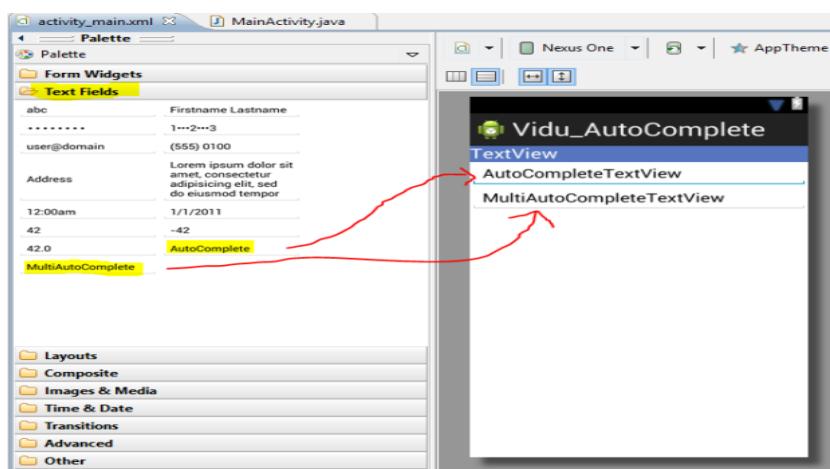
- Tôi đã giải thích ở ngay bên trong mã lệnh, bạn có thể đọc và rắng hiểu coding Tôi viết bên trên.
- Bạn có thể tải code mẫu đầy đủ ở đây: <http://www.mediafire.com/?i4lev5ek8944kop>
- Bài tập tiếp theo bạn sẽ học **AutoCompleteTextView** và **MultiAutoCompleteTextView**. 2 control này bản chất cũng giống như **EditText** nhưng nó hỗ trợ người sử dụng nhập dữ liệu được nhanh hơn, làm cho ứng dụng trở lên User Friendly hơn, bạn hãy chú ý theo dõi.

Bài tập 17: Thực hành về AutocompleteTextView và MultiAutocompleteTextView

- Đối với các thiết bị di động, việc hỗ trợ nhập dữ liệu nhanh cho người sử dụng là điều rất cần thiết.
- Android hỗ trợ 2 control này giúp chúng ta làm được điều đó. Bạn để ý là danh sách hiển thị lên nó tương tự như Spinner do đó một số bạn sẽ tưởng lầm là Spinner khi quan sát chưa kỹ.
- Tôi ví dụ một trường hợp cụ thể như sau: Bạn viết ứng dụng yêu cầu nhập vào quê quán, giả sử đất nước Việt Nam mình có 63 tỉnh thành, trong đó có các tỉnh như: Hà Nội, Huế, Hà Giang, Hà Nam Ninh,... bất kỳ tỉnh nào đó có chữ H hoặc một nhóm tỉnh thành nào đó có cùng một số ký tự đầu. Như vậy ứng dụng phải Thông minh tự đưa ra lời đề nghị nhập tỉnh thành theo đúng ký tự mà họ muốn nhập, xem hình dưới:



- Như hình trên: Bạn chỉ cần nhập ký tự **h** đầu tiên, nó sẽ lọc ra các tỉnh thành (hay thành phố) có ký tự đầu là **h**.
- Bạn nhớ đây **không phải là Spinner** vì bạn nhìn vào tưởng nó là Spinner. Mà nó là **AutoCompleteTextView**.
- Vậy 2 control này nó ở đâu? xem hình:



- Bạn xem cấu trúc XML của giao diện (**activity_main.xml**):

```
1 <LinearLayout  
2     xmlns:android="http://schemas.android.com/apk/res/android"  
3     xmlns:tools="http://schemas.android.com/tools"  
4     android:id="@+id/LinearLayout1"  
5     android:layout_width="match_parent"  
6     android:layout_height="match_parent"  
7     android:orientation="vertical"  
8     tools:context=".AutoCompleteTextViewActivity" >  
9         <TextView  
10            android:id="@+id/selection"  
11            android:layout_width="match_parent"  
12            android:layout_height="wrap_content"  
13            android:background="#5576BE"  
14            android:text="TextView"  
15            android:textColor="#FFFFFF"  
16            android:textSize="20sp" />  
17  
18         <AutoCompleteTextView  
19            android:id="@+id/editauto"  
20            android:layout_width="match_parent"  
21            android:layout_height="wrap_content"  
22            android:completionThreshold="1"  
23            android:ems="10" >  
24             <requestFocus />  
25         </AutoCompleteTextView>  
26  
27         <MultiAutoCompleteTextView  
28            android:id="@+id/multi.AutoCompleteTextView1"  
29            android:layout_width="match_parent"  
30            android:layout_height="wrap_content"  
31            android:completionThreshold="1"  
32            android:ems="10" />  
33  
34     </LinearLayout>
```

- Ở trên Tôi kéo 2 control ra luôn. Vì Tôi muốn demo code của 2 control này trong một chỗ để bạn dễ so sánh.

- Bạn nhìn vào dòng lệnh số 21 và 31:

android:completionThreshold= “1”

- Mục đích của nó là thiết lập số ký tự bắt đầu lọc trong AutoComplete. Ở đây Tôi nhập là số 1 tức là chỉ cần 1 ký tự là nó bắt đầu lọc, còn nếu như bạn sửa thành 3 thì bạn nhập tới 3 ký tự vào nó mới bắt đầu lọc.

- Xem class xử lý (**MainActivity.java**):

```
1 package tranduythanh.com;  
2
```

```

3 import android.os.Bundle;
4 import android.app.Activity;
5 import android.text.Editable;
6 import android.text.TextWatcher;
7 import android.widget.ArrayAdapter;
8 import android.widget.AutoCompleteTextView;
9 import android.widget.MultiAutoCompleteTextView;
10 import android.widget.TextView;
11
12 public class MainActivity extends Activity
13     implements TextWatcher /*TextWatcher để xử lý sự kiện
14 TextChange */
15 {
16     TextView selection;
17     //Khai báo 2 CompleteTextView
18     AutoCompleteTextView singleComplete;
19     MultiAutoCompleteTextView multiComplete;
20     //Khởi tạo mảng tạm để Test
21     String arr[]={ "hà nội", "Huế", "Sài gòn",
22     "hà giang", "Hội an", "Kiên giang",
23     "Lâm đồng", "Long khánh"};
24     protected void onCreate(Bundle savedInstanceState) {
25         super.onCreate(savedInstanceState);
26         setContentView(R.layout.activity_main);
27         selection =(TextView) findViewById(R.id.selection);
28         //lấy đối tượng AutoCompleteTextView ra
29         singleComplete=(AutoCompleteTextView)
30         findViewById(R.id.editauto);
31         //Thiết lập để lắng nghe TextChange
32         singleComplete.addTextChangedListener(this);
33         //Thiết lập ArrayAdapter
34         singleComplete.setAdapter(
35             new ArrayAdapter<String>
36             (
37                 this,
38                 android.R.layout.simple_list_item_1,
39                 arr
40             );
41         //Lấy đối tượng MultiAutoCompleteTextView ra
42         multiComplete=(MultiAutoCompleteTextView)
43         findViewById(R.id.multiAutoCompleteTextView1);
44         //Thiết lập ArrayAdapter
45         multiComplete.setAdapter(
46             new ArrayAdapter<String>
47             (
48                 this,
49                 android.R.layout.simple_list_item_1,
50                 arr
51             );
52         //Đối với MultiAutoCompleteTextView bắt buộc phải gọi dòng
53         lệnh này
54         multiComplete.setTokenizer(new MultiAutoCompleteTextView
55             .CommaTokenizer());
56     }
57     //Khi chọn trong AutoCompleteTextView hàm này sẽ tự động phát

```

```
57    sinh
58    public void onTextChanged(CharSequence arg0, int arg1,
59        int arg2, int arg3) {
60        selection.setText(singleComplete.getText());
61    }
62    public void afterTextChanged(Editable arg0) {
63    }
64    public void beforeTextChanged(CharSequence arg0, int arg1, int
arg2,
65        int arg3) {
66    }
67}
```

- Tôi giải thích thêm:

+ Việc gán **DataSource** vào **ArrayAdapter** rồi gán **ArrayAdapter** vào cho **ListView** như thế nào thì nó y xì như vậy đối với **AutoCompleteTextView**.

+ Đối với **MultiAutoCompleteTextView** cũng vậy, nó chỉ yêu cầu thêm dòng lệnh:

setTokenizer(new MultiAutoCompleteTextView.CommaTokenizer())

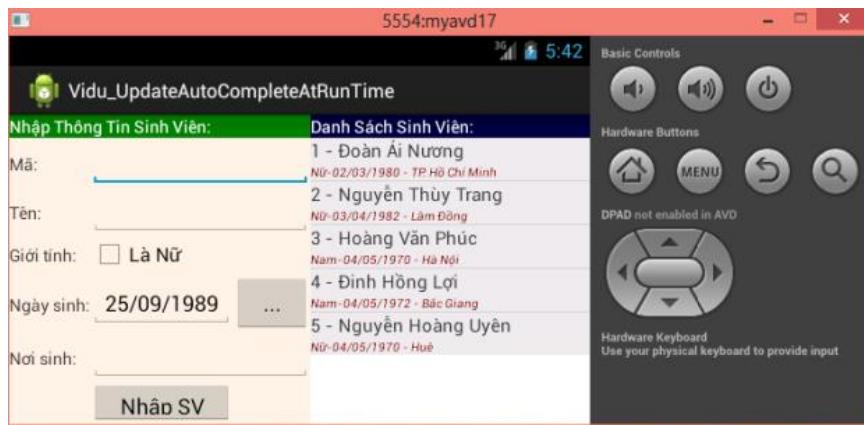
- Như vậy nếu bạn đã hiểu **ListView** thì không có lý do gì mà lại không hiểu **CompleteTextView**.

- Bạn tải code mẫu ở đây:<http://www.mediafire.com/?dos0a2v0bh6hp2b>

- Bài tập sau Tôi sẽ hướng dẫn các bạn cập nhật **DataSource** lúc Runtime cho **AutocompleteTextView**.

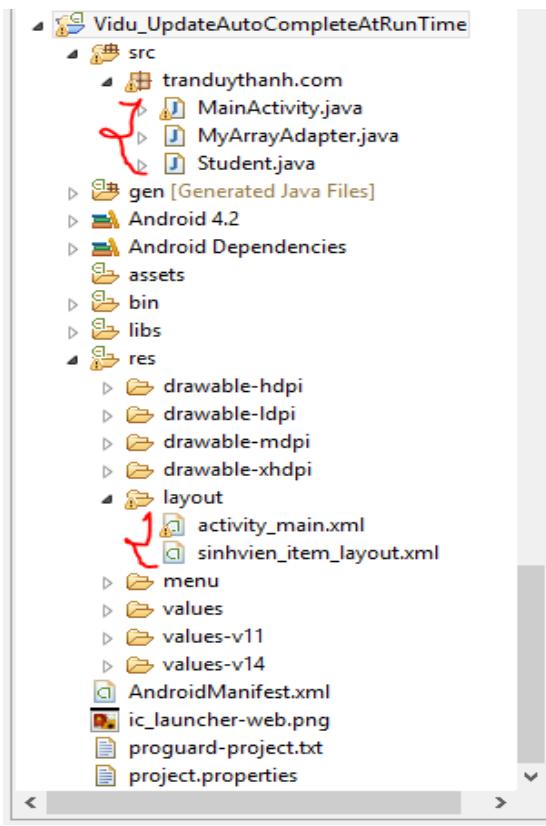
Bài tập 18: Cập nhật DataSource cho AutocompleteTextView lúc Runtime

- Trong [bài tập 17](#) bạn đã biết cách sử dụng **AutocompleteTextView** và **MultiAutocompleteTextView** nhưng DataSource được cố định sẵn trong Coding. Bài tập này Bạn sẽ học cách **cập nhật DataSource lúc ứng dụng đang chạy** (runtime).
- Tôi sẽ có một ví dụ cụ thể để bạn thực hành về trường hợp này, bạn hãy tập trung theo dõi và thực hành lại. Tôi nghĩ bài này rất thú vị.
- Bây giờ bạn mở AVD lên và nhấn tổ hợp phím ctrl+ F11 để xoay ngang màn hình, Tôi sẽ làm ví dụ về màn hình nằm ngang.



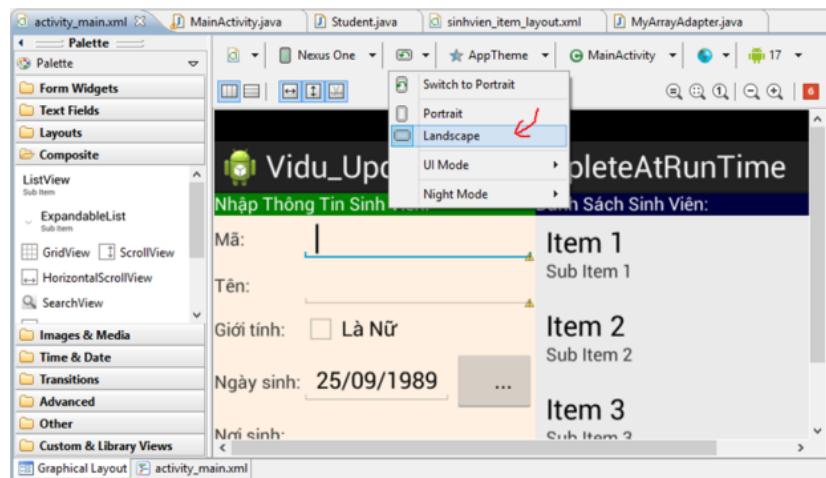
- Bạn xem màn hình trên Tôi chia ra làm 2. Bên trái cho phép nhập thông tin sinh viên, bên phải cho phép hiển thị danh sách sinh viên đã nhập. Bạn chú ý là ListView Tôi làm là Custom Layout (Mỗi dòng Lớn có 2 loại dữ liệu: mã và tên cùng 1 dòng. giới tính, năm sinh, quê quán cùng 1 dòng và chữ nhỏ hơn có màu đó in nghiêng).
- Ngày sinh dùng DatePickerDialog
- Nơi sinh dùng **AutoCompleteTextView**. Mỗi lần nhấn “Nhập Sv” thì chương trình sẽ cập nhật động nơi sinh vào DataSource của nó, chú ý là không cho phép trùng lặp. Trong các lần nhập nơi sinh tiếp theo thì nó phải tự động lọc giúp người sử dụng nhập lẹ hơn.
- Ta bắt đầu vào chi tiết ứng dụng.

1- Đây là cấu trúc thư mục trong ứng dụng:

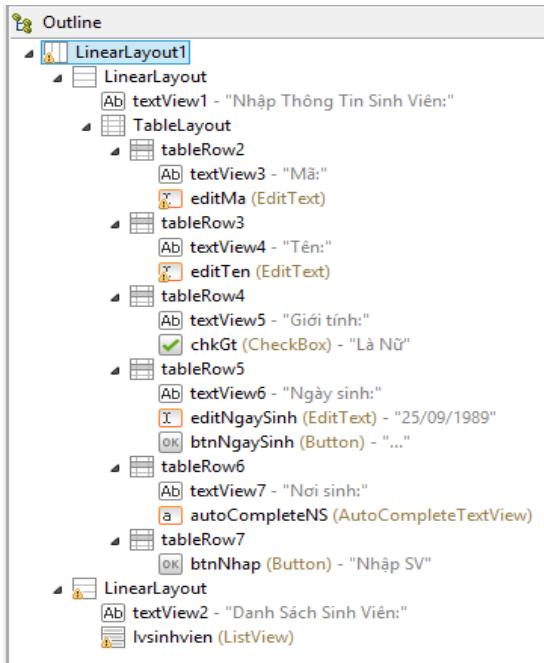


- Nhìn vào cấu trúc thì bạn thấy có 2 Layout, layout chính là **activity_main.xml**, còn **layout sinhvien_item_layout.xml** là Tôi dùng để custom lại ListView (nó sẽ được xử lý trong class**MyArrayAdapter**).

2- Thiết kế giao diện với màn hình nằm ngang (xem hình):



- Dưới đây là Outline **activity_main.xml**:



- Nếu như nhìn vào Outline mà bạn có thể thiết kế được giao diện như trên thì Tôi nghĩ bạn đã thật sự hiểu về Layout, tuy nhiên Tôi vẫn cung cấp Source XML, bạn xem tham khảo

```

1 <LinearLayout
2   xmlns:android="http://schemas.android.com/apk/res/android"
3   xmlns:tools="http://schemas.android.com/tools"
4   android:id="@+id/LinearLayout1"
5   android:layout_width="match_parent"
6   android:layout_height="match_parent"
7   tools:context=".MainActivity" >
8
9 <LinearLayout
10  android:layout_width="wrap_content"
11  android:layout_height="wrap_content"
12  android:background="#FFF0E1"
13  android:orientation="vertical" >
14
15 <TextView
16  android:id="@+id/textView1"
17  android:layout_width="match_parent"
18  android:layout_height="wrap_content"
19  android:background="#008000"
20  android:text="Nhập Thông Tin Sinh Viên:"
21  android:textColor="#FFFFFF" />
22
23 <TableLayout
24  android:layout_width="wrap_content"
25  android:layout_height="wrap_content"
26  >
27
28 <TableRow
29  android:id="@+id/tableRow2"
30  android:layout_width="wrap_content"
31  android:layout_height="wrap_content" >
32

```

```
33 <TextView  
34     android:id="@+id/textView3"  
35     android:layout_width="wrap_content"  
36     android:layout_height="wrap_content"  
37     android:text="Mã:" />  
38  
39 <EditText  
40     android:id="@+id/editMa"  
41     android:layout_width="wrap_content"  
42     android:layout_height="wrap_content"  
43     android:layout_span="2" >  
44  
45     <requestFocus />  
46 </EditText>  
47  
48 </TableRow>  
49  
50 <TableRow  
51     android:id="@+id/tableRow3"  
52     android:layout_width="wrap_content"  
53     android:layout_height="wrap_content" >  
54  
55 <TextView  
56     android:id="@+id/textView4"  
57     android:layout_width="wrap_content"  
58     android:layout_height="wrap_content"  
59     android:text="Tên:" />  
60  
61 <EditText  
62     android:id="@+id/editTen"  
63     android:layout_width="wrap_content"  
64     android:layout_height="wrap_content"  
65     android:layout_span="2" />  
66  
67 </TableRow>  
68  
69 <TableRow  
70     android:id="@+id/tableRow4"  
71     android:layout_width="wrap_content"  
72     android:layout_height="wrap_content" >  
73  
74 <TextView  
75     android:id="@+id/textView5"  
76     android:layout_width="wrap_content"  
77     android:layout_height="wrap_content"  
78     android:text="Giới tính:" />  
79  
80 <CheckBox  
81     android:id="@+id/chkGt"  
82     android:layout_width="wrap_content"  
83     android:layout_height="wrap_content"  
84     android:layout_span="2"  
85     android:text="Là Nữ" />  
86 </TableRow>
```

```
87
88 <TableRow
89     android:id="@+id/tableRow5"
90     android:layout_width="wrap_content"
91     android:layout_height="wrap_content" >
92
93     <TextView
94         android:id="@+id/textView6"
95         android:layout_width="wrap_content"
96         android:layout_height="wrap_content"
97         android:text="Ngày sinh:" />
98
99     <EditText
100        android:id="@+id/editNgaySinh"
101        android:layout_width="wrap_content"
102        android:layout_height="wrap_content"
103        android:inputType="date"
104        android:text="25/09/1989" />
105
106     <Button
107         android:id="@+id	btnNgaySinh"
108         android:layout_width="wrap_content"
109         android:layout_height="wrap_content"
110         android:text="..." />
111
112 </TableRow>
113
114 <TableRow
115     android:id="@+id/tableRow6"
116     android:layout_width="wrap_content"
117     android:layout_height="wrap_content" >
118
119     <TextView
120         android:id="@+id/textView7"
121         android:layout_width="wrap_content"
122         android:layout_height="wrap_content"
123         android:text="Nơi sinh:" />
124
125     <AutoCompleteTextView
126         android:id="@+id/autoCompleteNS"
127         android:layout_width="wrap_content"
128         android:layout_height="wrap_content"
129         android:completionThreshold="1"
130         android:layout_span="2"
131         android:ems="10" />
132
133 </TableRow>
134
135 <TableRow
136     android:id="@+id/tableRow7"
137     android:layout_width="wrap_content"
138     android:layout_height="wrap_content" >
139
140     <Button
141         android:id="@+id	btnNhaph"
```

```

141     android:layout_width="wrap_content"
142     android:layout_height="wrap_content"
143     android:layout_column="1"
144     android:gravity="center"
145     android:text="Nhập SV" />
146
147 </TableRow>
148
149 </TableLayout>
150 </LinearLayout>
151
152 <LinearLayout
153     android:layout_width="wrap_content"
154     android:layout_height="wrap_content"
155     android:layout_weight="1"
156     android:background="#EAEAEA"
157     android:orientation="vertical" >
158
159 <TextView
160     android:id="@+id/textView2"
161     android:layout_width="match_parent"
162     android:layout_height="wrap_content"
163     android:background="#000040"
164     android:text="Danh Sách Sinh Viên:"
165     android:textColor="#FFFFFF" />
166
167 <ListView
168     android:id="@+id/lvsinhvien"
169     android:layout_width="match_parent"
170     android:layout_height="wrap_content"
171     android:layout_weight="1" >
172 </ListView>
173
174 </LinearLayout>
175
176 </LinearLayout>
177

```

- còn đây là của **sinhvien_item_layout.xml**:

```

1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout
3     xmlns:android="http://schemas.android.com/apk/res/android"
4     android:layout_width="match_parent"
5     android:layout_height="match_parent"
6     android:orientation="vertical" >
7
8 <TextView
9     android:id="@+id/txtMaVaTen"
10    android:layout_width="match_parent"
11    android:layout_height="wrap_content"
12    android:textSize="15sp" />
13
14 <TextView

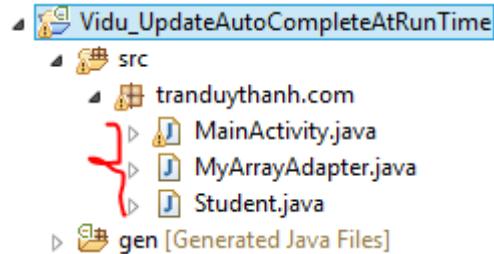
```

```

15     android:id="@+id/txtThongTinKhac"
16     android:layout_width="match_parent"
17     android:layout_height="wrap_content"
18     android:textColor="#800000"
19     android:textSize="10sp"
20     android:textStyle="italic" />
21
</LinearLayout>

```

-3 -Bạn xem cấu trúc class xử lý nghiệp vụ:



- class **Student**:

```

1  package tranduythanh.com;
2
3  import java.util.Date;
4  /**
5   * Class dùng để lưu trữ thông tin của sinh viên
6   * id: Mã
7   * name: Tên
8   * gender: giới tính, true là nữ
9   * birthday: lưu năm sinh
10  * placeOfBirth: nơi sinh
11  * @author drthanh
12  *
13  */
14 public class Student {
15     private String id;
16     private String name;
17     private boolean gender;
18     private Date birthday;
19     private String placeOfBirth;
20     public String getId() {
21         return id;
22     }
23     public void setId(String id) {
24         this.id = id;
25     }
26     public String getName() {
27         return name;
28     }
29     public void setName(String name) {
30         this.name = name;
31     }
32     public boolean isGender() {
33         return gender;
34     }
35 }

```

```

33 }
34 public void setGender(boolean gender) {
35     this.gender = gender;
36 }
37 public Date getBirthday() {
38     return birthday;
39 }
40 public void setBirthday(Date birthday) {
41     this.birthday = birthday;
42 }
43 public String getPlaceOfBirth() {
44     return placeOfBirth;
45 }
46 public void setPlaceOfBirth(String placeOfBirth) {
47     this.placeOfBirth = placeOfBirth;
48 }
49 public Student(String id, String name, boolean gender, Date
50 birthday,
51     String placeOfBirth) {
52     super();
53     this.id = id;
54     this.name = name;
55     this.gender = gender;
56     this.birthday = birthday;
57     this.placeOfBirth = placeOfBirth;
58 }
59 public Student() {
60     super();
61 }
62 }
63

```

- 4 class MyArrayAdapter (class dùng để Custom Listview):

```

1 package tranduythanh.com;
2
3 import java.text.SimpleDateFormat;
4 import java.util.ArrayList;
5 import java.util.Locale;
6
7 import android.app.Activity;
8 import android.view.View;
9 import android.view.ViewGroup;
10 import android.widget.ArrayAdapter;
11 import android.widget.TextView;
12 /**
13 * Đây là class dùng để custom layout
14 * Bạn đã được học trước đó
15 * Class này sẽ lấy layout: sinhvien_item_layout.xml
16 * @author drthanh
17 *
18 */
19 public class MyArrayAdapter extends ArrayAdapter<Student> {
20

```

```

21 Activity context;
22     int resourceId;
23     ArrayList<Student>arrStudent;
24     public MyArrayAdapter(Activity context, int resource,
25     ArrayList<Student> objects) {
26         super(context, resource, objects);
27         this.context=context;
28         this.resourceId=resource;
29         this.arrStudent=objects;
30     }
31     @Override
32     public View getView(int position, View convertView, ViewGroup
33 parent) {
34         if(convertView==null)
35         {
36             //gắn Layout vào Activity
37             convertView= context.getLayoutInflater().inflate(resourceId,
38             null);
39         }
40         //Lấy TextView để lưu mã và tên
41         TextView txtMaTen=(TextView)
42         convertView.findViewById(R.id.txtMaVaTen);
43         //Lấy TextView để lưu giới tính, năm sinh, nơi sinh
44         TextView txtKhac=(TextView)
45         convertView.findViewById(R.id.txtThongTinKhac);
46         //Lấy sinh viên thứ position
47         Student s=arrStudent.get(position);
48         txtMaTen.setText(s.getId()+" - "+s.getName());
49         //Dùng SimpleDateFormat để định dạng ngày tháng dd/MM/YYYY ->
50         22/12/2012
51         SimpleDateFormat dft=new
52         SimpleDateFormat("dd/MM/yyyy", Locale.getDefault());
53         txtKhac.setText((s.isGender())?"Nữ-":"Nam-") +
54         dft.format(s.getBirthday())+" - "+
55         s.getPlaceOfBirth());
56         return convertView;
57     }
58 }
```

-5- class MainActivity để xử lý nghiệp vụ:

```

1 package tranduythanh.com;
2
3 import java.text.ParseException;
4 import java.text.SimpleDateFormat;
5 import java.util.ArrayList;
6 import java.util.Date;
7 import java.util.Locale;
8
9 import android.os.Bundle;
10 import android.app.Activity;
11 import android.app.DatePickerDialog;
12 import android.app.DatePickerDialog.OnDateSetListener;
13 import android.view.Menu;
```

```

14 import android.view.View;
15 import android.view.View.OnClickListener;
16 import android.widget.ArrayAdapter;
17 import android.widget.AutoCompleteTextView;
18 import android.widget.Button;
19 import android.widget.CheckBox;
20 import android.widget.DatePicker;
21 import android.widget.EditText;
22 import android.widget.ListView;
23 import android.widget.Toast;
24
25 public class MainActivity extends Activity {
26
27     ListView lvSinhvien;
28     //Cặp đôi để Custom ListView
29     MyArrayAdapter adapterSinhvien;
30     ArrayList<Student> arrSinhvien=new ArrayList<Student>();
31
32     EditText editMa,editTen,editNamsinh;
33     CheckBox chkGender;
34     Button btnNamsinh,btnNhapsv;
35
36     AutoCompleteTextView autoTextNs;
37     //Cặp đôi để dùng cho AutoCompleteTextView
38     ArrayList<String>arrAuto=new ArrayList<String>();
39     ArrayAdapter<String>adapterAuto;
40
41     @Override
42     protected void onCreate(Bundle savedInstanceState) {
43         super.onCreate(savedInstanceState);
44         setContentView(R.layout.activity_main);
45         getFormWidgets();
46         addEventsForWidgets();
47         //fakeData();
48     }
49     /**
50      * hàm lấy các widgets
51      */
52     public void getFormWidgets()
53     {
54         editMa=(EditText) findViewById(R.id.editMa);
55         editTen=(EditText) findViewById(R.id.editTen);
56         editNamsinh=(EditText) findViewById(R.id.editNgaySinh);
57         chkGender=(CheckBox) findViewById(R.id.chkGt);
58         autoTextNs=(AutoCompleteTextView)
59         findViewById(R.id.autoCompleteNS);
60         btnNamsinh=(Button) findViewById(R.id.btnNgaySinh);
61         btnNhapsv=(Button) findViewById(R.id.btnNhap);
62
63         lvSinhvien=(ListView) findViewById(R.id.lvsinhvien);
64         //Gán DataSource cho Adapter ListView
65         adapterSinhvien=new MyArrayAdapter(this,
66             R.layout.sinhvien_item_layout,
67             arrSinhvien);
68         //Gán Adapter vào ListView

```

```

68     lvSinhvien.setAdapter(adapterSinhvien);
69 }
70 /**
71 * Hàm thiết lập sự kiện cho Button
72 */
73 public void addEventsForWidgets()
74 {
75     btnNamsinh.setOnClickListener(new OnClickListener() {
76
77     @Override
78     public void onClick(View arg0) {
79         // TODO Auto-generated method stub
80         processBirthday();
81     }
82 });
83     btnNhapsv.setOnClickListener(new OnClickListener() {
84
85     @Override
86     public void onClick(View v) {
87         processInput();
88     }
89 });
90 }
91 /**
92 * Hàm hiển thị DatePickerDialog để chọn năm sinh
93 */
94 public void processBirthday()
95 {
96     OnDateSetListener callBack=new OnDateSetListener() {
97
98     @Override
99     public void onDateSet(DatePicker arg0, int arg1, int arg2, int
100     arg3) {
101         editNamsinh.setText(arg3+"/"+arg2+"/"+arg1);
102     }
103 };
104     //Ở đây Tôi chưa xử lý lấy ngày tháng năm trên EditText đưa vào
105     DatePicker
106     //Bạn tự làm
107     DatePickerDialog dateDialog=new DatePickerDialog(this,
108     callBack, 1989, 9, 25);
109     dateDialog.setTitle("Birthday");
110     dateDialog.show();
111 }
112 /**
113 * Hàm xử lý nhập dữ liệu từ giao diện
114 */
115 public void processInput()
116 {
117     String ma=editMa.getText()+"";
118     String ten=editTen.getText()+"";
119     String ns=editNamsinh.getText()+"";
120     String nois=autoTextNs.getText()+"";
121     boolean gt=chkGender.isSelected();
122     SimpleDateFormat dft=new

```

```

122     SimpleDateFormat("dd/MM/yyyy", Locale.getDefault());
123     try {
124         Student s=new Student(ma, ten, gt, dft.parse(ns), nois);
125         arrSinhvien.add(s);
126         //Thêm mới xong thì gọi hàm dưới cập nhập lại giao diện
127         adapterSinhvien.notifyDataSetChanged();
128         //Xử lý cho Autocomplete về nơi sinh
129         processAutoComplete(nois);
130     } catch (ParseException e) {
131         Toast.makeText(this, e.toString(), Toast.LENGTH_LONG).show();
132     }
133 }
134 /**
135 * Hàm xử lý Autocomplete Nơi sinh
136 * @param data
137 */
138 public void processAutoComplete(String data)
139 {
140     //Chạy từ đầu chí cuối nếu trùng thì thoát khỏi hàm
141     for(int i=0;i<arrAuto.size();i++)
142     {
143         String x=arrAuto.get(i);
144         if(x.trim().equalsIgnoreCase(data.trim()))
145             return;
146     }
147     //nếu xuống đây được tức là chưa tồn tại-> đưa vào arrAuto
148     arrAuto.add(data);
149     //Đưa vào ADapter
150     adapterAuto=new ArrayAdapter<String>(this,
151         android.R.layout.simple_list_item_1,
152         arrAuto);
153     //đưa Adapter vào AutoComplete
154     autoTextNs.setAdapter(adapterAuto);
155 }
156 public void fakeData()
157 {
158     Student s1=new Student("1", "Đoàn Ái Nương", true, new
159     Date(1980-1900, 2, 2), "TP. Hồ Chí Minh");
160     Student s2=new Student("2", "Nguyễn Thùy Trang", true, new
161     Date(1982-1900, 3, 3), "Lâm Đồng");
162     Student s3=new Student("3", "Hoàng Văn Phúc", false, new
163     Date(1970-1900, 4, 4), "Hà Nội");
164     Student s4=new Student("4", "Đinh Hồng Lợi", false, new
165     Date(1972-1900, 4, 4), "Bắc Giang");
166     Student s5=new Student("5", "Nguyễn Hoàng Uyên", true, new
167     Date(1970-1900, 4, 4), "Huế");
168     arrSinhvien.add(s1);
169     arrSinhvien.add(s2);
170     arrSinhvien.add(s3);
171     arrSinhvien.add(s4);
172     arrSinhvien.add(s5);
173     adapterSinhvien.notifyDataSetChanged();
174 }
@Override
public boolean onCreateOptionsMenu(Menu menu) {

```

```
// Inflate the menu; this adds items to the action bar if it is
// present.
getMenuInflater().inflate(R.menu.activity_main, menu);
return true;
}
```

- Bạn xem Tôi giải thích trong coding luôn rồi.

- Chú ý là bạn có thể yêu cầu ứng dụng chạy theo màn hình nằm ngang trong code (bạn viết trong **onCreate**):

setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_LANDSCAPE);

- Bạn hãy cố gắng làm bài này vì Tôi cảm thấy nó rất hay về ý tưởng thiết kế và kỹ thuật lập trình.

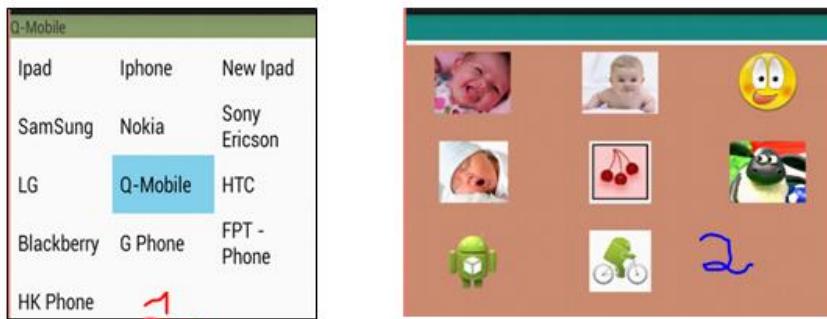
- Bạn có thể Tải code mẫu tại đây: <http://www.mediafire.com/?m0jeop2fb83ib3u>

- Bài tập tiếp theo bạn sẽ được thực hành về **GridView**, một control cũng cần phải lưu tâm.

Bài tập 19: Thực hành Gridview trong Android

- Bài tập này bạn sẽ làm quen với control Gridview, cũng tương tự như ListView. Gridview cũng dựa vào Datasource và ArrayAdapter. ListView bạn làm như thế nào thì Gridview y xì.

- Điểm khác nhau là GridView có thiết lập số cột. Dữ liệu luôn đưa vào dưới dạng hình ông (mảng, list một chiều), nhưng dựa vào số cột ta thiết lập mà nó tự động ngắt hàng, xem hình minh họa:



- Bạn thấy đấy, Tôi có thể hiển thị Text hoặc hình ảnh vào GridView.

- Bạn có thể thiết lập số cột cho GridView theo hình dưới đây:

```
<GridView  
    android:id="@+id/gridView1"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:numColumns="3" >  
</GridView>
```

- Nếu bạn thiết lập **android:numColumns="3"**, Tức là Gridview sẽ ngắt dòng khi đủ 3 phần tử, nó chỉ khác chỗ này, còn việc đưa dữ liệu lên như thế nào thì nó y xì như làm với ListView.

- Ví dụ 1: Hiển thị văn bản lên GridView (xem hình Tôi đánh số 1):

- Bạn tạo một Android Project tên tùy thích, ở đây Tôi đặt tên: **Vidu_Gridview_Text**

- Đây là **activity_main.xml** cho ứng dụng:

```
1  <LinearLayout  
2      xmlns:android="http://schemas.android.com/apk/res/android"  
3      xmlns:tools="http://schemas.android.com/tools"  
4      android:id="@+id/LinearLayout1"  
5      android:layout_width="match_parent"  
6      android:layout_height="match_parent"  
7      android:orientation="vertical"  
8      tools:context=".MainActivity" >  
9      <TextView  
10         android:id="@+id/selection"  
11         android:layout_width="match_parent"  
12         android:layout_height="wrap_content"
```

```

13     android:background="#8A9D6F"
14     android:hint="Selected here" />
15     <GridView
16     android:id="@+id/gridView1"
17     android:layout_width="match_parent"
18     android:layout_height="wrap_content"
19     android:numColumns="3" >
20   </GridView>
</LinearLayout>

```

- Bạn xem dòng 15 là id của GridView, Tôi để mặc định **gridView1**

- Dòng 18 có thuộc tính: **android:numColumns= “3”**, tức là dữ liệu sẽ được hiển thị trong Gridview với định dạng 3 cột.

- Tiếp theo bạn xem MainActivity.java:

```

1  package tranduythanh.com;
2  import android.os.Bundle;
3  import android.app.Activity;
4  import android.view.View;
5  import android.widget.AdapterView;
6  import android.widget.ArrayAdapter;
7  import android.widget.GridView;
8  import android.widget.TextView;
9
10 public class MainActivity extends Activity {
11
12 //Dùng mảng 1 chiều hoặc ArrayList để lưu một số dữ liệu
13 String arr[]={ "Ipad", "Iphone", "New Ipad",
14 "SamSung", "Nokia", "Sony Ericson",
15 "LG", "Q-Mobile", "HTC", "Blackberry",
16 "G Phone", "FPT - Phone", "HK Phone"
17 };
18 protected void onCreate(Bundle savedInstanceState) {
19     super.onCreate(savedInstanceState);
20     setContentView(R.layout.activity_main);
21     //Tôi tượng này dùng để hiển thị phần tử được chọn trong
22     //GridView
23     final TextView selection=(TextView)
24     findViewById(R.id.selection);
25     final GridView gv=(GridView) findViewById(R.id.gridView1);
26     //Gán DataSource vào ArrayAdapter
27     ArrayAdapter<String>da=new ArrayAdapter<String>
28     (
29     this,
30     android.R.layout.simple_list_item_1,
31     arr
32 );
33     //gán Datasource vào GridView
34     gv.setAdapter(da);
35     //Thiết lập sự kiện cho GridView,
36     gv.setOnItemClickListener(new AdapterView
37     .OnItemClickListener() {

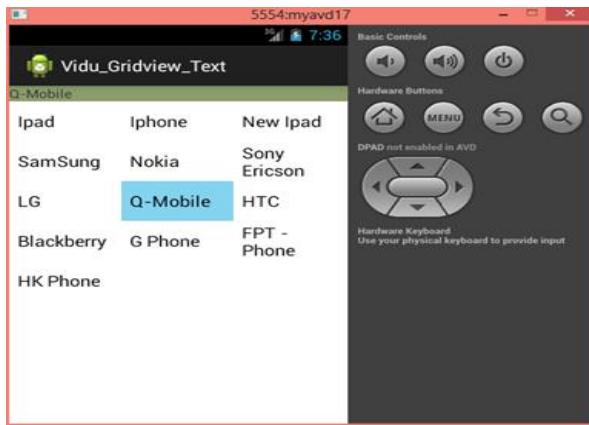
```

```

37     public void onItemClick(AdapterView<?> arg0,
38         View arg1, int arg2,
39         long arg3) {
40             //Hiển thị phần tử được chọn trong GridView lên TextView
41             selection.setText(arr[arg2]);
42         }
43     );
44 }
45 }
```

- Bạn thấy đó, cách sử dụng GridView là y xì như ListView, nên nếu như bạn đã rành về ListView rồi thì GridView hiển nhiên bạn cũng làm tốt.

- Thực thi chương trình bạn sẽ thấy như hình bên dưới:

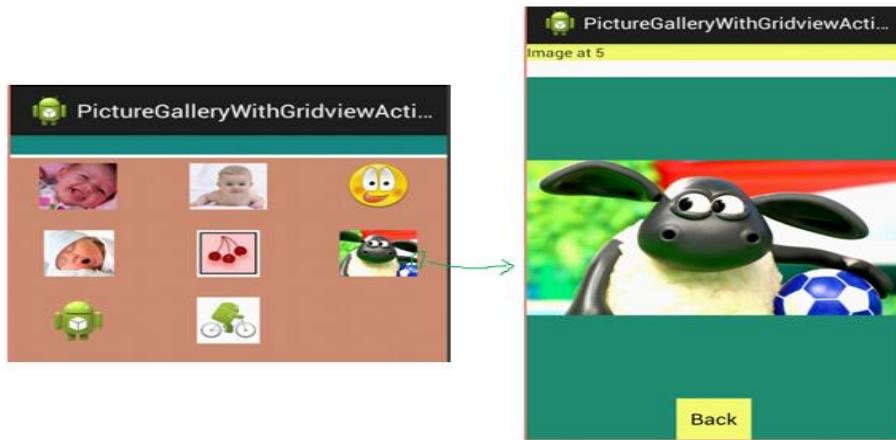


- Xem coding đầy đủ ở đây: <http://www.mediafire.com/?v3ww92ys1s5jth0>

- Ví dụ 2 sẽ phức tạp hơn, hiển thị danh sách hình ảnh có sẵn lên GridView, mỗi lần chọn vào hình ảnh nào thì sẽ hiển thị chi tiết ảnh đó trên một màn hình mới:

- Có thể Demo này đã có nhiều website và Ebooks làm rồi, ở đây Tôi cũng muốn demo lại cho các bạn.

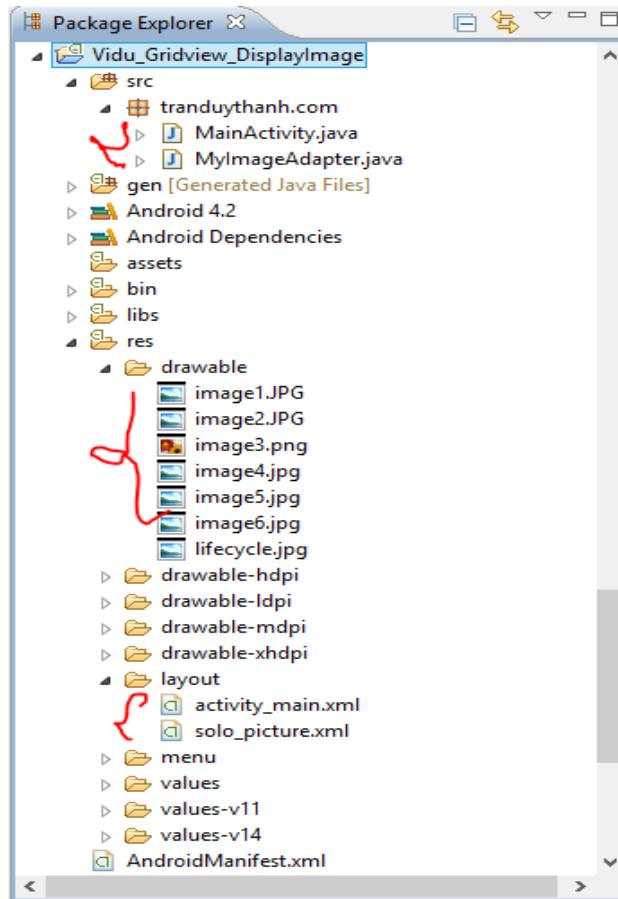
- Bạn xem giao diện của ứng dụng trước:



- Bên trái là màn hình chính cho phép hiển thị danh sách hình ảnh vào GridView, mỗi lần chọn vào từng hình trong GridView thì sẽ mở hình được chọn đó vào một màn hình mới (ví dụ như khi chọn hình chú Cừu thì nó sẽ hiển thị ra như màn hình bên phải), nhấn nút Back để trở về màn hình chính.

- Ở đây có một sự khác biệt lớn đó là chúng ta **chỉ sử dụng 1 MainActivity**, không hề tạo thêm bất kỳ một Activity nào khác, chúng ta chỉ thay đổi Layout mà thôi. Nên nó cũng là điểm nhấn của ứng dụng.

- Hãy tạo một Android Project tên: **Vidu_Gridview_DisplayImage** và xem cấu trúc của chương trình:



- Layout sẽ có 2 cái: **activity_main.xml** là của màn hình chính dùng để hiển thị danh sách hình ảnh.**solo_picture.xml** là dùng để hiển thị từng hình riêng lẻ.

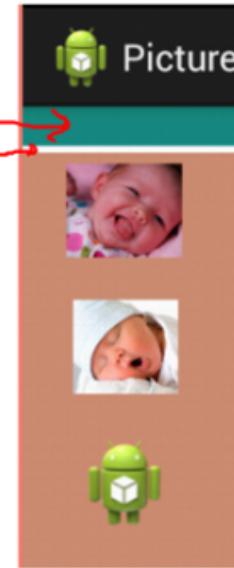
- Tạo thêm thư mục **drawable** và kéo thả một số hình ảnh vào.
- Phân class có 2 class: MainActivity và MyImageAdapter kế thừa từ BaseAdapter.
- Bây giờ ta đi vào chi tiết của từng cái:

- activity_main.xml:

```

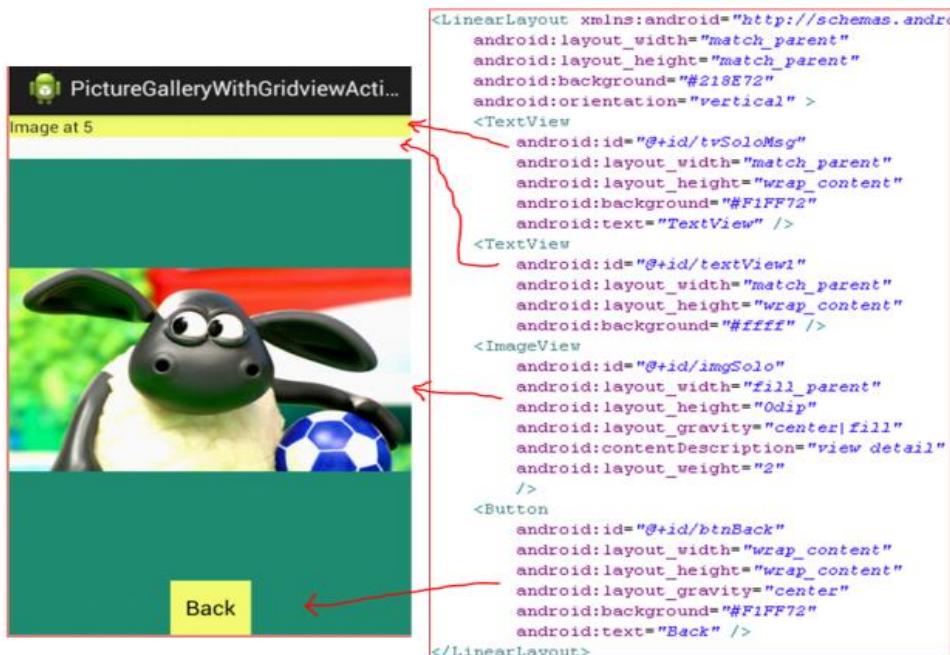
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/LinearLayout1"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context=".PictureGalleryWithGridviewActivity" >
    <TextView
        android:id="@+id/tvMsg"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:background="#1F8B82" />
    <TextView
        android:id="@+id/textView1"
        android:layout_width="match_parent"
        android:layout_height="3dp" />
    <GridView
        android:id="@+id/gridView1"
        android:layout_width="fill_parent"
        android:layout_height="fill_parent"
        android:verticalSpacing="10dp"
        android:horizontalSpacing="10dp"
        android:columnWidth="90dp"
        android:stretchMode="columnWidth"
        android:gravity="center"
        android:background="#CF8E72"
        android:numColumns="auto_fit" >
    </GridView>
</LinearLayout>

```

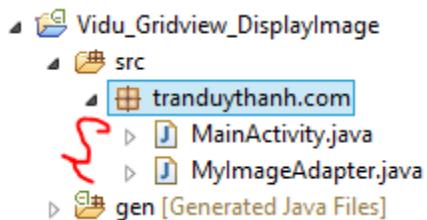


- Bạn có thể nhìn vào hình trên để làm hoặc tải coding mẫu ở bên dưới.

solo_picture.xml:



- Bây giờ ta vào các class xử lý nghiệp vụ:



- Thứ nhất là class **MyImageAdapter**:

+ class này sẽ kế thừa từ BaseAdapter, và dùng để hiển thị từng hình ảnh riêng lẻ:

```
1  package tranduythanh.com;
2
3  import android.content.Context;
4  import android.view.View;
5  import android.view.ViewGroup;
6  import android.widget.BaseAdapter;
7  import android.widget.GridView;
8  import android.widget.ImageView;
9  /**
10   * Class dùng để hiển thị từng hình ảnh riêng lẻ
11   * @author drthanh
12   *
13   */
14  public class MyImageAdapter extends BaseAdapter {
15      private Context mContext;
16      private Integer []mThumbIds;
17      public MyImageAdapter(Context c) {
18          mContext=c;
19      }
20      public MyImageAdapter(Context c, Integer []arrIds) {
21          mContext=c;
22          mThumbIds=arrIds;
23      }
24      public int getCount () {
25      }
26      return mThumbIds.length;
27  }
28  public Object getItem (int arg0) {
29  }
30  return null;
31  }
32  public long getItemId (int arg0) {
33  }
34  return 0;
35  }
36  /**
37   * Cần override lại hàm này để hiển thị hình ảnh
38   */
39  public View getView (int arg0, View convertView, ViewGroup arg2) {
40
41      ImageView imgView;
42      if(convertView==null){
43          imgView=new ImageView(mContext);
44          //cần chỉnh lại hình cho đẹp
45          imgView.setLayoutParams(new GridView.LayoutParams(85, 85));
46      }
47  }
```

```

46     imgView.setScaleType(ImageView.ScaleType.CENTER_CROP);
47     imgView.setPadding(8, 8, 8, 8);
48 }else{
49     imgView=(ImageView) convertView;
50 }
51 //lấy đúng vị trí hình ảnh được chọn
52 //gán lại ImageResource
53 imgView.setImageResource(mThumbIds[arg0]);
54 return imgView;
55 }
}

```

- Thứ hai là class **MainActivity**:

```

1  package tranduythanh.com;
2  import android.os.Bundle;
3  import android.app.Activity;
4  import android.view.View;
5  import android.widget.AdapterView;
6  import android.widget.Button;
7  import android.widget.GridView;
8  import android.widget.ImageView;
9  import android.widget.TextView;
10 import android.widget.AdapterView.OnItemClickListener;
11
12 public class MainActivity extends Activity
13     implements OnItemClickListener
14 {
15     TextView tvMsg;
16     GridView gv;
17     TextView tvSoloMsg;
18     //mảng lưu danh sách các id hình ảnh có sẵn
19     Integer[] mThumbIds;
20     //Adapter cho GridView
21     MyImageAdapter adapter=null;
22     //Vì không tạo thêm 1 Activity nên lấy luôn 2 Id ở bên
23     solo_picture.xml
24     ImageView ivSoloPicture;
25     Button btnBack;
26     //Lưu Bundle backup cho MainActivity
27     Bundle myBackupBundle;
28     protected void onCreate(Bundle savedInstanceState) {
29         super.onCreate(savedInstanceState);
30         //Lưu savedInstanceState trước vào myBackupBundle
31         myBackupBundle=savedInstanceState;
32         setContentView(R.layout.activity_main);
33         tvMsg=(TextView) findViewById(R.id.tvMsg);
34         //gán mảng các Id hình ảnh cho mThumbIds
35         mThumbIds=new
36         Integer[]{R.drawable.image1,R.drawable.image2,
37         R.drawable.image3,R.drawable.image4,
38         R.drawable.image5,R.drawable.image6,
39         R.drawable.ic_launcher,R.drawable.lifecycle};
40         gv=(GridView) findViewById(R.id.gridView1);
41         //thiết lập Datasource cho Adapter

```

```

41     adapter=new MyImageAdapter(this, mThumbIds);
42     //gán Adapter vào Gridview
43     gv.setAdapter(adapter);
44     //thiết lập sự kiện để mở từng hình ảnh chi tiết
45     gv.setOnItemClickListener(this);
46   }
47   public void onItemClick(AdapterView<?> arg0,
48   View arg1, int arg2, long arg3) {
49     //gọi hàm xem hình ảnh chi tiết tại vị trí thứ arg2
50     showdetail(arg2);
51   }
52   public void showdetail(int posistion)
53   {
54     //Không mở Activity mới mà chỉ thiết lập lại Layout
55     setContentView(R.layout.solo_picture);
56     //Vừa gọi hàm trên thì màn hình sẽ thay đổi qua cái mới
57     //ta lấy các control trong layout mới này
58     tvSoloMsg=(TextView) findViewById(R.id.tvSoloMsg);
59     tvSoloMsg.setText("Image at "+posistion);
60     ivSoloPicture=(ImageView) findViewById(R.id.imgSolo);
61     //thiết lập hình ảnh đang chọn lên ImageView mới
62     ivSoloPicture.setImageResource(mThumbIds[posistion]);
63     btnBack=(Button) findViewById(R.id.btnBack);
64     //Thiết lập sự kiện click Back để phục hồi lại
65     MainActivity
66     //bằng cách gọi lại onCreate(myBackupBundle);
67     btnBack.setOnClickListener(new View.OnClickListener() {
68       public void onClick(View arg0) {
69         onCreate(myBackupBundle);
70       }
71     });
72   }
}

```

- Khi chạy ứng dụng bạn sẽ có kết quả như mong muốn.

- Bài tập này bạn đã biết cách sử dụng GridView, biết cách đưa hình ảnh vào GridView, đặc biệt biết thêm một kỹ thuật mới đó là thay đổi Layout để đổi màn hình không cần chạy Activity.

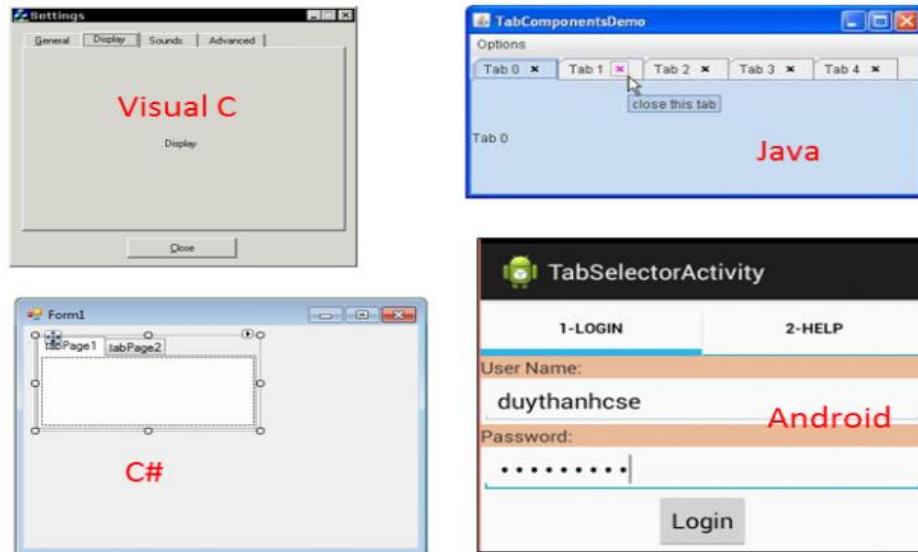
- Ngoài ra còn biết thêm về BaseAdapter.

- Bạn có thể tải code mẫu đầy đủ ở đây: <http://www.mediafire.com/?3o2jva4mzgp2kj8>

- Bài tập tiếp theo bạn sẽ được thực hành về **TimePickerDialog** và **DatePickerDialog**, 2 control này cũng đáng phải lưu tâm vì nó cũng được sử dụng thường xuyên trong Android.

Bài tập 21: Thực hành về Tab Selector trong Android

Tab Selector giống như Property Sheet trong Visual C, giống như Tab Control C#, hay trong Java:

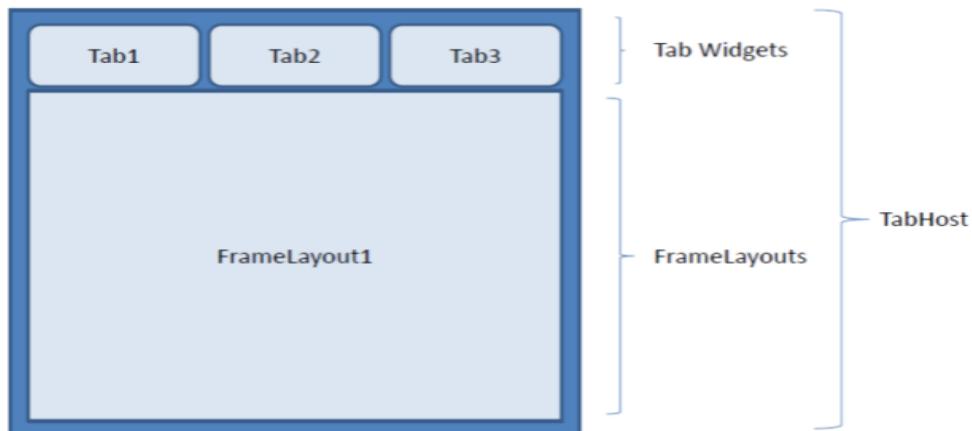


- Tôi sẽ làm một ví dụ cụ thể về Tab Selector để bạn có thể thực hành lại và hiểu được vấn đề.

- Đối với Android, mỗi Tab bạn nên thiết kế trên một Layout khác nhau, rồi trong Main Layout bạn include các tab đó vào (Tức là nếu như ứng dụng bạn có 3 Tab con thì sẽ tạo 3 Layout khác nhau rồi include chúng vào Main layout, chứ đừng thiết kế tất tần tật trong một Main Layout nó sẽ gây khó khăn trong việc sửa lỗi).

- Tôi trình bày sơ qua lý thuyết về Tab selector:

+ Tab selector gồm có 3 phần: **Tab Host**, **Tab Widgets** và **FrameLayout**.

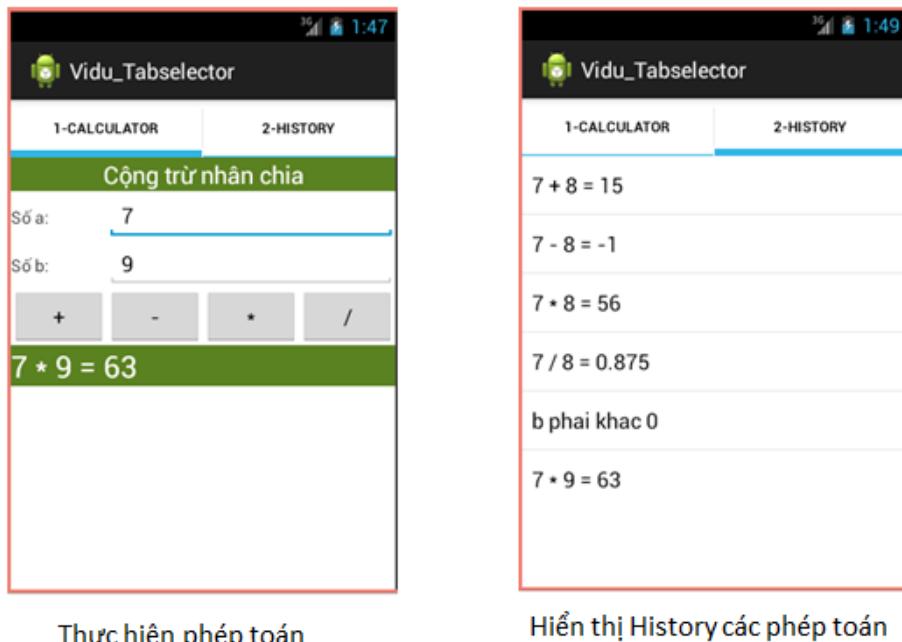


+ **Tab Host:** Là Container chính chứa các Tab buttons và Tab contents

+Tab Widget: Để định dạng cho các Tab buttons : Nhãn, Icon...

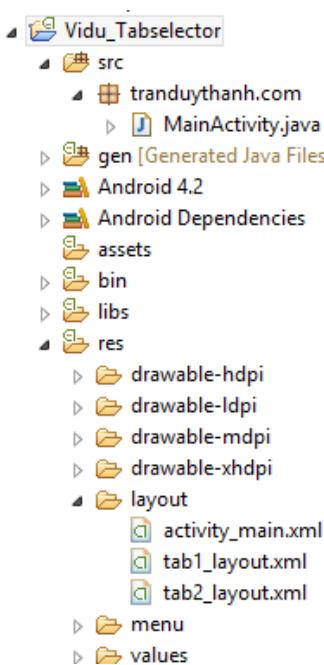
+FrameLayout: là Container để chứa các layout cho Tab contents, ta chỉ có thể dùng FrameLayout cho Tab contents, không thể dùng các loại Layout khác. Nếu bạn thắc mắc tại vì sao lại là FrameLayout mà không phải là các Layout khác? thì Tôi chỉ nói đơn giản thế này: Cho dù bạn có nhấn vào các Tab nào đi nữa thì layout tương ứng với mỗi Tab mà bạn vừa nhấn vào cũng chỉ xuất hiện cùng một chỗ trên màn hình điện thoại, điều này chỉ có FrameLayout mới giải quyết được.

*** Nay bạn hãy xem hình minh họa về giao diện trong bài ví dụ Tab Selector của Tôi như sau:



- Như hình trên thì bạn thấy đó: Tab đầu tiên “1-CALCULATOR” là giao diện cho phép tính cộng trừ nhân chia, Tab thứ 2 “2-HISTORY” dùng để hiển thị danh sách các phép toán đã thực hiện.

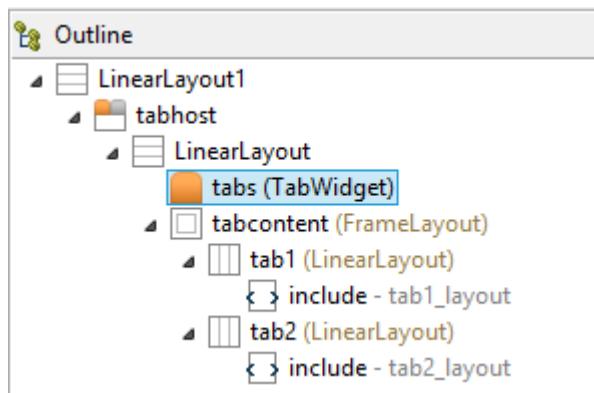
- Bạn xem cấu trúc tổng quan của ứng dụng:



- Vì ứng dụng của Tôi có 2 Tab nên Tôi sẽ tạo 2 tabs: **tab1_layout.xml** và **tab2_layout.xml**, 2 tabs này sẽ được include vào main layout **activity_main.xml**, vậy tổng cộng Tôi có 3 Layout.

- Ta vào xem main layout (**activity_main.xml**):

- xem Outline để dễ tưởng tượng:



-Còn đây là source XML:

```
1 <LinearLayout  
2     xmlns:android="http://schemas.android.com/apk/res/android"  
3     xmlns:tools="http://schemas.android.com/tools"  
4     android:id="@+id/LinearLayout1"  
5     android:layout_width="match_parent"  
6     android:layout_height="match_parent"  
7     android:orientation="vertical"  
8     tools:context=".MainActivity" >  
9  
10    <TabHost  
11        android:id="@+id/tabhost"  
12        android:layout_width="match_parent"  
13        android:layout_height="match_parent" >  
14        <LinearLayout  
15            android:layout_width="match_parent"  
16            android:layout_height="match_parent"  
17            android:orientation="vertical" >  
18            <TabWidget  
19                android:id="@+id/tabs"  
20                android:layout_width="match_parent"  
21                android:layout_height="wrap_content" >  
22            </TabWidget>  
23            <FrameLayout  
24                android:id="@+id/tabcontent"  
25                android:layout_width="match_parent"  
26                android:layout_height="match_parent" >  
27                <LinearLayout  
28                    android:id="@+id/tab1"  
29                    android:layout_width="match_parent"  
30                    android:layout_height="match_parent">  
31                    <include layout="@layout/tab1_layout"/>  
32                </LinearLayout>  
33            <LinearLayout
```

```

33     android:id="@+id/tab2"
34     android:layout_width="match_parent"
35     android:layout_height="match_parent" >
36     <include layout="@layout/tab2_layout"/>
37   </LinearLayout>
38   </FrameLayout>
39 </LinearLayout>
40 </TabHost>
41
42 </LinearLayout>
```

- Bạn nhìn vào dòng lệnh 30 và 36:

```
<include layout="@layout/tab1_layout"/>
```

```
<include layout="@layout/tab2_layout"/>
```

Đó chính là cách include một layout này vào trong một layout khác.

- Tiếp tục ta xem **tab1_layout.xml**, Tôi lấy lại bài tập trước về cộng trừ nhân chia:

```

1  <TableLayout
2   xmlns:android="http://schemas.android.com/apk/res/android"
3   xmlns:tools="http://schemas.android.com/tools"
4   android:id="@+id/TableLayout1"
5   android:layout_width="match_parent"
6   android:layout_height="match_parent"
7   android:stretchColumns="*" >
8
9  <TableRow
10   android:id="@+id/tableRow1"
11   android:layout_width="wrap_content"
12   android:layout_height="wrap_content" >
13
14  <TextView
15   android:id="@+id/textView1"
16   android:layout_width="wrap_content"
17   android:layout_height="wrap_content"
18   android:layout_span="4"
19   android:background="#5B8020"
20   android:gravity="center"
21   android:text="Cộng trừ nhân chia"
22   android:textColor="#FFFFFF"
23   android:textSize="20sp" />
24 </TableRow>
25
26  <TableRow
27   android:id="@+id/tableRow2"
28   android:layout_width="wrap_content"
29   android:layout_height="wrap_content" >
30
31  <TextView
32   android:id="@+id/textView2"
```

```
33     android:layout_width="wrap_content"
34     android:layout_height="wrap_content"
35     android:text="Số a:"/>
36
37 <EditText
38     android:id="@+id/editsoa"
39     android:layout_width="wrap_content"
40     android:layout_height="wrap_content"
41     android:layout_span="3"
42     android:ems="10"
43     android:inputType="number" >
44
45     <requestFocus />
46 </EditText>
47 </TableRow>
48
49 <TableRow
50     android:id="@+id/tableRow3"
51     android:layout_width="wrap_content"
52     android:layout_height="wrap_content" >
53
54     <TextView
55         android:id="@+id/textView3"
56         android:layout_width="wrap_content"
57         android:layout_height="wrap_content"
58         android:text="Số b:" />
59
60     <EditText
61         android:id="@+id/editsoob"
62         android:layout_width="wrap_content"
63         android:layout_height="wrap_content"
64         android:layout_span="3"
65         android:ems="10"
66         android:inputType="number" />
67 </TableRow>
68
69 <TableRow
70     android:id="@+id/tableRow4"
71     android:layout_width="wrap_content"
72     android:layout_height="wrap_content" >
73
74     <Button
75         android:id="@+id/btncong"
76         android:layout_width="wrap_content"
77         android:layout_height="wrap_content"
78         android:text="+" />
79
80     <Button
81         android:id="@+id/btntru"
82         android:layout_width="wrap_content"
83         android:layout_height="wrap_content"
84         android:text="-" />
85
86     <Button
87         android:id="@+id/btnnhan"
```

```

87     android:layout_width="wrap_content"
88     android:layout_height="wrap_content"
89     android:text="*" />
90
91 <Button
92     android:id="@+id/btnchia"
93     android:layout_width="wrap_content"
94     android:layout_height="wrap_content"
95     android:text="/" />
96 </TableRow>
97
98 <TableRow
99     android:id="@+id/tableRow5"
100    android:layout_width="wrap_content"
101    android:layout_height="wrap_content" >
102
103 <TextView
104     android:id="@+id/txtketqua"
105     android:layout_width="wrap_content"
106     android:layout_height="wrap_content"
107     android:layout_span="4"
108     android:background="#5B8020"
109     android:textColor="#FFFFFF"
110     android:textSize="25sp" />
111 </TableRow>
112
113 </TableLayout>

```

- Và xem tiếp **tab2_layout.xml**, đơn giản là chỉ có 1 ListView chứa danh sách các phép toán đã thực hiện bên Tab1:

```

1   <?xml version="1.0" encoding="utf-8"?>
2   <LinearLayout
3       xmlns:android="http://schemas.android.com/apk/res/android"
4       android:layout_width="match_parent"
5       android:layout_height="match_parent"
6       android:orientation="vertical" >
7
8   <ListView
9       android:id="@+id/lvhistory"
10      android:layout_width="match_parent"
11      android:layout_height="wrap_content" >
12
13 </LinearLayout>

```

- Giờ bạn xem **MainActivity** để biết được cách cấu hình Tabhost:

```

1 package tranduythanh.com;
2
3 import java.util.ArrayList;
4
5 import android.os.Bundle;

```

```

6 import android.app.Activity;
7 import android.view.View;
8 import android.view.View.OnClickListener;
9 import android.widget.ArrayAdapter;
10 import android.widget.Button;
11 import android.widget.EditText;
12 import android.widget.ListView;
13 import android.widget.TabHost;
14 import android.widget.TextView;
15 import android.widget.Toast;
16
17 public class MainActivity extends Activity {
18     //Enum để thực hiện phép toán
19     enum Operator {
20         {
21             Cong, //phép cộng
22             Tru, //phép trừ
23             Nhan, //phép nhân
24             Chia //phép chia
25         }
26         Button btncong,btntru,btnnhan,btnchia;
27         EditText editsoa,editsob;
28         TextView txtkq;
29         ListView lvHistory;
30         ArrayList<String>array_operator=new ArrayList<String>();
31         ArrayAdapter<String>adapter=null;
32         //Variable in listener
33         OnClickListener myclick=new OnClickListener() {
34
35             @Override
36             public void onClick(View arg0) {
37                 switch(arg0.getId())
38                 {
39                     case R.id.btncong:
40                         {
41                             processOperator(Operator.Cong);
42                         }
43                         break;
44                     case R.id.btntru:
45                         {
46                             processOperator(Operator.Tru);
47                         }
48                         break;
49                     case R.id.btnnhan:
50                         {
51                             processOperator(Operator.Nhan);
52                         }
53                         break;
54                     case R.id.btnchia:
55                         {
56                             processOperator(Operator.Chia);
57                         }
58                 }
59             };

```

```

60  /**
61  * Hàm xử lý phép toán theo operator
62  * @param op
63  */
64  public void processOperator(Operator op)
65  {
66  String sa=editsoa.getText()+"";
67  String sb=edit sob.getText().toString();
68  int a=Integer.parseInt(sa);
69  int b=Integer.parseInt(sb);
70  String kq="";
71  switch(op)
72  {
73  case Cong:
74  kq=a+" + "+b +" = "+(a+b);
75  break;
76  case Tru:
77  kq=a+" - "+b +" = "+(a-b);
78  break;
79  case Nhan:
80  kq=a+" * "+b +" = "+(a*b);
81  break;
82  case Chia:
83  if(b!=0)
84  kq=a+" / "+b +" = "+(a*1.0/b);
85  else
86  kq="b phai khac 0";
87  break;
88  default:
89  kq="Invalid operator!";
90  }
91  txtkq.setText(kq);
92  array_operator.add(kq);
93  adapter.notifyDataSetChanged();
94  }
95  @Override
96  protected void onCreate(Bundle savedInstanceState) {
97  super.onCreate(savedInstanceState);
98  setContentView(R.layout.activity_main);
99  loadTabs();
100 }
101 //Cấu hình tab
102 public void loadTabs()
103 {
104 //Lấy Tabhost id ra trước (cái này của built - in android
105 final TabHost tab=(TabHost) findViewById
106 (android.R.id.tabhost);
107 //gọi lệnh setup
108 tab.setup();
109 TabHost.TabSpec spec;
110 //Tạo tab1
111 spec=tab.newTabSpec("t1");
112 spec.setContent(R.id.tab1);
113 spec.setIndicator("1-Calculator");

```

```

114    tab.addTab(spec);
115    //Tạo tab2
116    spec=tab.newTabSpec("t2");
117    spec.setContent(R.id.tab2);
118    spec.setIndicator("2-History");
119    tab.addTab(spec);
120    //Thiết lập tab mặc định được chọn ban đầu là tab 0
121    tab.setCurrentTab(0);
122    //Ở đây Tôi để sự kiện này để các bạn tùy xử lý
123    //Ví dụ tab1 chưa nhập thông tin xong mà lại qua tab 2 thì
124    báo...
125    tab.setOnTabChangedListener(new
126        TabHost.OnTabChangeListener() {
127            public void onTabChanged(String arg0) {
128                String s="Tab tag ="+arg0 +" index ="+
129                tab.getCurrentTab();
130                Toast.makeText(getApplicationContext(),
131                s, Toast.LENGTH_LONG).show();
132            }
133        }
134    //Khởi tạo các đối tượng và gán ADapter cho ListView
135    public void doFormWidgets()
136    {
137        btncong=(Button) findViewById(R.id.btncong);
138        btntru=(Button) findViewById(R.id.btntru);
139        btnnhan=(Button) findViewById(R.id.bnhan);
140        btnchia=(Button) findViewById(R.id.btnchia);
141        editsoa=(EditText) findViewById(R.id.editsoa);
142        editsob=(EditText) findViewById(R.id.editsob);
143        txtkq=(TextView) findViewById(R.id.txtketqua);
144        lvHistory=(ListView) findViewById(R.id.lvhistory);
145        btncong.setOnClickListener(myclick);
146        btntru.setOnClickListener(myclick);
147        bttnhan.setOnClickListener(myclick);
148        btnchia.setOnClickListener(myclick);
149        adapter=new ArrayAdapter<String>
150            (this,
151            android.R.layout.simple_list_item_1,
152            array_operator);
153        lvHistory.setAdapter(adapter);
154    }
155
156

```

- Tôi có giải thích sơ qua trong coding cách tạo tab, bạn hãy cố gắng làm lại để hiểu nó.
- Bao nhiêu tab không quan trọng, cuối cùng chỉ cần đưa về xử lý bình thường như chỉ có 1 màn hình.
- Bạn có thể tải coding mẫu ở đây: <http://www.mediafire.com/?0xyr6ooi066dhiw>
- Bài tập kế tiếp bạn sẽ học về **Menu** và **Context menu**, 2 control này rất ưa chuộng nên ta phải biết nó.

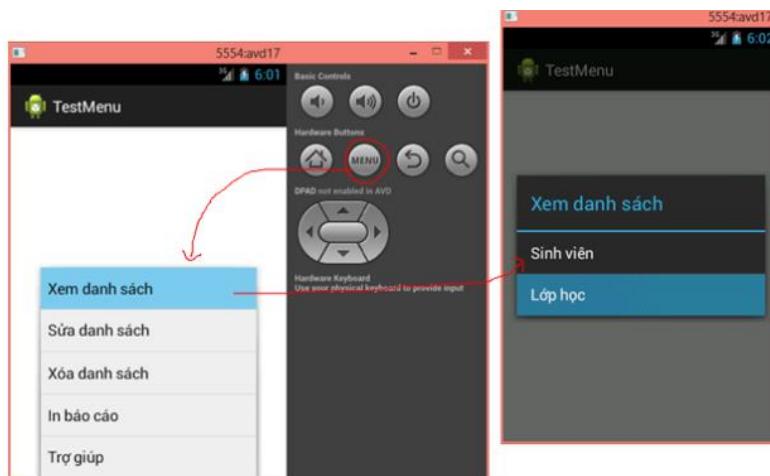
Bài tập 22: Thực hành về Menu trong Android

- Menu và Context menu là những control rất hữu dụng và nó rất được ưa chuộng, do đó chúng ta phải biết về nó để làm cho ứng dụng của Ta đạt hiệu suất cao hơn.

- Trước tiên Tôi sẽ trình bày về Menu:

+ Tùy vào từng phiên bản Android hay sản phẩm của hãng sản xuất mà chức năng hiển thị menu có khác nhau chút xíu.

+ Ở đây Tôi nói máy ảo thôi, để hiển thị Menu thì ta làm như sau:



- Nếu ứng dụng của bạn có sử dụng Menu thì chỉ cần click vào nút có chữ “Menu” là sẽ hiển thị ra.

0) Menu tự động phát sinh khi chúng ta tạo một Android Project:

- Bạn để ý là mỗi lần tạo một Android Project, nó sẽ tự tạo cho ta XML menu Resource đồng thời trong MainActivity cũng gọi sẵn hàm cho chúng ta để load Menu:



- Như bạn thấy thì xml resource menu (kiến trúc bên trong nó ta sẽ tính sau) của ứng dụng được lưu trong thư mục **menu**, và mặc nhiên nó cùng tên với Layout.

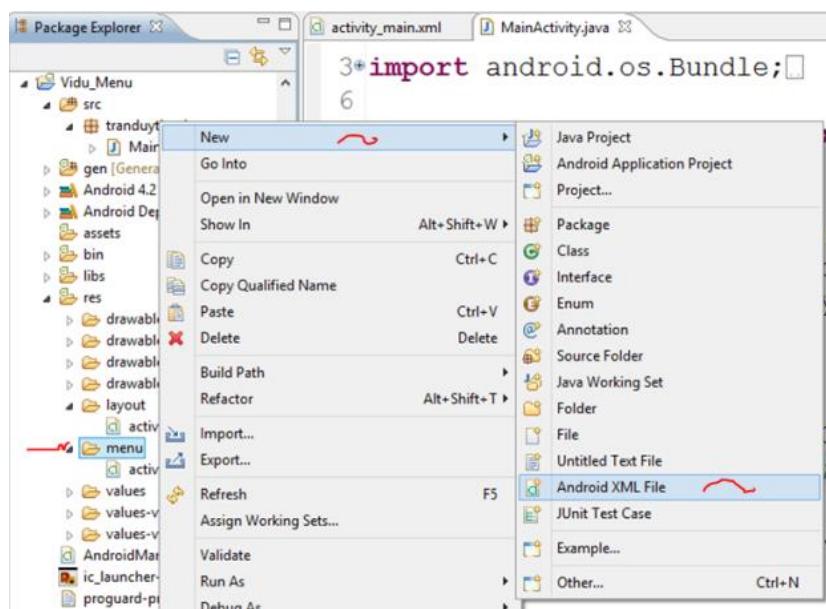
- Bạn mở **MainActivity** lên: Thấy nó override phương thức **onCreateOptionsMenu**, dòng lệnh **18, 19** của Hàm này được dùng để tải Menu vào ứng dụng khi người dùng click vào nút “Menu”.

1) Cách tạo Menu:

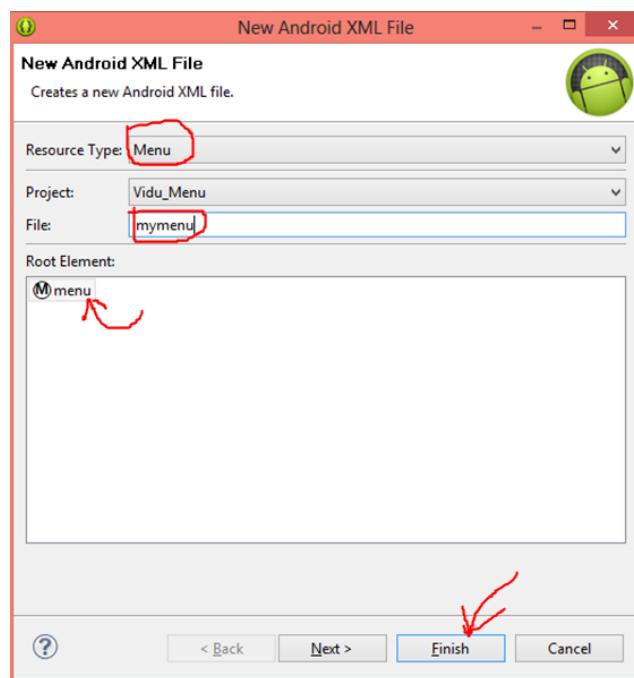
a) Tạo Menu bằng XML resource:

- Giờ tôi sẽ cố tình hướng dẫn bạn tạo 1 XML menu resource khác đồng thời giúp bạn tải Menu Resource mới này vào ứng dụng khi nhấn nút “Menu” cũng như viết sự kiện cho các Menu Item:

+ Bấm chuột phải vào thư mục **menu**/ chọn **New / chọn Android XML File**:



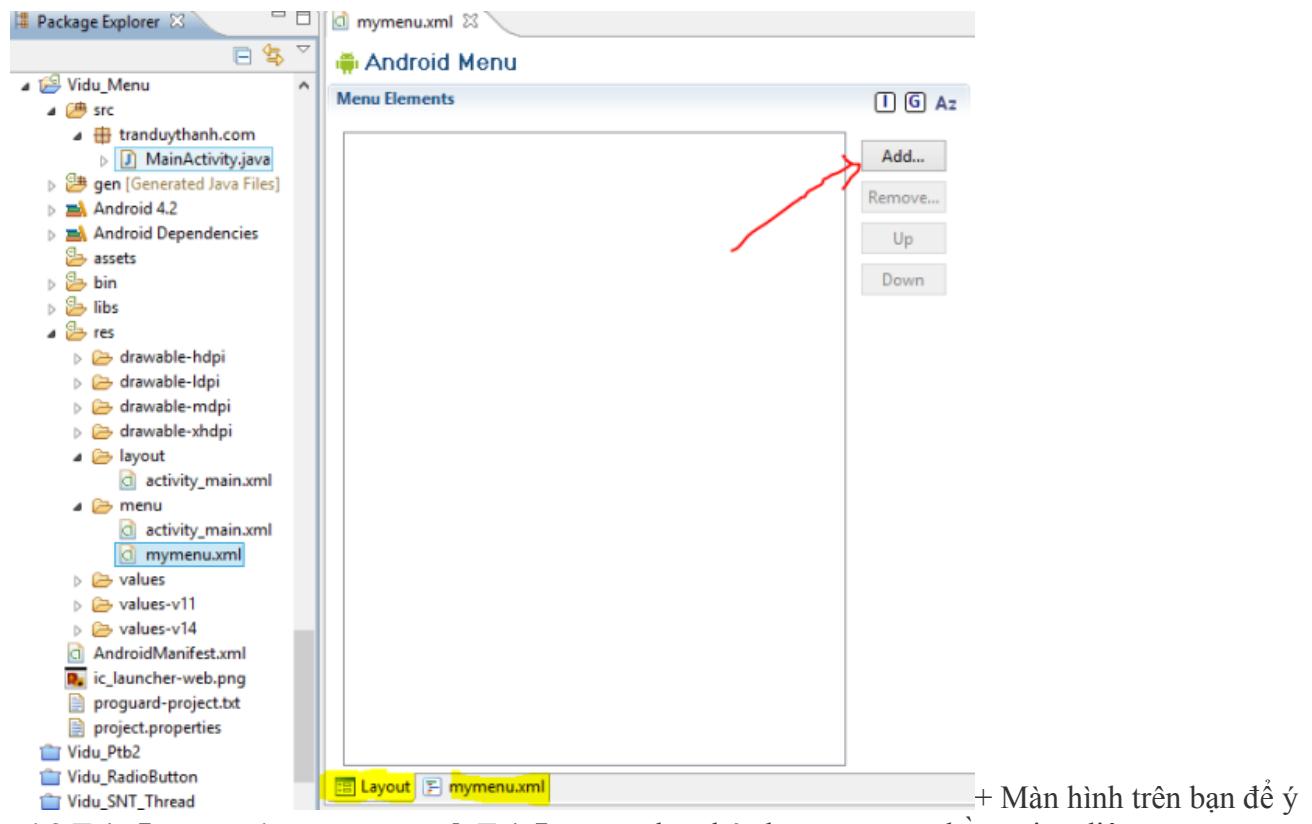
+Màn hình New Android XML hiện lên, ta chọn **Resource type** là **Menu** giống hình dưới đây, đặt tên cho nó là “**mymenu**” rồi nhấn **Finish**:



+ Sau khi nhấn Finish, bạn quan sát lại thư mục **menu** của ứng dụng sẽ xuất hiện **mymenu.xml**:



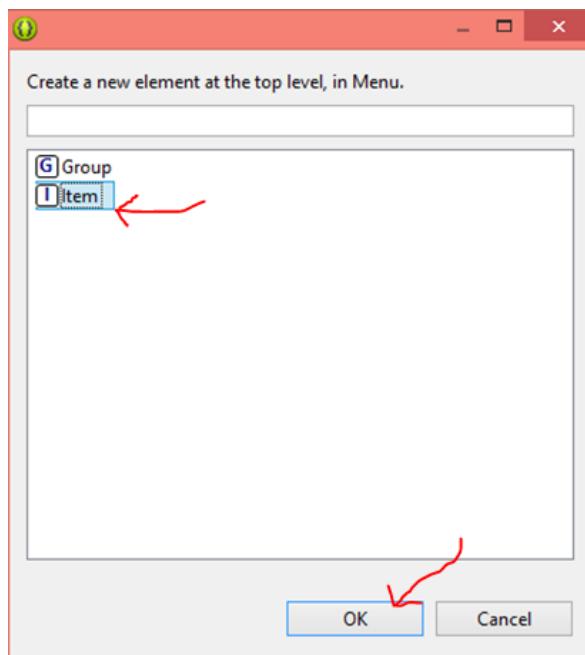
+ Để tạo các Menu Item, bạn double click vào **mymenu.xml**:



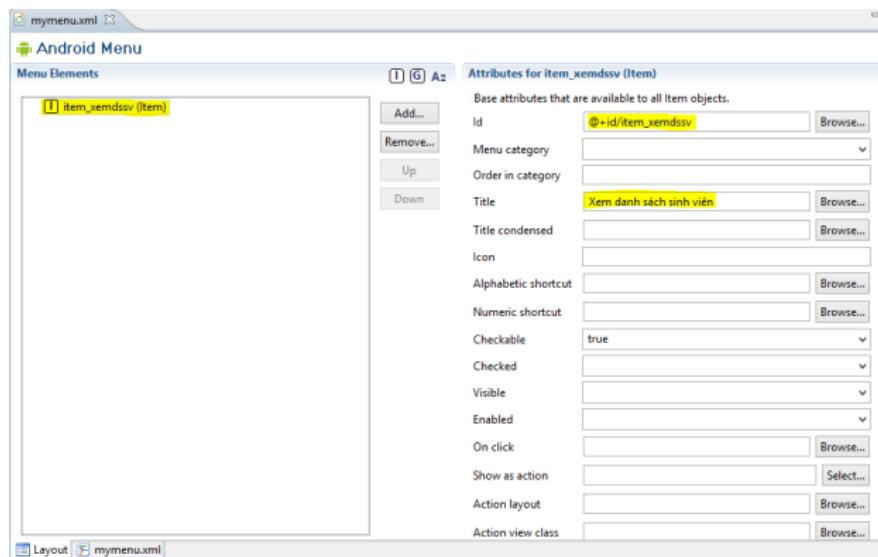
+ Màn hình trên bạn để ý

có 2 Tab: **Layout** và **mymenu.xml**, Tab **Layout** cho phép bạn tạo menu bằng giao diện, tab **mymenu.xml** cho phép bạn tạo menu bằng source code xml.

+ Nay ta sẽ làm menu bằng giao diện, bạn nhấp vào nút “**Add**” ở phía bên trên :



+ Ở màn hình này bạn chọn **Item**, rồi nhấn OK:



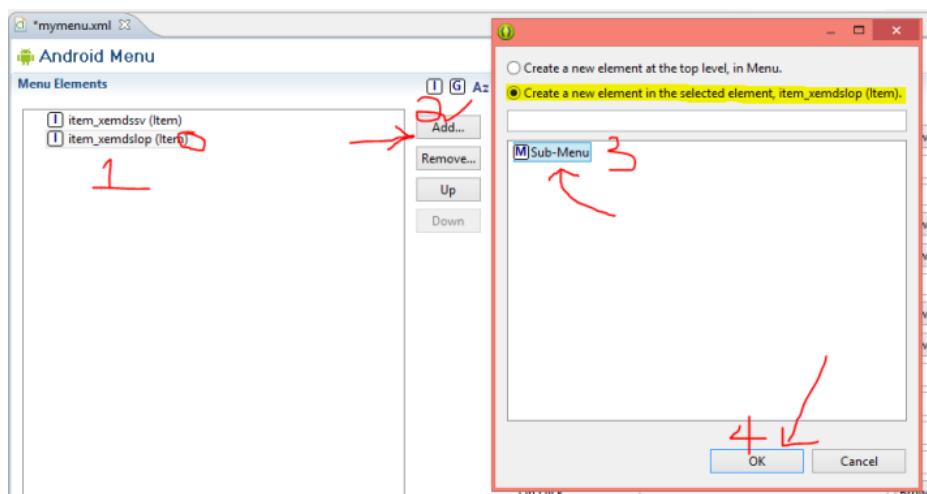
+ Khi tạo một Menu Item thì có 2 phần tử chính đó là Id và Title (Bạn xem phần Tôi tô màu vàng). Ta dựa vào Id để xử lý coding, bạn nên đặt Id nó cho khoa học, mang tính gợi nhớ. Như bạn thấy đó, Tôi đặt id cho Menu Item này là **item_xemdssv** thì id này sẽ được tự động cập nhập vào màn hình phía bên trái.

+ Các thuộc tính khác bạn tự nghiên cứu trên mạng, hoặc khi nào gặp vấn đề về nó Tôi sẽ trình bày sau.

+ Cứ như vậy bạn tuần tự tạo cho đầy đủ các Menu Item mà bạn mong muốn.

***** Trường hợp bạn muốn tạo Sub Menu Item (Menu Item chứa các Menu Item khác):**

+ Tôi có tạo thêm 1 Menu Item là “**Xem danh sách lớp**“, bây giờ Tôi muốn tạo thêm 3 Sub Menu Item cho nó, bao gồm “Lớp DHTH1A”, “Lớp DHTH1B” và “Lớp DHTH1C”:

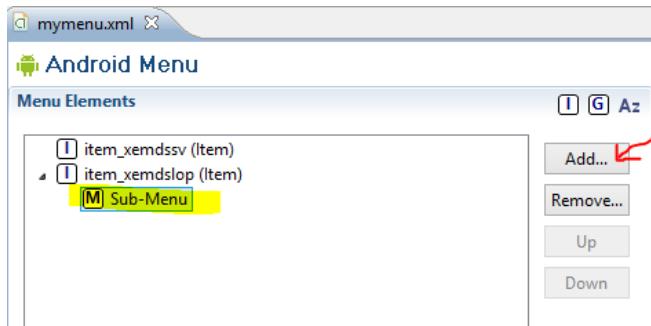


*Theo Hình trên thì bạn chọn Menu bạn muốn tạo Sub Menu cho nó trước, ở đây là **item_xemdsslop** (số 1 Tôi đánh dấu)

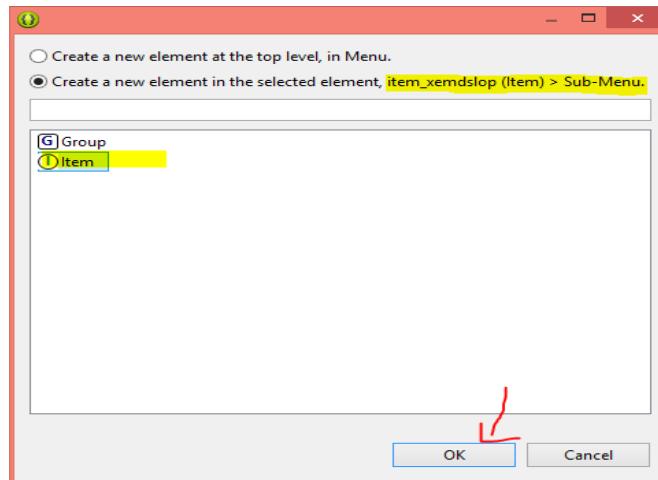
*Sau đó bạn chọn nút “**Add**” :

- Mục “Create a new element at the top level, in Menu” để tạo Menu cùng cấp
- Mục Tôi tô màu vàng “Create a element in the selected element...” để tạo Sub Menu.

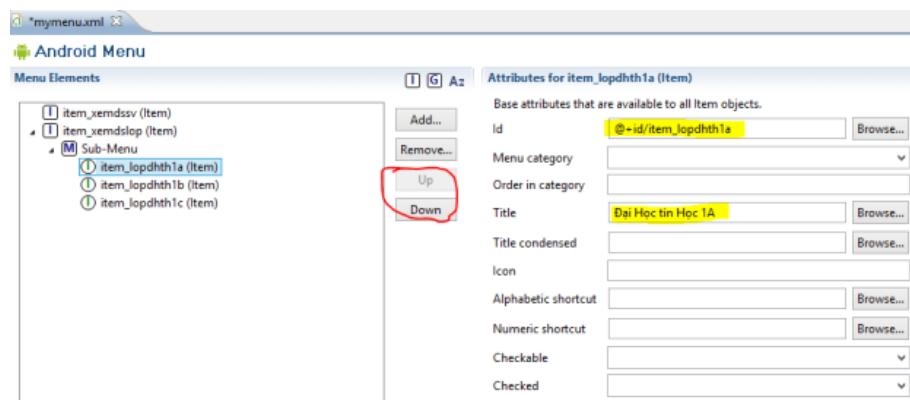
* Bấm OK để tiến hành tạo Sub Menu, màn hình sau sẽ xuất hiện:



+ Tiếp tục để tạo Menu Item cho Sub – Menu ta chọn nút “Add“, mọi thứ sẽ lại y xì như tạo Menu Item cha của nó vậy. Muốn xóa thì chọn nút “Remove“. Khi bạn nhấn nút “Add” thì bạn sẽ thấy màn hình quen thuộc sau:



+ Bạn chọn lựa như trên và nhấn OK, cứ lặp lại như vậy, mỗi lần tạo Menu Item thì bạn đặt Id và Title cho nó là xong:



+ Muốn thay đổi thứ tự xuất hiện của các Menu Item bạn chọn nút “Down”, “Up”.

* Nếu muốn xem XML resource của nó như thế nào thì bạn chọn vào tab “mymenu.xml“:

```

1 <?xml version="1.0" encoding="utf-8"?>
2 <menu xmlns:android="http://schemas.android.com/apk/res/android" >
3     <item
4         android:id="@+id/item_xemdssv"
5         android:checkable="true"
6         android:title="Xem danh sách sinh viên">
7     </item>
8     <item
9         android:id="@+id/item_xemdslop"
10        android:title="Xem danh sách lớp">
11        <menu>
12            <item
13                android:id="@+id/item_lopdhthia"
14                android:title="Đại Học tin Học 1A"/>
15            <item
16                android:id="@+id/item_lopdhthib"
17                android:title="Đại Học Tin Học 1B"/>
18            <item
19                android:id="@+id/item_lopdhthic"
20                android:title="Đại Học Tin Học 1C"/>
21        </menu>
22    </item>
23
24 </menu>

```

*** Bây giờ bạn xem cách gắn Menu vừa tạo vào ứng dụng như thế nào:

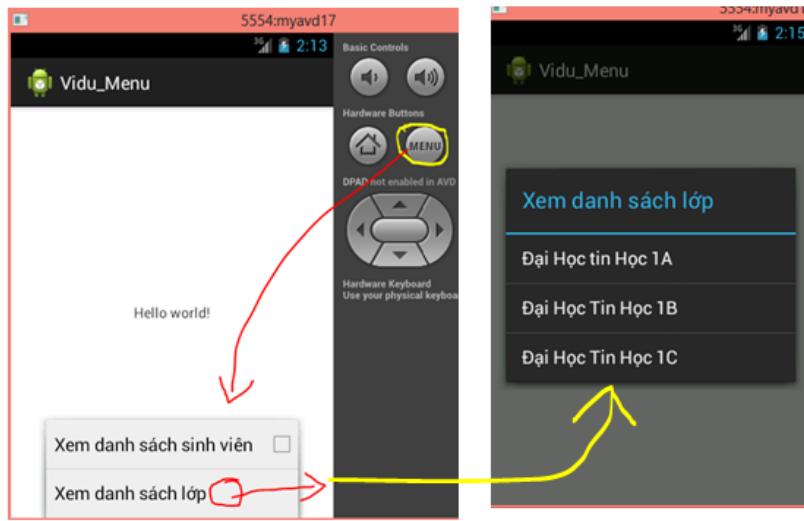
- Bạn mở **MainActivity.java** lên:

```

1 package tranduythanh.com;
2 import android.os.Bundle;
3 import android.app.Activity;
4 import android.view.Menu;
5 public class MainActivity extends Activity {
6     @Override
7     protected void onCreate(Bundle savedInstanceState) {
8         super.onCreate(savedInstanceState);
9         setContentView(R.layout.activity_main);
10    }
11    @Override
12    public boolean onCreateOptionsMenu(Menu menu) {
13        getMenuInflater()
14            .inflate(R.menu.mymenu, menu);
15        return true;
16    }
17 }

```

Dòng lệnh 13, 14: getMenuInflater().inflate(**R.menu.mymenu**, menu); dùng để gắn Menu XML Resource vào ứng dụng, bạn chạy máy ảo lên và nhấn vào nút Menu, bạn sẽ thấy được kết quả:



+ Bây giờ ta viết Coding để xử lý sự kiện cho các Menu Item:

- Để viết sự kiện cho các Menu Item bạn cần Override phương thức **onOptionsItemSelected**, Trong phương thức này ta dựa và Id của các Menu Item để xử lý:

```

1 package tranduythanh.com;
2 import android.os.Bundle;
3 import android.app.Activity;
4 import android.view.Menu;
5 import android.view.MenuItem;
6 public class MainActivity extends Activity {
7     @Override
8     protected void onCreate(Bundle savedInstanceState) {
9         super.onCreate(savedInstanceState);
10        setContentView(R.layout.activity_main);
11    }
12    @Override
13    public boolean onCreateOptionsMenu(Menu menu) {
14        getMenuInflater()
15            .inflate(R.menu.mymenu, menu);
16        return true;
17    }
18    @Override
19    public boolean onOptionsItemSelected(MenuItem item) {
20        switch(item.getItemId())
21        {
22            case R.id.item_xemdssv:
23                //Xử lý xem danh sách sinh viên
24                break;
25            case R.id.item_lopdhth1a:
26                //xử lý xem thông tin lớp DHTH1A
27                break;
28            case R.id.item_lopdhth1b:
29                break;
30            case R.id.item_lopdhth1c:
31                break;
32        }
33        return super.onOptionsItemSelected(item);

```

```
34      }
35  }
```

- Bạn nhìn vào dòng lệnh **19** trở xuống, Ở đây Tôi dùng switch case để xử lý theo đúng Id mà người sử dụng chọn lựa. Tùy vào yêu cầu của bài toán mà chúng ta xử lý trong này. Tôi khuyên bạn nên viết từng hàm riêng theo nghiệp vụ rồi cứ thế mà gọi hàm theo đúng Menu Item.

- Bạn tải code mẫu ở đây: <http://www.mediafire.com/?zawcff1ha6ap3ic>

b) Tạo Menu bằng Coding (Runtime):

- Chúng ta có thể tạo Menu lúc Runtime (không cần dùng XML Resource):

```
1  public boolean onCreateOptionsMenu(Menu menu) {
2      menu.add("Menu 1");
3      menu.add("Menu 2");
4      SubMenu sub3= menu.addSubMenu("Menu 3");
5      sub3.add("File 1 Menu 3");
6      sub3.add("File 2 Menu 3");
7      sub3.add("File 3 Menu 3");
8      return true;
9  }
```

- Bạn nhìn thấy đây, việc tạo Menu lúc Runtime rất dễ dàng. Nhưng nếu như bạn viết code tạo Menu như vậy thì việc xử lý gấp chút khó khăn vì bạn không biết Id của mỗi Menu Item.

- Do đó bạn nên viết lại theo cách sau:

```
1  package tranduythanh.com;
2
3  import android.os.Bundle;
4  import android.app.Activity;
5  import android.view.Menu;
6  import android.view.MenuItem;
7  import android.view.SubMenu;
8
9  public class MainActivity extends Activity {
10
11     @Override
12     protected void onCreate(Bundle savedInstanceState) {
13         super.onCreate(savedInstanceState);
14         setContentView(R.layout.activity_main);
15     }
16
17     @Override
18     public boolean onCreateOptionsMenu(Menu menu) {
19         //Đối số 1 là nhóm
20         //Đối số 2 là Id cho Menu Item
21         //Đối số 3 là thứ tự xuất hiện của Menu Item
22         //Đối số 4 là tiêu đề cho Menu Item
23         int itemId=113;
24         menu.add(0, itemId, 0, "Menu 1");
```

```

25     itemId=114;
26     menu.add(0,itemId,1,"Menu 2");
27     SubMenu sub3= menu.addSubMenu("Menu 3");
28     itemId=115;
29     sub3.add(0,itemId,0,"File 1 Menu 3");
30     itemId=116;
31     sub3.add(0,itemId,1,"File 2 Menu 3");
32     itemId=117;
33     sub3.add(0,itemId,2,"File 3 Menu 3");
34     return true;
35   }
36   @Override
37   public boolean onOptionsItemSelected(MenuItem item) {
38     switch(item.getItemId())
39     {
40       case 113:
41         //Xử lý Menu 1
42         break;
43       case 114:
44         //Xử lý Menu 2
45         break;
46       case 115:
47         //Xử lý File 1 của Menu 3
48         break;
49       case 116:
50         //Xử lý File 2 của Menu 3
51         break;
52       case 117:
53         //Xử lý File 3 của Menu 3
54         break;
55     }
56     return super.onOptionsItemSelected(item);
57   }
58 }
```

- Như vậy mỗi lần tạo Menu Item bạn nên gán Id cho nó và xử lý theo Id này ở trong hàm **onOptionsItemSelected**.

- Bạn tải code mẫu ở đây: <http://www.mediafire.com/?em82rb51dduvnou>

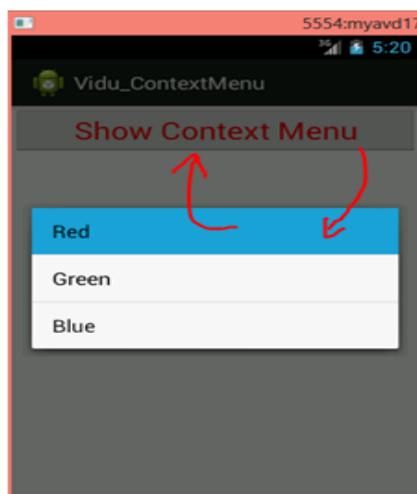
- Bài tập này Tôi sẽ không làm ví dụ mẫu lớn, Tôi để dành khi học xong Context Menu Tôi Demo luôn.

- Bài tập sau Tôi sẽ hướng dẫn cách làm Context Menu.

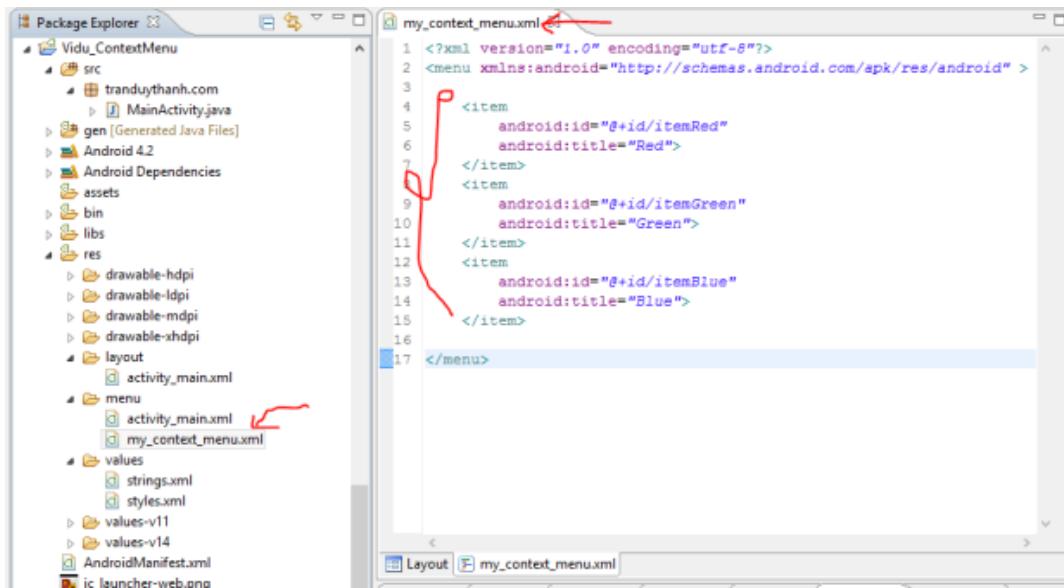
Bài tập 23: Thực hành về Context Menu trong Android

Tiếp nối bài 22, Bài tập này bạn sẽ học về Context Menu một trong những control hữu dụng mà Android cung cấp.

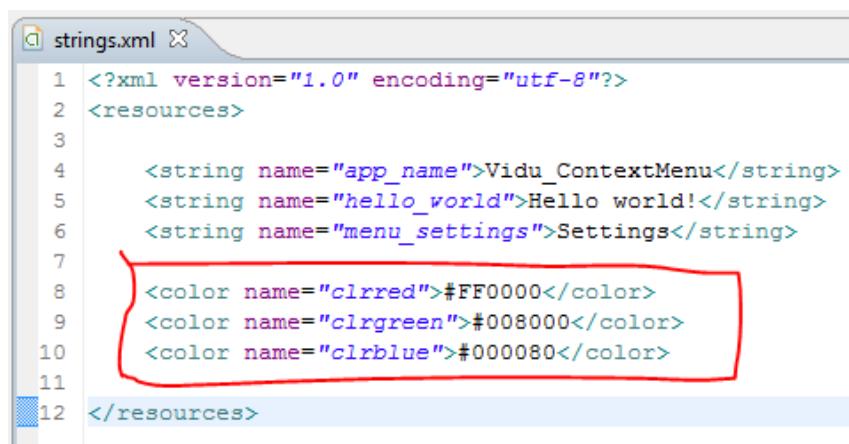
- Đối với C# , VB, hay Java để hiển thị Context Menu : Sau khi thiết lập Context Menu ta chỉ cần bấm chuột phải vào đối tượng thì sẽ có Menu Popup ra theo đúng yêu cầu.
- Đối với Android: Chúng ta cũng phải đăng ký Context Menu cho đối tượng sau đó muốn hiển thị lên thì ta nhấn thật lâu vào đối tượng (long click).
- Tương tự như với Menu, Ta có thể tạo Context menu trong XML hoặc trong Coding.
- Cách tạo Context Menu bằng XML thì nó y xì ... như Menu ... không khác 1 tí ADN nào cả, nên Tôi sẽ không nói lại cách tạo trong XML. Mà Tôi sẽ nói luôn cách đăng ký Context Menu cho đối tượng cũng như cách xử lý sự kiện cho từng Menu Item trong Context Menu.
- Ở đây Tôi làm một ví dụ đơn giản về Context Menu. Giao diện có 1 Button, khi nhấn thật lâu vào nó thì sẽ hiển thị Context Menu cho phép đổi màu chữ : Màu đỏ, màu xanh lá cây và xanh dương:



- Ở hình trên là khi Context Menu hiển thị ra, Tôi chọn Red -> Màu của Button sẽ chuyển thành màu đỏ.
- Bạn xem cấu trúc thư mục của Project này và nội dung bên trong của Context menu:



- Tiếp tục bạn Double click vào **strings.xml**, Tôi có định nghĩa một số color trong này:



- Ở trong strings.xml, Tôi tạo 3 tag color : Red, Green, Blue; Nội dung bạn phải để dạng Hex Color. 3 màu này sẽ được triết gọi trong hàm xử lý sự kiện khi người sử dụng chọn từng Menu Item trong Context Menu.

- Bạn xem nội dung **MainActivity.java**:

```

1 package tranduythanh.com;
2
3 import android.os.Bundle;
4 import android.app.Activity;
5 import android.view.ContextMenu;
6 import android.view.MenuItem;
7 import android.view.View;
8 import android.view.ContextMenu.ContextMenuItemInfo;
9 import android.widget.Button;
10
11 public class MainActivity extends Activity {
12     Button btnCtx;
13     @Override
14     protected void onCreate(Bundle savedInstanceState) {

```

```

16    super.onCreate(savedInstanceState);
17    setContentView(R.layout.activity_main);
18    btnCtx=(Button) findViewById(R.id.btnshowcontext);
19    registerForContextMenu(btnCtx);
20    }
21    @Override
22    public void onCreateContextMenu(ContextMenu menu, View v,
23        ContextMenuInfo menuInfo) {
24        // TODO Auto-generated method stub
25        super.onCreateContextMenu(menu, v, menuInfo);
26        getMenuInflater()
27            .inflate(R.menu.my_context_menu, menu);
28        }
29    @Override
30    public boolean onContextItemSelected(MenuItem item) {
31        switch(item.getItemId())
32        {
33            case R.id.itemRed:
34                btnCtx.setTextColor(
35                    getResources().getColor(R.color.clrred));
36                break;
37            case R.id.itemGreen:
38                btnCtx.setTextColor(
39                    getResources().getColor(R.color.clrgreen));
39                break;
40            case R.id.itemBlue:
41                btnCtx.setTextColor(
42                    getResources().getColor(R.color.clrblue));
43                break;
44        }
45        return super.onContextItemSelected(item);
46    }
47}
48

```

- Bạn quan sát dòng lệnh:

+ Dòng lệnh 22 , **onCreateContextMenu** dùng để nạp Context Menu XML vào ứng dụng

+ Dòng lệnh 30, **onContextItemSelected** dùng để xử lý sự kiện.

+ Để đăng ký Context Menu cho đối tượng nào thì bạn làm giống như dòng 19: **registerForContextMenu(btnCtx);**

+ **getResources().getColor(R.color.clrred)** dùng để lấy màu từ XML Resource

+ Như vậy bạn đã biết cách sử dụng Context Menu

+ Bạn có thể tải coding mẫu ở đây: <http://www.mediafire.com/?8s368sl4z594y85>

- Bài tập sau các bạn sẽ thực hành về Intent, Intent là linh hồn của Android, nó là hạt nhân chính trong Android, các bạn chú ý theo dõi.

Bài tập 24: Thực hành về Intent trong Android

- Các bạn đã đi qua được các thành phần giao diện trong Android, bài tập này các bạn sẽ làm quen với Intent. Intent rất quan trọng nó là linh hồn của Android, là hạt nhân để Android tồn tại, hầu như mọi thứ trong Android đều có dấu ấn của Intent.

- Các bạn cần hiểu rõ Intent, Tôi sẽ đưa ra hàng loạt ví dụ từ dễ tới khó để các bạn có thể thâm thấu nó từ từ.

- Đối với người bắt đầu tiếp cận Android thì Tôi có thể nói rằng các bạn chỉ dùng Intent để mở các Activity và truyền dữ liệu qua lại giữa chúng mà thôi. Khi đã nâng cao level thì bạn sẽ sử dụng Intent ở những mục đích khác nhau (Tôi sẽ có những ví dụ riêng về nó, nhưng trước tiên các bạn chỉ cần sử dụng tốt trong việc mở Activity và xử lý tác vụ này tốt là ok rồi).

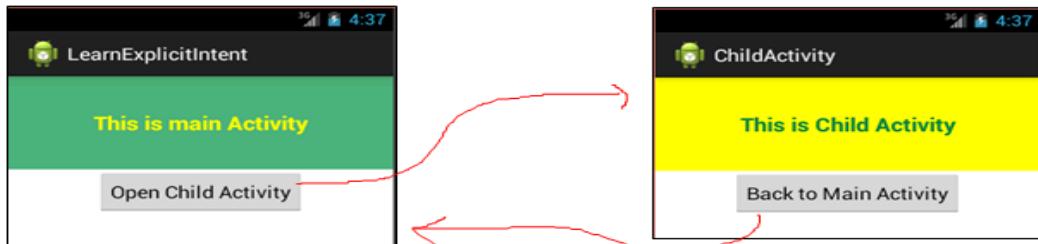
- Vì lý do đó, để phù hợp với các bạn thì Tôi chia Intent ra làm 2 công dụng chính:

1) Dùng Intent để mở một Activity khác nhưng không kiểm soát kết quả trả về.

2) Dùng Intent để mở một Activity khác và có kiểm soát kết quả trả về.

Bây giờ chúng ta vào từng ví dụ cụ thể:

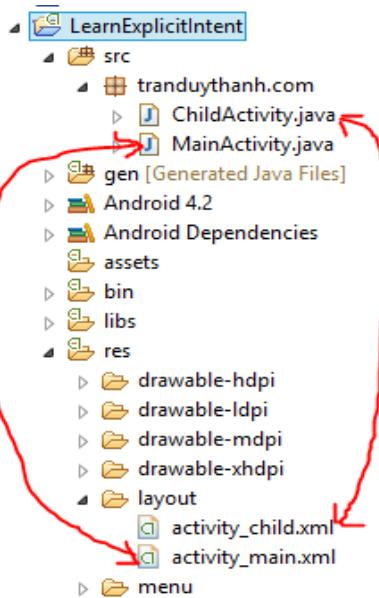
Ví dụ 1: Đơn thuần là mở một Activity khác.



- Tôi giải thích hình trên như sau: Từ Main Activity Tôi có 1 Button, khi nhấn vào Button này thì sẽ mở một Activity khác, tại Activity mới mở này sẽ có 1 Button cho phép quay trở lại MainActivity.

- Bạn phải nhớ lại vòng đời của một ứng dụng Android ở [bài tập 5](#) và [bài tập 6](#) để làm ví dụ này tốt hơn.

- Bạn tạo một Android Project có cấu trúc như hình bên dưới:



- Bạn chú ý là mọi Activity muốn được triệu gọi trong Android thì phải đăng ký trong Manifest, ở các bài tập trước Tôi cũng đã nói rồi, nhưng Tôi vẫn nhắc lại **AndroidManifest.xml**:

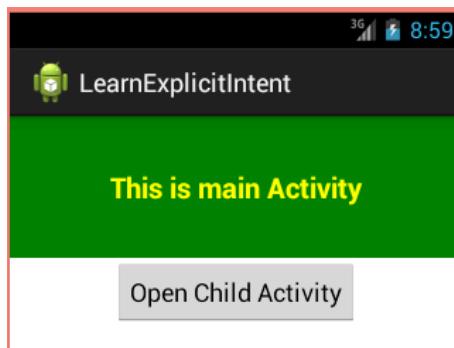
```

1   <?xml version="1.0" encoding="utf-8"?>
2   <manifest
3     xmlns:android="http://schemas.android.com/apk/res/android"
4       package="tranduythanh.com"
5       android:versionCode="1"
6       android:versionName="1.0" >
7       <uses-sdk
8         android:minSdkVersion="14"
9         android:targetSdkVersion="17" >
10      </uses-sdk>
11      <application
12        android:allowBackup="true"
13        android:icon="@drawable/ic_launcher"
14        android:label="@string/app_name"
15        android:theme="@style/AppTheme" >
16          <activity
17            android:name="tranduythanh.com.MainActivity"
18            android:label="@string/app_name" >
19              <intent-filter>
20                <action android:name="android.intent.action.MAIN" />
21                <category android:name="android.intent.category.LAUNCHER" />
22              </intent-filter>
23            </activity>
24            <activity
25              android:name="tranduythanh.com.ChildActivity"
26              android:label="@string/title_activity_child" >
27            </activity>
28          </application>
29      </manifest>

```

- Bạn thấy đó cả MainActivity (dòng 16) và ChildActivity (Dòng 24) đều phải được đăng ký trong Manifest.

- Giờ ta xem giao diện và cách xử lý coding trong MainActivity:



- Đây là cấu trúc XML của activity_main.xml ở trên:

```
1 <LinearLayout
2     xmlns:android="http://schemas.android.com/apk/res/android"
3     xmlns:tools="http://schemas.android.com/tools"
4     android:id="@+id/LinearLayout1"
5     android:layout_width="match_parent"
6     android:layout_height="match_parent"
7     android:orientation="vertical"
8     tools:context=".MainActivity" >
9
10    <TextView
11        android:id="@+id/textView1"
12        android:layout_width="match_parent"
13        android:layout_height="100dp"
14        android:background="#008000"
15        android:gravity="center"
16        android:text="This is main Activity"
17        android:textColor="#FFFF00"
18        android:textSize="20sp"
19        android:textStyle="bold" />
20
21    <Button
22        android:id="@+id/btnOpenChildActivity"
23        android:layout_width="wrap_content"
24        android:layout_height="wrap_content"
25        android:layout_gravity="center"
26        android:text="Open Child Activity" />
27
28 </LinearLayout>
```

- Tiếp hành xem coding **MainActivity.java**:

```
1 package tranduythanh.com;
2
3 import android.os.Bundle;
4 import android.app.Activity;
5 import android.content.Intent;
6 import android.view.Menu;
7 import android.view.View;
8 import android.widget.Button;
```

```

9
10 public class MainActivity extends Activity {
11     protected void onCreate(Bundle savedInstanceState) {
12         super.onCreate(savedInstanceState);
13         setContentView(R.layout.activity_main);
14         Button btnOpen=(Button)
15         findViewById(R.id.btnOpenChildActivity);
16         btnOpen.setOnClickListener(new
17             View.OnClickListener() {
18                 public void onClick(View arg0) {
19                     doOpenChildActivity();
20                 }
21             });
22     }
23     public void doOpenChildActivity()
24     {
25         Intent myIntent=new Intent(this, ChildActivity.class);
26         startActivity(myIntent);
27     }
28     public boolean onCreateOptionsMenu(Menu menu) {
29         // Inflate the menu; this adds items to the action bar if
30         it is present.
31         getMenuInflater().inflate(R.menu.activity_main, menu);
32         return true;
33     }
}

```

- Quan sát dòng lệnh 25:

Intent myIntent=new Intent(this, ChildActivity.class);

Đối số 1 là context hiện tại, nếu bạn muốn chắc ăn ở đâu nó cũng hiểu thì bạn gán tên class Activity như thế này:

Intent myIntent=new Intent(MainActivity.this, ChildActivity.class);

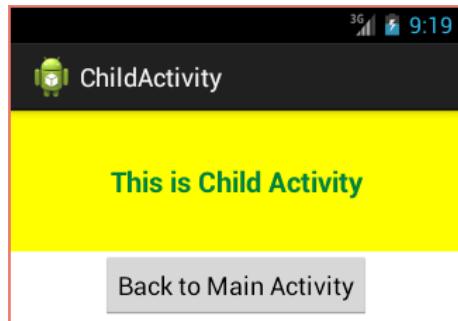
Với **MainActivity** là một Activity tùy ý (trong Context hiện tại), viết như trên thì bạn sẽ luôn luôn đúng trong mọi trường hợp.

Đối số 2 là : **ChildActivity.class**, Tức là bạn muốn mở Activity nào thì lấy tên Activity đó .class; trong trường hợp này Tôi muốn mở ChildActivity nên Tôi ghi là **ChildActivity.class**

- Quan sát dòng lệnh 26

startActivity(myIntent); Hàm này có tác dụng mở Activity được truyền vào đối số thứ 2 ở trên.

- **Tiến hành khám phá ChildActivity:**



-Đây là source XML của ChildActivity:

```
1 <LinearLayout
2     xmlns:android="http://schemas.android.com/apk/res/android"
3     xmlns:tools="http://schemas.android.com/tools"
4     android:id="@+id/LinearLayout1"
5     android:layout_width="match_parent"
6     android:layout_height="match_parent"
7     android:orientation="vertical"
8     tools:context=".ChildActivity" >
9
10    <TextView
11        android:id="@+id/textView1"
12        android:layout_width="match_parent"
13        android:layout_height="100dp"
14        android:background="#FFFF00"
15        android:gravity="center"
16        android:text="This is Child Activity"
17        android:textColor="#008040"
18        android:textSize="20sp"
19        android:textStyle="bold" />
20
21    <Button
22        android:id="@+id	btnBacktoMainActivity"
23        android:layout_width="wrap_content"
24        android:layout_height="wrap_content"
25        android:layout_gravity="center"
26        android:text="Back to Main Activity" />
27
28 </LinearLayout>
```

-Giao diện của ChildActivity chỉ có một Button cho phép quay lại MainActivity.

- Bạn xem coding của **ChildActivity**:

```
1 package tranduythanh.com;
2
3 import android.os.Bundle;
4 import android.app.Activity;
5 import android.view.Menu;
6 import android.view.View;
7 import android.widget.Button;
8
9 public class ChildActivity extends Activity {
```

```

10     protected void onCreate(Bundle savedInstanceState) {
11         super.onCreate(savedInstanceState);
12         setContentView(R.layout.activity_child);
13         Button btnBack= (Button)
14             findViewById(R.id.btnBacktoMainActivity);
15         btnBack.setOnClickListener(new
16             View.OnClickListener() {
17                 public void onClick(View v) {
18                     finish();
19                 }
20             });
21         }
22         public boolean onCreateOptionsMenu(Menu menu) {
23             // Inflate the menu; this adds items to the action bar if it
24             is present.
25             getMenuInflater().inflate(R.menu.activity_child, menu);
26             return true;
27         }

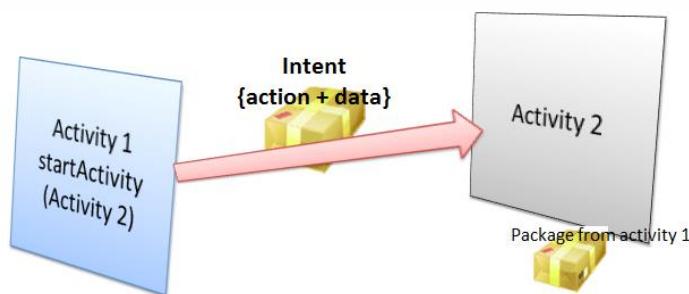
```

-Dòng lệnh 18: finish() có nhiệm vụ đóng Activity hiện tại.

- Bạn vừa đi qua 1 ví dụ vô cùng đơn giản.
 - Tải coding ở đây:<http://www.mediafire.com/?awoq5mkffm00qes>
-

Ví dụ 2: Mở một Activity khác đồng thời truyền dữ liệu qua.

- Bạn xem hình Tôì minh họa dưới này:



- Ở đây Ta thường dùng đối tượng Bundle để đóng gói dữ liệu để truyền tải qua các Activity khác.

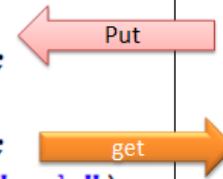
- Tại sao lại nên dùng Bundle để truyền tải dữ liệu qua lại giữa các Activity?

+ Tôi lấy ví dụ sau: Giả sử bạn phải chuyển toàn bộ hàng hóa từ Việt Nam sang Singapore, trường hợp này Ta sẽ bỏ tất cả các hàng hóa vào trong 1 Container rồi chuyển Container này đi, Không phải chuyển từng món hàng 1 qua Singapore. Tương tự như vậy, đối với Android khi truyền dữ liệu chúng ta cũng có thể truyền từng thông số riêng lẻ nhưng chúng ta không nên, chúng ta phải bỏ tất cả các dữ liệu riêng lẻ đó vào 1 Bundle rồi gửi Bundle này qua Activity nào đó. Bên kia chỉ cần lấy Bundle này ra, mọi dữ liệu đều nằm trong Bundle và ta dễ dàng xử lý.

```

Bundle bundle=new Bundle();
bundle.putInt("soa", 5);
bundle.putDouble("sob", 5.7);
//...
int soa=bundle.getInt("soa");
double sob=bundle.getDouble("sob");

```



- Bundle có 2 loại phương thức: **putXXX** dùng để cho bên gửi đi, ứng với kiểu dữ liệu nào thì put đúng kiểu đó. Ở ví dụ trên.putInt và putDouble là tương ứng với 2 loại dữ liệu khác nhau (phải đi kèm với **key** để bên nhận có thể xử lý đúng)

- Ở bên nhận Bundle thì dùng các phương thức **getXXX** tương ứng để lấy dữ liệu theo **key** bên gửi.
- key này phải giống nhau cho cả 2 nơi. Nơi nhận đặt key=**soa** thì bên nhận cũng phải dùng key=**soa**.
- Ta cũng có thể truyền Đối Tượng qua lại giữa các Activity, các đối tượng này phải được Serialize

```

class Person implements Serializable
{
    private int id;
    private String name;
    public Person(int id, String name)
    {
        this.id=id;
        this.name=name;
    }
    public String toString()
    {
        return this.id+" - "+this.name;
    }
}

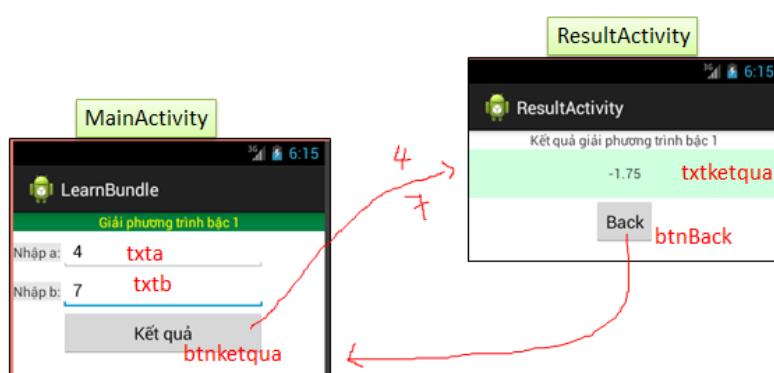
```

```

Bundle bundle=new Bundle();
Person p=new Person(1, "teo");
bundle.putSerializable("t1", p);
//...
Person t1=(Person) bundle.getSerializable("t1");

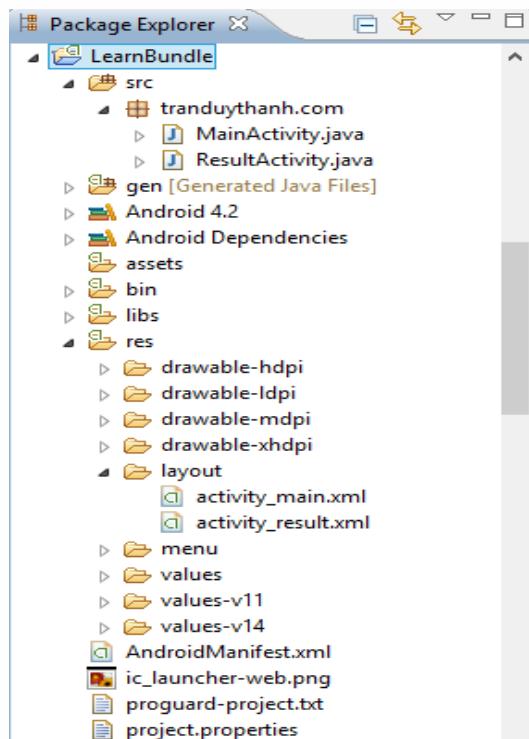
```

- Trong trường hợp truyền đối tượng thì ta dùng **putSerializable** và **getSerializable**
- Tôi sẽ làm một ví dụ đơn giản với giao diện như bên dưới:

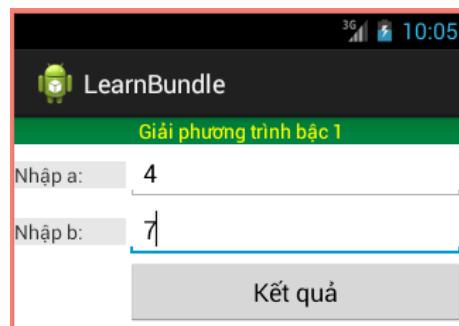


- Từ MainActivity có 2 thông số là a và b. Khi nhấn nút **Kết quả** nó sẽ truyền 2 thông số này qua ResultActivity và tiến hành giải phương trình bậc 1. Muốn trở về để tiếp tục giải phương trình khác khi nhấn nút **Back**.

- Bạn xem cấu trúc chương trình để dễ xử lý:



- Bạn xem giao diện của MainActivity (**activity_main.xml**):



- Đây là Source XML của MainActivity:

```
1 <LinearLayout  
2     xmlns:android="http://schemas.android.com/apk/res/android"  
3     xmlns:tools="http://schemas.android.com/tools"  
4     android:id="@+id/LinearLayout1"  
5     android:layout_width="match_parent"  
6     android:layout_height="match_parent"  
7     android:orientation="vertical"  
8     tools:context=".MainActivity" >  
9     <TextView  
10        android:id="@+id/textView1"  
11        android:layout_width="match_parent"  
12        android:layout_height="wrap_content"
```

```
13    android:layout_gravity="center"
14    android:background="#008040"
15    android:gravity="center"
16    android:text="Giải phương trình bậc 1"
17    android:textColor="#FFFF00" />
18    <TableLayout
19        android:layout_width="match_parent"
20        android:layout_height="wrap_content"
21        android:stretchColumns="*"
22    >
23    <TableRow
24        android:id="@+id/tableRow1"
25        android:layout_width="wrap_content"
26        android:layout_height="wrap_content" >
27        <TextView
28            android:id="@+id/textView2"
29            android:layout_width="wrap_content"
30            android:layout_height="wrap_content"
31            android:background="#E5E5E5"
32            android:text="Nhập a:" />
33        <EditText
34            android:id="@+id/txta"
35            android:layout_width="wrap_content"
36            android:layout_height="wrap_content"
37            android:inputType="text"
38            android:ems="10" >
39            <requestFocus />
40        </EditText>
41    </TableRow>
42    <TableRow
43        android:id="@+id/tableRow2"
44        android:layout_width="wrap_content"
45        android:layout_height="wrap_content" >
46        <TextView
47            android:id="@+id/textView3"
48            android:layout_width="wrap_content"
49            android:layout_height="wrap_content"
50            android:background="#E5E5E5"
51            android:text="Nhập b:" />
52        <EditText
53            android:id="@+id/txtb"
54            android:layout_width="wrap_content"
55            android:layout_height="wrap_content"
56            android:inputType="text"
57            android:ems="10" />
58    </TableRow>
59    <TableRow
60        android:id="@+id/tableRow3"
61        android:layout_width="wrap_content"
62        android:layout_height="wrap_content" >
63        <Button
64            android:id="@+id/btnketqua"
65            android:layout_width="wrap_content"
66            android:layout_height="wrap_content"
```

```

67     android:layout_column="1"
68     android:text="Kết quả" />
69   </TableRow>
70 </TableLayout>
71 </LinearLayout>

```

- Trong xml layout Bạn có thể chọn layout nào cũng được, ở trên Tôi dùng TableLayout

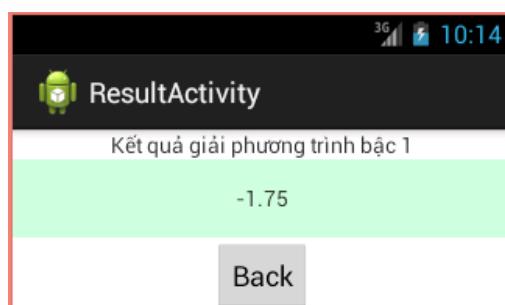
- Tiếp theo xem coding của **MainActivity.java**:

```

1  package tranduythanh.com;
2  import android.os.Bundle;
3  import android.app.Activity;
4  import android.content.Intent;
5  import android.view.Menu;
6  import android.view.View;
7  import android.widget.Button;
8  import android.widget.EditText;
9
10 public class MainActivity extends Activity {
11     EditText txta,txtb;
12     Button btnketqua;
13     protected void onCreate(Bundle savedInstanceState) {
14         super.onCreate(savedInstanceState);
15         setContentView(R.layout.activity_main);
16         txta=(EditText) findViewById(R.id.txta);
17         txtb=(EditText) findViewById(R.id.txtb);
18         btnketqua=(Button) findViewById(R.id.btnketqua);
19         btnketqua.setOnClickListener(new View.OnClickListener() {
20             public void onClick(View arg0) {
21                 //Tạo Intent để mở ResultActivity
22                 Intent myIntent=new Intent(MainActivity.this,
23                 ResultActivity.class);
24                 //Khai báo Bundle
25                 Bundle bundle=new Bundle();
26                 int a=Integer.parseInt(txta.getText().toString());
27                 int b=Integer.parseInt(txtb.getText().toString());
28                 //đưa dữ liệu riêng lẻ vào Bundle
29                 bundle.putInt("soa", a);
30                 bundle.putInt("sob", b);
31                 //Đưa Bundle vào Intent
32                 myIntent.putExtra("MyPackage", bundle);
33                 //Mở Activity ResultActivity
34                 startActivity(myIntent);
35             }
36         });
37         public boolean onCreateOptionsMenu(Menu menu) {
38             // Inflate the menu; this adds items to the action bar if it is
39             present.
40             getMenuInflater().inflate(R.menu.activity_main, menu);
41             return true;
42         }
}

```

- Tôi đã giải thích mã lệnh trong đoạn code ở trên rồi.
- Ở dòng 31, key nhận dạng Bundle là **MyPackage**, bên ResultActivity sẽ dựa vào key này để lấy Bundle ra.
- Sau khi có Bundle rồi thì dựa vào **soa**, **sob** ở dòng lệnh 28,29 để lấy đúng dữ liệu.
- Nay ta qua giao diện **ResultActivity**:



- Bạn xem XML Resource của ResultActivity:

```

1 <LinearLayout
2   xmlns:android="http://schemas.android.com/apk/res/android"
3   xmlns:tools="http://schemas.android.com/tools"
4   android:id="@+id/LinearLayout1"
5   android:layout_width="match_parent"
6   android:layout_height="match_parent"
7   android:orientation="vertical"
8   tools:context=".ResultActivity" >
9
10 <TextView
11   android:id="@+id/textView1"
12   android:layout_width="match_parent"
13   android:layout_height="wrap_content"
14   android:gravity="center"
15   android:text="Kết quả giải phương trình bậc 1" />
16
17 <TextView
18   android:id="@+id/txtketqua"
19   android:layout_width="match_parent"
20   android:layout_height="50sp"
21   android:gravity="center"
22   android:background="#CCFFD9"
23   />
24
25 <Button
26   android:id="@+id/btnBack"
27   android:layout_width="wrap_content"
28   android:layout_height="wrap_content"
29   android:layout_gravity="center"
30   android:text="Back" />
31 </LinearLayout>

```

- Tiếp tục xem xử lý coding của **ResultActivity**:

```
1  package tranduythanh.com;
2  import java.text.DecimalFormat;
3
4  import android.os.Bundle;
5  import android.app.Activity;
6  import android.content.Intent;
7  import android.view.Menu;
8  import android.view.View;
9  import android.widget.Button;
10 import android.widget.TextView;
11
12
13 public class ResultActivity extends Activity {
14     TextView txtketqua;
15     Button btnBack;
16     protected void onCreate(Bundle savedInstanceState) {
17         super.onCreate(savedInstanceState);
18         setContentView(R.layout.activity_result);
19         btnBack=(Button) findViewById(R.id.btnBack);
20         txtketqua=(TextView) findViewById(R.id.txtketqua);
21         //lấy intent gọi Activity này
22         Intent callerIntent=getIntent();
23         //có intent rồi thì lấy Bundle dựa vào MyPackage
24         Bundle packageFromCaller=
25             callerIntent.getBundleExtra("MyPackage");
26         //Có Bundle rồi thì lấy các thông số dựa vào soa, sob
27         int a=packageFromCaller.getInt("soa");
28         int b=packageFromCaller.getInt("sob");
29         //tiến hành xử lý
30        giaipt(a, b);
31         btnBack.setOnClickListener(new View.OnClickListener() {
32             public void onClick(View v) {
33                 finish();
34             }
35         });
36         public void giaipt(int a,int b)
37         {
38             String kq="";
39             if(a==0 && b==0)
40             {
41                 kq="Vô số nghiệm";
42             }
43             else if(a==0 && b!=0)
44             {
45                 kq="Vô nghiệm";
46             }
47             else
48             {
49                 DecimalFormat dcf=new DecimalFormat("0.##");
50                 kq=dfc.format(-b*1.0/a);
51             }
52             txtketqua.setText(kq);
53         }
54     }
55 }
```

```

53     }
54     @Override
55     public boolean onCreateOptionsMenu(Menu menu) {
56         // Inflate the menu; this adds items to the action bar if
57         it is present.
58         getMenuInflater().inflate(R.menu.activity_result, menu);
59         return true;
60     }
61 }

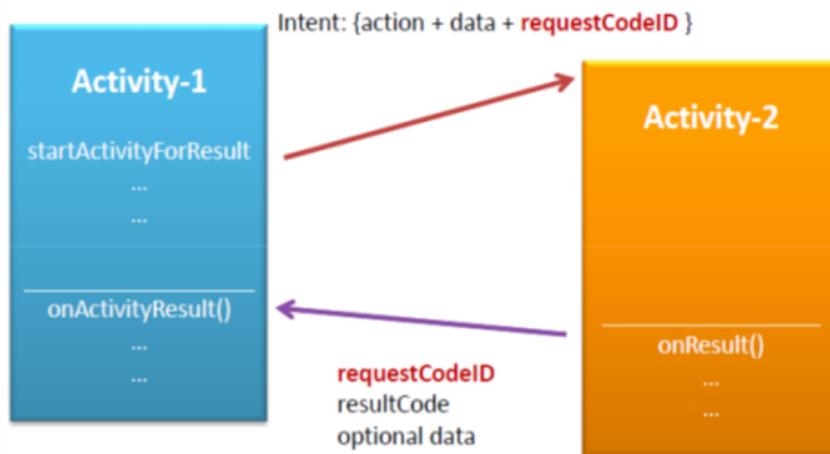
```

- Bạn xem dòng lệnh 22: **Intent callerIntent=getIntent();** lệnh này cho phép lấy Intent start Activity này. Tức là lấy Intent mà ta khai báo bên MainActivity để start ResultActivity.

- Có được Intent này rồi thì mọi thứ bạn làm như Tôi giải thích trong Coding ở trên.
 - Bạn có thể tải coding mẫu ở đây:<http://www.mediafire.com/?vb8ddcoos7dupv7>
-

Ví dụ 3: Mở một Activity khác đồng thời truyền dữ liệu qua và xử lý kết quả trả về.

- Ví dụ này bạn sẽ biết cách xử lý kết quả trả về như thế nào.
- Bạn xem hình minh họa:



- Chúng ta sẽ dựa vào requestCodeID và resultCode để xử lý.
- Việc tạo Intent trong trường hợp này cũng y xì như trường hợp trước. Nó chỉ khác hàm gọi :



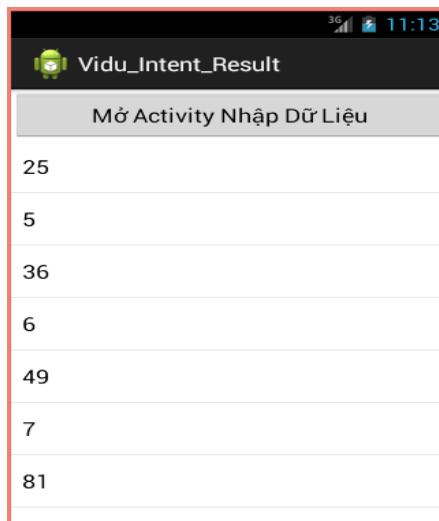
- Và phải thêm hàm xử lý kết quả trả về:

onActivityResult (requestCodeID, resultCode, Intent)

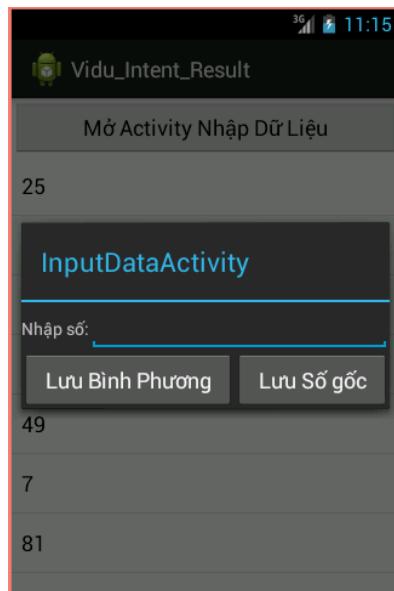


- Tôi làm ví dụ sau để bạn dễ hiểu:

- Chương trình đơn giản là MainActivity sẽ có 1 Button và ListView như hình bên dưới:

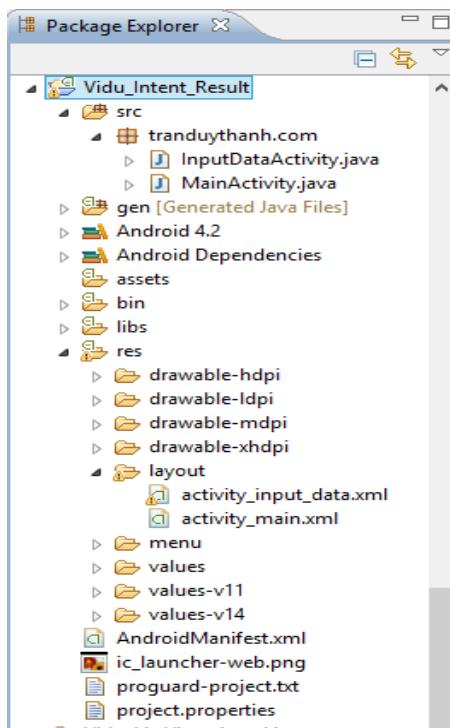


- Khi nhấn “Mở Activity nhập dữ liệu”, chương trình sẽ mở Activity mới dưới dạng Dialog:



- Khi nhập số và nhấn “Lưu bình phương” thì nó sẽ truyền số này qua MainActivity và cập nhật ListView với số này là nhân bình phương.
- Khi nhập số và nhấn “lưu số gốc” thì nó sẽ truyền số này qua MainActivity và cập nhật ListView với đúng số gốc này.
- Bạn chú ý là phải đóng Dialog ngay, vì nếu không đóng thì **onActivityResult** sẽ không xảy ra. onActivityResult chỉ xảy ra trong foreground life time.
- Ta tiến hành khám phá ứng dụng:

- Xem cấu trúc chương trình:



- Source XML của Mainactivity:

```
1 <LinearLayout
2   xmlns:android="http://schemas.android.com/apk/res/android"
3   xmlns:tools="http://schemas.android.com/tools"
4   android:id="@+id/LinearLayout1"
5   android:layout_width="match_parent"
6   android:layout_height="match_parent"
7   android:orientation="vertical"
8   tools:context=".MainActivity" >
9
10  <Button
11    android:id="@+id/btnopenactivity"
12    android:layout_width="match_parent"
13    android:layout_height="wrap_content"
14    android:text="Mở Activity Nhập Dữ Liệu" />
15
16  <ListView
17    android:id="@+id/lvldata"
18    android:layout_width="match_parent"
19    android:layout_height="wrap_content" >
20
21  </LinearLayout>
```

- Source XML của **activity_input_data.xml**:

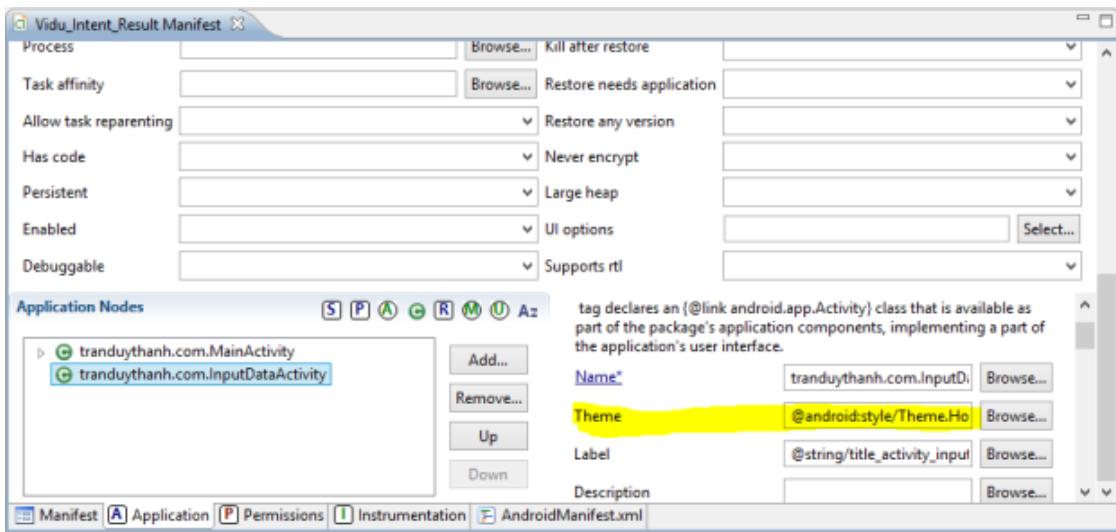
```
1 <LinearLayout
2   xmlns:android="http://schemas.android.com/apk/res/android"
3   xmlns:tools="http://schemas.android.com/tools"
4   android:id="@+id/LinearLayout1"
```

```

5      android:layout_width="match_parent"
6      android:layout_height="match_parent"
7      android:orientation="vertical"
8      tools:context=".InputDataActivity" >
9
10     <LinearLayout
11         android:layout_width="match_parent"
12         android:layout_height="wrap_content" >
13
14     <TextView
15         android:id="@+id/textView1"
16         android:layout_width="wrap_content"
17         android:layout_height="wrap_content"
18         android:text="Nhập số:" />
19
20     <EditText
21         android:id="@+id/editNumber"
22         android:layout_width="wrap_content"
23         android:layout_height="wrap_content"
24         android:layout_weight="1"
25         android:ems="10" >
26
27     <requestFocus />
28     </EditText>
29
30 </LinearLayout>
31
32 <LinearLayout
33     android:layout_width="match_parent"
34     android:layout_height="wrap_content" >
35
36     <Button
37         android:id="@+id	btnSave1"
38         android:layout_width="wrap_content"
39         android:layout_height="wrap_content"
40         android:layout_weight="1"
41         android:text="Lưu Bình Phương" />
42
43     <Button
44         android:id="@+id btnSave2"
45         android:layout_width="wrap_content"
46         android:layout_height="wrap_content"
47         android:layout_weight="1"
48         android:text="Lưu Số gốc" />
49
50 </LinearLayout>
51 </LinearLayout>

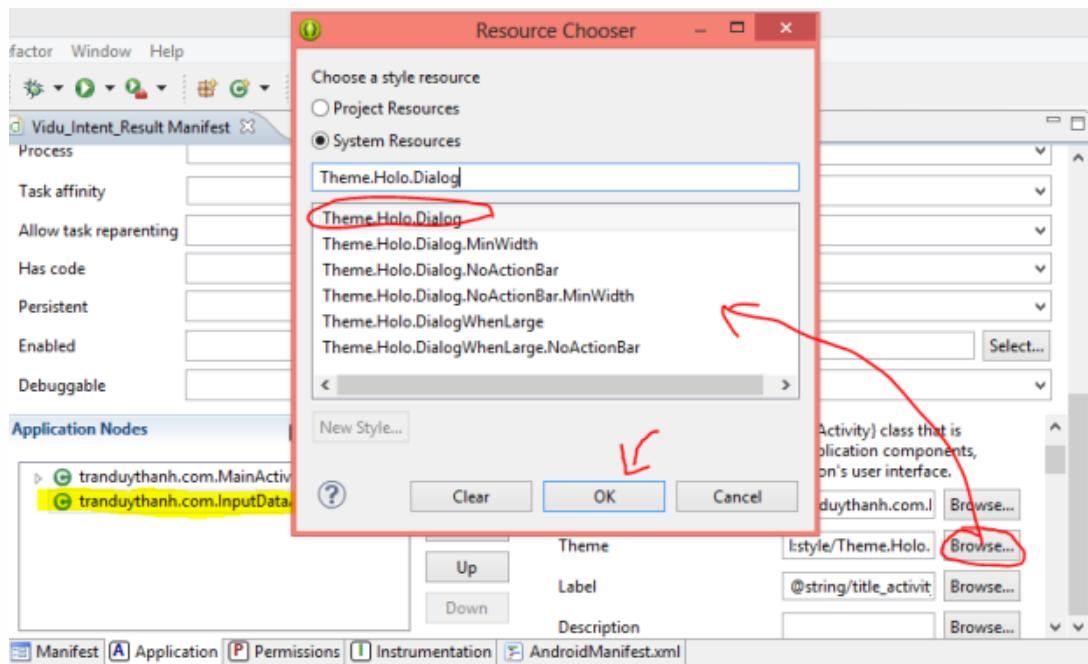
```

- Nhớ là phải cấu hình InputDataActivity dưới dạng Dialog: Mở Manifest để cấu hình như hình bên dưới:



- chọn Theme: @android:style/Theme.Holo.Dialog

- Tại màn hình trên ta chọn đúng Activity muốn làm Dialog rồi tìm tới thuộc tính Theme, nhấp vào nút “Browse”...:



- Bây giờ bạn xem xử lý code của **MainActivity.java**:

```

1 package tranduythanh.com;
2
3 import java.util.ArrayList;
4
5 import android.os.Bundle;
6 import android.app.Activity;
7 import android.content.Intent;
8 import android.view.Menu;
9 import android.view.View;
10 import android.view.View.OnClickListener;
11 import android.widget.ArrayAdapter;
12 import android.widget.Button;
```

```

12 import android.widget.ListView;
13
14 public class MainActivity extends Activity {
15
16     public static final int REQUEST_CODE_INPUT=113;
17     public static final int RESULT_CODE_SAVE1=115;
18     public static final int RESULT_CODE_SAVE2=116;
19     Button btnInputData;
20     ListView lvData;
21     ArrayList<Integer>arrData=new ArrayList<Integer>();
22     ArrayAdapter<Integer>adapter=null;
23     @Override
24     protected void onCreate(Bundle savedInstanceState) {
25         super.onCreate(savedInstanceState);
26         setContentView(R.layout.activity_main);
27         btnInputData =(Button) findViewById(R.id.btnopenactivity);
28         btnInputData.setOnClickListener(new OnClickListener() {
29
30             @Override
31             public void onClick(View arg0) {
32                 //Mở Activity với REQUEST_CODE_INPUT
33                 Intent intent=new Intent(MainActivity.this,
34                 InputDataActivity.class);
35                 //gọi startActivityForResult
36                 startActivityForResult(intent, REQUEST_CODE_INPUT);
37             }
38         });
39         //đoạn code dưới này học nhiều rồi, ko nói lại
40         lvData=(ListView) findViewById(R.id.lvdata);
41         adapter=new ArrayAdapter<Integer>
42             (this,
43             android.R.layout.simple_list_item_1,
44             arrData);
45         lvData.setAdapter(adapter);
46     }
47     /**
48      * Xử lý kết quả trả về ở đây
49     */
50     @Override
51     protected void onActivityResult(int requestCode, int resultCode,
52     Intent data) {
53         // TODO Auto-generated method stub
54         super.onActivityResult(requestCode, resultCode, data);
55         //Kiểm tra có đúng requestCode =REQUEST_CODE_INPUT hay không
56         //Vì ta có thể mở Activity với những RequestCode khác nhau
57         if(requestCode==REQUEST_CODE_INPUT)
58         {
59             //Kiểm trả ResultCode trả về, cái này ở bên InputDataActivity
60             //truyền về
61             //có nó rồi thì xử lý trả lên thông thường
62             switch(resultCode)
63             {
64                 case RESULT_CODE_SAVE1:
65                     //giá trị từ InputDataActivity
66                     int v1= data.getIntExtra("data", 0);

```

```

66     arrData.add(v1*v1);
67     adapter.notifyDataSetChanged();
68     break;
69     case RESULT_CODE_SAVE2:
70     //giá trị từ InputDataActivity
71     int v2= data.getIntExtra("data", 0);
72     arrData.add(v2);
73     adapter.notifyDataSetChanged();
74     break;
75   }
76 }
77 }
78 @Override
79 public boolean onCreateOptionsMenu(Menu menu) {
80   // Inflate the menu; this adds items to the action bar if it
81   is present.
82   getMenuInflater().inflate(R.menu.activity_main, menu);
83   return true;
84 }

}

```

- Chúng ta nên khai báo hằng số cho requestCode và resultCode như ở trên, dựa vào nó để chúng ta xử lý kết quả trả về.

- Bạn qua xử lý code của **InputDataActivity.java**:

```

1  package tranduythanh.com;
2
3  import android.os.Bundle;
4  import android.app.Activity;
5  import android.content.Intent;
6  import android.view.Menu;
7  import android.view.View;
8  import android.view.View.OnClickListener;
9  import android.widget.Button;
10 import android.widget.EditText;
11
12 public class InputDataActivity extends Activity {
13
14     Button btnSave1,btnSave2;
15     EditText editNumber;
16     @Override
17     protected void onCreate(Bundle savedInstanceState) {
18         super.onCreate(savedInstanceState);
19         setContentView(R.layout.activity_input_data);
20         btnSave1=(Button) findViewById(R.id.btnSave1);
21         btnSave2=(Button) findViewById(R.id.btnSave2);
22         btnSave1.setOnClickListener(new OnClickListener() {
23
24             @Override
25             public void onClick(View arg0) {
26                 //Gửi thông điệp là lưu bình phương
27                 sendToMain(MainActivity.RESULT_CODE_SAVE1);

```

```

28     }
29   });
30   editNumber=(EditText) findViewById(R.id.editNumber);
31   btnSave2.setOnClickListener(new OnClickListener() {
32
33     @Override
34     public void onClick(View v) {
35       //Gửi thông điệp là lưu số gốc
36       sendToMain(MainActivity.RESULT_CODE_SAVE2);
37     }
38   });
39 }
40 /**
41 * hàm xử lý gửi kết quả về mainactivity
42 * khi hàm này được gọi thì lập tức onActivityResult
43 * ở MainActivity sẽ xảy ra kèm theo resultCode và Intent
44 * @param resultcode
45 */
46 public void sendToMain(int resultcode)
47 {
48   Intent intent=getIntent();
49   int value= Integer.parseInt(editNumber.getText().toString());
50   intent.putExtra("data", value);
51   setResult(resultcode, intent);
52   finish();
53 }
54 @Override
55 public boolean onCreateOptionsMenu(Menu menu) {
56   // Inflate the menu; this adds items to the action bar if it is
57   // present.
58   getMenuInflater().inflate(R.menu.activity_input_data, menu);
59   return true;
60 }
61 }

```

- Bạn nhìn vào hàm sendToMain, khi hàm này vừa kết thúc thì lập tập onActivityResult của MainActivity sẽ xảy ra

onActivityResult(int requestCode, int resultCode, Intent data)

Ta kiểm tra requestCode và resultCode cho đúng

requestCode là bên MainActivity dùng để triệu gọi một Activity bất kỳ nào đó

resultCode là kết quả trả về trong hàm **setResult(resultcode, intent)**; của sub Activity nào đó. Hàm này cho chúng ta biết kết quả trả về là code nào và đồng thời cho ta biết luôn Intent của nó. Dựa vào Intent này mà trong **onActivityResult** ta có thể dễ dàng lấy thông số ra (đối số thứ 3).

- Như vậy bạn đã biết sơ qua về Intent, bạn phải lại đi làm lại 3 ví dụ của Tôi vừa nêu ra bên trên.

- Trong bài tập tiếp theo, Tôi sẽ cung cấp kiến thức Intent cho bạn bằng một ví dụ cụ thể đầy đủ và hơi khó về Intent, bạn hãy chú ý theo dõi.

- Bạn có thể tải code mẫu đầy đủ ở đây: <http://www.mediafire.com/?d8if4q9asgso18q>

Bài tập 25: Tiếp tục củng cố kiến thức Intent, ví dụ tổng hợp Quản Lý Nhân Viên

- Ở [bài tập 24](#) bạn đã hiểu cơ bản về Intent
- Bài tập này Tôi chủ ý viết thật phức tạp lên để các bạn tập trung công lực tìm hiểu nó.
- Xin lỗi vì để các bạn chờ lâu mới cập nhật bài tập này.
- Bây giờ Tôi tiếp tục hướng dẫn các bạn tìm hiểu sâu về Android cũng như ôn tập lại các thành phần đã học trước đó.
- Bài tập này Tôi cũng chủ ý giải thích ít đi (vì đã giải thích quá nhiều ở các bài tập trước rồi). Bài tập này bạn phải tập suy luận logic để hiểu nó sâu hơn.
- Tôi tin chắc rằng nếu như bạn hiểu bài tập 25 này thì bạn có thể viết được một chương trình Android ra trò rồi. Vì nó là tổng hợp của các kiến thức về Android mà Tôi đã trình bày từ trước tới giờ.

Mục tiêu của bài tập này:

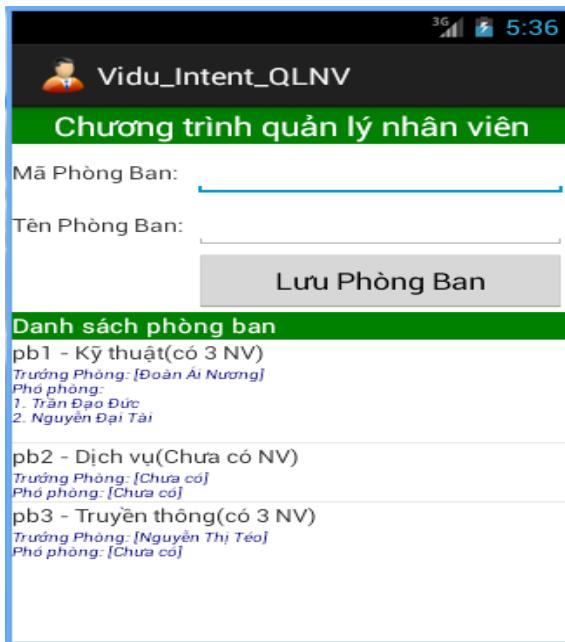
- Thao tác nhuần nhuyễn với Intent
 - Ôn tập lại các control thường hay sử dụng
 - Nhuần nhuyễn với ListView
 - Xử lý Checked (radio, Checkbox) trong ListView
 - Nhuần nhuyễn với Custom Layout
 - Biết cách sử dụng Enum
 - Cách sử dụng chung Layout cho các Activity
 - ... và đón nhận thêm một số ý tưởng sáng tạo trong lập trình.
-

- Bài này rất là khó và phức tạp, nhưng đổi lại nó rất hay. Tôi hi vọng bài tập này sẽ giúp ích cho các bạn rất nhiều trong việc khám phá Android. **Vì vậy Tôi khuyên các bạn hãy chăm chỉ làm đi làm lại bài này.**

- Mô tả chương trình:

- + Chương trình cho phép quản lý nhân viên + phòng ban.
- + Mỗi phòng ban sẽ có nhiều nhân viên, phòng ban cần lưu trữ (mã phòng ban, tên phòng ban)
- + Một phòng ban sẽ có Trưởng phòng và Phó phòng. Trưởng phòng thì chỉ có 1, phó phòng thì số lượng không giới hạn (hoặc không có)
- + Thông tin nhân viên : Mã nhân viên, tên nhân viên, giới tính (nam, nữ), chức vụ (trưởng phòng, phó phòng, nhân viên)
- + Chương trình cho phép:
 - Thêm/sửa/ xóa phòng ban
 - Thêm / sửa / xóa nhân viên
 - cho phép nhân viên chuyển phòng ban
 - cho phép thiết lập ban lãnh đạo phòng ban (trưởng phó)
 - và một số chức năng khác

Sau đây là demo giao diện chính của chương trình:



- Tôi giải thích về màn hình chính như sau:

- + Nút “**Lưu Phòng Ban**”: Cho phép lưu phòng ban và cập nhật vào ListView. Nếu mã phòng ban đã tồn tại thì tự động cập nhật, còn mã phòng ban chưa tồn tại thì phải thêm mới.
- + Sự kiện nhấn trên **ListView**: nhấp vào phòng ban nào thì hiển thị lại thông tin của phòng ban đó vào phần edittext bên trên.
- + ListView phải sử dụng Custom Layout, Ta phải tự thiết kế Layout cho ListView bên trên:

* Mỗi phần tử ListView có chứa 2 TextView, TextView 1 dùng để hiển thị mã phòng ban và tên phòng ban, chú ý phải cho người sử dụng biết Phòng ban này có nhân viên hay chưa có, nếu có nhân viên thì có bao nhiêu người (xem hình trên).

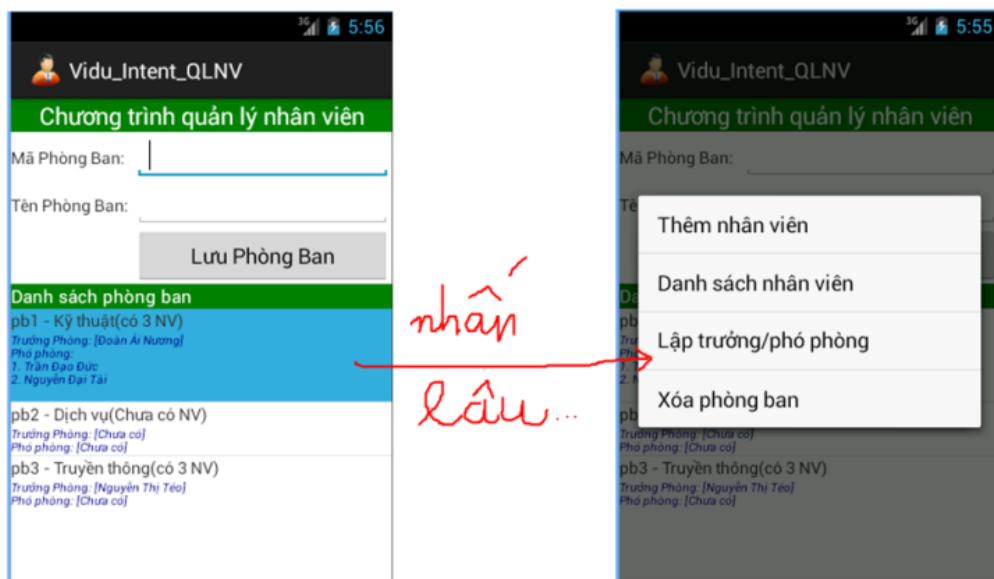
* Mỗi phòng ban phải cho biết Trưởng Phòng và phó phòng là ai (xem hình)

* Ví dụ bạn nhìn vào phòng Kỹ Thuật: Bạn thấy thông tin phòng ban và số nhân viên trong phòng ban là 3. Và Trưởng Phòng tên là “**Đoàn Ái Nương**“. Có 2 Phó Phòng là “**Trần Đạo Đức**” và “**Nguyễn Đại Tài**“.

* Phòng **dịch vụ** thì chưa có nhân viên nào cả

* Phòng **Truyền Thông** thì có 3 nhân viên nhưng chỉ có 1 trưởng phòng là “**Nguyễn Thị Téo**“.

- Tiếp tục xử lý Context Menu cho ListView Phòng ban:



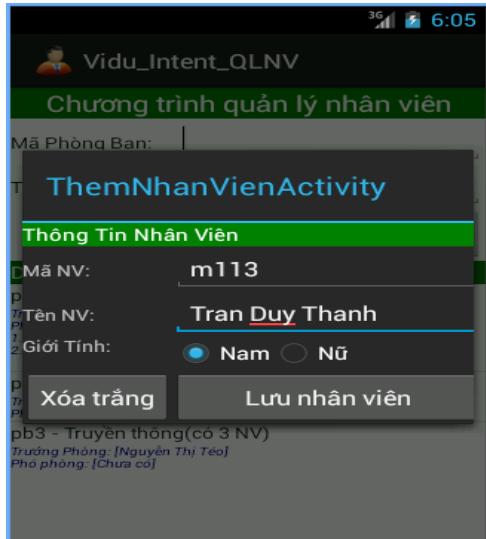
- Khi nhấn thật lâu vào một phòng ban nào đó, ví dụ trường hợp trên là phòng Kỹ Thuật, chương trình sẽ hiển thị context Menu có 4 chức năng:

1. Thêm nhân viên
2. Xem danh sách nhân viên
3. Lập trưởng/ phó phòng
4. và xóa phòng ban.

- Ứng với mỗi chức năng ta có:

1. Thêm nhân viên:

- Mặc định mỗi một nhân viên mới được tạo ra thì chỉ có chức vụ là Nhân Viên (không có chức quản lý):

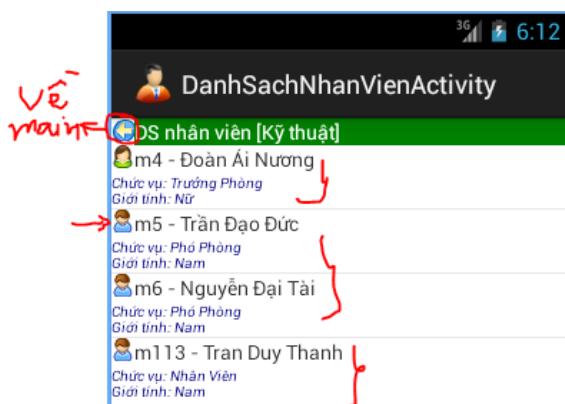


- Nút “xóa trắng”: cho phép xóa dữ liệu trên EditText và focus tới mã nhân viên
- Nút “Lưu nhân viên”, cho phép thêm mới nhân viên vào phòng ban, Giới Tính Nam, Nữ để hiển thị hình ảnh cho đúng.
- Sau khi bấm nút “Lưu nhân viên”, thì cập nhật lại danh sách phòng ban (cập nhập số lượng nhân viên cho phòng ban).
- Ví dụ ở hình trên bây giờ Tôi bấm “Lưu nhân viên”, màn hình này sẽ tắt đi để hiển thị lại màn hình chính, nhưng trong phòng ban Kỹ Thuật số lượng nhân viên sẽ tăng lên là 4, xem hình:



2. Xem danh sách nhân viên:

- Chương trình sẽ hiển thị danh sách nhân viên của phòng ban đang chọn, ví dụ trong trường hợp này là xem danh sách nhân viên của phòng ban Kỹ thuật:



- Như bạn thấy đó. Màn hình xem danh sách nhân viên có các chức năng sau:

+ Nút mũi tên “BacK” cho phép trở về màn hình chính.

+ Nhân viên có hình theo giới tính (nam, nữ)

+ Nhân viên có mô tả: Chức vụ, giới tính

+ Vậy ta cũng phải dùng Custom layout

- Nay giờ ta xem chức năng Context Menu cho ListView của mục xem danh sách nhân viên:



- Vậy màn hình này có 3 chức năng:

2.1 Sửa nhân viên

2.2 Chuyển phòng ban

2.3 Xóa nhân viên

- Ta vào từng chức năng:

2.1 Sửa nhân viên:

- Ở ví dụ trên là ta nhấn thật lâu vào nhân viên tên “Trần Đạo Đức“, sau đó Context Menu hiển thị lên, ta chọn sửa nhân viên, màn hình sửa xuất hiện như sau (chú ý là Activity sửa nhân viên sử dụng chung Layout với Activity thêm nhân viên):



- Thông tin của nhân viên “Trần Đạo Đức” sẽ được hiển thị như trên. Mã nhân viên sẽ改成 thay đổi, còn các chức năng khác là y xì như trong màn hình thêm mới nhân viên.

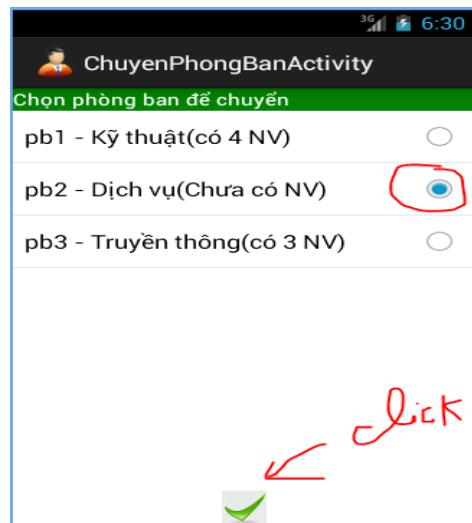
- Trong trường hợp này Tôi sửa “Trần Đạo Đức” thành “Trần Thé” và nhấn nút “Lưu nhân viên” thì màn hình này đóng lại để trở về màn hình xem danh sách, bạn quan sát sự thay đổi như sau:



2.2 Chuyển phòng ban:

- Bây giờ ta vào chức năng chuyển phòng ban cho nhân viên. Chương trình sẽ hiển thị danh sách phòng ban (có RadioButton ở bên cạnh cho phép chuyển phòng ban).

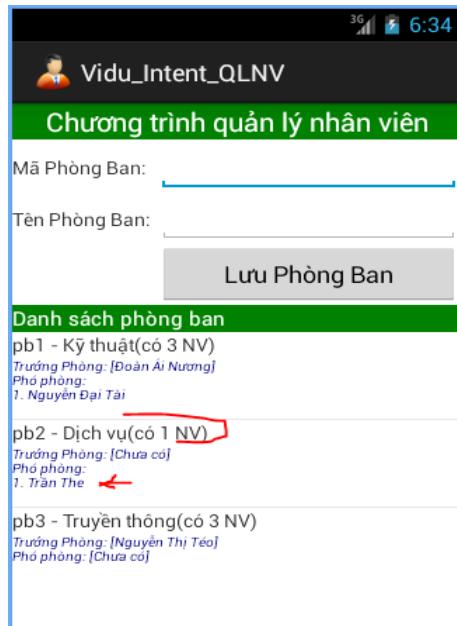
- Bây giờ Tôi chọn “Trần Thé” và chuyển qua phòng **Dịch vụ** (Nhấn thật lâu vào Trần Thé, context menu hiển thị ra thì ta chọn Chức năng **Chuyển phòng ban**):



- Sau khi chọn phòng ban để chuyển, nhấn nút có hình “Apply”, chương trình sẽ quay lại màn hình xem danh sách nhân viên, lúc này Bạn không còn thấy nhân viên Trần Thé nữa, bởi vì nhân viên này đã được chuyển qua phòng dịch vụ:



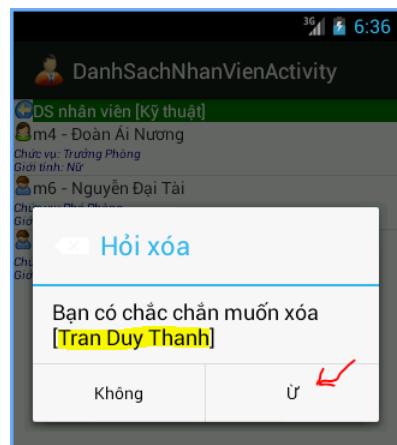
- Bạn có thể kiểm tra lại bằng cách bấm nút “Back” để trở về màn hình chính, trong màn hình chính sẽ xuất hiện 1 nhân viên mà ta vừa chuyển:



2.3 xóa nhân viên

- bây giờ ta qua chức năng Xóa nhân viên trong màn hình xem danh sách, chương trình sẽ hiển thị màn hình hỏi xem có chắc chắn muốn xóa hay không? nhấn đồng ý để xóa:

- Ví dụ Tôi chọn nhân viên tên “Trần Duy Thành” rồi xóa:



3. Lập trưởng/ phó phòng:

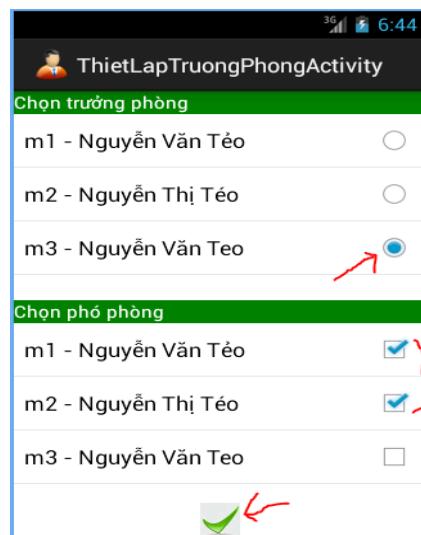
- bây giờ ta qua chức năng Lập trưởng / phó phòng cho mỗi phòng ban (tại màn hình chính).

- Khi context Menu hiển thị lên thì ta chọn “Lập trưởng/ phó phòng”:

- Ví dụ, bây giờ Tôi nhấn thật lâu vào phòng ban “Truyền Thông”, sau đó Tôi thiết lập trưởng phó phòng lại cho phòng ban này, xem giao diện bên dưới:



- Sau khi chọn “lập trưởng/ phó phòng”, thì chương trình hiển thị màn hình thiết lập như bên dưới:

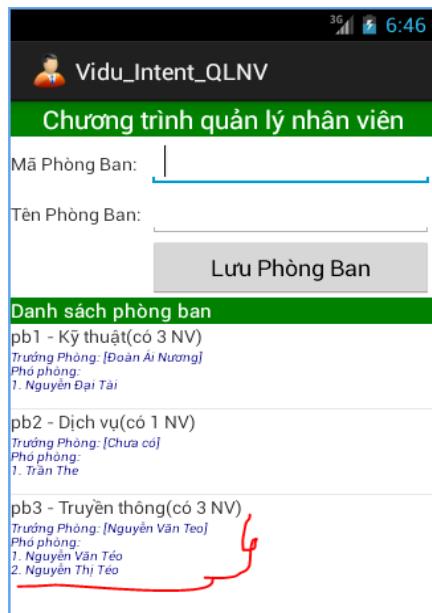


- Trưởng phòng thì chỉ cho phép chọn 1 (dùng Radio Button)

- Phó phòng cho phép chọn tùy ý (dùng Checkbox)

- Ta dùng layout có sẵn của android (không phải custom layout), chỉ viết lại sự kiện onclick cho ListView nó hiểu mà thôi.

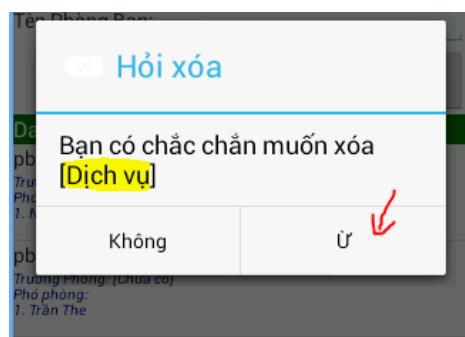
- Sau khi chọn Trưởng/ phó, click “Apply” thì chương trình trở về màn hình chính, quan sát:



- Bạn thấy đó, Trưởng phòng và phó phòng đã được cập nhật.

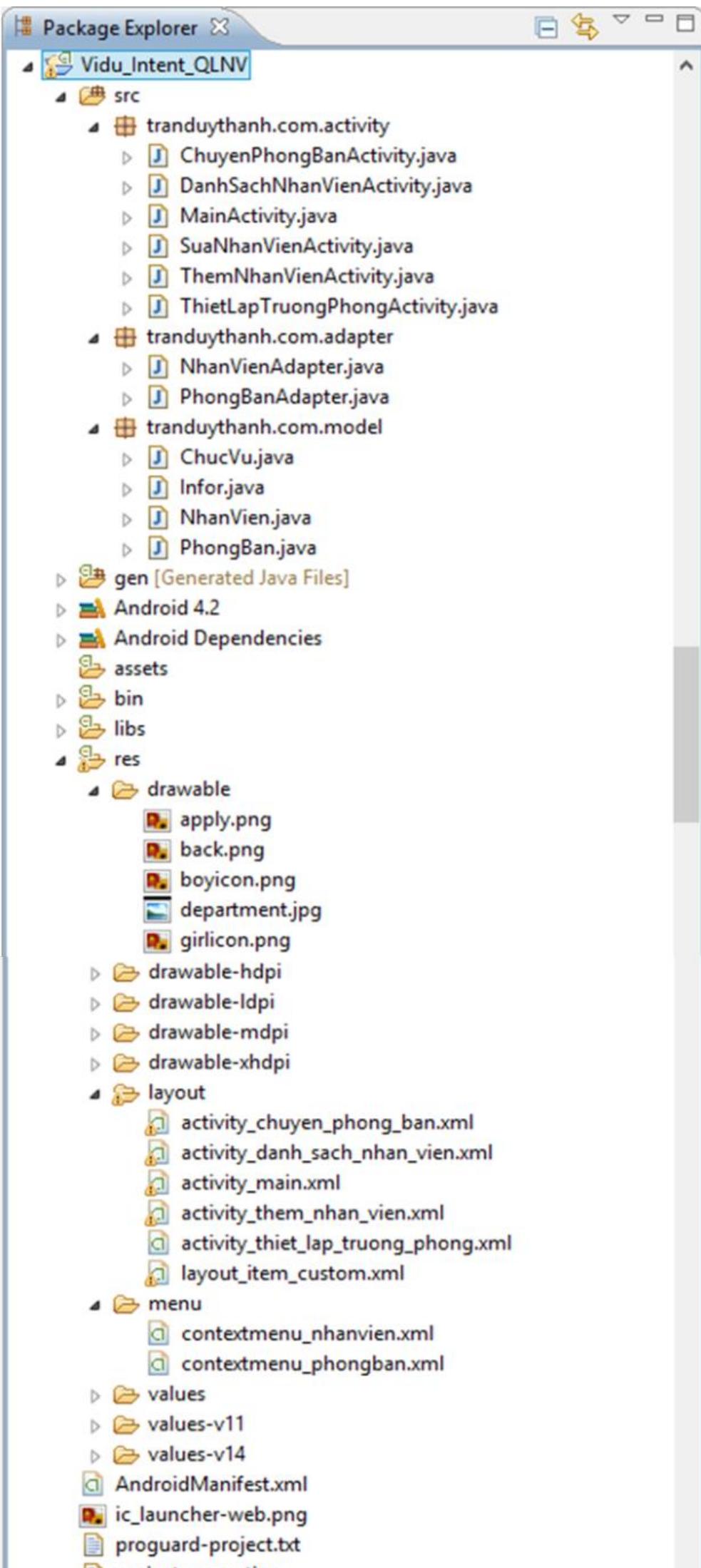
4. và xóa phòng ban.

- Ta qua chức năng cuối cùng: Xóa phòng ban, tưởng tự như xóa nhân viên. Chương trình cũng hỏi ta có chắc chắn muốn xóa hay không?



- Như vậy Tôi đã mô tả xong yêu cầu của bài toán. Bây giờ Tôi đi chi tiết vào coding:

- Bạn xem cấu trúc thư mục của Android Project:



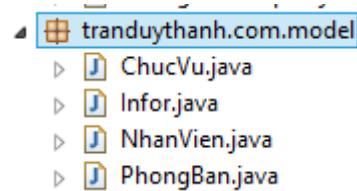
- Bạn quan sát trong phần Source Code Tôi chia ra làm 3 Package:

1) **tranduythanh.com.model** -> dùng để viết mô hình hóa đối tượng

2) **tranduythanh.com.adapter** -> dùng để viết Customlayout cho ListView

3) **tranduythanh.com.activity** -> dùng để lưu trữ các Activity

- Nay ta vào package : **tranduythanh.com.model** :



- Package này có 3 class và 1 enum chức vụ (thực ra enum cũng là một class đặc biệt).

- Coding enum ChucVu.java:

```
1 package tranduythanh.com.model;
2 /**
3  * Enum này để thiết lập chức vụ cho nhân viên
4  * @author drthanh
5  *
6  */
7 public enum ChucVu {
8     //Muốn gán được như thế này: TruongPhong("Trưởng Phòng")
9     //thì phải có constructor ChucVu(String cv)
10    TruongPhong("Trưởng Phòng"),
11    PhoPhong("Phó Phòng"),
12    NhanVien("Nhân Viên");
13    private String cv;
14    ChucVu(String cv)
15    {
16        this.cv=cv;
17    }
18    public String getChucVu()
19    {
20        return this.cv;
21    }
22 }
```

- Coding Infor.java:

```
1 package tranduythanh.com.model;
2
3 import java.io.Serializable;
4 /**
5  * Class này để lưu mã và tên là class cha của NhanVien
6  * và PhongBan. Vì cả nhân viên hay phòng ban đều có mã và
```

```

7  tên
8   * @author drthanh
9   *
10  */
11 public class Infor implements Serializable{
12  /**
13  *
14  */
15  private static final long serialVersionUID = 1L;
16  private String ma;
17  private String ten;
18  public Infor(String ma, String ten) {
19      super();
20      this.ma = ma;
21      this.ten = ten;
22  }
23  public Infor() {
24      super();
25  }
26  public String getMa() {
27      return ma;
28  }
29  public void setMa(String ma) {
30      this.ma = ma;
31  }
32  public String getTen() {
33      return ten;
34  }
35  public void setTen(String ten) {
36      this.ten = ten;
37  }
38  @Override
39  public String toString() {
40      // TODO Auto-generated method stub
41      return this.ma+" - "+this.ten;
42  }
}

```

- Coding NhanVien.java:

```

1  package tranduythanh.com.model;
2
3  import java.io.Serializable;
4  /**
5   * Class lưu trữ thông tin nhân viên
6   * @author drthanh
7   *
8   */
9  public class NhanVien extends Infor implements Serializable{
10  /**
11  *
12  */
13  private static final long serialVersionUID = 1L;
14  private boolean gioitinh;
    private ChucVu chucvu;

```

```

15     private PhongBan phongban;
16     public NhanVien(String ma, String ten, boolean gioitinh,
17 ChucVu chucvu,
18     PhongBan phongban) {
19     super(ma,ten);
20     this.gioitinh = gioitinh;
21     this.chucvu = chucvu;
22     this.phongban = phongban;
23   }
24   public NhanVien(String ma, String ten, boolean gioitinh) {
25     super(ma,ten);
26     this.gioitinh = gioitinh;
27   }
28   public NhanVien() {
29     super();
30   }
31   public boolean isGioitinh() {
32     return gioitinh;
33   }
34   public void setGioitinh(boolean gioitinh) {
35     this.gioitinh = gioitinh;
36   }
37   public ChucVu getChucvu() {
38     return chucvu;
39   }
40   public void setChucvu(ChucVu chucvu) {
41     this.chucvu = chucvu;
42   }
43   public PhongBan getPhongban() {
44     return phongban;
45   }
46   public void setPhongban(PhongBan phongban) {
47     this.phongban = phongban;
48   }
49   @Override
50   public String toString() {
51     // TODO Auto-generated method stub
52     return super.toString();
53   }
54 }
```

- Coding PhongBan.java:

```

1 package tranduythanh.com.model;
2
3 import java.io.Serializable;
4 import java.util.ArrayList;
5 /**
6  * Class để lưu thông tin phòng ban
7  * @author drthanh
8  *
9  */
10 public class PhongBan extends Infor implements Serializable{
11 }
```

```

12 private static final long serialVersionUID = 1L;
13 private ArrayList<NhanVien>dsnv=new ArrayList<NhanVien>();
14 public PhongBan(String ma, String ten) {
15     super(ma,ten);
16 }
17 public PhongBan() {
18     super();
19 }
20 /**
21 * Hàm cho phép thêm một nhân viên mới
22 * nếu nhân viên có mã tồn tại rồi thì
23 * tự động trở thành cập nhật
24 * @param nv
25 */
26 public void themNv(NhanVien nv)
27 {
28     int i=0;
29     for(;i<dsnv.size();i++)
30     {
31         NhanVien nvOld=dsnv.get(i);
32         if(nvOld.getMa().trim().equalsIgnoreCase(nv.getMa().trim()))
33         {
34             break;
35         }
36     }
37     if(i<dsnv.size())
38         dsnv.set(i, nv);
39     else
40         dsnv.add(nv);
41 }
42 public NhanVien get(int index)
43 {
44     return dsnv.get(index);
45 }
46 public int size()
47 {
48     return dsnv.size();
49 }
50 /**
51 * hàm trả về trưởng phòng của phòng ban
52 * Nếu chưa có trưởng phòng thì trả về null
53 * @return
54 */
55 public NhanVien getTruongPhong()
56 {
57     for(int i=0;i<dsnv.size();i++)
58     {
59         NhanVien nv=dsnv.get(i);
60         if(nv.getChucVu()==ChucVu.TruongPhong)
61             return nv;
62     }
63     return null;
64 }
65 /**
66 * hàm trả về danh sách phó trưởng phòng

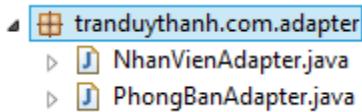
```

```

66 * vì 1 phòng ban có thể có nhiều phó phòng
67 * Nếu không có phó phòng nào thì danh sách
68 * có kích thước =0
69 * @return
70 */
71 public ArrayList<NhanVien>getPhoPhong()
72 {
73 ArrayList<NhanVien> dsPhoPhong=new ArrayList<NhanVien>();
74 for(NhanVien nv: dsnv)
75 {
76 if(nv.getChucvu ()==ChucVu.PhoPhong)
77 dsPhoPhong.add(nv);
78 }
79 return dsPhoPhong;
80 }
81 public ArrayList<NhanVien> getListNhanVien()
82 {
83 return this.dsnv;
84 }
85 @Override
86 public String toString() {
87 // TODO Auto-generated method stub
88 String str=super.toString();
89 if(dsnv.size ()==0)
90 str+="(Chưa có NV)";
91 else
92 str+="(có "+dsnv.size ()+" NV)";
93 return str;
94 }
95 }
96

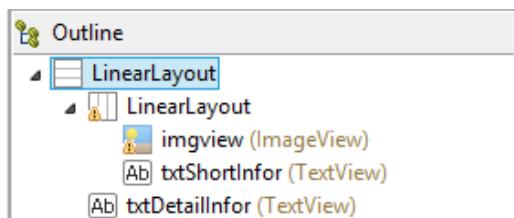
```

- Bây giờ ta qua package **tranduythanh.com.adapter**:



- 2 Adapter này dùng để custom layout cho danh sách Phòng ban và danh sách nhân viên.

- Chú ý là cả 2 Adapter này đều dùng chung 1 Layout tên “**layout_item_custom.xml**“:



- Bạn xem source XML của ”**layout_item_custom.xml**“:

```
1 <?xml version="1.0" encoding="utf-8"?>
```

```

2 <LinearLayout
3     xmlns:android="http://schemas.android.com/apk/res/android"
4         android:layout_width="match_parent"
5         android:layout_height="match_parent"
6         android:orientation="vertical" >
7
8     <LinearLayout
9         android:layout_width="match_parent"
10        android:layout_height="wrap_content" >
11
12    <ImageView
13        android:id="@+id/imgview"
14        android:layout_width="wrap_content"
15        android:layout_height="wrap_content"
16    />
17
18    <TextView
19        android:id="@+id/txtShortInfor"
20        android:layout_width="match_parent"
21        android:layout_height="wrap_content" />
22
23 </LinearLayout>
24
25 <TextView
26     android:id="@+id/txtDetailInfor"
27     android:layout_width="match_parent"
28     android:layout_height="wrap_content"
29     android:textColor="#000080"
30     android:textSize="10sp"
31     android:textStyle="italic" />
32
33 </LinearLayout>

```

- Bây giờ ta xem cách custom layout thông qua ADapter (phần customlayout Tôi đã hướng dẫn ở các bài tập trước, bạn tự xem lại):

- Coding **PhongBanAdapter.java**:

```

1 package tranduythanh.com.adapter;
2
3 import java.util.ArrayList;
4
5 import tranduythanh.com.activity.R;
6 import tranduythanh.com.model.NhanVien;
7 import tranduythanh.com.model.PhongBan;
8 import android.app.Activity;
9 import android.view.View;
10 import android.view.ViewGroup;
11 import android.widget.ArrayAdapter;
12 import android.widget.TextView;
13 /**
14 * Class này dùng để custom layout cho Danh sách phòng ban
15 * @author drthanh
16 */

```

```

17     */
18     public class PhongBanAdapter extends ArrayAdapter<PhongBan> {
19         Activity context;
20         int layoutId;
21         ArrayList<PhongBan> arrPhongBan;
22         public PhongBanAdapter(Activity context, int
23             textViewResourceId,
24             ArrayList<PhongBan> objects) {
25             super(context, textViewResourceId, objects);
26             this.context=context;
27             this.layoutId=textViewResourceId;
28             this.arrPhongBan=objects;
29         }
30         @Override
31         public View getView(int position, View convertView, ViewGroup
32             parent) {
33             //gán layout vào coding
34             convertView=context.getLayoutInflater().inflate(layoutId,
35             null);
36             //lấy các control ra theo id
37             TextView txtpb= (TextView)
38             convertView.findViewById(R.id.txtShortInfor);
39             TextView txtmotapb= (TextView)
40             convertView.findViewById(R.id.txtDetailInfor);
41             //Lấy phòng ban thứ position
42             PhongBan pb=arrPhongBan.get(position);
43             txtpb.setText(pb.toString());
44             /**
45             * Các Dòng lệnh dưới này để kiểm tra Trưởng phòng, phó phòng
46             */
47             String strMota="";
48             String tp="Trưởng Phòng: [Chưa có]";
49             NhanVien nv=pb.getTruongPhong();
50             if(nv!=null)
51             {
52                 tp="Trưởng Phòng: ["+nv.getTen()+"]";
53             }
54             ArrayList<NhanVien> dsPp=pb.getPhoPhong();
55             String pp="Phó phòng: [Chưa có]";
56             if(dsPp.size()>0)
57             {
58                 pp="Phó phòng:\n";
59                 for(int i=0;i<dsPp.size();i++)
60                 {
61                     pp+=(i+1)+". "+dsPp.get(i).getTen()+"\n";
62                 }
63             }
64             strMota=tp+"\n"+pp;
65             //gán thông tin cho phần chi tiết
66             txtmotapb.setText(strMota);
67             return convertView;
68         }
69     }

```

- Coding **NhanVienAdapter.java**, Ở phòng ban Tôi không sử dụng ImageView, còn ở Nhân viên Tôi sử dụng ImageView :

```
1  package tranduythanh.com.adapter;
2
3  import java.util.ArrayList;
4
5  import tranduythanh.com.activity.R;
6  import tranduythanh.com.model.NhanVien;
7  import android.app.Activity;
8  import android.view.View;
9  import android.view.ViewGroup;
10 import android.widget.ArrayAdapter;
11 import android.widget.ImageView;
12 import android.widget.TextView;
13 /**
14  * Class để tạo customlayout cho danh sách nhân viên
15  * ImageView để hiển thị hình ảnh theo giới tính
16  * @author drthanh
17  *
18 */
19 public class NhanVienAdapter extends ArrayAdapter<NhanVien> {
20
21     Activity context;
22     int layoutId;
23     ArrayList<NhanVien> arrNhanVien;
24     public NhanVienAdapter(Activity context, int
25     textViewResourceId,
26     ArrayList<NhanVien> objects) {
27         super(context, textViewResourceId, objects);
28         this.context=context;
29         this.layoutId=textViewResourceId;
30         this.arrNhanVien=objects;
31     }
32     @Override
33     public View getView(int position, View convertView, ViewGroup
34     parent) {
35         //gán XML layout vào coding
36         convertView=context.getLayoutInflater().inflate(layoutId,
37         null);
38         //lấy các control ra
39         TextView txtnv= (TextView)
40         convertView.findViewById(R.id.txtShortInfor);
41         TextView txtmotanv= (TextView)
42         convertView.findViewById(R.id.txtDetailInfor);
43         ImageView img=(ImageView)
44         convertView.findViewById(R.id.imgview);
45         //lấy nhân viên thứ position
46         NhanVien nv=arrNhanVien.get(position);
47         txtnv.setText(nv.toString());
48         String strMota="";
49         String cv="Chức vụ: "+nv.getChucvu().getChucVu();
50         String gt="Giới tính: "+(nv.isGioitinh())?"Nữ":"Nam";
51         //Kiểm tra giới tính để gán cho đúng hình đại diện
52         img.setImageResource(R.drawable.girlicon);
```

```

52     if(!nv.isGioitinh())
53         img.setImageResource(R.drawable.boyicon);
54         strMota=cv+"\n"+gt;
55         txtmotanv.setText(strMota);
56         return convertView;
57     }

}

```

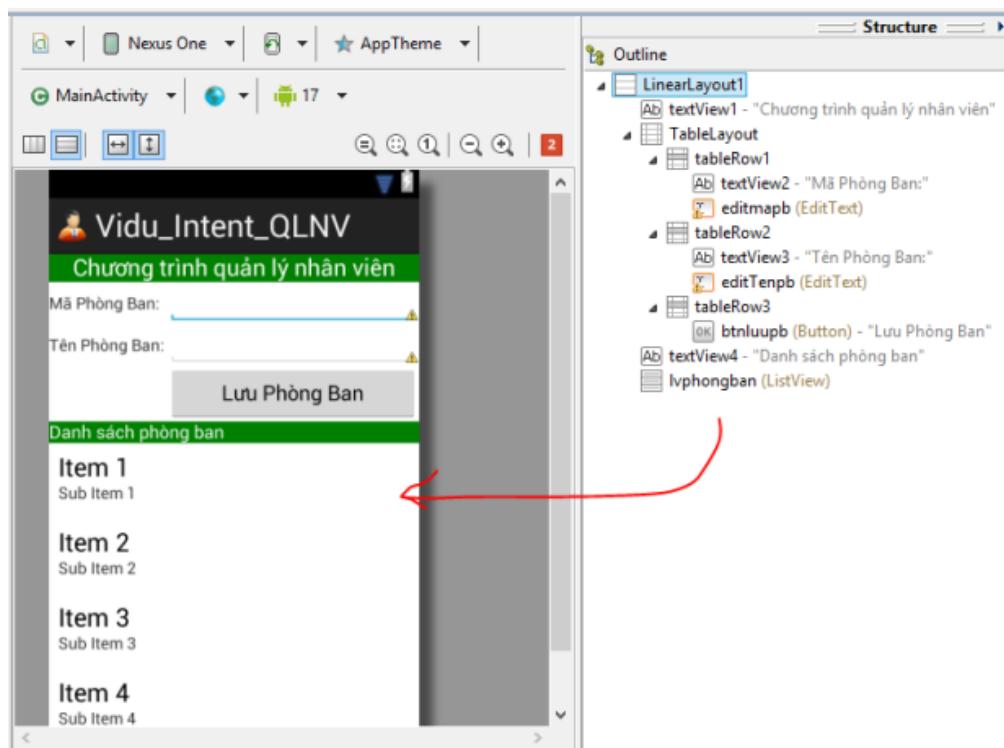
- Bây giờ ta qua package **tranduythanh.com.activity**:



- Ta lần lượt vào từng màn hình theo thứ tự.

- Màn hình chính (**MainActivity.java**):

- Bạn xem Outline của màn hình chính chính:



- Đây là XML source của MainActivity:

```

1 <LinearLayout
2     xmlns:android="http://schemas.android.com/apk/res/android"
3     xmlns:tools="http://schemas.android.com/tools"
4     android:id="@+id/LinearLayout1"

```

```
5     android:layout_width="match_parent"
6     android:layout_height="match_parent"
7     android:orientation="vertical"
8     tools:context=".MainActivity" >
9
10    <TextView
11        android:id="@+id/textView1"
12        android:layout_width="match_parent"
13        android:layout_height="wrap_content"
14        android:background="#008000"
15        android:gravity="center"
16        android:text="Chương trình quản lý nhân viên"
17        android:textColor="#FFFFFF"
18        android:textSize="20sp" />
19
20    <TableLayout
21        android:layout_width="match_parent"
22        android:layout_height="wrap_content"
23        android:stretchColumns="*"
24    >
25
26    <TableRow
27        android:id="@+id/tableRow1"
28        android:layout_width="wrap_content"
29        android:layout_height="wrap_content" >
30
31        <TextView
32            android:id="@+id/textView2"
33            android:layout_width="wrap_content"
34            android:layout_height="wrap_content"
35            android:text="Mã Phòng Ban:" />
36
37        <EditText
38            android:id="@+id/editmapb"
39            android:layout_width="wrap_content"
40            android:layout_height="wrap_content"
41            android:ems="10" >
42
43            <requestFocus />
44        </EditText>
45
46    </TableRow>
47
48    <TableRow
49        android:id="@+id/tableRow2"
50        android:layout_width="wrap_content"
51        android:layout_height="wrap_content" >
52
53        <TextView
54            android:id="@+id/textView3"
55            android:layout_width="wrap_content"
56            android:layout_height="wrap_content"
57            android:text="Tên Phòng Ban:" />
58
59        <EditText
```

```

59     android:id="@+id/editTenpb"
60     android:layout_width="wrap_content"
61     android:layout_height="wrap_content"
62     android:ems="10" />
63
64 </TableRow>
65
66 <TableRow
67     android:id="@+id/tableRow3"
68     android:layout_width="wrap_content"
69     android:layout_height="wrap_content" >
70
71 <Button
72     android:id="@+id/btnluupb"
73     android:layout_width="wrap_content"
74     android:layout_height="wrap_content"
75     android:layout_column="1"
76     android:text="Lưu Phòng Ban" />
77
78 </TableRow>
79 </TableLayout>
80
81 <TextView
82     android:id="@+id/textView4"
83     android:layout_width="match_parent"
84     android:layout_height="wrap_content"
85     android:background="#008000"
86     android:text="Danh sách phòng ban"
87     android:textColor="#FFFFFF"
88     android:textSize="15sp" />
89
90 <ListView
91     android:id="@+id/lvphongban"
92     android:layout_width="match_parent"
93     android:layout_height="wrap_content" >
94 </ListView>
95
96 </LinearLayout>

```

- Ta xem xử lý coding của MainActivity:

```

1 package tranduythanh.com.activity;
2
3 import java.util.ArrayList;
4
5 import tranduythanh.com.activity.R;
6 import tranduythanh.com.adapter.PhongBanAdapter;
7 import tranduythanh.com.model.ChucVu;
8 import tranduythanh.com.model.NhanVien;
9 import tranduythanh.com.model.PhongBan;
10 import android.os.Bundle;
11 import android.view.ContextMenu;
12 import android.view.MenuItem;
13 import android.view.View;

```

```

14 import android.view.ContextMenu.ContextMenuItemInfo;
15 import android.view.View.OnClickListener;
16 import android.widget.AdapterView;
17 import android.widget.AdapterView.OnItemLongClickListener;
18 import android.widget.Button;
19 import android.widget.EditText;
20 import android.widget.ListView;
21 import android.app.Activity;
22 import android.app.AlertDialog;
23 import android.content.DialogInterface;
24 import android.content.Intent;
25 /**
26 * Class MainActivity của ứng dụng
27 * @author drthanh
28 *
29 */
30 public class MainActivity extends Activity {
31 //Khai báo các Request + Result code để xử lý Intent for
32 result
33 public static final int MO_ACTIVITY_THEME_NHAN_VIEN=1;
34 public static final int MO_ACTIVITY_SUA_NHAN_VIEN=2;
35 public static final int THEME_NHAN_VIEN_THANHCONG=3;
36 public static final int SUA_NHAN_VIEN_THANHCONG=4;
37 public static final int XEM_DS_NHAN_VIEN=5;
38 public static final int CAPNHAT_DS_NHAN_VIEN_THANHCONG=6;
39 public static final int MO_ACTIVITY_THIET_LAP_TP_PP=7;
40 public static final int THIET_LAP_TP_PP_THANHCONG=8;
41 public static final int MO_ACTIVITY_CHUYENPHONG=9;
42 public static final int CHUYENPHONG_THANHCONG=10;
43
44 private Button btnLuuPb;
45 private EditText editMapb,editTenpb;
46 private ListView lvpb;
47 private static ArrayList<PhongBan>arrPhongBan=new
48 ArrayList<PhongBan>();
49 //ở đây dùng PhongBanAdapter
50 private PhongBanAdapter adapterPb=null;
51 private PhongBan pbSelected=null;
52 @Override
53 protected void onCreate(Bundle savedInstanceState) {
54 super.onCreate(savedInstanceState);
55 setContentView(R.layout.activity_main);
56 getFormWidgets();
57 addEvents();
58 fakeData();
59 }
60 /**
61 * Hàm này dùng để tải dữ liệu tạm (đó phải nhập tay)
62 * Tôi dùng hàm này để ví dụ cho lẹ
63 * bạn có thể xóa nó đi (nói chung vô thường vô phạt)
64 */
65 public void fakeData()
66 {
67 NhanVien nv=null;
68 PhongBan pb=new PhongBan("pb1", "Kỹ thuật");

```

```

68     nv=new NhanVien("m4", "Đoàn Ái Nương", true);
69     nv.setChucvu(ChucVu.TruongPhong);
70     pb.themNv(nv);
71     nv=new NhanVien("m5", "Trần Đạo Đức", false);
72     nv.setChucvu(ChucVu.PhopPhong);
73     pb.themNv(nv);
74     nv=new NhanVien("m6", "Nguyễn Đại Tài", false);
75     nv.setChucvu(ChucVu.PhopPhong);
76     pb.themNv(nv);
77     arrPhongBan.add(pb);
78     pb=new PhongBan("pb2", "Dịch vụ");
79     arrPhongBan.add(pb);
80     pb=new PhongBan("pb3", "Truyền thông");
81     arrPhongBan.add(pb);
82     nv=new NhanVien("m1", "Nguyễn Văn Tèo", false);
83     nv.setChucvu(ChucVu.NhanVien);
84     pb.themNv(nv);
85     nv=new NhanVien("m2", "Nguyễn Thị Téo", true);
86     nv.setChucvu(ChucVu.TruongPhong);
87     pb.themNv(nv);
88     nv=new NhanVien("m3", "Nguyễn Văn Teo", false);
89     nv.setChucvu(ChucVu.NhanVien);
90     pb.themNv(nv);

91     adapterPb.notifyDataSetChanged();
92 }
93 /**
94 * - Hàm này để load các control theo Id
95 * - Thiết lập Adapter cho ListView phòng ban
96 * - thiết lập context Menu cho ListView
97 */
98 public void getFormWidgets()
99 {
100     btnLuuPb=(Button) findViewById(R.id.btnluupb);
101     editMapb=(EditText) findViewById(R.id.editmapb);
102     editTenpb=(EditText) findViewById(R.id.edittenpb);
103     lvpb=(ListView) findViewById(R.id.lvphongban);
104     //khởi tạo đối tượng phòng ban adapter
105     //dùng layout_item_custom.xml
106     adapterPb=new PhongBanAdapter(this,
107             R.layout.layout_item_custom,
108             arrPhongBan);
109     lvpb.setAdapter(adapterPb);
110     //Đăng ký contextmenu cho Listview
111     registerForContextMenu(lvpb);
112 }
113 /**
114 * hàm gán sự kiện cho các control: button, ListView
115 */
116 public void addEvents()
117 {
118     //Bấm lưu để lưu phòng ban
119     btnLuuPb.setOnClickListener(new OnClickListener() {
120
121     @Override

```

```

122 public void onClick(View arg0) {
123     // TODO Auto-generated method stub
124     doLuuPhongBan();
125 }
126 });
127 //xử lý lưu biến tạm khi nhấn long - click
128 //phải dùng cái này để biết được trước đó đã chọn item nào
129 lvpb.setOnItemLongClickListener(new OnItemLongClickListener()
130 {
131
132     @Override
133     public boolean onItemLongClick(AdapterView<?> arg0, View arg1,
134     int arg2, long arg3) {
135         //lưu vết lại đối tượng thứ arg2 trong danh sách
136         pbSelected=arrPhongBan.get(arg2);
137         return false;
138     }
139 });
140 }
141 /**
142 * hàm xử lý lưu phòng ban
143 * đơn thuần là đưa phòng ban mới vào ArrayList
144 * chú ý bạn phải kiểm tra trùng lắp Id, hay các
145 * thông tin đã đầy đủ chưa ở đây
146 * gọi adapterPb.notifyDataSetChanged(); để cập nhật ListView
147 */
148 public void doLuuPhongBan()
149 {
150     String mapb=editMapb.getText()+"";
151     String tenpb=editTenpb.getText()+"";
152     PhongBan pb=new PhongBan(mapb, tenpb);
153     arrPhongBan.add(pb);
154     adapterPb.notifyDataSetChanged();
155 }
156 /**
157 * hàm này để gán COntextMenu vào ứng dụng
158 * đã học từ lâu rồi, phải tự hiểu
159 */
160 @Override
161 public void onCreateContextMenu(ContextMenu menu, View v,
162 ContextMenuItem menuInfo) {
163     super.onCreateContextMenu(menu, v, menuInfo);
164     getMenuInflater().inflate(R.menu.contextmenu_phongban, menu);
165 }
166 /**
167 * hàm này để xử lý người sử dụng vừa chọn menuitem nào
168 * ta dựa vào Id để xử lý
169 */
170 @Override
171 public boolean onContextItemSelected(MenuItem item) {
172     switch(item.getItemId())
173     {
174     case R.id.mnuthemnv:
175         doThemNhanVien();

```

```

176    break;
177    case R.id.mnudanhhsachnv:
178        doDanhSachNhanVien();
179        break;
180    case R.id.mnutruongphong:
181        doThietLapLanhDao();
182        break;
183    case R.id.mnuxoapb:
184        doXoaPhongBan();
185        break;
186    }
187    return super.onContextItemSelected(item);
188 }
189 /**
190 * khi chọn menu Thêm nhân viên, đơn thuần chỉ là hiển thị
191 * màn hình thêm nhân viên (với dạng Dialog)
192 * nhớ là dùng startActivityForResult
193 * do đó ta lắng nghe kết quả tại hàm onActivityResult
194 */
195 public void doThemNhanVien()
196 {
197     Intent i=new Intent(this, ThemNhanVienActivity.class);
198     startActivityForResult(i, MO_ACTIVITY_THEME_NHAN_VIEN);
199 }
200 /**
201 * hàm onActivityResult để xử lý kết quả trả về
202 * sau khi gọi startActivityForResult kết thúc
203 */
204 @Override
205 protected void onActivityResult(int requestCode, int resultCode,
Intent data) {
206     // TODO Auto-generated method stub
207     super.onActivityResult(requestCode, resultCode, data);
208     //màn hình thêm mới nhân viên trả kết quả về
209     if(resultCode==THEM_NHAN_VIEN_THANHCONG)
210     {
211         //Cách lấy dữ liệu đã học rồi, ko nói lại
212         Bundle bundle= data.getBundleExtra("DATA");
213         NhanVien nv= (NhanVien) bundle.getSerializable("NHANVIEN");
214         pbSelected.themNv(nv);
215         adapterPb.notifyDataSetChanged();
216     }
217     //Màn hình thiết lập lạnh đảo/ cập nhật danh sách trả
218     //kết quả về
219     else if(resultCode==THIET_LAP_TP_PP_THANHCONG ||
220     resultCode==CAPNHAT_DS_NHAN_VIEN_THANHCONG)
221     {
222         //Cách lấy dữ liệu đã học rồi, ko nói lại
223         Bundle bundle= data.getBundleExtra("DATA");
224         PhongBan pb= (PhongBan) bundle.getSerializable("PHONGBAN");
225         //đơn thuần là xóa danh sách cũ
226         pbSelected.getListNhanVien().clear();
227         //rồi cập nhật lại toàn bộ danh sách mới
228         //sinh viên có thể chọn giải pháp thông minh hơn
229         //tức là phần tử nào bị ảnh hưởng thì xử lý phần tử đó

```

```

230 pbSelected.getListNhanVien().addAll(pb.getListNhanVien());
231 adapterPb.notifyDataSetChanged();
232 }
233 }
234 /**
235 * hàm cho phép danh xem danh sách nhân viên của phòng
236 * ban đang chọn, đơn thuần là mở DanhSachNhanVienActivity
237 * và truyền phòng ban qua -->toàn bộ danh sách nhân viên
238 * sẽ được hiển thị trong DanhSachNhanVienActivity
239 */
240 public void doDanhSachNhanVien()
241 {
242 Intent i=new Intent(this, DanhSachNhanVienActivity.class);
243 Bundle bundle=new Bundle();
244 bundle.putSerializable("PHONGBAN", pbSelected);
245 i.putExtra("DATA", bundle);
246 startActivityForResult(i, XEM_DS_NHAN_VIEN);
247 }
248 /**
249 * hàm cho phép mở màn hình thiết lập lãnh đạo phòng ban
250 * Trưởng phòng và phó phòng
251 * ThietLapTruongPhongActivity sẽ có 2 ListView
252 * - ListView 1 hiển thị dạng radiobutton để chỉ chọn 1 trưởng
253 phòng
254 * - ListView 2 hiển thị dạng Checkbox cho phép chọn nhiều phó
255 phòng
256 */
257 public void doThietLapLanhDao()
258 {
259 Intent i=new Intent(this, ThietLapTruongPhongActivity.class);
260 Bundle bundle=new Bundle();
261 bundle.putSerializable("PHONGBAN", pbSelected);
262 i.putExtra("DATA", bundle);
263 startActivityForResult(i, MO_ACTIVITY_THIET_LAP_TP_PP);
264 }
265 /**
266 * hàm cho phép xóa phòng ban đang chọn
267 */
268 public void doXoaPhongBan()
269 {
270 AlertDialog.Builder builder=new AlertDialog.Builder (this);
271 builder.setTitle("Hỏi xóa");
272 builder.setMessage("Bạn có chắc chắn muốn xóa
[ "+pbSelected.getTen()+" ]");
273 builder.setIcon(android.R.drawable.ic_input_delete);
274 builder.setNegativeButton("Không", new
DialogInterface.OnClickListener() {
275
276
277 @Override
278 public void onClick(DialogInterface arg0, int arg1) {
279 // TODO Auto-generated method stub
280 arg0.cancel();
281 }
282 });
283 builder.setPositiveButton("Yes", new

```

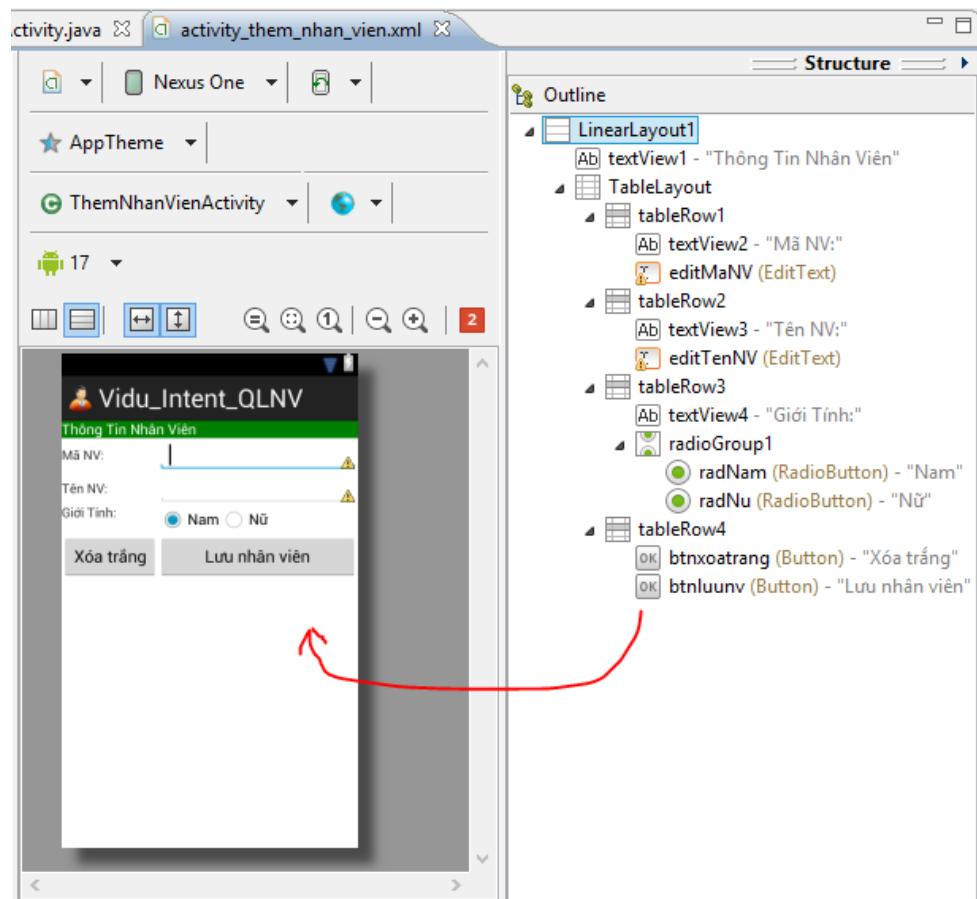
```

284     DialogInterface.OnClickListener() {
285
286     @Override
287     public void onClick(DialogInterface dialog, int which) {
288         // TODO Auto-generated method stub
289         arrPhongBan.remove(pbSelected);
290         adapterPb.notifyDataSetChanged();
291     }
292 });
293 builder.show();
294 }
295 /**
 * Tôi cố tình cung cấp hàm này để ở các Activity khác đều
 * có thể truy suất được danh sách phòng ban tổng thể
 * @return
 */
public static ArrayList<PhongBan> getListPhongBan()
{
    return arrPhongBan;
}
}

```

- Ta qua màn hình thêm nhân viên. Chú ý là màn hình này bạn phải cấu hình mở dạng Dialog.

- xem layout:



- Bạn xem XML source code của màn hình này:

```
1 <LinearLayout  
2     xmlns:android="http://schemas.android.com/apk/res/android"  
3     xmlns:tools="http://schemas.android.com/tools"  
4     android:id="@+id/LinearLayout1"  
5     android:layout_width="match_parent"  
6     android:layout_height="match_parent"  
7     android:orientation="vertical"  
8     tools:context=".ThemNhanVienActivity" >  
9  
10    <TextView  
11        android:id="@+id/textView1"  
12        android:layout_width="match_parent"  
13        android:layout_height="wrap_content"  
14        android:background="#008000"  
15        android:text="Thông Tin Nhân Viên"  
16        android:textColor="#FFFFFF"  
17        android:textSize="15sp" />  
18  
19    <TableLayout  
20        android:layout_width="match_parent"  
21        android:layout_height="wrap_content"  
22        android:stretchColumns="*"  
23    >  
24  
25    <TableRow  
26        android:id="@+id/tableRow1"  
27        android:layout_width="wrap_content"  
28        android:layout_height="wrap_content" >  
29  
30    <TextView  
31        android:id="@+id/textView2"  
32        android:layout_width="wrap_content"  
33        android:layout_height="wrap_content"  
34        android:text="Mã NV:" />  
35  
36    <EditText  
37        android:id="@+id/editMaNV"  
38        android:layout_width="wrap_content"  
39        android:layout_height="wrap_content"  
40        android:ems="10" >  
41  
42        <requestFocus />  
43    </EditText>  
44  
45    </TableRow>  
46  
47    <TableRow  
48        android:id="@+id/tableRow2"  
49        android:layout_width="wrap_content"  
50        android:layout_height="wrap_content" >  
51  
52    <TextView  
53        android:id="@+id/textView3"  
54        android:layout_width="wrap_content"  
55        android:layout_height="wrap_content"
```

```
55     android:text="Tên NV:" />
56
57 <EditText
58     android:id="@+id/editTenNV"
59     android:layout_width="wrap_content"
60     android:layout_height="wrap_content"
61     android:ems="10" />
62
63 </TableRow>
64
65 <TableRow
66     android:id="@+id/tableRow3"
67     android:layout_width="wrap_content"
68     android:layout_height="wrap_content" >
69
70 <TextView
71     android:id="@+id/textView4"
72     android:layout_width="wrap_content"
73     android:layout_height="wrap_content"
74     android:text="Giới Tính:" />
75
76 <RadioGroup
77     android:id="@+id/radioGroup1"
78     android:layout_width="wrap_content"
79     android:layout_height="wrap_content"
80     android:orientation="horizontal" >
81
82 <RadioButton
83     android:id="@+id/radNam"
84     android:layout_width="wrap_content"
85     android:layout_height="wrap_content"
86     android:checked="true"
87     android:text="Nam" />
88
89 <RadioButton
90     android:id="@+id/radNu"
91     android:layout_width="wrap_content"
92     android:layout_height="wrap_content"
93     android:text="Nữ" />
94 </RadioGroup>
95
96 </TableRow>
97
98 <TableRow
99     android:id="@+id/tableRow4"
100    android:layout_width="wrap_content"
101    android:layout_height="wrap_content" >
102
103 <Button
104     android:id="@+id/btnxoatrang"
105     android:layout_width="wrap_content"
106     android:layout_height="wrap_content"
107     android:text="Xóa trống" />
108 <Button
```

```

109     android:id="@+id/btnluunv"
110     android:layout_width="wrap_content"
111     android:layout_height="wrap_content"
112     android:text="Lưu nhân viên" />
113
114 </TableRow>
115 </TableLayout>
116
117 </LinearLayout>
118

```

- Giờ ta xem xử lý coding cho phần thêm nhân viên (**ThemNhanVienActivity**):

```

1 package tranduythanh.com.activity;
2
3 import tranduythanh.com.activity.R;
4 import tranduythanh.com.model.ChucVu;
5 import tranduythanh.com.model.NhanVien;
6 import android.os.Bundle;
7 import android.app.Activity;
8 import android.content.Intent;
9 import android.view.View;
10 import android.view.View.OnClickListener;
11 import android.widget.Button;
12 import android.widget.EditText;
13 import android.widget.RadioButton;
14 /**
15  * class này cho phép thêm mới một nhân viên vào
16  * Phòng ban đang chọn
17  * @author drthanh
18  *
19  */
20 public class ThemNhanVienActivity extends Activity {
21
22     private Button btnXoaTrang,btnLuuNhanVien;
23     private EditText editManv,editTenNv;
24     private RadioButton radNam;
25
26     @Override
27     protected void onCreate(Bundle savedInstanceState) {
28         super.onCreate(savedInstanceState);
29         setContentView(R.layout.activity_them_nhan_vien);
30         getFormWidgets();
31         addEvents();
32     }
33 /**
34  * hàm lấy các control theo id
35  */
36     public void getFormWidgets()
37     {
38         btnXoaTrang=(Button) findViewById(R.id.btnxoatrang);
39         btnLuuNhanVien=(Button) findViewById(R.id.btnluunv);
40         editManv=(EditText) findViewById(R.id.editMaNV);
41         editTenNv=(EditText) findViewById(R.id.editTenNV);

```

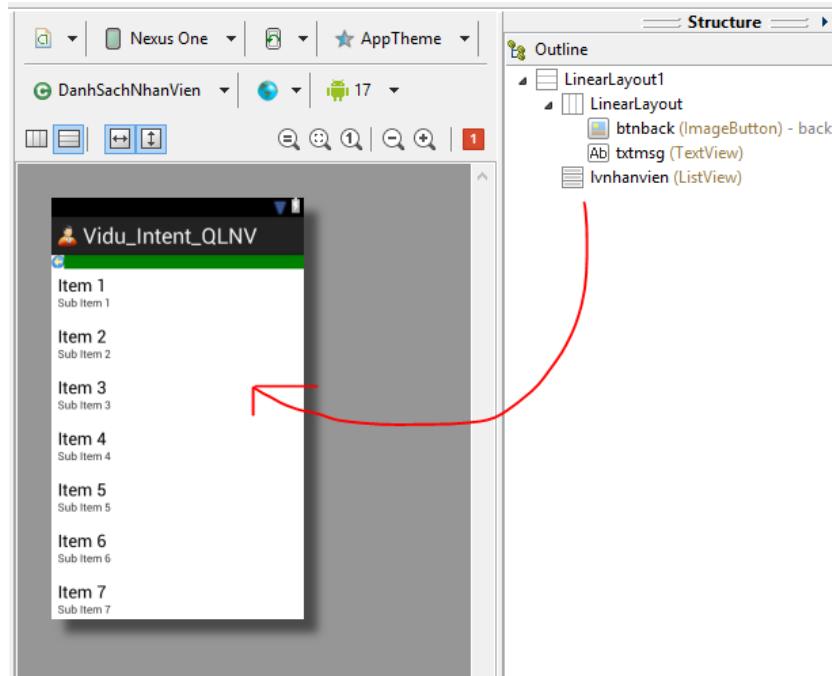
```

42     radNam=(RadioButton) findViewById(R.id.radNam);
43 }
44 /**
45 * hàm gán sự kiện
46 */
47 public void addEvents()
48 {
49     btnXoaTrang.setOnClickListener(new OnClickListener() {
50
51     @Override
52     public void onClick(View arg0) {
53         // TODO Auto-generated method stub
54         doXoaTrang();
55     }
56 });
57     btnLuuNhanVien.setOnClickListener(new OnClickListener() {
58
59     @Override
60     public void onClick(View v) {
61         // TODO Auto-generated method stub
62         doLuuNhanVien();
63     }
64 });
65 }
66 /**
67 * đơn thuần là xóa trắng dữ liệu của các control
68 * rồi focus tới mã
69 */
70 public void doXoaTrang()
71 {
72     editManv.setText("");
73     editTenNv.setText("");
74     editManv.requestFocus();
75 }
76 /**
77 * hàm lưu nhân viên theo phòng ban
78 * truyền nhân viên qua MainActivity
79 * MainActivity có nhiệm vụ lấy được nhân viên này
80 * và đưa họ vào phòng ban chọn lúc nãy
81 */
82 public void doLuuNhanVien()
83 {
84     NhanVien nv=new NhanVien();
85     nv.setMa(editManv.getText()+"");
86     nv.setTen(editTenNv.getText()+"");
87     nv.setChucvu(ChucVu.NhanVien);
88     nv.setGioitinh(!radNam.isChecked());
89     Intent i=getIntent();
90     Bundle bundle=new Bundle();
91     bundle.putSerializable("NHANVIEN", nv);
92     i.putExtra("DATA", bundle);
93     setResult(MainActivity.THEM_NHAN_VIEN_THANHCONG, i);
94     finish();
95 }

```

- Tiếp tục ta xem màn hình Xem danh sách nhân viên:

- bạn xem outline:



- Xem XML Resource của **activity_danh_sach_nhan_vien.xml**:

```

1 <LinearLayout
2   xmlns:android="http://schemas.android.com/apk/res/android"
3   xmlns:tools="http://schemas.android.com/tools"
4   android:id="@+id/LinearLayout1"
5   android:layout_width="match_parent"
6   android:layout_height="match_parent"
7   android:orientation="vertical"
8   tools:context=".DanhSachNhanVienActivity" >
9
10  <LinearLayout
11    android:layout_width="match_parent"
12    android:layout_height="wrap_content" >
13
14    <ImageButton
15      android:id="@+id/btnback"
16      android:layout_width="16dp"
17      android:layout_height="16dp"
18      android:contentDescription="TODO"
19      android:src="@drawable/back" />
20
21    <TextView
22      android:id="@+id/txtmsg"
23      android:layout_width="match_parent"
24      android:layout_height="wrap_content"
25      android:background="#008000"
26      android:textColor="#FFFFFF" />

```

```

26
27 </LinearLayout>
28
29 <ListView
30     android:id="@+id/lvnhhanvien"
31     android:layout_width="match_parent"
32     android:layout_height="wrap_content" >
33 </ListView>
34
35 </LinearLayout>

```

- Xem coding xử lý phần xem danh sách nhân viên (**DanhSachNhanVienActivity**) :

- Class này rất phức tạp, nó còn chứa Context menu để cho phép:

+ Sửa nhân viên

+ Chuyển phòng ban cho nhân viên

+ Xóa nhân viên:

```

1 package tranduythanh.com.activity;
2
3 import java.util.ArrayList;
4
5 import tranduythanh.com.activity.R;
6 import tranduythanh.com.adapter.NhanVienAdapter;
7 import tranduythanh.com.model.NhanVien;
8 import tranduythanh.com.model.PhongBan;
9 import android.os.Bundle;
10 import android.view.ContextMenu;
11 import android.view.MenuItem;
12 import android.view.View;
13 import android.view.ContextMenu.ContextMenuItemInfo;
14 import android.view.View.OnClickListener;
15 import android.widget.AdapterView;
16 import android.widget.ImageButton;
17 import android.widget.ListView;
18 import android.widget.TextView;
19 import android.widget.AdapterView.OnItemLongClickListener;
20 import android.app.Activity;
21 import android.app.AlertDialog;
22 import android.content.DialogInterface;
23 import android.content.Intent;
24 /**
25 * class này dùng để hiển thị danh sách nhân viên
26 * Class này cũng phức tạp không kém MainActivity
27 * nhớ là dùng custom layout
28 * - Hiển thị danh sách
29 * - cho phép chỉnh sửa
30 * - cho phép chuyển phòng ban
31 * - cho phép xóa nhân viên
32 * @author drthanh

```

```

33     *
34     */
35     public class DanhSachNhanVienActivity extends Activity {
36
37     TextView txtmsg;
38     ImageButton btnback;
39     ListView lvNhanvien;
40     ArrayList<NhanVien> arrNhanvien=null;
41     //Nhân viên Adapter để hiển thị thông tin
42     //và chi tiết : chức vụ, giới tính
43     NhanVienAdapter adapter=null;
44     PhongBan pb=null;
45     private NhanVien nvSelected=null;
46     private int position=-1;
47     @Override
48     protected void onCreate(Bundle savedInstanceState) {
49         super.onCreate(savedInstanceState);
50         setContentView(R.layout.activity_danh_sach_nhan_vien);
51         txtmsg=(TextView) findViewById(R.id.txtmsg);
52         btnback=(ImageButton) findViewById(R.id.btnback);
53         lvNhanvien=(ListView) findViewById(R.id.lvnhanvien);
54         getDataFromMain();
55         addEvents();
56         registerForContextMenu(lvNhanvien);
57     }
58     /**
59     * hàm lấy giá trị từ MainActivity
60     * sẽ truyền phòng ban qua
61     * bên này lấy phòng ban và dựa vào phòng ban này
62     * lấy ra danh sách nhân viên
63     */
64     public void getDataFromMain()
65     {
66         Intent i=getIntent();
67         Bundle b=i.getBundleExtra("DATA");
68         pb= (PhongBan) b.getSerializable("PHONGBAN");
69         arrNhanvien=pb.getListNhanVien();
70         adapter=new NhanVienAdapter(this,
71             R.layout.layout_item_custom,
72             arrNhanvien);
73         lvNhanvien.setAdapter(adapter);
74         txtmsg.setText("DS nhân viên ["+pb.getTen()+"]");
75     }
76     /**
77     * hàm gán sự kiện (đã quá quen thuộc rồi)
78     * có lưu vết lại nhân viên vừa chọn để xử lý
79     * cho contextmenu
80     */
81     public void addEvents()
82     {
83         btnback.setOnClickListener(new OnClickListener() {
84
85             @Override
86             public void onClick(View arg0) {

```

```

87    }
88    });
89    lvNhanvien.setOnItemLongClickListener(new
90    OnItemLongClickListener() {
91
92    @Override
93    public boolean onItemLongClick(AdapterView<?> arg0, View
94    arg1,
95    int arg2, long arg3) {
96    nvSelected=arrNhanvien.get(arg2);
97    position=arg2;
98    return false;
99    }
100   });
101  }
102  /**
103  * hàm gán context menu (đã quá quen thuộc)
104  */
105  @Override
106  public void onCreateContextMenu(ContextMenu menu, View v,
107  ContextMenuItemInfo menuInfo) {
108  // TODO Auto-generated method stub
109  super.onCreateContextMenu(menu, v, menuInfo);
110  getMenuInflater().inflate(R.menu.contextmenu_nhanvien,
111  menu);
112  }
113  /**
114  * hàm xử lý sự kiện chọn menuitem (đã quen thuộc quá)
115  */
116  @Override
117  public boolean onContextItemSelected(MenuItem item) {
118  // TODO Auto-generated method stub
119  switch(item.getItemId())
120  {
121  case R.id.mnusuavn:
122  doSuaNhanVien();
123  break;
124  case R.id.mnuchuyenpb:
125  doChuyenPhongBan();
126  break;
127  case R.id.mnuxoany:
128  doXoaNhanVien();
129  break;
130  }
131  return super.onContextItemSelected(item);
132  }
133  /**
134  * hàm onActivityResult xử lý kết quả trả về
135  * cho trường hợp sử dụng COntext Menu để mở các
136  * Activity khác
137  */
138  @Override
139  protected void onActivityResult(int requestCode, int
140  resultCode, Intent data) {

```

```

141 // TODO Auto-generated method stub
142 super.onActivityResult(requestCode, resultCode, data);
143 //lấy kết quả sửa nhân viên thành công
144 if(resultCode==MainActivity.SUA_NHAN_VIEN_THANHCONG)
145 {
146 Bundle b=data.getBundleExtra("DATA");
147 NhanVien nv= (NhanVien) b.getSerializable("NHANVIEN");
148 arrNhanvien.set(position, nv);
149 adapter.notifyDataSetChanged();
150 }
151 //lấy kết quả chuyển phòng ban thành công
152 else if(resultCode==MainActivity.CHUYENPHONG_THANHCONG)
153 {
154 arrNhanvien.remove(nvSelected);
155 adapter.notifyDataSetChanged();
156 }
157 }
158 /**
159 * hàm sửa nhân viên
160 * đơn giản là mở Activity sửa nhân viên lên
161 * rồi truyền nhân viên đang chọn qua Activity đó
162 *
163 */
164 public void doSuaNhanVien()
165 {
166 Intent i=new Intent(this, SuaNhanVienActivity.class);
167 Bundle b=new Bundle();
168 b.putSerializable("NHANVIEN", nvSelected);
169 i.putExtra("DATA", b);
170 startActivityForResult(i,
171 MainActivity.MO_ACTIVITY_SUA_NHAN_VIEN);
172 }
173 /**
174 * hàm chuyển phòng ban cho nhân viên đang chọn
175 * đơn thuần là mở Activity chuyển phòng ban
176 * Activity này có nhiệm vụ hiển thị toàn bộ phòng ban
177 * rồi cho phép người sử dụng chọn phòng ban để chuyển
178 */
179 public void doChuyenPhongBan()
180 {
181 Intent i=new Intent(this, ChuyenPhongBanActivity.class);
182 Bundle b=new Bundle();
183 b.putSerializable("NHANVIEN", nvSelected);
184 i.putExtra("DATA", b);
185 startActivityForResult(i,
186 MainActivity.MO_ACTIVITY_CHUYENPHONG);
187 }
188 /**
189 * hàm cho phép xóa nhân viên hiện tại
190 * (đã quen thuộc quá)
191 */
192 public void doXoaNhanVien()
193 {
194 AlertDialog.Builder builder=new AlertDialog.Builder (this);
builder.setTitle("Hỏi xóa");

```

```

195     builder.setMessage("Bạn có chắc chắn muốn xóa
196     [" + nvSelected.getTen() + "]");
197     builder.setIcon(android.R.drawable.ic_input_delete);
198     builder.setNegativeButton("Không", new
199     DialogInterface.OnClickListener() {
200
201         @Override
202         public void onClick(DialogInterface arg0, int arg1) {
203             // TODO Auto-generated method stub
204             arg0.cancel();
205         }
206     });
207     builder.setPositiveButton("Đồng ý", new
208     DialogInterface.OnClickListener() {
209
210         @Override
211         public void onClick(DialogInterface dialog, int which) {
212             // TODO Auto-generated method stub
213             arrNhanvien.remove(nvSelected);
214             adapter.notifyDataSetChanged();
215         }
216     });
217     builder.show();
218 }
219 /**
220 * khi nhấn nút Back (hình back)
221 * thì truyền thông số về MainActivity để cập nhật
222 * phòng ban
223 */
224 public void doUpdateToMain()
225 {
226     Intent i = getIntent();
227     Bundle b = new Bundle();
228     b.putSerializable("PHONGBAN", pb);
229     i.putExtra("DATA", b);
230     setResult(MainActivity.CAPNHAT_DS_NHAN_VIEN_THANHCONG, i);
231     finish();
232 }
}

```

- Ta tiếp tục làm việc với màn hình sửa nhân viên, vì màn hình này sử dụng chung Layout với màn hình thêm nhân viên. Nên Tôi chỉ mô tả phần xử lý coding mà thôi (**SuaNhanVienActivity**):

```

1 package tranduythanh.com.activity;
2
3 import tranduythanh.com.model.NhanVien;
4 import android.app.Activity;
5 import android.content.Intent;
6 import android.os.Bundle;
7 import android.view.View;
8 import android.view.View.OnClickListener;
9 import android.widget.Button;
10 import android.widget.EditText;
11 import android.widget.RadioButton;

```

```

12 /**
13  * class này cho phép sửa nhân viên được chọn trong
14 Danhsachnhanvien
15  * sử dụng chung Layout với Thêm nhân viên
16  * Không cho sửa mã số nhân viên
17  * khi bấm lưu thì cập nhật lại (rất đơn giản)
18  * @author drthanh
19  *
20 */
21 public class SuaNhanVienActivity extends Activity{
22     EditText editMa,editTen;
23     RadioButton radNam;
24     Button btnClear,btnSave;
25     NhanVien nv=null;
26     @Override
27     protected void onCreate(Bundle savedInstanceState) {
28         // TODO Auto-generated method stub
29         super.onCreate(savedInstanceState);
30         setContentView(R.layout.activity_them_nhan_vien);
31         getFormWidgets();
32         getDefaultData();
33         addEvents();
34     }
35     /**
36      * hàm lấy các control theo id
37     */
38     public void getFormWidgets()
39     {
40         editMa=(EditText) findViewById(R.id.editMaNV);
41         editTen=(EditText) findViewById(R.id.editTenNV);
42         radNam=(RadioButton) findViewById(R.id.radNam);
43         editMa.setEnabled(false);
44         editTen.requestFocus();

45         btnClear=(Button) findViewById(R.id.btnxoatrang);
46         btnSave=(Button) findViewById(R.id.btnluunv);
47     }
48     /**
49      * hàm lấy giá trị từ DanhSachNhanVienActivity truyền qua
50      * để hiển thị
51     */
52     public void getDefaultData()
53     {
54         Intent i =getIntent();
55         Bundle b=i.getBundleExtra("DATA");
56         nv=(NhanVien) b.getSerializable("NHANVIEN");
57         editMa.setText(nv.getMa());
58         editTen.setText(nv.getTen());
59         radNam.setChecked(true);
60         if(nv.isGioitinh())
61             radNam.setChecked(false);
62     }
63     public void addEvents()
64     {
65         btnClear.setOnClickListener(new OnClickListener() {

```

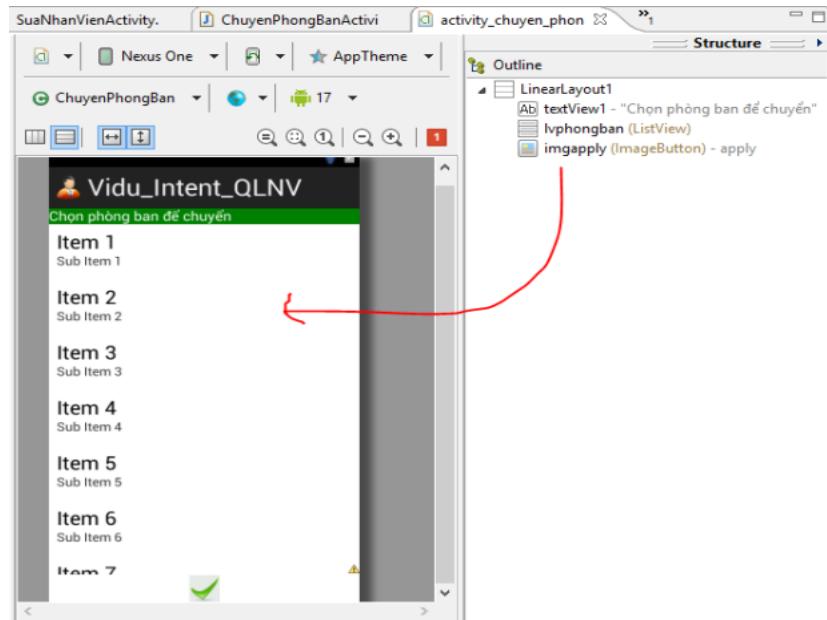
```

66
67     @Override
68     public void onClick(View arg0) {
69         // TODO Auto-generated method stub
70         editTen.setText("");
71         editTen.requestFocus();
72     }
73 }
74 /**
75 * bấm lưu thì đóng cửa sổ sửa này
76 * và truyền dữ liệu qua màn hình cha để nó tự cập nhật
77 */
78 btnSave.setOnClickListener(new OnClickListener() {
79
80     @Override
81     public void onClick(View v) {
82         // TODO Auto-generated method stub
83         Intent i =getIntent();
84         nv.setTen(editTen.getText()+"");
85         nv.setGioitinh(!radNam.isChecked());
86         Bundle b=new Bundle();
87         b.putSerializable("NHANVIEN", nv);
88         i.putExtra("DATA", b);
89         setResult(MainActivity.SUA_NHAN_VIEN_THANHCONG, i);
90         finish();
91     }
92 }
93 }
94 }

```

- Ta qua màn hình chuyển phòng ban cho nhân viên:

- Bạn xem layout XML:



- Màn hình này là màn hình con của màn hình Xem danh sách nhân viên (lúc dùng Context Menu).

- Khi chọn nhân viên nào thì sẽ tiến hành mở màn hình chuyển phòng ban cho nhân viên đó.

- xem XML Resource:

```
1 <LinearLayout  
2     xmlns:android="http://schemas.android.com/apk/res/android"  
3     xmlns:tools="http://schemas.android.com/tools"  
4     android:id="@+id/LinearLayout1"  
5     android:layout_width="match_parent"  
6     android:layout_height="match_parent"  
7     android:orientation="vertical"  
8     tools:context=".ChuyenPhongBanActivity" >  
9  
10    <TextView  
11        android:id="@+id/textView1"  
12        android:layout_width="match_parent"  
13        android:layout_height="wrap_content"  
14        android:background="#008000"  
15        android:text="Chọn phòng ban để chuyển"  
16        android:textColor="#FFFFFF"  
17        android:textSize="15sp" />  
18  
19    <ListView  
20        android:id="@+id/lvphongban"  
21        android:layout_width="match_parent"  
22        android:layout_height="wrap_content"  
23        android:layout_weight="3" >  
24    </ListView>  
25  
26    <ImageButton  
27        android:id="@+id/imgapply"  
28        android:layout_width="32dp"  
29        android:layout_height="32dp"  
30        android:layout_gravity="center"  
31        android:contentDescription="Thiết lập"  
32        android:src="@drawable/apply" />  
33    </LinearLayout>
```

- Bạn xem xử lý coding cho phần chuyển phòng ban:

```
1 package tranduythanh.com.activity;  
2  
3 import java.util.ArrayList;  
4  
5 import tranduythanh.com.model.NhanVien;  
6 import tranduythanh.com.model.PhongBan;  
7 import android.os.Bundle;  
8 import android.app.Activity;  
9 import android.content.Intent;  
10 import android.view.View;  
11 import android.view.View.OnClickListener;  
12 import android.widget.AdapterView;  
13 import android.widget.AdapterView.OnItemClickListener;
```

```

14 import android.widget.ArrayAdapter;
15 import android.widget.CheckedTextView;
16 import android.widget.ImageButton;
17 import android.widget.ListView;
18 /**
19 * Class dùng để xử lý chuyển phòng ban cho nhân viên
20 * @author drthanh
21 *
22 */
23 public class ChuyenPhongBanActivity extends Activity {
24
25 ListView lvPb;
26 private static ArrayList<PhongBan>arrPhongBan=null;
27 ArrayAdapter<PhongBan>adapter;
28 ImageButton btnApply;
29 NhanVien nv=null;
30 @Override
31 protected void onCreate(Bundle savedInstanceState) {
32 super.onCreate(savedInstanceState);
33 setContentView(R.layout.activity_chuyen_phong_ban);
34 getFormWidgets();
35 //lấy nhân viên từ màn hình xem danh sách nhân viên
36 Intent i=getIntent();
37 Bundle b= i.getBundleExtra("DATA");
38 nv=(NhanVien) b.getSerializable("NHANVIEN");
39 }
40 /**
41 * hàm lấy control theo id
42 * đồng thời load toàn bộ danh sách phòng ban ở MainActivity
43 * lên ListView để sử dụng
44 * android.R.layout.simple_list_item_single_choice ->dùng Radio
45 * Bắt buộc phải xử lý hàm: lvPb.setOnItemClickListener
46 * để gán checked cho Radio
47 */
48 public void getFormWidgets()
49 {
50 lvPb=(ListView) findViewById(R.id.lvphongban);
51 btnApply=(ImageButton) findViewById(R.id.imgapply);
52 arrPhongBan=MainActivity.getListPhongBan();
53 adapter=new ArrayAdapter<PhongBan>
54 (this, android.R.layout.simple_list_item_single_choice,
55 arrPhongBan);
56 lvPb.setAdapter(adapter);
57
58 lvPb.setOnItemClickListener(new OnItemClickListener() {
59
60 boolean somethingChecked = false;
61 @Override
62 public void onItemClick(AdapterView<?> arg0, View arg1, int
63 arg2,
64 long arg3) {
65 //hiển nhiên View arg1 là CheckedTextView
66 if(somethingChecked){
67 CheckedTextView cv = (CheckedTextView) arg1;

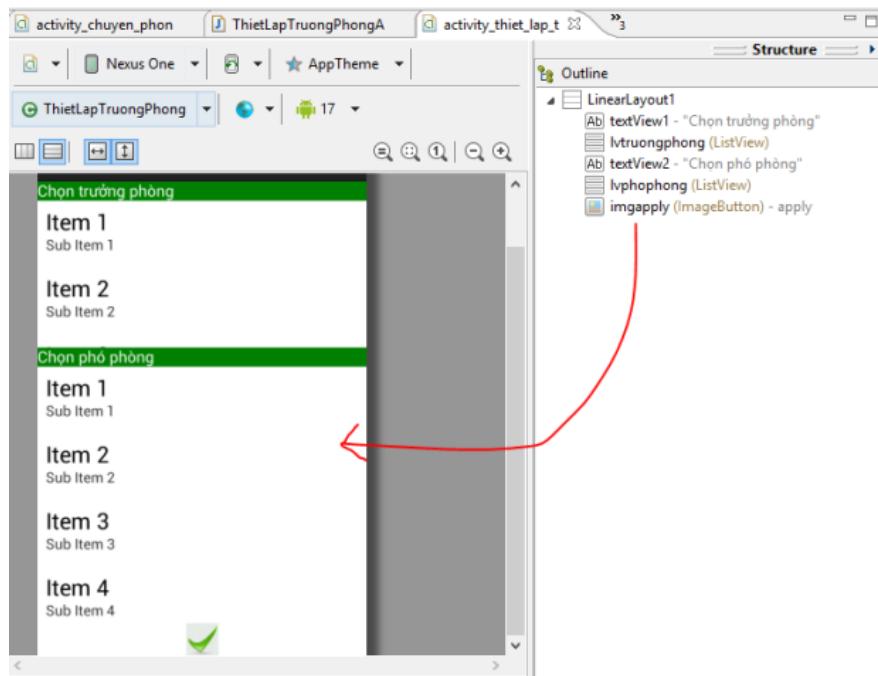
```

```

68     cv.setChecked(false);
69 }
70 CheckedTextView cv = (CheckedTextView) arg1;
71 if(!cv.isChecked())
72 {
73     cv.setChecked(true);
74     arrPhongBan.get(arg2).themNv(nv);
75 }
76 somethingChecked=true;
77 }
78 });
79 //khi chọn nút Apply thì tiến hành đóng màn hình này
80 //và truyền lệnh về cho DanhSachNhanVienActivity
81 btnApply.setOnClickListener(new OnClickListener() {
82
83     @Override
84     public void onClick(View arg0) {
85         doApply();
86     }
87 }
88 });
89 }
90 public void doApply()
91 {
92     setResult(MainActivity.CHUYENPHONG_THANHCONG);
93     finish();
94 }
95 }
96 }

```

- Ta qua màn hình cuối cùng: Thiết lập trưởng phó phòng cho nhân viên
- Màn hình này là màn hình con của MainActivity.
- Khi chọn phòng ban nào thì thiết lập trưởng phó phòng cho phòng ban đó.
- xem Layout (**activity_thiet_lap_truong_phong.xml**):



- Xem XML resource:

```
1    </pre>
2    <LinearLayout
3        xmlns:android="http://schemas.android.com/apk/res/android"
4        xmlns:tools="http://schemas.android.com/tools"
5        android:id="@+id/LinearLayout1"
6        android:layout_width="match_parent"
7        android:layout_height="match_parent"
8        android:orientation="vertical"
9        tools:context=".ThietLapTruongPhongActivity" >
10
11    <TextView
12        android:id="@+id/textView1"
13        android:layout_width="match_parent"
14        android:layout_height="wrap_content"
15        android:background="#008000"
16        android:text="Chọn trưởng phòng"
17        android:textColor="#FFFFFF"
18        android:textSize="15sp" />
19
20    <ListView
21        android:id="@+id/lvtruongphong"
22        android:layout_width="match_parent"
23        android:layout_height="wrap_content"
24        android:layout_weight="3" >
25
26    </ListView>
27
28    <TextView
29        android:id="@+id/textView2"
30        android:layout_width="match_parent"
31        android:layout_height="wrap_content"
32        android:background="#008000"
33        android:text="Chọn phó phòng"
```

```

34     android:textColor="#FFFFFF"
35     android:textSize="15sp" />
36
37 <ListView
38     android:id="@+id/lvphophong"
39     android:layout_width="match_parent"
40     android:layout_height="wrap_content"
41     android:layout_weight="2" >
42 </ListView>
43
44 <ImageButton
45     android:id="@+id/imgapply"
46     android:layout_width="32dp"
47     android:layout_height="32dp"
48     android:layout_gravity="center"
49     android:contentDescription="Thiết lập"
50     android:src="@drawable/apply" />
51
52 </LinearLayout>
<pre>
```

- xem xử lý coding:

```

1  package tranduythanh.com.activity;
2
3  import java.util.ArrayList;
4
5  import tranduythanh.com.activity.R;
6  import tranduythanh.com.model.ChucVu;
7  import tranduythanh.com.model.NhanVien;
8  import tranduythanh.com.model.PhongBan;
9  import android.os.Bundle;
10 import android.view.View;
11 import android.view.View.OnClickListener;
12 import android.widget.AdapterView;
13 import android.widget.AdapterView.OnItemClickListener;
14 import android.widget.ArrayAdapter;
15 import android.widget.CheckedTextView;
16 import android.widget.ImageButton;
17 import android.widget.ListView;
18 import android.app.Activity;
19 import android.content.Intent;
20 /**
21 * màn hình sẽ hiển thị danh sách nhân viên vào 2 ListView
22 * khác nhau
23 * * ListView 1 dùng Radio để chọn trưởng phòng
24 * * ListView 2 dùng Checkbox để chọn phó phòng
25 * @author drthanh
26 * *
27 */
28 public class ThietLapTruongPhongActivity extends Activity {
29
30     ListView lvtruongphong, lvphophong;
31     ArrayList<NhanVien> arrNvForTP=new ArrayList<NhanVien>();
```

```

32     ArrayAdapter<NhanVien> adapterForTP;
33     ArrayList<NhanVien> arrNvForPP=new ArrayList<NhanVien>();
34     ArrayAdapter<NhanVien> adapterForPP;
35     ImageButton btnApply;
36     int lastChecked=-1;
37     PhongBan pb=null;
38     @Override
39     protected void onCreate(Bundle savedInstanceState) {
40         super.onCreate(savedInstanceState);
41         setContentView(R.layout.activity_thiet_lap_truong_phong);
42         getFormWidgets();
43     }
44     /**
45      * hàm lấy control theo id
46      * xử lý sự kiện checked (chọn chức vụ cho nhân viên)
47      * Mọi thứ là hướng đối tượng nên nó tự tham chiếu
48      */
49     public void getFormWidgets()
50     {
51         lvtruongphong=(ListView) findViewById(R.id.lvtruongphong);
52         lvtruongphong.setTextFilterEnabled(true);
53         lvtruongphong.setChoiceMode(ListView.CHOICE_MODE_SINGLE);
54         lvtruongphong.setOnItemClickListener(new
55         OnItemClickListener() {
56             boolean somethingChecked = false;
57             @Override
58             public void onItemClick(AdapterView<?> arg0, View arg1, int
59             arg2,
60             long arg3) {
61                 arrNvForTP.get(arg2).setChucvu(ChucVu.TruongPhong);
62                 if(somethingChecked) {
63                     CheckedTextView cv = (CheckedTextView) arg1;
64                     cv.setChecked(false);
65                 }
66                 CheckedTextView cv = (CheckedTextView) arg1;
67                 if(!cv.isChecked())
68                 {
69                     cv.setChecked(true);
70                     arrNvForTP.get(arg2).setChucvu(ChucVu.TruongPhong);
71                 }
72                 else
73                 {
74                     arrNvForTP.get(arg2).setChucvu(ChucVu.NhanVien);
75                 }
76                 lastChecked = arg2;
77                 somethingChecked=true;
78             }
79         });
80         lvpophong=(ListView) findViewById(R.id.lvpophong);
81         lvpophong.setOnItemClickListener(new OnItemClickListener() {
82             @Override
83             public void onItemClick(AdapterView<?> arg0, View arg1, int
84             arg2,

```

```

86     long arg3) {
87     CheckedTextView cv = (CheckedTextView) arg1;
88     if (!cv.isChecked())
89     {cv.setChecked(true);
90     arrNvForPP.get(arg2).setChucvu(ChucVu.PhoPhong);
91     }
92     else
93     {cv.setChecked(false);
94     arrNvForPP.get(arg2).setChucvu(ChucVu.NhanVien);
95     }
96     }
97   });
98
99   adapterForTP=new ArrayAdapter<NhanVien>(this,
100   android.R.layout.simple_list_item_single_choice,
101   arrNvForTP);
102   adapterForPP=new ArrayAdapter<NhanVien>(this,
103   android.R.layout.simple_list_item_multiple_choice,
104   arrNvForPP);
105   lvtruongphong.setAdapter(adapterForTP);
106   lvpophong.setAdapter(adapterForPP);
107   //Lấy được phòng ban gửi qua từ MainActivity
108   Intent i= getIntent();
109   Bundle bundle= i.getBundleExtra("DATA");
110   pb= (PhongBan) bundle.getSerializable("PHONGBAN");
111   addNvToListTP(pb);
112   addNvToListPP(pb);
113   adapterForTP.notifyDataSetChanged();
114   adapterForPP.notifyDataSetChanged();
115
116   btnApply=(ImageButton) findViewById(R.id.imgapply);
117   btnApply.setOnClickListener(new OnClickListener() {
118
119     @Override
120     public void onClick(View arg0) {
121       doApply();
122
123     }
124   );
125   }
126   /**
127   * gửi thông tin lại Mainactivity sau khi thiết lập
128   */
129   public void doApply()
130   {
131     Intent i=getIntent();
132     Bundle bundle=new Bundle();
133     bundle.putSerializable("PHONGBAN", pb);
134     i.putExtra("DATA", bundle);
135     setResult(MainActivity.THIET_LAP_TP_PP_THANHCONG, i);
136     finish();
137   }
138   /**
139   * duyệt toàn bộ nhân viên vào danh sách ứng viên Trường phòng

```

```

140 * @param pb
141 */
142 public void addNvToListTP(PhongBan pb)
143 {
144 arrNvForTP.clear();
145 for(NhanVien nv:pb.getListNhanVien())
146 {
147 arrNvForTP.add(nv);
148 }
149 }
150 /**
151 * duyệt toàn bộ nhân viên vào danh sách ứng viên phó phòng
152 * @param pb
153 */
154 public void addNvToListPP(PhongBan pb)
155 {
156 arrNvForPP.clear();
157 for(NhanVien nv:pb.getListNhanVien())
158 {
159 arrNvForPP.add(nv);
160 }
}

```

- Như vậy Tôi đã trình bày xong ví dụ 25.
- Bài tập này nó rất khó và phức tạp, bạn phải đọc đi đọc lại và làm đi làm lại
- Nếu như bài tập này bạn hiểu thì coi như bạn hiểu được phần lớn Intent và đã biết cách ứng dụng nó.
- Bạn tải coding mẫu toàn bộ tại đây:<http://www.mediafire.com/?x8yyy2rewcf9su>
- Bài tập sau Tôi sẽ có thêm 1 ví dụ về implicit intent (cũng là intent). Mục đích của ví dụ này là Tôi viết chương trình gọi điện thoại và nhắn tin sms (không sử dụng chương trình có sẵn của Android).

Bài tập 26: Dùng Implicit Intent để viết chương trình gọi điện thoại và nhắn tin SMS

- Ở [bài tập 24](#) và [bài tập 25](#) Tôi đã hướng dẫn rất kỹ về cách sử dụng Intent (các intent này còn gọi là **Explicit Intent**).

- Trong bài tập 26 Tôi sẽ hướng dẫn bạn cách sử dụng **Implicit Intent** (Bạn cứ hiểu nôm na là những Intent được xây dựng sẵn, được cung cấp sẵn trong hệ điều hành Android). Tạm thời bạn cứ hiểu như vậy đi, nếu như Tôi sai thì Tôi sẽ sửa lại sau, vì có thể có nhiều cách hiểu khác hay hơn Tôi nhiều.

- Tôi đặt vấn đề như sau:

+ Một ngày đẹp trời khủng khiếp nào đó chặng hạn 3h giờ ngày 13 tháng 3 năm 2013, “Dế” đáng ghét của bạn bị rót xuống nền nhà cái bạch bạch.... bị mẻ, méo mẩy miếng và hư luôn chương trình gọi điện thoại và nhắn tin SMS (bị hư 2 chức năng này, nhưng mà “Dế Mẻ” đó vẫn hoạt động). Không ai thèm sửa Dế cho bạn, xui cái bạn đang học Android lớp của Tôi, Tôi yêu cầu bạn phải tự lập trình ra chức năng Gọi điện thoại và nhắn tin SMS cho Dế đó (dĩ nhiên thiết bị của bạn sử dụng Android OS). Vậy bạn phải làm rồi, vì không làm sẽ bị 0 điểm.

+ Vậy dùng những kỹ thuật nào để có thể viết được chức năng nghe gọi+ nhắn tin SMS thay thế cho chương trình có sẵn của Android?

+ Tôi sẽ hướng dẫn các bạn vào buổi sau, bạn hãy chú ý theo dõi. Trong thời gian chờ đợi thì Tôi khuyên các bạn phải làm đi làm lại bài tập 24 và bài tập 25.

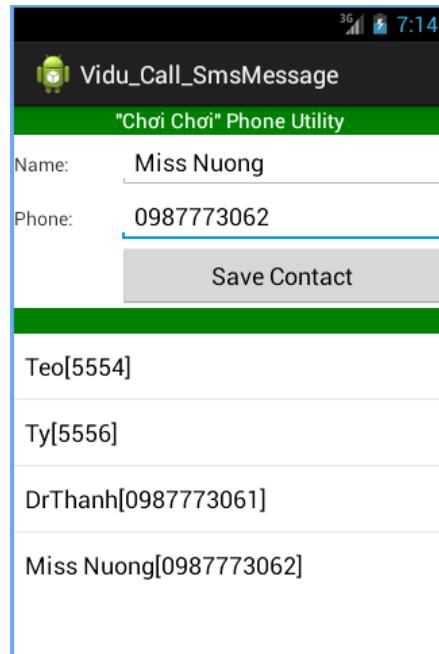
- **Tôi giới hạn bài tập này như sau:** Bài tập này đơn thuần chỉ là cho phép nhập tên và số điện thoại vào bộ nhớ (không phải danh bạ điện thoại) sau đó hiển thị danh sách này vào ListView, ListView sẽ cung cấp ContextMenu có 2 mục: “Gọi điện thoại” và “Nhắn tin”. Qua phần Content Provider và Telephony API các bạn sẽ được học kỹ hơn về cách lấy danh bạ từ điện thoại ra, cách đọc tin nhắn trong inbox ra, cách tương tác với danh bạ và tin nhắn... Nên bài tập này các bạn xem nó như là “Chơi chơi” vui thôi.

- Mẫu chốt của bài này là các bạn phải tự biết cách tạo một cuộc gọi, cách gửi tin nhắn trong Android như thế nào, mấy phần “râu ria” khác các bạn đừng bận tâm.

- Để có thể gọi điện thoại và nhắn tin thì các bạn phải khai báo trong Manifest các permission sau:

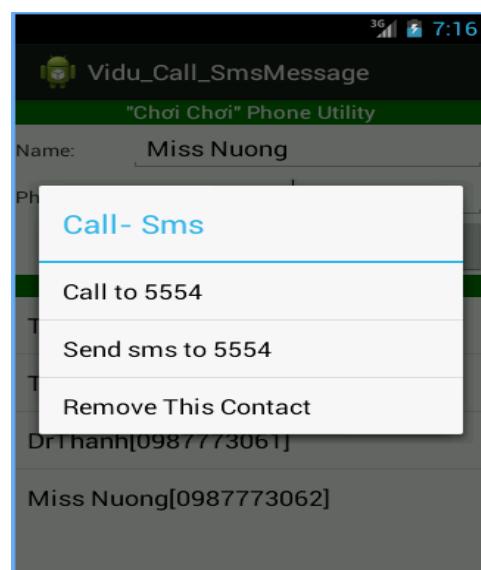
```
<uses-permission android:name="android.permission.CALL_PHONE" />
<uses-permission android:name="android.permission.SEND_SMS" />
```

- Bạn xem giao diện của chương trình:

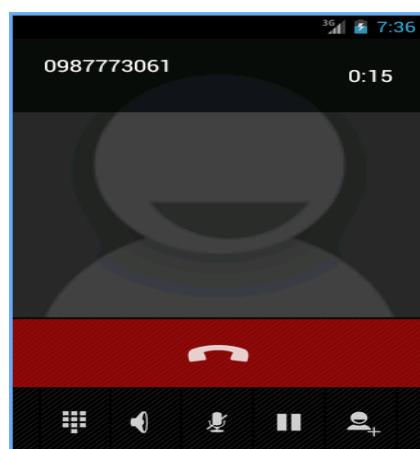


- Nhấn “Save Contact” để lưu thông tin vào ListView như hình.

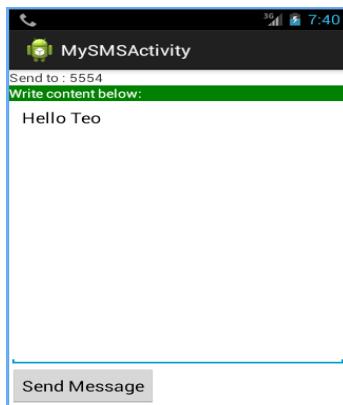
- Xử lý context menu cho ListView:



- Chức năng “Call to ...” để gọi, màn hình gọi sẽ hiện ra như bên dưới:



- Chức năng “Send Sms to ...“, cho ra màn hình gửi tin nhắn như bên dưới:



- Nhấn “Send Message” để gửi. Ở đây Tôi muốn nói các bạn rằng có 2 hành vi gửi tin nhắn.

1) **gửi tin nhắn mà không biết kết quả thành công hay không** (dễ dàng, nhưng mà không nên. Trường hợp này thường xuyên xảy ra. **Tôi lấy cự thể trường hợp của Tôi, nhà Tôi ở trên Núi sóng thường yếu và không liên lạc được, nếu viết theo cách này thì Bạn sẽ không biết là gửi cho Tôi thành công hay không.**)

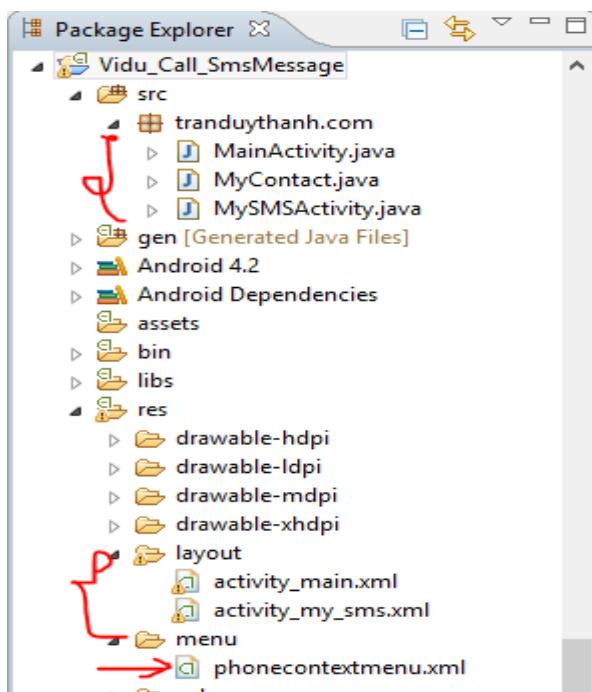
Ví dụ: Viết như bên dưới thì có thể gửi được tin nhắn ở trường hợp bình thường, không kiểm tra kết quả trả về (Còn kiểm tra kết quả trả về bạn xem trong coding đầy đủ).

```
final SmsManager sms = SmsManager.getDefault();
```

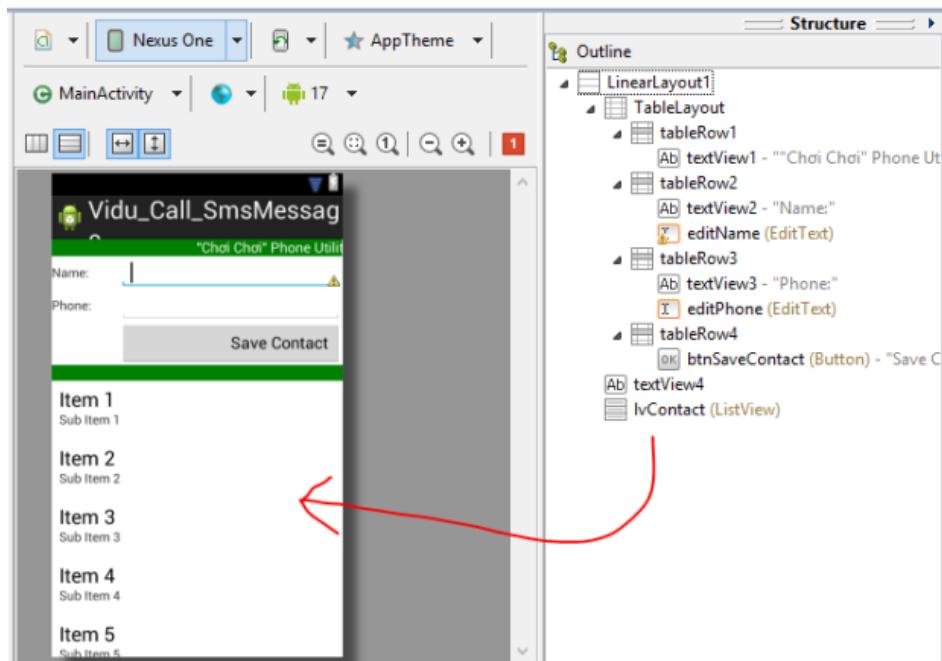
```
sms.sendTextMessage( "0987773061", null, "Hello teo teo!", null, null);
```

2) **gửi tin nhắn có kiểm tra kết quả thành công** (ta nên dùng cách này, nhưng mà khó vì các bạn chưa học IntentFilter, nhưng Tôi vẫn đưa vào, có gì các bạn tìm hiểu sau)

- Nay giờ bạn xem cấu tạo chương trình:



- Giao diện chính **activity_main.xml**:



- Xem source XML **activity_main.xml**:

```
1 <LinearLayout
2     xmlns:android="http://schemas.android.com/apk/res/android"
3     xmlns:tools="http://schemas.android.com/tools"
4     android:id="@+id/LinearLayout1"
5     android:layout_width="match_parent"
6     android:layout_height="match_parent"
7     android:orientation="vertical"
8     tools:context=".MainActivity" >
9
10    <TableLayout
11        android:layout_width="match_parent"
12        android:layout_height="wrap_content"
13        android:stretchColumns="*" >
14
15        <TableRow
16            android:id="@+id/tableRow1"
17            android:layout_width="wrap_content"
18            android:layout_height="wrap_content" >
19
20            <TextView
21                android:id="@+id/textView1"
22                android:layout_width="wrap_content"
23                android:layout_height="wrap_content"
24                android:layout_span="2"
25                android:background="#008000"
26                android:gravity="center"
27                android:text=""Chơi Chơi" Phone Utility"
28                android:textColor="#FFFFFF"
29                android:textSize="15sp" />
30
31    </TableRow>
```

```
32 <TableRow
33     android:id="@+id/tableRow2"
34     android:layout_width="wrap_content"
35     android:layout_height="wrap_content" >
36
37     <TextView
38         android:id="@+id/textView2"
39         android:layout_width="wrap_content"
40         android:layout_height="wrap_content"
41         android:text="Name:" />
42
43     <EditText
44         android:id="@+id/editName"
45         android:layout_width="wrap_content"
46         android:layout_height="wrap_content"
47         android:ems="10" >
48
49     <requestFocus />
50     </EditText>
51 </TableRow>
52
53 <TableRow
54     android:id="@+id/tableRow3"
55     android:layout_width="wrap_content"
56     android:layout_height="wrap_content" >
57
58     <TextView
59         android:id="@+id/textView3"
60         android:layout_width="wrap_content"
61         android:layout_height="wrap_content"
62         android:text="Phone:" />
63
64     <EditText
65         android:id="@+id/editPhone"
66         android:layout_width="wrap_content"
67         android:layout_height="wrap_content"
68         android:ems="10"
69         android:inputType="phone" />
70 </TableRow>
71
72 <TableRow
73     android:id="@+id/tableRow4"
74     android:layout_width="wrap_content"
75     android:layout_height="wrap_content" >
76
77     <Button
78         android:id="@+id/btnSaveContact"
79         android:layout_width="wrap_content"
80         android:layout_height="wrap_content"
81         android:layout_column="1"
82         android:text="Save Contact" />
83     </TableRow>
84 </TableLayout>
85
<TextView
```

```

86     android:id="@+id/textView4"
87     android:layout_width="match_parent"
88     android:layout_height="wrap_content"
89     android:background="#008000" />
90
91 <ListView
92     android:id="@+id/lvContact"
93     android:layout_width="match_parent"
94     android:layout_height="wrap_content" >
95 </ListView>
96
97 </LinearLayout>

```

- Các XML resource khác các bạn tự xem trong project.

- Class **MyContact**:

```

1  package tranduythanh.com;
2
3  import java.io.Serializable;
4
5  public class MyContact implements Serializable{
6      /**
7      *
8      */
9      private static final long serialVersionUID = 1L;
10     private String name;
11     private String phone;
12     public String getName() {
13         return name;
14     }
15     public void setName(String name) {
16         this.name = name;
17     }
18     public String getPhone() {
19         return phone;
20     }
21     public void setPhone(String phone) {
22         this.phone = phone;
23     }
24     public MyContact(String name, String phone) {
25         super();
26         this.name = name;
27         this.phone = phone;
28     }
29     public MyContact() {
30         super();
31     }
32     @Override
33     public String toString() {
34         return this.name+"["+this.phone+"]";
35     }
36 }

```

- Class xử lý MainActivity:

```
1  package tranduythanh.com;
2
3  import java.util.ArrayList;
4
5  import android.net.Uri;
6  import android.os.Bundle;
7  import android.app.Activity;
8  import android.content.Intent;
9  import android.view.ContextMenu;
10 import android.view.MenuItem;
11 import android.view.View;
12 import android.view.ContextMenu.ContextMenuItemInfo;
13 import android.view.View.OnClickListener;
14 import android.widget.AdapterView;
15 import android.widget.AdapterView.OnItemLongClickListener;
16 import android.widget.ArrayAdapter;
17 import android.widget.Button;
18 import android.widget.EditText;
19 import android.widget.ListView;
20
21 public class MainActivity extends Activity {
22
23     EditText editName,editPhone;
24     Button btnSave;
25     ListView lvContact;
26     //Danh sách contact để đưa vào ListView
27     ArrayList<MyContact>arrContact=new ArrayList<MyContact>();
28     ArrayAdapter<MyContact>adapter=null;
29     MyContact selectedContact=null;
30     @Override
31     protected void onCreate(Bundle savedInstanceState) {
32         super.onCreate(savedInstanceState);
33         setContentView(R.layout.activity_main);
34         doGetFormWidgets();
35     }
36     public void doGetFormWidgets()
37     {
38         btnSave=(Button) findViewById(R.id.btnSaveContact);
39         editName=(EditText) findViewById(R.id.editName);
40         editPhone=(EditText) findViewById(R.id.editPhone);
41         lvContact=(ListView) findViewById(R.id.lvContact);
42         //tạo đối tượng adapter
43         adapter=new ArrayAdapter<MyContact>
44             (this, android.R.layout.simple_list_item_1,arrContact);
45         //gán Adapter vào cho ListView
46         lvContact.setAdapter(adapter);
47         //thiết lập contextmenu cho ListView
48         registerForContextMenu(lvContact);
49     }
50     public void doAddEvents()
51     {
52         btnSave.setOnClickListener(new OnClickListener() {
```

```

53
54     @Override
55     public void onClick(View arg0) {
56         // TODO Auto-generated method stub
57         doSaveContact();
58     }
59 });
60     //lấy contact được chọn trước đó trong ListView
61     //Vì khi mở context menu sẽ làm mất focus nên ta phải lưu
62     //lại trước
63     //khi mở context menu
64     lvContact.setOnItemLongClickListener(new
65     OnItemLongClickListener() {
66
67         @Override
68         public boolean onItemLongClick(AdapterView<?> arg0, View
69         arg1,
70         int arg2, long arg3) {
71             //lưu vết contact được chọn trong ListView
72             selectedContact=arrContact.get(arg2);
73             return false;
74         }
75     });
76 }
77 /**
78 * mỗi lần nhấn Save contact thì gọi hàm này
79 * để cập nhật contact vào List view
80 * bạn lưu ý là ta chỉ làm trong bộ nhớ
81 * không phải lưu vào Danh Bạ (phần này học sau)
82 */
83 public void doSaveContact()
84 {
85     MyContact ct=new MyContact();
86     ct.setName(editName.getText().toString());
87     ct.setPhone(editPhone.getText().toString());
88     arrContact.add(ct);
89     adapter.notifyDataSetChanged();
90 }
91 @Override
92 public void onCreateContextMenu(ContextMenu menu, View v,
93 ContextMenuItemInfo menuInfo) {
94     // TODO Auto-generated method stub
95     super.onCreateContextMenu(menu, v, menuInfo);
96     //gắn context menu vào
97     getMenuInflater().inflate(R.menu.phonecontextmenu, menu);
98     menu.setHeaderTitle("Call- Sms");
99     menu.getItem(0).setTitle("Call to
100 "+selectedContact.getPhone());
101     menu.getItem(1).setTitle("Send sms to
102 "+selectedContact.getPhone());
103 }
104 @Override
105 public boolean onContextItemSelected(MenuItem item) {
106     //kiểm tra xem Menu Item nào được chọn
107     switch(item.getItemId())

```

```

107    {
108        case R.id.mnuCall:
109            doMakeCall();
110            break;
111        case R.id.mnuSms:
112            doMakeSms();
113            break;
114        case R.id.mnuRemove:
115            arrContact.remove(selectedContact);
116            adapter.notifyDataSetChanged();
117            break;
118        }
119        return super.onContextItemSelected(item);
120    }
121    /**
122     * Thực hiện gọi điện thoại
123     */
124    public void doMakeCall()
125    {
126        Uri uri=Uri.parse("tel:"+selectedContact.getPhone());
127        Intent i=new Intent(Intent.ACTION_CALL, uri);
128        startActivity(i);
129    }
130    /**
131     * thực hiện mở giao diện gửi tin nhắn
132     * Truyền thông tin contact đang chọn qua
133     * activity mới
134     */
135    public void doMakeSms()
136    {
137        Intent i=new Intent(this, MySMSActivity.class);
138        Bundle b=new Bundle();
139        b.putSerializable("CONTACT", selectedContact);
140        i.putExtra("DATA", b);
141        startActivity(i);
142    }
}

```

-Bạn xem dòng lệnh 96 – thiết lập tiêu đề cho context menu

- Dòng lệnh 97,98 để sửa lại title cho các menuitem ứng với mỗi contact.

- Dòng lệnh 121 có hàm **doMakeCall** để gọi điện thoại cho số nào đó. Bạn nhìn vào nội dung bên trong để xem quy tắc tạo lệnh gọi.

- Class xử lý gửi tin nhắn **MySMSActivity**:

```

1 package tranduythanh.com;
2
3 import android.os.Bundle;
4 import android.app.Activity;
5 import android.app.PendingIntent;
6 import android.content.BroadcastReceiver;

```

```

7 import android.content.Context;
8 import android.content.Intent;
9 import android.content.IntentFilter;
10 import android.telephony.SmsManager;
11 import android.view.View;
12 import android.view.View.OnClickListener;
13 import android.widget.Button;
14 import android.widget.EditText;
15 import android.widget.TextView;
16 import android.widget.Toast;
17
18 public class MySMSActivity extends Activity {
19
20     Button btnSendSMS;
21     EditText editContent;
22     TextView txtSendTo;
23     @Override
24     protected void onCreate(Bundle savedInstanceState) {
25         super.onCreate(savedInstanceState);
26         setContentView(R.layout.activity_my_sms);
27         btnSendSMS = (Button) findViewById(R.id.btnSendSms);
28         editContent = (EditText) findViewById(R.id.editSMS);
29         txtSendTo = (TextView) findViewById(R.id.txtSendTo);
30         //Lấy thông tin từ Intent
31         Intent i = getIntent();
32         Bundle b = i.getBundleExtra("DATA");
33         final MyContact c = (MyContact) b.getSerializable("CONTACT");
34         btnSendSMS.setOnClickListener(new OnClickListener() {
35
36             @Override
37             public void onClick(View v) {
38                 // TODO Auto-generated method stub
39                 sendSms(c);
40             }
41         });
42         txtSendTo.setText("Send to : " + c.getPhone());
43     }
44     /**
45      * hàm dùng để gửi tin nhắn có kiểm tra kết quả trả về
46      * Tôi chưa giải thích ở đây được vì nó liên quan rất nhiều
47      * đến kiến thức, khi nào tới Broadcast Receiver, telephony Tôi sẽ
48      * giải thích lại
49      * @param c
50      */
51     public void sendSms(MyContact c)
52     {
53         //lấy mặc định SmsManager
54         final SmsManager sms = SmsManager.getDefault();
55         Intent msgSent = new Intent("ACTION_MSG_SENT");
56         //Khai báo pendingintent để kiểm tra kết quả
57         final PendingIntent pendingMsgSent =
58             PendingIntent.getBroadcast(this, 0, msgSent, 0);
59         registerReceiver(new BroadcastReceiver() {
60             public void onReceive(Context context, Intent intent) {
61                 int result = getResultCode();

```

```

61     String msg="Send OK";
62     if (result != Activity.RESULT_OK) {
63         msg="Send failed";
64     }
65     Toast.makeText(MySMSActivity.this, msg,
66     Toast.LENGTH_LONG).show();
67 }
68 }, new IntentFilter("ACTION_MSG_SENT"));
69 //Gọi hàm gửi tin nhắn đi
70 sms.sendTextMessage(c.getPhone(), null,
71 editContent.getText()+"",
72 pendingMsgSent, null);
73 finish();
74 }
75 }
```

- Dòng lệnh 51 có hàm **sendSms** để gửi tin nhắn đi. hàm này Tôi viết có kiểm tra kết quả gửi thành công hay không. Tôi không giải thích nhiều vì nó liên quan tới quá nhiều kiến thức. Tạm thời bạn có gắng hiểu từng dòng lệnh Tôi giải thích ở bên trong là ok rồi.

- Bạn có thể tải coding mẫu đầy đủ ở đây:<http://www.mediafire.com/?ilpeqrycsbdswzg>

- Bài này là tiền đề để làm các ứng dụng khác liên quan tới Telephony.

- Đến đây coi như bạn đã tạm ổn về Intent. Bài tập kế tiếp Tôi sẽ hướng dẫn các bạn làm về Đa ngôn ngữ trong Android, đã nói tới Mobile thì thường nó phải hỗ trợ đa ngôn ngữ (cả thế giới đều sài, với những ngôn ngữ khác nhau).

Bài tập 27: Đa ngôn ngữ trong Android

- Bài tập này Tôi sẽ hướng dẫn các bạn cách tạo đa ngôn ngữ trong Android.
- Mobile trở thành thiết bị không thể thiếu được trên thế giới này. Người làm ruộng, bán vé số, bán bánh mì, nhân viên văn phòng, giám đốc, nhân viên bán hàng, bán tùng lum... every body.... đều sở hữu 1 chú Dé.
- Hỗ trợ đa ngôn ngữ giúp phần mềm ta bán chạy, mục tiêu viết phần mềm ra để bán kiếm tiền chứ không phải viết ra để tự kỷ 1 mình. Nên các bạn phải biết làm đa ngôn ngữ.

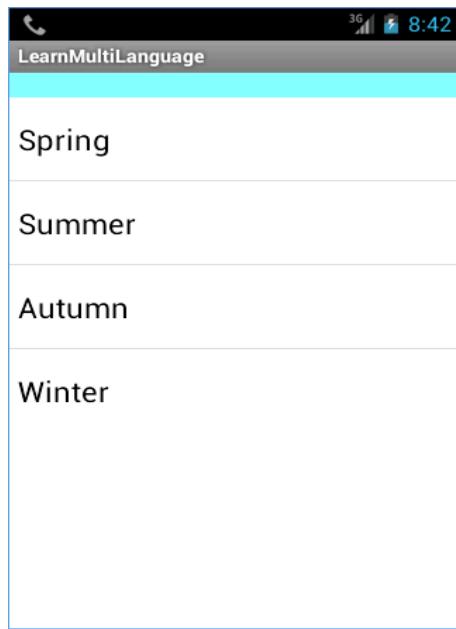


- Bài tập này Tôi sẽ làm 3 ngôn ngữ: Tiếng Anh, Tiếng Pháp, Tiếng Việt.
- Tiếng Pháp Tôi không biết gì cả (1 chữ cắn đôi không biết, như chữ “Không” bẻ đôi (dọc hoặc ngang) ra đọc bằng tiếng Pháp thì Tôi không biết được), Tôi dịch trên google.
- Đa ngôn ngữ chỉ là làm trên XML resource là chính (coding không phải sửa đổi, chỉ sửa resource theo đúng ngôn ngữ). Khi ta làm đúng ngôn ngữ theo resource thì khi người sử dụng chuyển ngôn ngữ trong Android nó sẽ tự tải đúng resource ngôn ngữ mà ta đã tạo.
- Cấu hình ngôn ngữ trong Android (bạn thử mò), Tôi chụp màn hình bên trong để chọn ngôn ngữ:

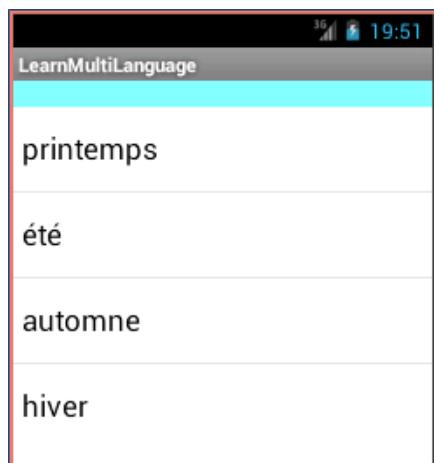


- Tôi có ví dụ về 4 mùa trong năm: xuân, hạ, thu, đông:

- Khi Setting trong Android là tiếng Anh:



- Tiếng Pháp:



- Tiếng việt:

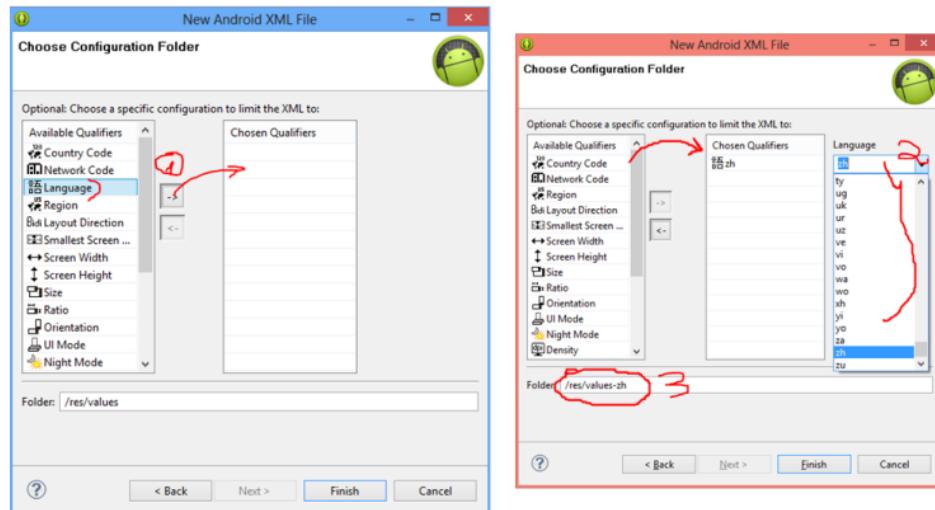


- Bạn chú ý là các hình trên Tôi hoàn toàn không sửa đổi coding, Tôi chỉ đổi ngôn ngữ trong Android, nó tự động tải đúng Resource XML ngôn ngữ.

- Cách tạo Resource XML ngôn ngữ:

- Cần chú ý: tất cả các string resource id ở các ngôn ngữ khác nhau đều phải có cùng id.

- Mỗi một Quốc gia sẽ có một Code riêng với 2 ký tự:



- Bấm chuột phải vào Android Project/ chọn Android XML resource:

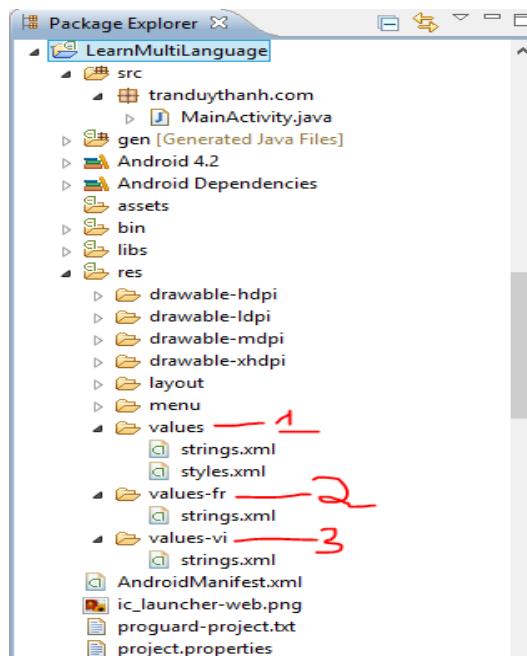
- Màn hình trên hiển thị ra. Bạn chọn vào mục Language rồi click biểu tượng số 1.

- Language sẽ hiển thị vào danh sách ở giữa

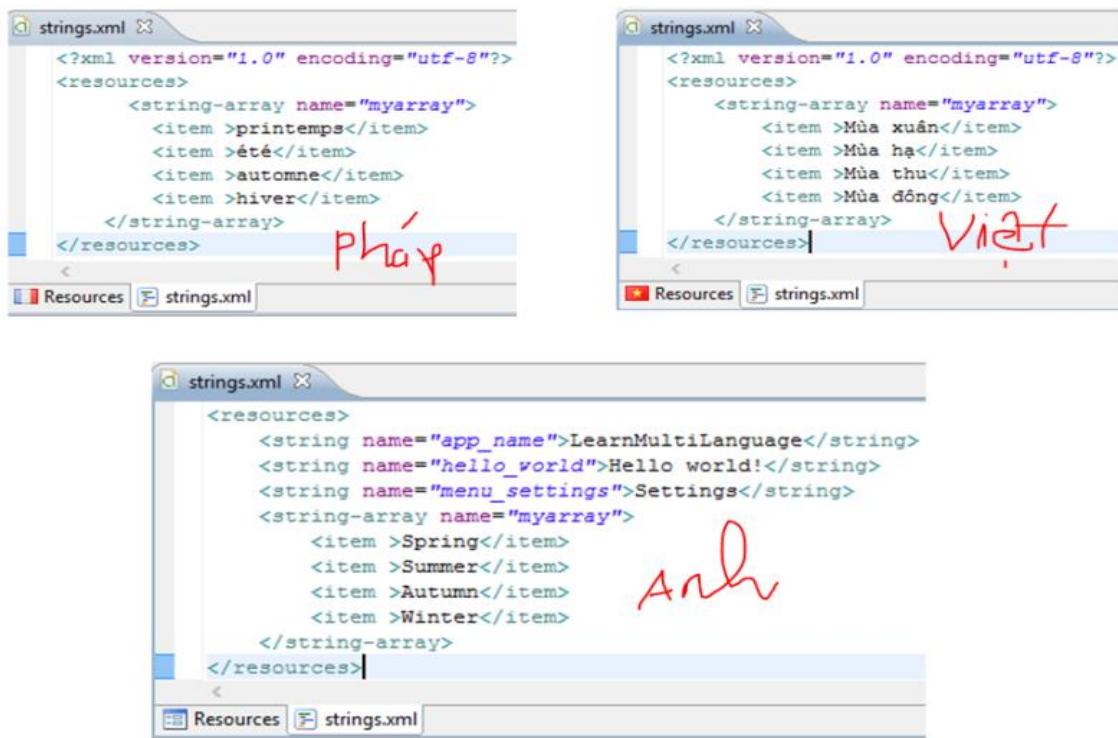
- Ở mục số 2 bạn chọn Code Quốc gia (theo ngôn ngữ)

- Mục Folder ở số 3 sẽ được nối kèm ở đằng sau values.

- Bạn xem cấu trúc thư mục của Android Project sau khi bạn chọn ngôn ngữ xong (ở đây là tiếng anh, tiếng pháp, tiếng việt):



-Bạn xem 3 tập tin strings.xml ở 3 ngôn ngữ khác nhau (nhớ là phải cùng id):



- Xem XML layout:

```
1  <LinearLayout
2      xmlns:android="http://schemas.android.com/apk/res/android"
3      xmlns:tools="http://schemas.android.com/tools"
4      android:id="@+id/LinearLayout1"
5      android:layout_width="match_parent"
6      android:layout_height="match_parent"
7      android:orientation="vertical"
8      tools:context=".MainActivity" >
9
10     <TextView
11         android:id="@+id/textView1"
12         android:layout_width="match_parent"
13         android:layout_height="wrap_content"
14         android:background="#80FFFF" />
15
16     <ListView
17         android:id="@+id/listView1"
18         android:layout_width="match_parent"
19         android:layout_height="wrap_content" >
20     </ListView>
21
22 </LinearLayout>
```

- Xem coding:

```
1  package tranduythanh.com;
2  import android.os.Bundle;
3  import android.app.Activity;
4  import android.view.Menu;
```

```
5 import android.widget.ArrayAdapter;
6 import android.widget.ListView;
7
8 public class MainActivity extends Activity {
9     protected void onCreate(Bundle savedInstanceState) {
10         super.onCreate(savedInstanceState);
11         setContentView(R.layout.activity_main);
12         ListView lv=(ListView) findViewById(R.id.listView1);
13         ArrayAdapter<String>adapter=new ArrayAdapter<String>
14             (this,android.R.layout.simple_list_item_1,
15             getResources().getStringArray(R.array.myarray));
16         lv.setAdapter(adapter);
17     }
18     @Override
19     public boolean onCreateOptionsMenu(Menu menu) {
20         // Inflate the menu; this adds items to the action bar if it is
21         // present.
22         getMenuInflater().inflate(R.menu.activity_main, menu);
23         return true;
24     }
25 }
```

- Bây giờ bạn chỉ cần việc đổi chuyển ngôn ngữ trong Setting là xong, tự nhiên nó sẽ lấy đúng XML resource ngôn ngữ mà bạn tạo.
- Bạn tải coding mẫu ở đây: <http://www.mediafire.com/?owbph611tqxtt50>
- Bài tập này không khó, các bạn chỉ cần làm theo hướng dẫn.
- Bài tập sau các bạn sẽ học về xử lý tập tin trong Android, cũng khá quan trọng và hay, các bạn hãy chú ý theo dõi.

Bài tập 28: Xử lý tập tin trong Android

- Bài tập này Tôi sẽ hướng dẫn các bạn cách xử lý tập tin trong Android, bao gồm:

+ Internal Storage

+ External Storage

+ Saving Cache Files

- Các bạn chú ý là Android sử dụng cấu trúc tập tin giống như trong ứng dụng Java thuần túy.

- Tập tin trong Android có thể được lưu trữ trong **Main Memory** (thường là lưu trữ tập tin có dung lượng nhỏ) hoặc **SD Card** (thường là lưu trữ tập tin có dung lượng lớn).

- Các tập tin được lưu trữ cùng với các tài nguyên khác trong bộ nhớ trong (chẳng hạn như Icons, hình ảnh, nhạc,...) thì người ta gọi chung là **Resource Files**.

- Trên mạng có đầy rẫy website hướng dẫn về thao tác với tập tin trong Android, nhưng ở đây Tôi muốn trình bày theo cách của Tôi.

1) Internal Storage

- Ở đây có 2 trường hợp:

- Thứ nhất: đọc Resource File (Read only)

- Thứ nhì : đọc và ghi tập tin (Read & Write).

* Đọc Resource File:

- Bạn xem giao diện bên dưới: Tôi tạo 1 tập tin và nhập đại vào một số dữ liệu rồi kéo thả tập tin đó vào thư mục **drawable**. Nếu như ứng dụng của bạn không có thư mục này thì bạn tự tạo nó nhớ là viết y xì “**drawable**”, tên file cũng đừng để khoảng trắng, đừng để chữ số đằng trước.



- Khi nhấn chọn “**Read Data**“, chương trình sẽ đọc nội dung từ **myfile.txt** và hiển thị lên **EditText**.

- Ở đây Tôi muốn hỏi các bạn là : “**Tại sao các resource được đóng gói trong APK lại là Read Only?**“. Tôi nghĩ câu trả lời đơn giản nhất đó là sử dụng tối ưu bộ nhớ. Vì điện thoại không giống như máy tính, bộ nhớ trong của nó không có bao la bát ngát như PC, mỗi phần mềm (.APK) được cài đặt vào thì nó sẽ không muốn “Co Giãn” dung lượng vì nó cần quản lý phần mềm để tối ưu nhất (sử dụng bộ nhớ ít nhất).

- Bạn xem cấu trúc XML của giao diện (**activity_main.xml**):

```
1 <LinearLayout
2     xmlns:android="http://schemas.android.com/apk/res/android"
3     xmlns:tools="http://schemas.android.com/tools"
4     android:id="@+id/LinearLayout1"
5     android:layout_width="match_parent"
6     android:layout_height="match_parent"
7     android:orientation="vertical"
8     tools:context=".MainActivity" >
9
10    <Button
11        android:id="@+id/btnreaddata"
12        android:layout_width="match_parent"
13        android:layout_height="wrap_content"
14        android:text="Read Data" />
15
16    <EditText
17        android:id="@+id/editdata"
18        android:layout_width="match_parent"
19        android:layout_height="0dip"
20        android:layout_weight="2"
21        android:ems="10"
22        android:gravity="top|left"
23        android:inputType="textMultiLine" >
24
25    <requestFocus />
26    </EditText>
```

- Đây là source code để đọc Resource File:

- Bạn chỉ cần copy đoạn code này vào sự kiện của nút đọc là xong:

```
1 public void readData()
2 {
3     String data;
4     InputStream in=
5     getResources().openRawResource(R.drawable myfile);
6     InputStreamReader inreader=new InputStreamReader(in);
7     BufferedReader bufreader=new BufferedReader(inreader);
8     StringBuilder builder=new StringBuilder();
9     if(in!=null)
10    {
11        try
12        {
```

```

13     while((data=bufreader.readLine())!=null)
14     {
15         builder.append(data);
16         builder.append("\n");
17     }
18     in.close();
19     editdata.setText(builder.toString());
20 }
21 catch(IOException ex) {
22     Log.e("ERROR", ex.getMessage());
23 }
24 }
}

```

- Ở đây Tôi không cung cấp source nguồn, vì nó chỉ có 1 hàm xí xi đó. Các bạn chỉ cần thiết kế giao diện cho giống như XML Layout rồi copy code vào sự kiện của nút nhấn là xong.

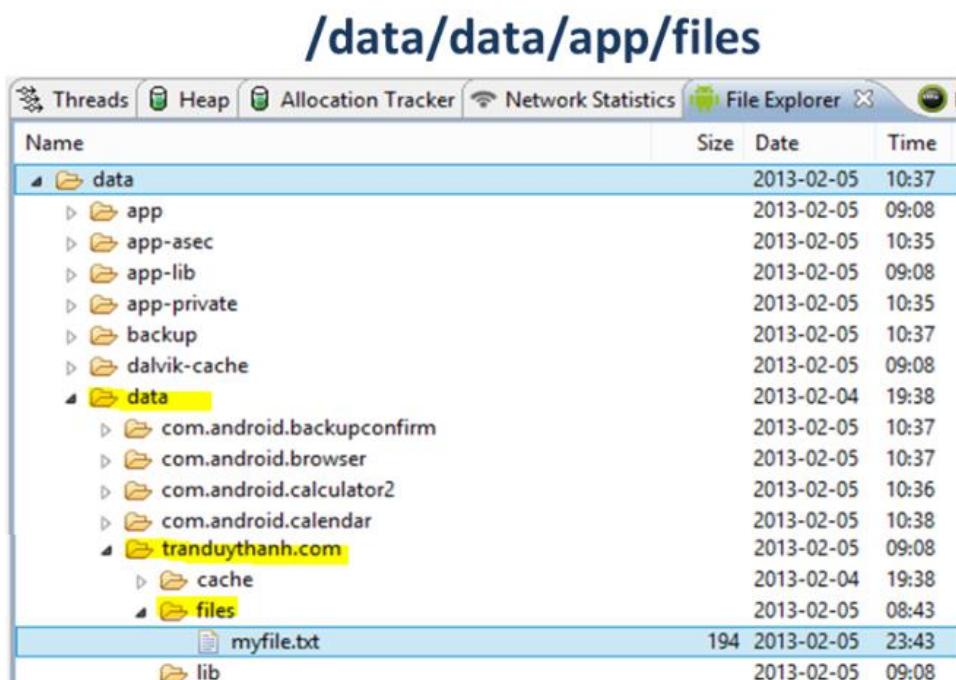
- Cách đọc tập tin trong Android y chang như trong Java thuần túy, ở đây bạn chỉ để ý dòng lệnh số 4:

`InputStream in= getResources().openRawResource(R.drawable myfile);`

Dòng lệnh này để đọc Resource File và trả về InputStream. Khi có InputStream rồi thì mọi xử lý sẽ giống như Java.

* Đọc và ghi tập tin:

- Trường hợp này bạn kéo thả tập tin vào cấu trúc thư mục như hình bên dưới (dùng DDMS):
- Nhìn vào những vệt Tôi tô màu vàng, các bạn kéo thả vào đúng vị trí ứng dụng của bạn.



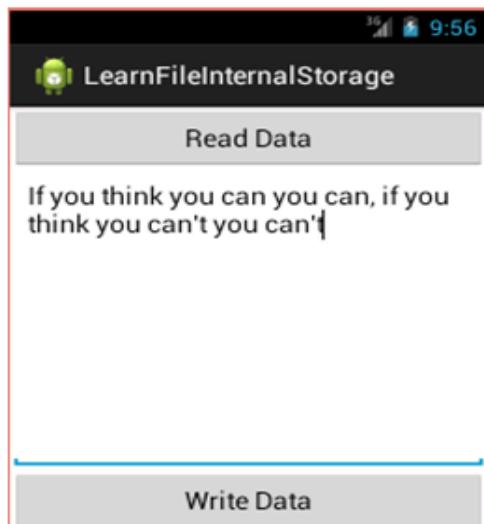
- XML Resource cho bài tập này:

```

1 <LinearLayout
2   xmlns:android="http://schemas.android.com/apk/res/android"
3     xmlns:tools="http://schemas.android.com/tools"
4     android:id="@+id/LinearLayout1"
5     android:layout_width="match_parent"
6     android:layout_height="match_parent"
7     android:orientation="vertical"
8     tools:context=".MainActivity" >
9
10 <Button
11   android:id="@+id/btnreaddata"
12   android:layout_width="match_parent"
13   android:layout_height="wrap_content"
14   android:text="Read Data" />
15
16 <EditText
17   android:id="@+id/editdata"
18   android:layout_width="match_parent"
19   android:layout_height="0dip"
20   android:layout_weight="2"
21   android:ems="10"
22   android:gravity="top|left"
23   android:inputType="textMultiLine" >
24 <requestFocus />
25 </EditText>
26
27 <Button
28   android:id="@+id/btnwritedata"
29   android:layout_width="match_parent"
30   android:layout_height="wrap_content"
31   android:text="Write Data" />
32
33 </LinearLayout>

```

- XML resource trên sẽ tạo giao diện như hình bên dưới:



- Source Activity

```

1  package tranduythanh.com;
2  import java.io.BufferedReader;
3  import java.io.FileInputStream;
4  import java.io.FileNotFoundException;
5  import java.io.FileOutputStream;
6  import java.io.IOException;
7  import java.io.InputStream;
8  import java.io.InputStreamReader;
9  import java.io.OutputStreamWriter;
10
11 import android.os.Bundle;
12 import android.app.Activity;
13 import android.util.Log;
14 import android.view.Menu;
15 import android.view.View;
16 import android.view.View.OnClickListener;
17 import android.widget.Button;
18 import android.widget.EditText;
19
20 public class MainActivity extends Activity implements
21 OnClickListener{
22
23     Button btnread,btnwrite;
24     EditText editdata;
25     @Override
26     protected void onCreate(Bundle savedInstanceState) {
27         super.onCreate(savedInstanceState);
28         setContentView(R.layout.activity_main);
29         btnread=(Button) findViewById(R.id.btnreaddata);
30         btnwrite=(Button) findViewById(R.id.btnwritedata);
31         editdata=(EditText) findViewById(R.id.editdata);
32         btnread.setOnClickListener(this);
33         btnwrite.setOnClickListener(this);
34     }
35     public void onClick(View v) {
36         if(v.getId()==R.id.btnreaddata)
37         {
38             readData();
39         }
40         else if(v.getId()==R.id.btnwritedata)
41         {
42             writeData();
43         }
44     /**
45      * Hàm đọc tập tin trong Android
46      * Dùng openFileInput trong Android để đọc
47      * ra FileInputStream
48      */
49     public void readData()
50     {
51         try {
52             FileInputStream in= openFileInput("myfile.txt");
53             BufferedReader reader=new
54             BufferedReader(new InputStreamReader(in));

```

```

55     String data="";
56     StringBuilder builder=new StringBuilder();
57     while((data=reader.readLine())!=null)
58     {
59         builder.append(data);
60         builder.append("\n");
61     }
62     in.close();
63     editdata.setText(builder.toString());
64     } catch (FileNotFoundException e) {
65         e.printStackTrace();
66     } catch (IOException e) {
67         e.printStackTrace();
68     }
69 }
70 /**
71 * Hàm ghi tập tin trong Android
72 * dùng openFileOutput để ghi
73 * openFileOutput tạo ra FileOutputStream
74 */
75 public void writeData()
76 {
77     try {
78         FileOutputStream out=
79             openFileOutput("myfile.txt",0);
80         OutputStreamWriter writer=
81             new OutputStreamWriter(out);
82         writer.write(editdata.getText().toString());
83         writer.close();
84     } catch (FileNotFoundException e) {
85         e.printStackTrace();
86     } catch (IOException e) {
87         e.printStackTrace();
88     }
89 }
90 /**
91 * hàm này là đọc Resource File, Tôi gom chung vào đây
92 */
93 public void readData2()
94 {
95     String data;
96     InputStream in= getResources()
97         .openRawResource(R.drawable myfile);
98     InputStreamReader inreader=new InputStreamReader(in);
99     BufferedReader bufreader=new BufferedReader(inreader);
100    StringBuilder builder=new StringBuilder();
101    if(in!=null)
102    {
103        try{
104            while((data=bufreader.readLine())!=null)
105            {
106                builder.append(data);
107                builder.append("\n");
108            }
109            in.close();

```

```

109     editdata.setText(builder.toString());
110 }
111 catch(IOException ex) {
112     Log.e("ERROR", ex.getMessage());
113 }
114 }
115 }
116
117 @Override
118 public boolean onCreateOptionsMenu(Menu menu) {
119     // Inflate the menu; this adds items to the action bar if it
120     // is present.
121     getMenuInflater().inflate(R.menu.activity_main, menu);
122     return true;
123 }
124 }
```

- Bạn nhìn vào dòng lệnh 52: FileInputStream in= **openFileInput**("myfile.txt");

Lệnh ở dòng này mở tập tin lên để đọc, nó trả về FileInputStream nên mọi thứ sẽ giống như Java thuần túy.

- Tiếp tục nhìn vào dòng lệnh 78: FileOutputStream out=**openFileOutput**("myfile.txt",0);

Lệnh ở dòng này mở tập tin lên để ghi. đối số thứ 2 Tôi để là số 0 (**MODE_PRIVATE**) là mặc định. Bạn có thể dùng **MODE_APPEND**, **MODE_WORLD_READABLE**,**MODE_WORLD_WRITEABLE**. Ý nghĩa của mỗi mode bạn có thể lên mạng tìm. Hoặc chỉ cần di chuyển chuột vào hàm trên là eclipse tự giải thích cho các bạn (nếu eclipse bạn đã cài Javadoc).

- Bạn tải source code mẫu phần này ở đây:

<http://www.mediafire.com/download/toi0ht8ht0t5gd3/LearnFileInternalStorage.rar>

2. External Storage:

- Phần này Tôi muốn hướng dẫn các bạn cách tương tác với tập tin được lưu trữ trên **SD Card**.
- Để thao tác được trên SD Card, bạn phải cấp quyền cho ứng dụng (**uses – permission**)
- Để vừa ghi vừa đọc trên **SD Card** thì ta phải sử dụng 2 permission bên dưới trong **AndroidManifest**:

```
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />
```

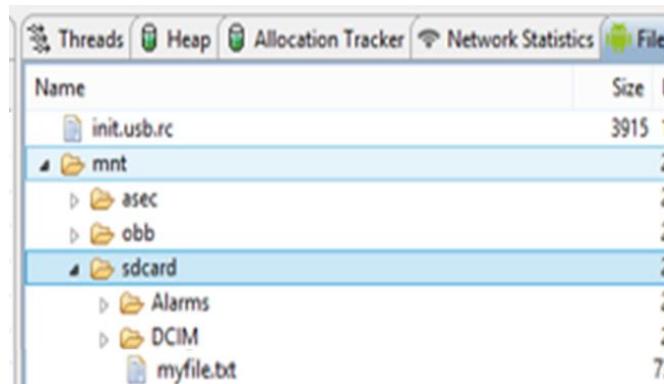
- Để lấy được đường dẫn của SD Card ta dùng lệnh:

```
String sdcard=Environment.getExternalStorageDirectory().getAbsolutePath();
```

- Sau đó mọi thứ tương tác với tập tin thì y xì như Java thuần túy, ở đây Tôi dùng **Scanner** để đọc và **OutputStreamWriter** để ghi.

- Vì vậy giao diện Tôi làm y chang như Internal Storage:

- Bạn tạo Project và kéo thả tập tin vào SD card (dĩ nhiên khi tạo AVD thì bạn phải checked vào using SD card thì mới có). Bạn xem hình để biết cách kéo thả vào SD Card:



- Ở hình trên bạn vào DDMS, tới thư mục **mnt/** chọn **sdcard**. Phải chọn cho đúng vì nó có nhiều thư mục cùng tên **sdcard**, nhưng bạn chọn sdcard ở trong thư mục **mnt** thôi. Sau đó kéo thả tập tin vào đây là xong.

- XML Resource y xì như phần internal Storage nên Tôi không đưa vào đây, bạn xem Source code Activity:

```
1 package tranduythanh.com;
2
3 import java.io.File;
4 import java.io.FileNotFoundException;
5 import java.io.FileOutputStream;
6 import java.io.IOException;
7 import java.io.OutputStreamWriter;
8 import java.util.Scanner;
9
10 import android.os.Bundle;
11 import android.os.Environment;
12 import android.app.Activity;
13 import android.view.Menu;
14 import android.view.View;
15 import android.view.View.OnClickListener;
16 import android.widget.Button;
17 import android.widget.EditText;
18
19 public class MainActivity extends Activity implements
20 OnClickListener{
21
22     Button btnread,btnwrite;
23     EditText editdata;
24     @Override
25     protected void onCreate(Bundle savedInstanceState) {
26         super.onCreate(savedInstanceState);
```

```

27     setContentView(R.layout.activity_main);
28     btnread=(Button) findViewById(R.id.btnreaddata);
29     btnwrite=(Button) findViewById(R.id.btnwritedata);
30     editdata=(EditText) findViewById(R.id.editdata);
31     btnread.setOnClickListener(this);
32     btnwrite.setOnClickListener(this);
33 }
34 public void onClick(View v) {
35     if(v.getId()==R.id.btnreaddata)
36     {
37         readData();
38     }
39     else if(v.getId()==R.id.btnwritedata)
40     {
41         writeData();
42     }
43 }
44 /*
45 * đọc từ SD Card
46 */
47 Environment.getExternalStorageDirectory().getAbsolutePath()
48     * để lấy đường dẫn trên SD Card
49     */
50 public void readData()
51 {
52     String sdcard=Environment
53         .getExternalStorageDirectory()
54         .getAbsolutePath() + "/myfile.txt";
55     try {
56         Scanner scan=new Scanner(new File(sdcard));
57         String data="";
58         while(scan.hasNext())
59         {
60             data+=scan.nextLine() + "\n";
61         }
62         scan.close();
63         editdata.setText(data+"");
64     } catch (FileNotFoundException e) {
65         e.printStackTrace();
66     }
67 /**
68 * ghi tập tin trên SD Card
69 */
70 public void writeData()
71 {
72     String sdcard=Environment
73         .getExternalStorageDirectory()
74         .getAbsolutePath() + "/myfile.txt";
75     try {
76         OutputStreamWriter writer=
77             new OutputStreamWriter(
78                 new FileOutputStream(sdcard));
79         writer.write(editdata.getText()+"");
80         writer.close();

```

```

81     } catch (FileNotFoundException e) {
82         e.printStackTrace();
83     } catch (IOException e) {
84         e.printStackTrace();
85     }
86 }
87
88 @Override
89 public boolean onCreateOptionsMenu(Menu menu) {
90     // Inflate the menu; this adds items to the action bar if it
91     is present.
92     getMenuInflater().inflate(R.menu.activity_main, menu);
93     return true;
94 }

```

}

- Vì có sử dụng Permission nên bạn xem AndroidManifest:

```

<?xml version="1.0" encoding="utf-8"?>
<manifest
1    xmlns:android="http://schemas.android.com/apk/res/android"
2        package="tranduythanh.com"
3        android:versionCode="1"
4        android:versionName="1.0" >
5
6    <uses-sdk
7        android:minSdkVersion="14"
8        android:targetSdkVersion="17" />
9
10   <uses-permission
11       android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
12       <uses-permission
13           android:name="android.permission.READ_EXTERNAL_STORAGE" />
14
15   <application
16       android:allowBackup="true"
17       android:icon="@drawable/ic_launcher"
18       android:label="@string/app_name"
19       android:theme="@style/AppTheme" >
20       <activity
21           android:name="tranduythanh.com.MainActivity"
22           android:label="@string/app_name" >
23           <intent-filter>
24               <action android:name="android.intent.action.MAIN" />
25
26           <category android:name="android.intent.category.LAUNCHER" />
27           </intent-filter>
28       </activity>
29   </application>
30
</manifest>

```

- Khi thực thi ứng dụng, bạn sẽ có giao diện như trình bày ở trên, với kết quả như mong muốn.

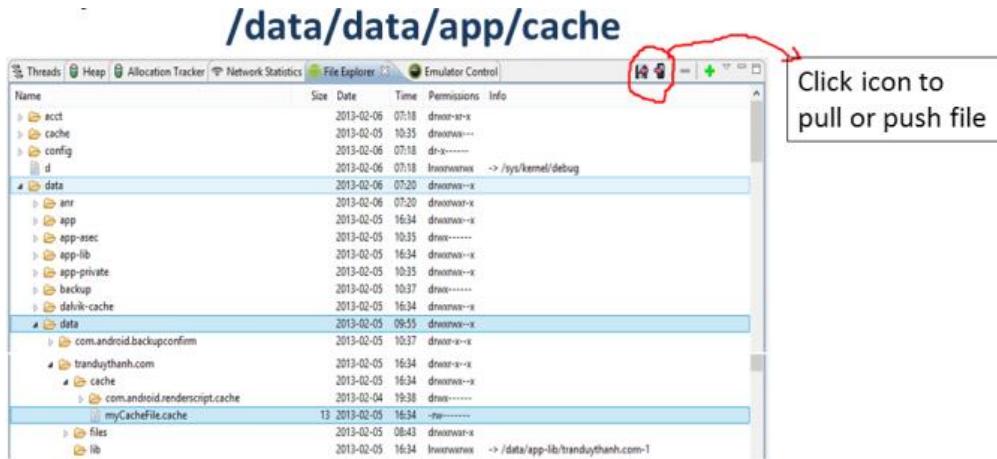
- Bạn có thể tải coding mẫu ở đây:

<http://www.mediafire.com/download/1ydvjc8wvizzjcm/LearnFileExternalStorage.rar>

3) Saving cache files:

- **Lợi ích của cache files là gì?** Để tăng tốc độ xử lý của ứng dụng khi bạn thường xuyên truy cập internet thì bạn nên lưu cache.

- Khi lưu cache thì nó sẽ để ở đâu trong điện thoại? bạn xem hình:



- Khi lưu cache thì mặc định nó sẽ lưu vào thư mục **cache** như hình bên trên. Hình trên Tôi có tập tin tên là **myCacheFile.Cache**.

- Lưu như thế nào? và xóa như thế nào?

Khi bạn lưu cache, Android thường chỉ cho tối đa file cache là **1MB**, khi bạn gỡ bỏ phần mềm thì nó cũng tự động xóa cache của phần mềm này đi.

- Để lấy được đường dẫn của thư mục Cache trong ứng dụng, ta dùng lệnh : **getCacheDir()** . Sau đó cách lưu cache tương tự như Java thuận túy.

- Dưới đây là phần mềm minh họa Cache, giao diện tương tự như trên:

- XML resource:

```

1 <LinearLayout
2     xmlns:android="http://schemas.android.com/apk/res/android"
3     xmlns:tools="http://schemas.android.com/tools"
4     android:id="@+id/LinearLayout1"
5     android:layout_width="match_parent"
6     android:layout_height="match_parent"
7     android:orientation="vertical"
8     tools:context=".MainActivity" >
9
10    <Button
11        android:id="@+id/btncreatecache"

```

```

12     android:layout_width="match_parent"
13     android:layout_height="wrap_content"
14     android:text="Create Cache" />
15
16 <EditText
17     android:id="@+id/editdata"
18     android:layout_width="match_parent"
19     android:layout_height="0dip"
20     android:layout_weight="2"
21     android:ems="10"
22     android:gravity="top|left"
23     android:inputType="textMultiLine" >
24
25 <requestFocus />
26 </EditText>
27
28 <Button
29     android:id="@+id/btnreadcache"
30     android:layout_width="match_parent"
31     android:layout_height="wrap_content"
32     android:text="Read Cache" />
33
34 </LinearLayout>

```

- Xử lý code Activity:

```

1  package tranduythanh.com;
2
3  import java.io.File;
4  import java.io.FileNotFoundException;
5  import java.io.FileOutputStream;
6  import java.io.IOException;
7  import java.util.Scanner;
8
9  import android.os.Bundle;
10 import android.app.Activity;
11 import android.view.Menu;
12 import android.view.View;
13 import android.view.View.OnClickListener;
14 import android.widget.Button;
15 import android.widget.EditText;
16
17 public class MainActivity extends Activity implements
18 OnClickListener{
19
20     Button btncreatecache,btnreadcache;
21     EditText editdata;
22     @Override
23     protected void onCreate(Bundle savedInstanceState) {
24         super.onCreate(savedInstanceState);
25         setContentView(R.layout.activity_main);
26         btncreatecache=(Button) findViewById(R.id.btncreatecache);
27         btnreadcache=(Button) findViewById(R.id.btnreadcache);
28         editdata =(EditText) findViewById(R.id.editdata);

```

```

29     btncreatecache.setOnClickListener(this);
30     btnreadcache.setOnClickListener(this);
31 }
32
33 @Override
34     public boolean onCreateOptionsMenu(Menu menu) {
35         // Inflate the menu; this adds items to the action bar if it
36         is present.
37         getMenuInflater().inflate(R.menu.activity_main, menu);
38         return true;
39     }
40
41 @Override
42     public void onClick(View v) {
43         // TODO Auto-generated method stub
44         if(v.getId()==R.id.btncreatecache)
45         {
46             createCache();
47         }
48         else if(v.getId()==R.id.btnreadcache)
49         {
50             readCache();
51         }
52     }
53 /**
54 * Lấy toàn bộ file cache
55 */
56 public void loadAllCache()
57 {
58     File pathCacheDir = getCacheDir();
59     File []listCache= pathCacheDir.listFiles();
60     for(File f :listCache)
61     {
62         //process f here
63         f.delete();
64     }
65 }
66 /**
67 * đọc cache file
68 * getCacheDir() trả về đúng thư mục cache
69 */
70 public void readCache() {
71     try {
72         File pathCacheDir = getCacheDir();
73         String strCacheFileName = "myCacheFile.cache";
74         File newCacheFile = new
75         File(pathCacheDir, strCacheFileName);
76         Scanner sc=new Scanner(newCacheFile);
77         String data="";
78         while(sc.hasNext())
79         {
80             data+=sc.nextLine()+"\n";
81         }
82         editdata.setText(data);
83         sc.close();

```

```

83     } catch (FileNotFoundException e) {
84     e.printStackTrace();
85   }
86 }
87 /**
88 * Lưu cache file
89 */
90 public void createCache()
91 {
92   try {
93     File pathCacheDir = getCacheDir();
94     String strCacheFileName = "myCacheFile.cache";
95     String strFileContents = editdata.getText()+"";
96     File newCacheFile = new
97       File(pathCacheDir, strCacheFileName);
98     newCacheFile.createNewFile();
99     FileOutputStream foCache =
100      new FileOutputStream(
101        newCacheFile.getAbsolutePath());
102      foCache.write(strFileContents.getBytes());
103      foCache.close();
104    } catch (IOException e) {
105      e.printStackTrace();
106    }
107  }
108 }

```

- Vì giao diện và cách chạy tương tự như ví dụ trước nên Tôi không chụp hình minh họa,

- Bạn có thể tải coding mẫu ở
đây: <http://www.mediafire.com/download/u0smn6v68xf6uc5/LearnCacheFile.rar>

- Như vậy tôi đã giới thiệu xong tương tác tập tin trong Android: internal Storage, external storage, caching file

- Trong bài tập kế tiếp Tôi sẽ hướng dẫn các bạn cách tương tác với XML file trong Android, trong bài tập này Tôi muốn trình bày về 2 kỹ thuật duyệt tập tin XML đó là : **DOM** và **SAX**. Tôi thấy nó rất hay, các bạn nên chú ý theo dõi.

Bài tập 29: XML Parser trong Android

- Tiếp tục chuỗi xử lý tập tin trong Android, bài tập này Tôi mong muốn các bạn sẽ làm được những công việc sau:

- 1) Biết cách tạo XML (tất nhiên rất nhiều bạn đã biết nó từ lâu)
- 2) Biết cách sử dụng XML
- 3) Biết cách duyệt XML bằng kỹ thuật DOM
- 4) Biết cách duyệt XML bằng kỹ thuật SAX.

- Dĩ nhiên bạn phải thực hiện được 2 bài ví dụ bên dưới thì mới có thể nói là hiểu về XML trong Android.

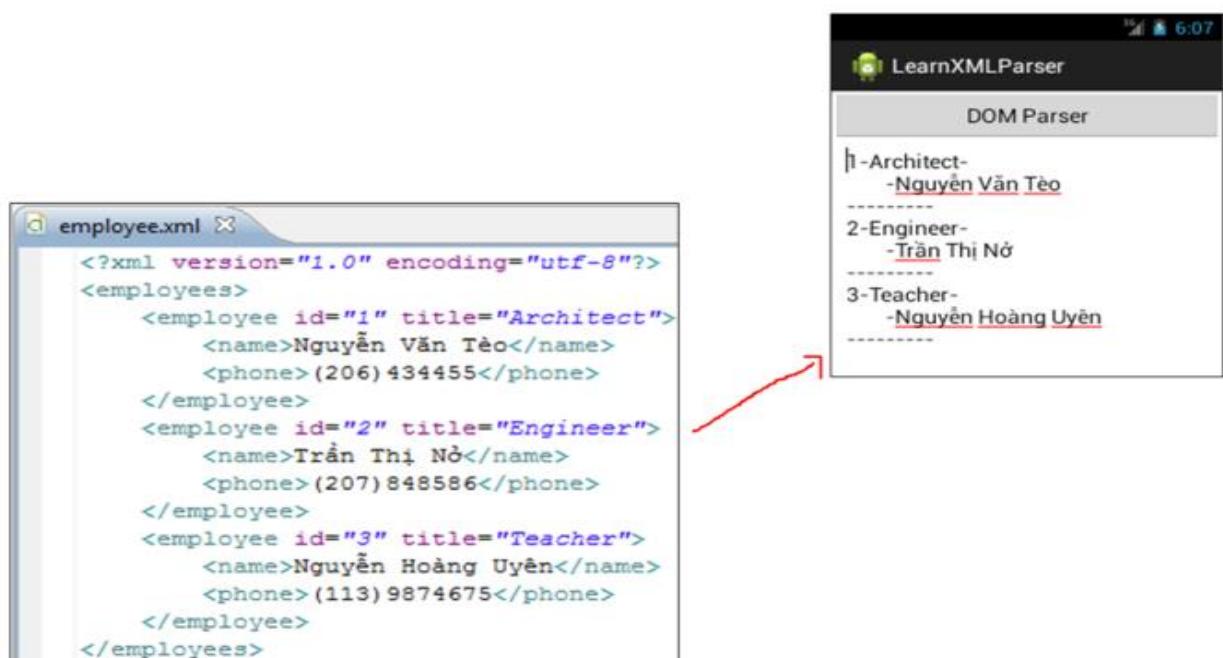
-Extensible Markup Language (XML): Các bạn xem chi tiết tại <http://www.w3schools.com/xml/default.asp>

A) Kỹ thuật dùng DOM:

- DOM (**D**ocument **O**bject **M**odel): Cache all – cơ chế của nó là đọc toàn bộ nội dung tập tin XML vào bộ nhớ (do đó nếu XML lớn thì làm chậm chương trình và có thể phung phí bộ nhớ vì không phải lúc nào ta cũng muốn đọc hết nội dung XML).

- DOM cho phép lấy: NodeLists, .getElementsByTagName() , .item(i), .getName() , .getValue() , .getFirstChild() , .getAttributes() ,...

Ta lướt qua ví dụ này (cần dùng DOM để đọc XML bên dưới lên EditText):



- Ta đi vào các bước cụ thể để đọc được tập tin XML bằng DOM như sau:

Bước 1:

Để có thể sử dụng được kỹ thuật DOM trong Android, chúng ta import các thư viện sau (kể cả những thư viện để bắt lỗi):

```
1 import javax.xml.parsers.DocumentBuilder;
2 import javax.xml.parsers.DocumentBuilderFactory;
3 import javax.xml.parsers.ParserConfigurationException;
4 import org.w3c.dom.Document;
5 import org.w3c.dom.Element;
6 import org.w3c.dom.Node;
7 import org.w3c.dom.NodeList;
8 import org.xml.sax.SAXException;
9
10 import java.io.FileInputStream;
11 import java.io.FileNotFoundException;
12 import java.io.IOException;
```

- Các bạn chú ý là trong quá trình viết code trong Eclipse, nó sẽ tự import tất cả các thư viện trên giúp ta (**Bạn không phải gõ import dòng nào cả**).

Bước 2:

Tạo đối tượng DocumentBuilder (**builder**) bằng lệnh dưới đây:

```
1 DocumentBuilderFactory
2 fac=DocumentBuilderFactory.newInstance();
3 DocumentBuilder builder=fac.newDocumentBuilder();
```

Bước 3:

Tạo FileInputStream từ tập tin XML nguồn (ở đây Tôi để XML nguồn trong SD Card):

```
1 String
2 sdcard=Environment.getExternalStorageDirectory().getAbsolutePath();
3 String xmlfile=sdcard+"/employee.xml";
4 FileInputStream fIn=new FileInputStream(xmlfile);
```

Bước 4:

Dùng phương thức parse của đối tượng **builder** ở bước 2 để tạo Document

```
1 Document doc=builder.parse(fIn);
```

Bước 5:

Ta dựa vào đối tượng **doc** được tạo ra ở bước 4 để duyệt từng node trong XML:

```
Element root= doc.getDocumentElement(); //lấy tag Root ra
NodeList list= root.getChildNodes(); // lấy toàn bộ node con
của Root
String datashow=""; //biến để lưu thông tin
for(int i=0;i<list.getLength();i++) // duyệt từ node đầu
tiên cho tới node cuối cùng
1 {
2     Node node=list.item(i); // mỗi lần duyệt thì lấy ra 1 node
3     if(node instanceof Element) // kiểm tra xem node đó có phải
4     là Element hay không, vì ta dựa vào element để lấy dữ liệu
5     bên trong
6     {
7         Element employee=(Element) node; // lấy được tag Employee ra
8         String id=employee.getAttribute("id"); //id là thuộc tính
9         của tag Employee
10        String title=employee.getAttribute("title"); //title là
11        thuộc tính của tag employee
12        NodeList listChild=
13        employee.getElementsByTagName("name"); // lấy tag name bên
14        trong của tag Employee
15        String name=listChild.item(0).getTextContent(); //lấy nội
16        dung của tag name
17        listChild=employee.getElementsByTagName("phone"); // lấy tag
18        phone bên trong của tag Employee
19        String phone=listChild.item(0).getTextContent(); // lấy nội
dung của tag phone</span>
        datashow+=id+"-"+title+"-"+name+"-"+phone+"\n-----\n";
//lưu vào biến lưu thông tin
    }
}
//ta dựa vào datashow để hiển thị lên giao diện
```

- Bạn có thể tải code mẫu ở
đây: http://www.mediafire.com/download/04q4rlg33nmrina/LearnXMLParser_DOM.rar (Tôi có đính kèm employee.xml trong project này, bạn chỉ việc kéo thả nó vào SD Card trong máy của bạn là ok).

B) kỹ thuật dùng SAX:

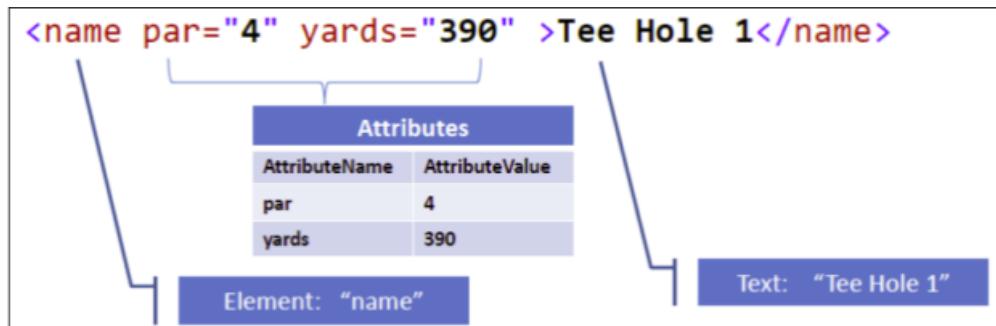
Cũng với ví dụ ở mục A, nhưng Ta sẽ viết theo kỹ thuật SAX.

- Simple API for XML , scan the document , Đỡ tốn bộ nhớ , Chạy nhanh , viết phức tạp hơn DOM
- Các tag là element trong SAX có thể dùng các hàm:

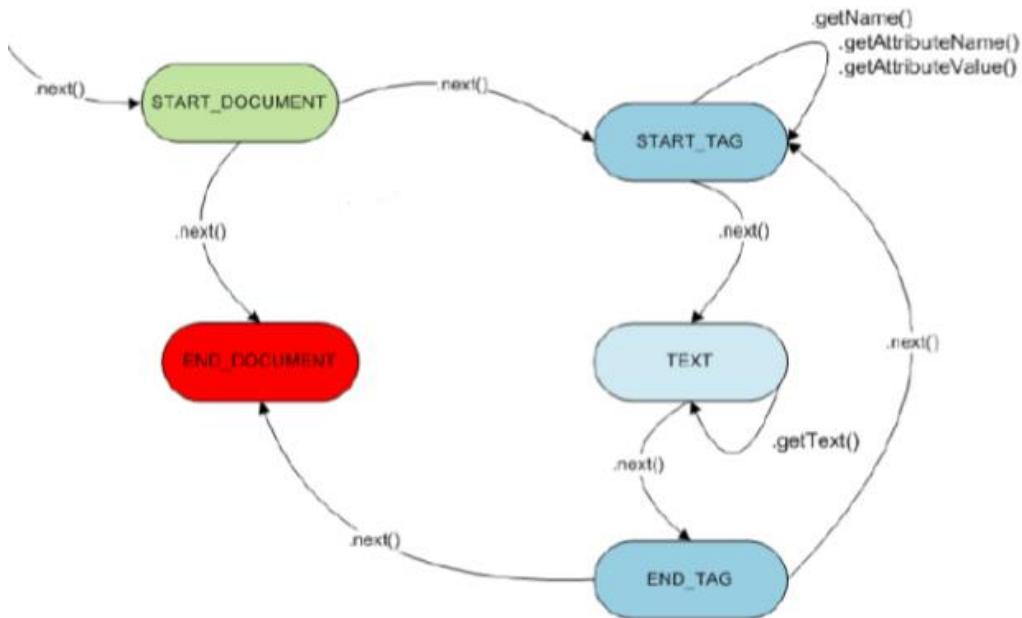
.getAttributeCount()

.getAttributeName()

.getAttributeValue()



- Sử dụng XmlPullParser để scan tài liệu:



- Bước 1:

- Ta cần import các thư viện sau:

```
1 import org.xmlpull.v1.XmlPullParser;
2 import org.xmlpull.v1.XmlPullParserException;
3 import org.xmlpull.v1.XmlPullParserFactory;
4
5 import java.io.FileInputStream;
6 import java.io.FileNotFoundException;
7 import java.io.IOException;
```

- Ta cũng không phải gõ từng dòng import mà eclipse tự động import giúp.

- Bước 2:

Tạo đối tượng parser từ class **XmlPullParser**

```
1 XmlPullParserFactory fc=XmlPullParserFactory.newInstance();
2 XmlPullParser parser= fc.newPullParser();
```

- Bước 3:

- Tạo FileInputStream từ xml source (XML để trong SD Card)

```
1 String sdcard=Environment.  
2     getExternalStorageDirectory().getAbsolutePath();  
3     String xmlfile=sdcard+"/employee.xml";  
4     FileInputStream fIn=new FileInputStream(xmlfile);
```

- Bước 4:

- Tiến hành duyệt

```
parser.setInput(fIn, "UTF-8");  
1 int eventType=-1;  
2     String nodeName;  
3     String datashow="";  
4     while(eventType!=XmlPullParser.END_DOCUMENT) //chưa kết thúc  
tài liệu  
5     {  
6         eventType=parser.next(); // bắt đầu duyệt để  
switch(eventType)  
7         {  
8             case XmlPullParser.START_DOCUMENT:  
9                 break;  
10            case XmlPullParser.END_DOCUMENT:  
11                break;  
12            case XmlPullParser.START_TAG://là tag mở  
13                nodeName=parser.getName();  
14                if(nodeName.equals("employee")){// kiểm tra đúng tag mình muốn  
hay không  
15                    datashow+=parser.getAttributeValue(0)+"-";//lấy giá trị của  
thuộc tính  
16                    datashow+=parser.getAttributeValue(1)+"-";  
17                }  
18                else if(nodeName.equals("name")){  
19                    datashow+=parser.nextText()+"-";//lấy nội dung tag element  
20                }  
21                else if(nodeName.equals("phone")){  
22                    datashow+=parser.nextText()+"-";  
23                }  
24                break;  
25            case XmlPullParser.END_TAG://là tag đóng  
26                nodeName=parser.getName();  
27                if(nodeName.equals("employee")){  
28                    datashow+="\n-----\n";  
29                }  
30                break;  
31            }  
32        }  
33    }  
34 }  
35 }  
36 }  
37  
//dựa vào datashow để hiển thị lên giao diện
```

- Bạn có thể tải code mẫu ở đây
: http://www.mediafire.com/download/fh22pkib490n7lf/LearnXMLParser_SAX.rar

Bài tập dành cho độc giả:

- Duyệt XML bằng DOM và SAX
- Title của các tag XML ta đưa vào Spinner (chú ý không đưa trùng lặp)
- Các thông tin khác đưa vào ListView
- Mỗi lần chọn Title trong Spinner thì chỉ hiển thị những employee theo Title này trong XML mà thôi

```
<?xml version="1.0" encoding="UTF-8"?>
<employees>
    <employee id="1" title="Architect">
        <name>Đoàn Ái Nương</name>
        <phone>01656152042</phone>
    </employee>
    <employee id="2" title="Engineer">
        <name>Đoàn Chính Thuần</name>
        <phone>01655654534</phone>
    </employee>
    <employee id="3" title="Teacher">
        <name>Trần Chính Nghĩa</name>
        <phone>0981234567</phone>
    </employee>
    <employee id="4" title="Architect">
        <name>Châu Bá Thông</name>
        <phone>0978976665</phone>
    </employee>
    <employee id="5" title="Engineer">
        <name>Kiều Phong</name>
        <phone>0901334556</phone>
    </employee>
</employees>
```



Bài tập 30: Thực hành về Shared Preferences

Trong bài tập này Tôi sẽ trình bày về cách lưu trạng thái của ứng dụng (liên quan tới **Shared Preferences**) và cách tạo các màn hình cấu hình (liên quan tới **Shared Preference Change Listeners**)

-A- Cách lưu trạng thái của ứng dụng:

Bước 1:

- Gọi hàm **getSharedPreferences**, hàm này trả về SharedPreferences và nó có 2 đối số. Đối số 1 là tên tập tin để lưu trạng thái, đối số 2 là kiểu lưu. Chú ý là đối số 1 ta chỉ ghi tên tập tin (không cần ghi phần mở rộng, vì phần mở rộng mặc nhiên của nó là **.xml** khi ta lưu thành công), đối số 2 thường ta để là MODE_PRIVATE:

SharedPreferences pre=getSharedPreferences("my_data", MODE_PRIVATE);

Bước 2:

- Tạo đối tượng Editor để cho phép chỉnh sửa dữ liệu:

SharedPreferences.Editor edit=pre.edit();

Bước 3:

- Đưa dữ liệu muốn lưu trữ vào **edit** bằng các phương thức **edit.putXXX("key","value")**;

Tùy vào kiểu dữ liệu ta muốn lưu trữ mà XXX được thay thế bởi các kiểu dữ liệu phù hợp:

editor.putString("user", "drthanh");

editor.putString("pwd", "hoilamgi");

editor.putBoolean("checked", true);

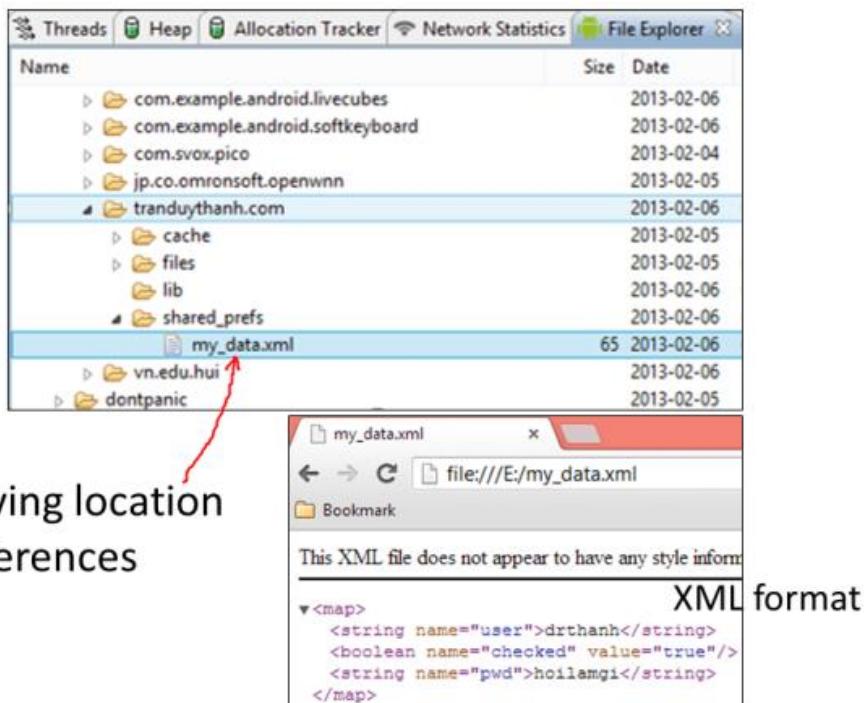
Bước 4:

- Lưu trạng thái bằng cách gọi dòng lệnh:

editor.commit();

Sau khi bạn làm đúng 4 bước trên thì chương trình sẽ lưu được trạng thái, như trên đã nói mặc định phần mở rộng là **.xml** (tức là trạng thái được lưu dưới định dạng tập tin XML). Bên trên ta đặt tên

là **my_data** có nghĩa là chương trình sẽ tạo ra tập tin **my_data.xml** (bạn mở DDMS lên để xem) – tự động nó sẽ lưu vào thư mục **shared_prefs** như hình bên dưới:



-B- Cách đọc trạng thái đã lưu:

Rất đơn giản

Bước 1:

- Gọi hàm `getSharedPreferences` để trả về đối tượng `SharedPreferences` (giống như phần lưu trạng thái mà Tôi nói ở trên)

`SharedPreferences pre=getSharedPreferences ("my_data", MODE_PRIVATE);`

Bước 2:

- Gọi các phương thức `getXXX("key", giá trị mặc định)` để lấy các giá trị lúc trước được lưu

```
boolean bchk=pre.getBoolean("checked", false); //đối số 2 Tôi để false là giá trị mặc định khi nó tìm không thấy key =checked
```

```
String user=pre.getString("user", ""); //lấy giá trị được lưu trong key=user, nếu không thấy thì gán giá trị mặc định là chuỗi rỗng
```

```
String pwd=pre.getString("pwd", ""); //giống trên
```

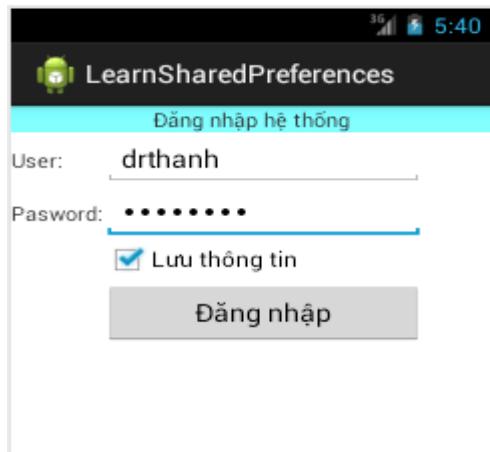
***** Gợi ý *****

- Lưu và đọc trạng thái bạn nên viết thành các hàm riêng biệt cho dễ sử dụng
- Hàm lưu bạn gọi trong sự kiện **onPause**
- Hàm đọc bạn gọi trong sự kiện **onResume**.
- Đừng hỏi tại vì sao???? sẽ bị 0 điểm lý do là Tôi đã giải thích rõ ở [bài tập 5](#) và [bài tập 6](#) rồi. Bạn tự xem lại.

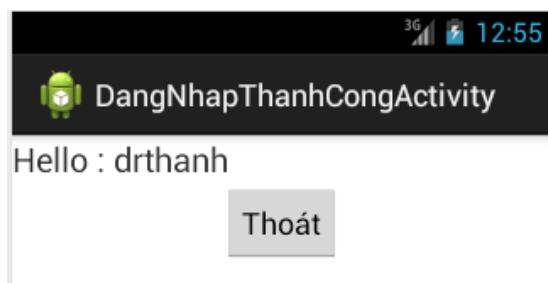
***** END *****

Tôi sẽ làm ví dụ cụ thể dưới đây để bạn hiểu rõ hơn về lý thuyết:

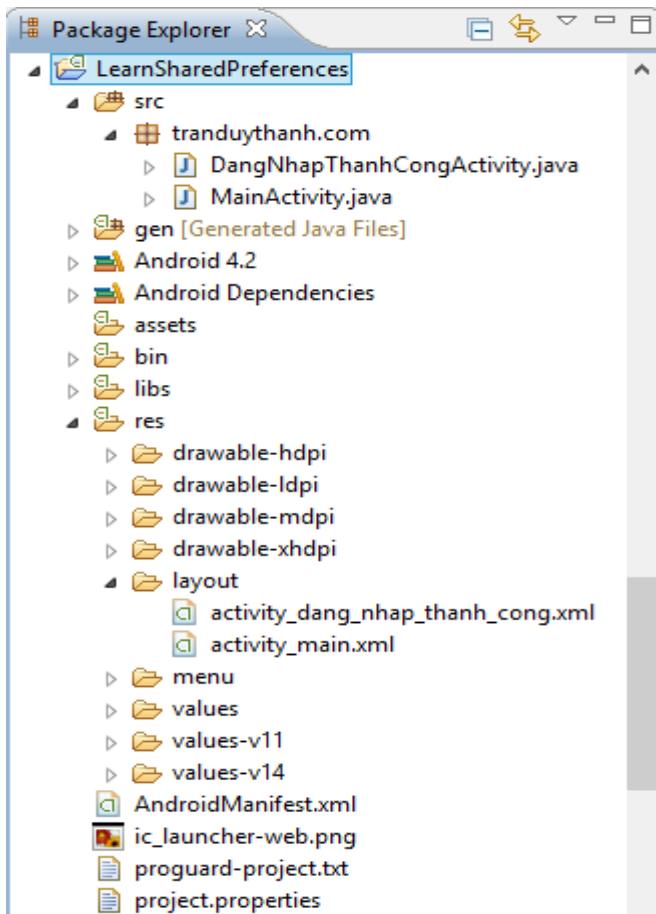
Ví dụ 1: bạn muốn tạo một màn hình đăng nhập có checkbox cho phép lưu lại thông tin đăng nhập, lần sau khởi động lại thì nó sẽ lấy lại thông tin nhập lúc trước để người sử dụng đỡ phải mất công nhập lại (biết về **Shared Preferences**), xem hình minh họa:



- Khi chọn nút đăng nhập, chương trình sẽ đóng Activity hiện tại và hiển thị Activity dưới đây:



- Chọn nút Thoát, chương trình tiếp tục đóng Activity này giúp tắt hẳn các Activity trong ứng dụng.
- Khởi động lại chương trình sẽ phải tự động load lại thông tin đăng nhập trước đó (Nếu khi đăng nhập có chọn “**Lưu thông tin**”).
- Bạn xem cấu trúc của Project:



- Layout XML của màn hình chính (**activity_main.xml**):

```
1 <LinearLayout
2     xmlns:android="http://schemas.android.com/apk/res/android"
3     xmlns:tools="http://schemas.android.com/tools"
4     android:id="@+id/LinearLayout1"
5     android:layout_width="match_parent"
6     android:layout_height="match_parent"
7     android:orientation="vertical"
8     tools:context=".MainActivity" >
9
10    <TextView
11        android:id="@+id/textView1"
12        android:layout_width="match_parent"
13        android:layout_height="wrap_content"
14        android:background="#80FFFF"
15        android:gravity="center"
16        android:text="Đăng nhập hệ thống" />
17
18    <TableLayout
19        android:layout_width="match_parent"
20        android:layout_height="wrap_content"
21        android:stretchColumns="*"
22    >
23
24    <TableRow
25        android:id="@+id/tableRow1"
26        android:layout_width="wrap_content"
```

```
27     android:layout_height="wrap_content" >
28
29 <TextView
30     android:id="@+id/textView2"
31     android:layout_width="wrap_content"
32     android:layout_height="wrap_content"
33     android:text="User:" />
34
35 <EditText
36     android:id="@+id/editUser"
37     android:layout_width="wrap_content"
38     android:layout_height="wrap_content"
39     android:inputType="text"
40     android:ems="10" >
41
42 <requestFocus />
43 </EditText>
44 </TableRow>
45
46 <TableRow
47     android:id="@+id/tableRow2"
48     android:layout_width="wrap_content"
49     android:layout_height="wrap_content" >
50 </TableRow>
51
52 <TableRow
53     android:id="@+id/tableRow3"
54     android:layout_width="wrap_content"
55     android:layout_height="wrap_content" >
56
57 <TextView
58     android:id="@+id/textView3"
59     android:layout_width="wrap_content"
60     android:layout_height="wrap_content"
61     android:text="Pasword:" />
62
63 <EditText
64     android:id="@+id/editPassword"
65     android:layout_width="wrap_content"
66     android:layout_height="wrap_content"
67     android:ems="10"
68     android:inputType="textPassword" />
69 </TableRow>
70
71 <TableRow
72     android:id="@+id/tableRow4"
73     android:layout_width="wrap_content"
74     android:layout_height="wrap_content" >
75
76 <CheckBox
77     android:id="@+id/chksaveacount"
78     android:layout_width="wrap_content"
79     android:layout_height="wrap_content"
80     android:layout_column="1"
81     android:text="Lưu thông tin" />
```

```

81    </TableRow>
82
83    <TableRow
84        android:id="@+id/tableRow5"
85        android:layout_width="wrap_content"
86        android:layout_height="wrap_content" >
87
88        <Button
89            android:id="@+id	btnlogin"
90            android:layout_width="wrap_content"
91            android:layout_height="wrap_content"
92            android:layout_column="1"
93            android:text="Đăng nhập" />
94        </TableRow>
95    </TableLayout>
96
97    </LinearLayout>

```

- Layout XML của **activity_dang_nhap_thanhCong.xml**:

```

1    <LinearLayout
2        xmlns:android="http://schemas.android.com/apk/res/android"
3        xmlns:tools="http://schemas.android.com/tools"
4        android:id="@+id/LinearLayout1"
5        android:layout_width="match_parent"
6        android:layout_height="match_parent"
7        android:orientation="vertical"
8        tools:context=".DangNhapThanhCongActivity" >
9
10       <TextView
11           android:id="@+id/txtmsg"
12           android:layout_width="match_parent"
13           android:layout_height="wrap_content"
14           android:text="TextView"
15           android:textSize="20sp" />
16
17       <Button
18           android:id="@+id/btnThoat"
19           android:layout_width="wrap_content"
20           android:layout_height="wrap_content"
21           android:layout_gravity="center"
22           android:text="Thoát" />
23

```

- Bạn xem Source code **MainActivity.java**:

```

1    package tranduythanh.com;
2
3    import android.os.Bundle;
4    import android.app.Activity;
5    import android.content.Intent;
6    import android.content.SharedPreferences;
7    import android.view.Menu;

```

```

7 import android.view.View;
8 import android.widget.Button;
9 import android.widget.CheckBox;
10 import android.widget.EditText;
11
12 public class MainActivity extends Activity {
13     Button btnlogin;
14     EditText edituser,editpassword;
15     CheckBox chksaveaccount;
16     //đặt tên cho tập tin lưu trạng thái
17     String prefname="my_data";
18     protected void onCreate(Bundle savedInstanceState) {
19         super.onCreate(savedInstanceState);
20         setContentView(R.layout.activity_main);
21         btnlogin=(Button) findViewById(R.id.btnlogin);
22         edituser =(EditText)
23         findViewById(R.id.editUser);
24         editpassword=(EditText)
25         findViewById(R.id.editPassword);
26         chksaveaccount=(CheckBox)
27         findViewById(R.id.chksaveacount);
28         btnlogin.setOnClickListener(
29             new View.OnClickListener() {
30                 public void onClick(View arg0) {
31                     doLogin();
32                 }
33             });
34         }
35         /**
36         * hàm đăng nhập hệ thống
37         */
38         public void doLogin()
39         {
40             finish();//đóng màn hình hiện tại
41             Intent i=new Intent(this, DangNhapThanhCongActivity.class);
42             //truyền dữ liệu qua màn hình mới
43             i.putExtra("user", edituser.getText().toString());
44             startActivity(i);//mở màn hình mới
45         }
46         @Override
47         protected void onPause() {
48             // TODO Auto-generated method stub
49             super.onPause();
50             //gọi hàm lưu trạng thái ở đây
51             savingPreferences();
52         }
53         @Override
54         protected void onResume() {
55             // TODO Auto-generated method stub
56             super.onResume();
57             //gọi hàm đọc trạng thái ở đây
58             restoringPreferences();
59         }
60         /**
61         * hàm lưu trạng thái

```

```

61     */
62     public void savingPreferences()
63     {
64         //tạo đối tượng getSharedPreferences
65         SharedPreferences pre=getSharedPreferences
66             (prefname, MODE_PRIVATE);
67         //tạo đối tượng Editor để lưu thay đổi
68         SharedPreferences.Editor editor=pre.edit();
69         String user=edituser.getText().toString();
70         String pwd=editpassword.getText().toString();
71         boolean bchk=chksaveaccount.isChecked();
72         if(!bchk)
73         {
74             //xóa mọi lưu trữ trước đó
75             editor.clear();
76         }
77         else
78         {
79             //lưu vào editor
80             editor.putString("user", user);
81             editor.putString("pwd", pwd);
82             editor.putBoolean("checked", bchk);
83         }
84         //chấp nhận lưu xuống file
85         editor.commit();
86     }
87     /**
88     * hàm đọc trạng thái đã lưu trước đó
89     */
90     public void restoringPreferences()
91     {
92         SharedPreferences pre=getSharedPreferences
93             (prefname, MODE_PRIVATE);
94         //lấy giá trị checked ra, nếu không thấy thì giá trị mặc
95         định là false
96         boolean bchk=pre.getBoolean("checked", false);
97         if(bchk)
98         {
99             //lấy user, pwd, nếu không thấy giá trị mặc định là rỗng
100            String user=pre.getString("user", "");
101            String pwd=pre.getString("pwd", "");
102            edituser.setText(user);
103            editpassword.setText(pwd);
104            chksaveaccount.setChecked(bchk);
105        }
106        @Override
107        public boolean onCreateOptionsMenu(Menu menu) {
108            // Inflate the menu; this adds items to the action bar if it
109            is present.
110            getMenuInflater().inflate(R.menu.activity_main, menu);
111            return true;
112        }
113    }
114

```

- Bạn xem source code **DangNhapThanhCongActivity.java**:

```
1  package tranduythanh.com;
2
3  import android.os.Bundle;
4  import android.app.Activity;
5  import android.content.Intent;
6  import android.view.Menu;
7  import android.view.View;
8  import android.view.View.OnClickListener;
9  import android.widget.Button;
10 import android.widget.TextView;
11
12 public class DangNhapThanhCongActivity extends Activity {
13
14     TextView txtMsg;
15     Button btnThoat;
16     @Override
17     protected void onCreate(Bundle savedInstanceState) {
18         super.onCreate(savedInstanceState);
19         setContentView(R.layout.activity_dang_nhap_thanh_cong);
20         txtMsg=(TextView) findViewById(R.id.txtmsg);
21         btnThoat=(Button) findViewById(R.id.btnThoat);
22         btnThoat.setOnClickListener(new OnClickListener() {
23
24             @Override
25             public void onClick(View arg0) {
26                 // TODO Auto-generated method stub
27                 finish();
28             }
29         });
30
31         Intent i=getIntent();
32         txtMsg.setText("Hello : "+i.getStringExtra("user"));
33     }
34     @Override
35     public boolean onCreateOptionsMenu(Menu menu) {
36         // Inflate the menu; this adds items to the action bar if it is
37         present.
38         getMenuInflater().inflate(R.menu.activity_dang_nhap_thanh_cong,
39         menu);
39         return true;
40     }
41 }
42 }
```

- Như vậy là đã xong, bạn đã biết được cách lưu trạng thái. Chú ý là lưu trạng thái chỉ lưu Primitive Data, không cho phép lưu Object Serialize.

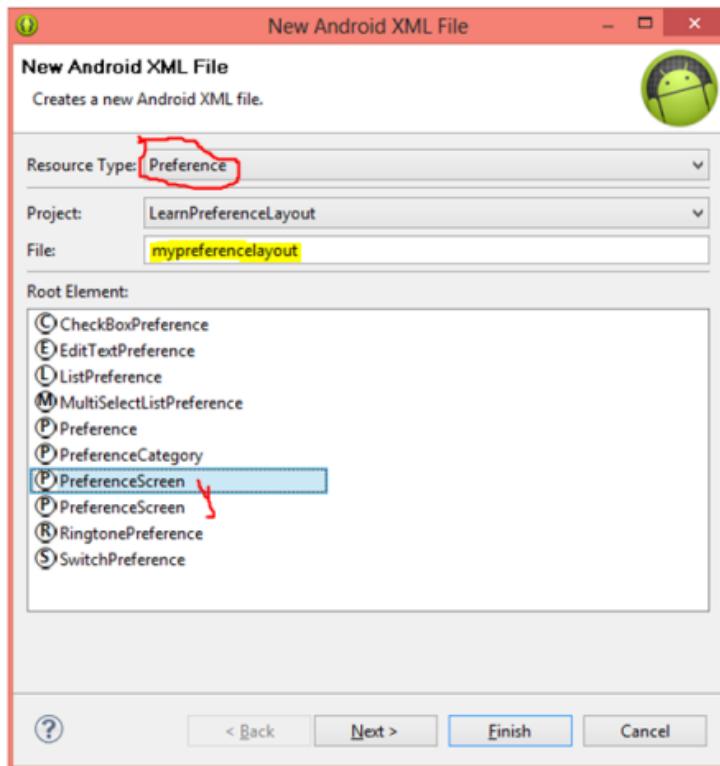
- Bạn click vào đây để tải coding

mẫu: <http://www.mediafire.com/download/4e4llyykdukgo82/LearnSharedPreferences.rar>

-B- cách tạo Shared Preference Change Listeners để tạo giao diện cấu hình:

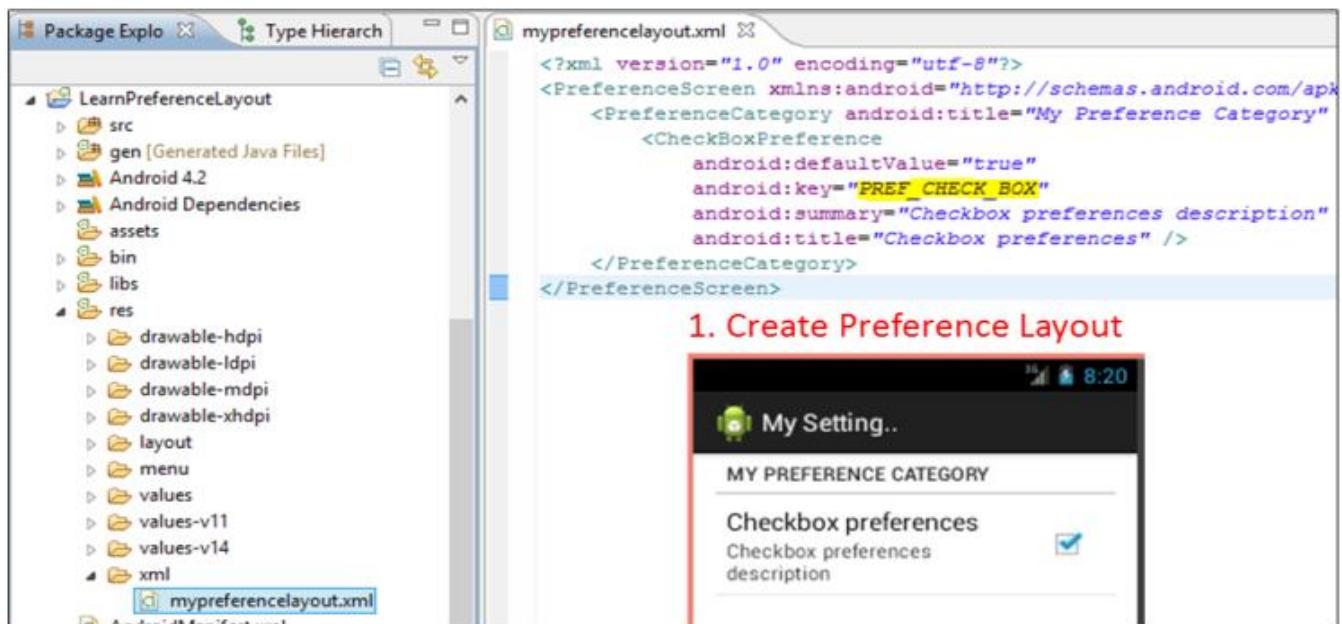
Bước 1:

- Tạo một Project Android bất kỳ, sau đó tạo thêm một layout PreferenceScreen:



- đặt tên tùy ý (ở đây Tôi đặt tên **myprefencelayout**)

- Không giống như layout bình thường, preference screen sẽ nằm trong thư mục xml như hình bên dưới:



- Tiếp theo tạo Checkbox giống như trên. (nó có nhiều control khác, tùy vào nhu cầu)

Bước 2:

- Tạo Activity cho Prefence layout trên (Ké thừa từ **PreferenceActivity**):

```
public class MySettingActivity
    extends PreferenceActivity {
protected void onCreate
    (Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
addPreferencesFromResource
    (R.xml.mypreferencelayout);
}
}
```

Bước 3:

- Cấu hình Manifest XML cho Preference Activity ở trên:

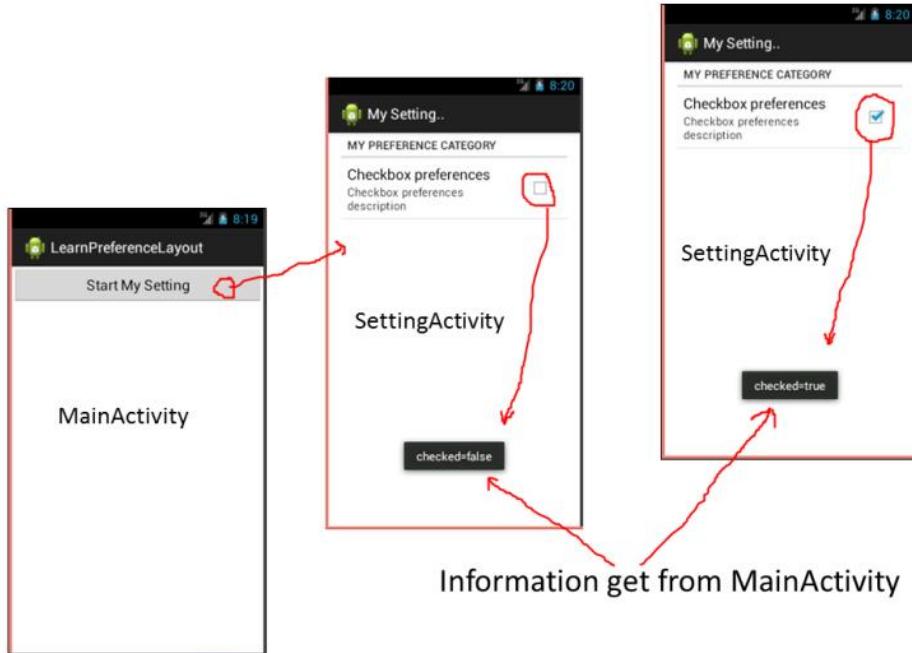
```
<activity
    android:name="tranduythanh.com.MySettingActivity"
    android:label="My Setting...">
</activity>
```

Bước 4:

- Tiến hành sửa MainActivity để sử dụng Preference Activity (Đăng ký**OnSharedPreferenceChangeListener**):

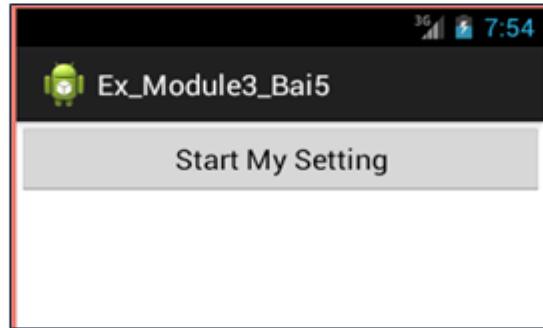
```
public class MainActivity extends Activity
    implements OnSharedPreferenceChangeListener
{
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        Button btn=(Button) findViewById(R.id.btnsetting);
        btn.setOnClickListener(new View.OnClickListener() {
            public void onClick(View arg0) {
                Intent intent=new Intent(MainActivity.this,
                    MySettingActivity.class);
                startActivityForResult(intent, 113);
            }
        });
        Context ctx=getApplicationContext();
        SharedPreferences pre=PreferenceManager.
            getDefaultSharedPreferences(ctx);
        pre.registerOnSharedPreferenceChangeListener(this);
    }
    public void onSharedPreferenceChanged(
        SharedPreferences sharedPreferences, String key) {
        boolean data=sharedPreferences.getBoolean(key, false);
        Toast.makeText(this, "checked="+data, Toast.LENGTH_LONG).show();
    }
}
```

- Bạn có thể xem kết quả như bên dưới:

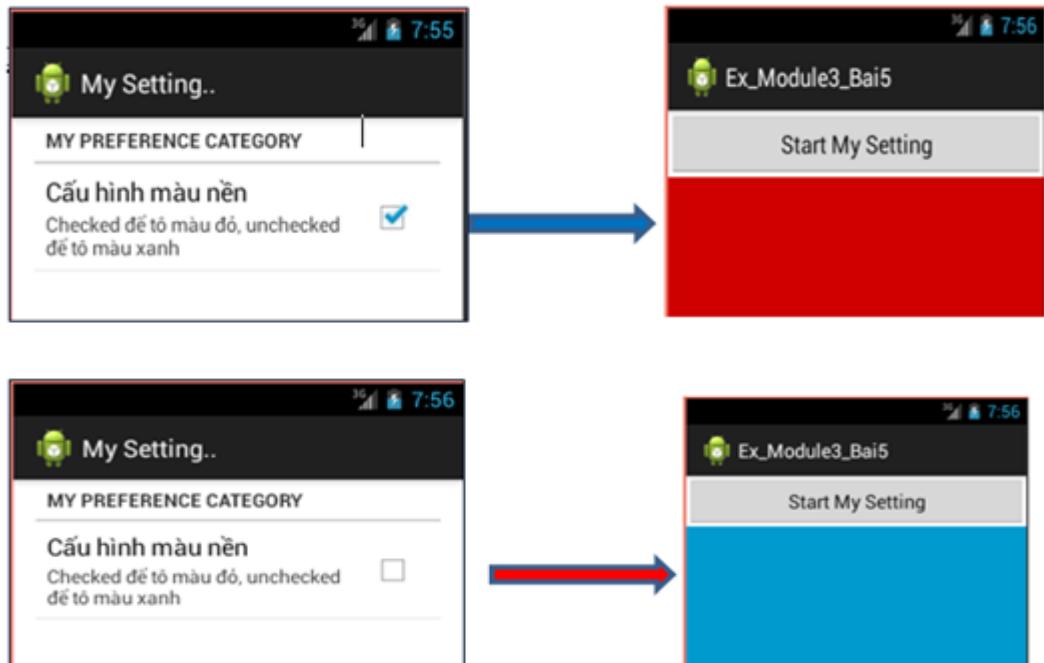


Ví dụ 2:

- Bạn cố gắng tự xem lại lý thuyết và đọc coding để hiểu. Ví dụ này Tôi không giải thích coding.
- Hay bạn cần tạo ra các màn hình để cấu hình (giống như màn hình Setting của thiết bị chẵng hạn – cần biết **Shared Preference Change Listeners**):



- Khi bấm Start my Setting sẽ hiển thị màn hình Setting:



- o Nếu checked màn hình chính sẽ có màu đỏ
- o Nếu unchecked màn hình chính sẽ có màu xanh

- Bạn tải source code ở

đây: http://www.mediafire.com/download/lg8881lz8jrp1np/Ex_Module3_Bai5.rar

Bài tập 31: Cách sử dụng SQLite trong Android

Bài tập này Tôi sẽ hướng dẫn các bạn cách sử dụng SQLite trong Android.

Thay vì lưu trữ bằng text file, XML hay SharePreference thì bạn cũng có thể lưu trữ thông tin bằng SQLite. SQLite đã được tích hợp sẵn trong Android SDK.

Trong bài này các bạn sẽ học các phần sau:

1) Cách tạo / xóa một cơ sở dữ liệu SQLite trong Android

2) Cách tạo / xóa bảng trong SQLite

3) Cách thêm/ sửa/ xóa dữ liệu trong bảng

4) Cách truy vấn dữ liệu trong bảng.

- Tất nhiên còn rất nhiều chức năng khác, nhưng theo Tôi các bạn chỉ cần làm tốt 4 phần này thì có thể viết ứng dụng Android có SQLite một cách chuyên nghiệp rồi.

- Theo Tôi thì các bạn nên sử dụng công cụ SQLite

Administrator: <http://download.orbmw2k.de/files/sqliteadmin.zip> để tạo hoàn chỉnh 1 CSDL sau đó kéo thả tập tin đó vào DDMS cho lẹ (cái này bạn tải về và tự tạo, rồi kéo thả vào DDMS). Còn các hướng dẫn dưới này Tôi muốn giúp các bạn hiểu được sâu xa bên trong (hướng programmer) CSDL SQLite.

- Giả sử các bạn cần tạo CSDL như mô tả dưới đây (**qlquanlysinhvien.db**):

- Bảng Lớp học (**tbllop**):

tbllop		
Tên cột	Kiểu dữ liệu	Mô tả
malop	TEXT	Khóa chính
tenlop	TEXT	
siso	INTEGER	

- Bảng sinh viên (**tblsinhvien**):

tblsinhvien		
Tên cột	Kiểu dữ liệu	Mô tả
masv	TEXT	Khóa chính
tensv	TEXT	
malop	TEXT	khóa ngoại

- Để sử dụng SQLite, bạn import thư viện sau:

import android.database.sqlite.SQLiteDatabase;

1) Cách tạo / xóa một cơ sở dữ liệu SQLite trong Android:

- Tao moi 1 CSDL:

```

public void doCreateDb()
{
    SQLiteDatabase database;
    database=openOrCreateDatabase(
        "qlsinhvien.db",
        MODE_PRIVATE,
        null);
}

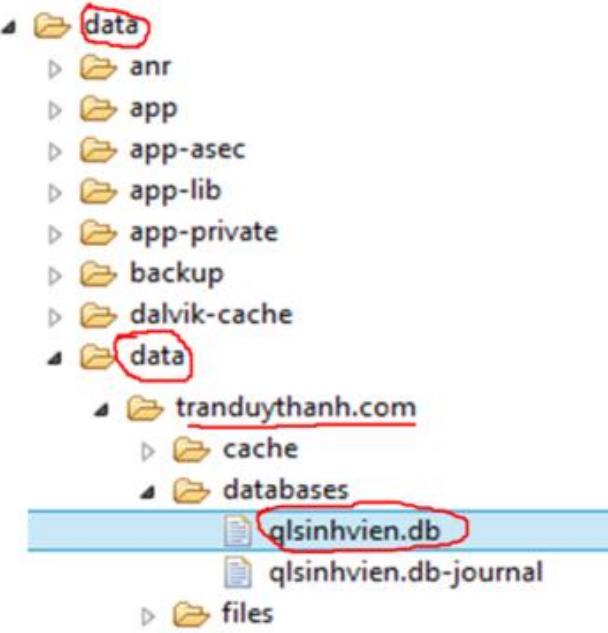
database= SQLiteDatabase.openDatabase(
    "qlsinhvien.db",
    null,
    SQLiteDatabase.CREATE_IF_NECESSARY);

```

Khi lưu thành công, nó sẽ lưu CSDL vào:

/data/data/app/databases/<dbname>

cu the:



- Nếu bạn muốn lưu trữ trên SD Card thì bắt buộc bạn phải cấp quyền giống như đã đề cập tới ở những bài trước:

`<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />`

- Bạn chỉ việc lấy đường dẫn của SD Card ra rồi lưu tên CSDL vào đúng đường dẫn SD Card là ok. (Bạn tự xem lại các bài tập trước mà Tôi đã hướng dẫn cách làm).

- Xóa 1 CSDL:

```
public void doDeleteDb()
{
    String msg="";
    if(deleteDatabase("qlsinhvien.db")==true)
    {
        msg="Delete database [qlsinhvien.db] is successful";
    }
    else
    {
        msg="Delete database [qlsinhvien.db] is failed";
    }
    Toast.makeText(this, msg, Toast.LENGTH_LONG).show();
}
```

- Như trên thì ta chỉ cần gọi lệnh **deleteDatabase**(tên CSDL). Nếu xóa thành công thì trả về **true**, xóa thất bại trả về **false**;

2) Cách tạo / xóa bảng trong SQLite:

- Ở đây các bạn sẽ tạo 2 bảng **tbllop** và **tblsinhvien**. Chú ý là chúng có mối ràng buộc toàn vẹn.

- Bạn xem cách tạo bảng lớp:

```

public void doCreateLopTable()
{
    String sql="CREATE TABLE tbllop ("+
               "malop TEXT primary key,"+
               "tenlop TEXT,"+
               "siso INTEGER)";
    database.execSQL(sql);
}

```

- Bạn chú ý là tên đối tượng **database** (chỗ **database.execSQL(sql)**) là đối tượng SQLiteDatabase được tạo ra ở bước tạo CSDL. Bạn phải khai báo cho phù hợp để ở trong hàm này cũng có thể truy suất.

- Tao bảng sinh viên:

```

public void doCreateSinhvienTable()
{
    String sql="CREATE TABLE tblsinhvien ("+
               "masv TEXT PRIMARY KEY ,"+ 
               "tensv TEXT,"+
               "malop TEXT NOT NULL CONSTRAINT malop "+
               " REFERENCES tbllop(malop) ON DELETE CASCADE )";
    database.execSQL(sql);
}

```

- Vì bảng sinh viên có chứa khóa ngoại để tham chiếu tới bảng lớp, nên bạn phải chú ý dòng lệnh tham chiếu ở trên.

3) Cách thêm/ sửa/ xóa dữ liệu trong bảng:

- Cách thêm một dòng dữ liệu vào bảng:

+ Dùng đối tượng **ContentValues** để đưa dữ liệu vào bảng. Đối tượng này có các phương thức **put(tên cột, dữ liệu)**

+ Sau đó gọi phương thức **insert** để đưa đối tượng (dòng này) vào bảng.

+ Bạn chú ý là phương thức **insert** có rất nhiều loại đối số khác nhau, nhưng ở đây Tôi chỉ nói 1 loại đơn giản nhất (các kiểu khác bạn tự tìm hiểu thêm). Loại mà Tôi muốn đề cập tới đó là không liên quan gì tới kiểm tra các điều kiện, chỉ cần đưa đối tượng **ContentValues** vào **insert** là bạn sẽ có được 1 dòng mới:

```

public void doInsertRecord()
{
    ContentValues values=new ContentValues();
    values.put("malop", "DHTH7C");
    values.put("tenlop", "Đại học 7C");
    values.put("siso", 30);
    String msg="";
    if(database.insert("tbllop", null, values)==-1) {
        msg="Failed to insert record";
    }
    else{
        msg="insert record is successful";
    }
    Toast.makeText(this, msg, Toast.LENGTH_LONG).show();
}

```

- Nhìn vào đoạn code ở trên, bạn thấy đó Tôi sử dụng cả 3 cột **malop**, **tenlop**, **siso** của bảng lớp học:

values.put("malop","DHTH7C"); tức là đưa giá trị “DHTH7C” vào cột **malop**.

- dòng lệnh **database.insert("tbllop",null,values)**; Đối số 1 là tên bảng, đối số 2 bạn truyền null, đối số 3 bạn truyền đối tượng values

- Nếu thêm thành công thì sẽ trả về giá trị khác -1. Nếu bảng -1 là thất bại.

- Cách cập nhật dữ liệu:

- Ta dùng hàm **update** để cập nhật dữ liệu theo một điều kiện bất kỳ nào đó.

public int update
(String table, ContentValues values, String whereClause, String[] whereArgs)

- Đối số 1 là tên bảng

- Đối số 2 là đối tượng muốn chỉnh sửa (với giá trị mới)

- Đối số 3 là tập các điều kiện lọc (dùng dấu chấm hỏi ? để tạo điều kiện lọc)

- Đối số 4 là tập các giá trị của điều kiện lọc (lấy theo đúng thứ tự)

- Hàm này trả về số dòng bị ảnh hưởng. Ví dụ nếu có 3 dòng bị thay đổi thì nó trả về 3. nếu không có dòng nào bị ảnh hưởng thì nó trả về 0.

Ví dụ: Tôi viết hàm chỉnh sửa (bạn cũng nên tách thành từng hàm giống vậy):

```

public void updateTenlopName(String malop, String new_tenlop)
{
    ContentValues values=new ContentValues();
    values.put("tenlop", new_tenlop);
    int ret=database.update("tbllop", values,
                           "malop=?", new String[]{malop});
    if(ret==0){
        //failed;
    }
    else{
        //ok
    }
}

```

- Hàm trên : Đối số 1 là mã nào muốn được chỉnh sửa . Đối số 2 là giá trị mới. Cụ thể bảng lớp của chúng ta có 3 cột: mã lớp, tên lớp và sĩ số, nhưng mà Tôi chỉ muốn chỉnh sửa Tên lớp mà thôi. Do đó bạn thấy như vậy (ở đây Tôi chủ ý viết khuyết như vậy để các bạn hiểu rằng không nhất thiết phải sử dụng hết các cột).

- Xóa dữ liệu:

- Ta dùng hàm delete để xóa:

```
public int delete (String table, String whereClause, String[] whereArgs)
```

- Đối số 1 là tên bảng
- Đối số 2 là tập điều kiện lọc (dùng ? để tạo)
- Đối số 3 là tập các giá trị của điều kiện lọc
- Hàm trả về số dòng bị ảnh hưởng.
- Muốn xóa toàn bộ dữ liệu trong bảng thì ta truyền null vào 2 đối số cuối:

```
database.delete("tbllop", null, null);
```

- Muốn xóa theo 1 mã nào đó:

```

String malop="DHTH7C";
database.delete("tbllop",
               "malop=?",
               new String [] {malop});

```

4) Cách truy vấn dữ liệu trong bảng.

- Là thao tác phức tạp nhất trong truy suất SQLite
- Ta dùng Cursor để lưu trữ giá trị trả về của hàm dưới đây:

```
public Cursor query
(String table, String[] columns, String selection, String[] selectionArgs, String groupBy, String having, String orderBy)
```

Xem bảng mô tả chi tiết:

table	The table name to compile the query against.
columns	A list of which columns to return. Passing null will return all columns, which is discouraged to prevent reading data from storage that isn't going to be used.
selection	A filter declaring which rows to return, formatted as an SQL WHERE clause (excluding the WHERE itself). Passing null will return all rows for the given table.
selectionArgs	You may include ?s in selection, which will be replaced by the values from selectionArgs, in order that they appear in the selection. The values will be bound as Strings.
groupBy	A filter declaring how to group rows, formatted as an SQL GROUP BY clause (excluding the GROUP BY itself). Passing null will cause the rows to not be grouped.
having	A filter declare which row groups to include in the cursor, if row grouping is being used, formatted as an SQL HAVING clause (excluding the HAVING itself). Passing null will cause all row groups to be included, and is required when row grouping is not being used.
orderBy	How to order the rows, formatted as an SQL ORDER BY clause (excluding the ORDER BY itself). Passing null will use the default sort order, which may be unordered.

- Ở đây Tôi làm ví dụ đơn giản nhất là truy vấn không phải lọc theo điều kiện nào cả:

- Ví dụ đọc tất cả danh sách lớp học trong bảng **tbllop**:

```

public void loadallloop()
{
    Cursor c=database.query("tblloop",
                           null, null, null, null, null, null);
    c.moveToFirst();
    String data="";
    while(c.isAfterLast()==false)
    {
        data+=c.getString(0)+"-"++
            c.getString(1)+"-"++
            c.getString(2);
        data+="\n";
        c.moveToNext();
    }
    Toast.makeText(this, data, Toast.LENGTH_LONG).show();
    c.close();
}

```

- database.query sẽ trả về một Cursor, Lúc này Cursor đầu đọc chưa trỏ tới dòng dữ liệu nào cả. Do đó ta phải gọi lệnh .moveToFirst để Cursor trỏ đầu đọc tới dòng đầu tiên. Sau đó ta dùng vòng lặp while để duyệt từng dòng dữ liệu. Chú ý là Cursor này giống như Pointer nó cho phép truy suất ngẫu nhiên.

- Bạn có thể tải coding mẫu đầy đủ của phần hướng dẫn này ở đây: <http://www.mediafire.com/download/leuuld4a225tw5c/LearnSQLite.rar>
- Bài tập tiếp theo Tôi sẽ yêu cầu các bạn viết chương trình quản lý sách sử dụng SQLite để bạn củng cố thêm kiến thức về nó.

Bài tập 32: Tiếp tục củng cố kiến thức SQLite, ví dụ tổng hợp quản lý sách

Để làm được bài tập 32 này thì bắt buộc bạn phải đạt **Đai Đen** [bài tập 31](#), đồng thời bạn phải rành custom layout, intent ... Trong bài tập này Tôi chủ ý không giải thích coding nhiều là để các bạn tự suy luận logic, tự tổng hợp lại các kiến thức đã học được từ các bài tập trước để hoàn thành bài này. Vì thực ra bài tập này là kết hợp của tất cả các kiến thức trước đó.

- **Viết chương trình quản lý sách được mô tả như sau:**

- Một tác giả sẽ có nhiều cuốn sách, thông tin tác giả gồm: mã , tên
- Mỗi cuốn sách thuộc về một nhà tác giả nào đó, thông tin mỗi cuốn sách gồm: mã sách, tên sách, ngày xuất bản
- Hãy chọn kiểu dữ liệu hợp lý để tạo cơ sở dữ liệu (sử dụng SQLite) cho đúng với mô tả trên

- **Thiết kế giao diện như hình bên dưới:**



- Phải tạo Tác giả trước khi vào chức năng quản lý Sách
- Khi chọn chức năng thêm Tác giả, chương trình sẽ xuất hiện màn hình dưới đây (chú ý mở dưới dạng Dialog):



- Chọn “Lưu tác giả” để lưu, chọn xóa trắng để xóa dữ liệu vừa nhập, focus tới mã.
- Khi chọn “xem danh sách Tác giả”, chương trình sẽ mở một màn hình mới để hiển thị danh sách Tác giả đã tạo – chú ý dùng CustomLayout cho ListView:

STT	Mã tác giả	Tên tác giả
1	tg01	Nguyen Nhat Anh
2	tg02	Hoang Trung Thong
3	tg03	Han Mac Tu
4	tg04	Ho Xuan Huong

- Khi chọn từng tác giả trong danh sách, sẽ hiển thị màn hình cho phép chỉnh sửa tác giả:



- Chọn Update để cập nhật lại Tác giả
- Khi nhấn Thật Lâu (Long time) vào từng tác giả trong danh sách, chương trình sẽ hiển thị Alert Dialog hỏi xem có muốn xóa Tác giả này hay không?

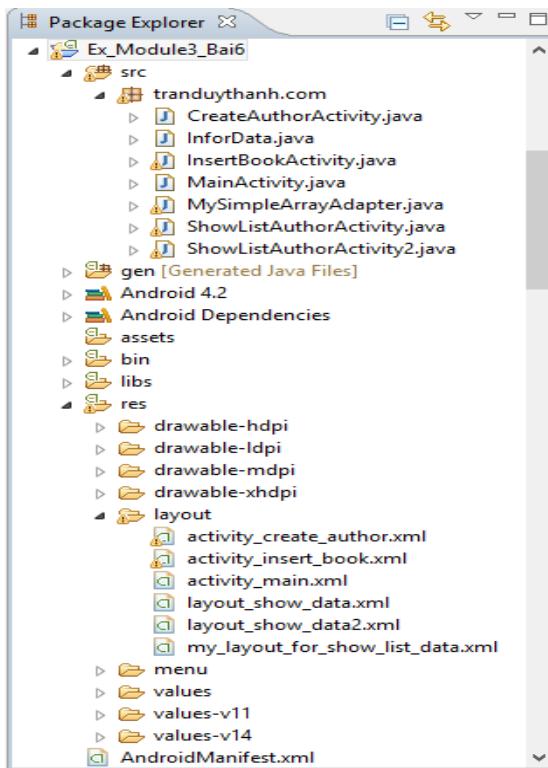


- Chọn Có để xóa tác giả hiện tại,
- Chọn Không để trở về màn hình xem danh sách
- Khi chọn chức năng “quản lý sách” ở màn hình chính chương trình sẽ hiển thị:



- Load danh sách Tác giả vào Spinner
 - Sử dụng DatePickerDialog để chọn ngày xuất bản
 - Chọn “Thêm sách” lưu vào CSDL đúng với tác giả chọn trong Spinner, đồng thời cập nhập vào ListView bên dưới.
-

Bạn xem cấu trúc thư mục của ứng dụng:



- Bài này rất phức tạp nên các bạn phải tập trung 12 thành công lực để nghiên cứu + ngồi thiền để tĩnh tâm làm bài nếu không sẽ bị Tàu Hỏa Nhập Ma.

- Bạn xem **activity_main.xml** layout – Giao diện chính của chương trình:

```

1 <LinearLayout
2   xmlns:android="http://schemas.android.com/apk/res/android"
3   xmlns:tools="http://schemas.android.com/tools"
4   android:id="@+id/LinearLayout1"
5   android:layout_width="match_parent"
6   android:layout_height="match_parent"
7   android:orientation="vertical" >
8
9   <TextView
10    android:id="@+id/textView1"
11    android:layout_width="match_parent"
12    android:layout_height="wrap_content"
13    android:background="#004000"
14    android:gravity="center"
15    android:text="Quản Lý Sách"
16    android:textColor="#FFFFFF"
17    android:textSize="20sp" />
```

```

18
19    <Button
20        android:id="@+id/btnInsertAuthor"
21        android:layout_width="match_parent"
22        android:layout_height="wrap_content"
23        android:text="Thêm Tác Giả" />
24
25    <Button
26        android:id="@+id/buttonShowAuthorList"
27        android:layout_width="match_parent"
28        android:layout_height="wrap_content"
29        android:text="Xem danh sách Tác Giả" />
30
31    <Button
32        android:id="@+id/buttonInsertBook"
33        android:layout_width="match_parent"
34        android:layout_height="wrap_content"
35        android:layout_marginTop="14dp"
36        android:text="Quản lý Sách" />
37
</LinearLayout>

```

và Source code xử lý **MainActivity.java**:

```

1 package tranduythanh.com;
2
3 import java.util.Locale;
4
5 import android.os.Bundle;
6 import android.app.Activity;
7 import android.content.ContentValues;
8 import android.content.Intent;
9 import android.database.Cursor;
10 import android.database.sqlite.SQLiteDatabase;
11 import android.view.Menu;
12 import android.view.View;
13 import android.view.View.OnClickListener;
14 import android.widget.Button;
15 import android.widget.Toast;
16 /**
17 * hàm hình chính cho phép chọn các thao tác
18 * @author drthanh
19 *
20 */
21 public class MainActivity extends Activity {
22
23     Button btnCreateDatabase=null;
24     Button btnInsertAuthor=null;
25     Button btnShowAuthorList=null;
26     Button btnShowAuthorList2=null;
27     Button btnTransaction=null;
28     Button btnShowDetail=null;
29     Button btnInsertBook=null;
30     public static final int OPEN_AUTHOR_DIALOG=1;

```

```

31 public static final int SEND_DATA_FROM_AUTHOR_ACTIVITY=2;
32 SQLiteDatabase database=null;
33 @Override
34 public void onCreate(Bundle savedInstanceState) {
35     super.onCreate(savedInstanceState);
36     setContentView(R.layout.activity_main);
37     btnInsertAuthor=(Button) findViewById(R.id.btnInsertAuthor);
38     btnInsertAuthor.setOnClickListener(new MyEvent());
39     btnShowAuthorList=(Button)
40     findViewById(R.id.buttonShowAuthorList);
41     btnShowAuthorList.setOnClickListener(new MyEvent());
42     btnInsertBook=(Button) findViewById(R.id.buttonInsertBook);
43     btnInsertBook.setOnClickListener(new MyEvent());
44     getDatabase();
45 }
46 /**
47 * hàm kiểm tra xem bảng có tồn tại trong CSDL hay chưa
48 * @param database - cơ sở dữ liệu
49 * @param tableName - tên bảng cần kiểm tra
50 * @return trả về true nếu tồn tại
51 */
52 public boolean isTableExists(SQLiteDatabase database, String
53 tableName) {
54     Cursor cursor = database.rawQuery("select DISTINCT tbl_name
55 from sqlite_master where tbl_name = '" +tableName+ "'", null);
56     if(cursor!=null) {
57         if(cursor.getCount()>0) {
58             cursor.close();
59             return true;
60         }
61         cursor.close();
62     }
63     return false;
64 }
65 /**
66 * hàm tạo CSDL và các bảng liên quan
67 * @return
68 */
69 public SQLiteDatabase getDatabase()
70 {
71     try
72     {
73         database=openOrCreateDatabase("mydata.db",
74 SQLiteDatabase.CREATE_IF_NECESSARY, null);
75         if(database!=null)
76         {
77             if(isTableExists(database,"tblAuthors"))
78                 return database;
79             database.setLocale(Locale.getDefault());
80             database.setVersion(1);
81             String sqlAuthor="create table tblAuthors (
82             id integer primary key autoincrement,
83             firstname text,
84             lastname text)";
85             database.execSQL(sqlAuthor);
86     }

```

```

85 String sqlBook="create table tblBooks (
86     +id integer primary key autoincrement,
87     +title text, "
88     +"dateadded date,"
89     +"authorid integer not null constraint authorid references
90     tblAuthors(id) on delete cascade)";
91 database.execSQL(sqlBook);
92 //Cách tạo trigger khi nhập dữ liệu sai ràng buộc quan hệ
93 String sqlTrigger="create trigger fk_insert_book before insert
on tblBooks "
94     +" for each row "
95     +" begin "
96     +" select raise(rollback,'them du lieu tren bang tblBooks bi
97 sai') "
98     +" where (select id from tblAuthors where id=new.authorid) is
99 null ;"
100    +" end; ";
101 database.execSQL(sqlTrigger);
102 Toast.makeText(MainActivity.this, "OK OK",
103 Toast.LENGTH_LONG).show();
104 }
105 }
106 catch(Exception e)
107 {
108     Toast.makeText(this, e.toString(), Toast.LENGTH_LONG).show();
109 }
110 return database;
111 }
112 public void createDatabaseAndTrigger()
113 {
114     if(database==null)
115     {
116         getDatabase();
117         Toast.makeText(MainActivity.this, "OK OK",
118         Toast.LENGTH_LONG).show();
119     }
120 }
121 /**
122 * hàm mở màn hình nhập Tác giả
123 */
124 public void showInsertAuthorDialog()
125 {
126     Intent intent=new Intent(MainActivity.this,
127 CreateAuthorActivity.class);
128     startActivityForResult(intent, OPEN_AUTHOR_DIALOG);
129 }
130 /**
131 * hàm xem danh sách tác giả dùng Activity
132 * Tôi làm 2 cách để các bạn ôn tập lại ListView
133 * bạn gọi hàm nào thì gọi 1 thôi showAuthorList1 hoặc
134 showAuthorList2
135 */
136 public void showAuthorList1()
137 {
138     Intent intent=new Intent(MainActivity.this,

```

```

139 ShowListAuthorActivity.class);
140     startActivity(intent);
141 }
142 /**
143 * hàm xem danh sách tác giả dùng ListActivity
144 * Tôi làm 2 cách để các bạn ôn tập lại ListView
145 * bạn gọi hàm nào thì gọi 1 thôi showAuthorList1 hoặc
146 showAuthorList2
147 */
148 public void showAuthorList2()
149 {
150 Intent intent=new Intent(MainActivity.this,
151 ShowListAuthorActivity2.class);
152     startActivity(intent);
153 }
154 /**
155 * Tôi cung cấp thêm hàm này để các bạn nghiên cứu thêm về
156 transaction
157 */
158 public void interactDBWithTransaction()
159 {
160 if(database!=null)
161 {
162 database.beginTransaction();
163 try
164 {
165 //làm cái gì đó tùm lum ở đây,
166 //chỉ cần có lỗi xảy ra thì sẽ kết thúc transaction
167 ContentValues values=new ContentValues();
168 values.put("firstname", "xx");
169 values.put("lastname", "yyy");
170 database.insert("tblAuthors", null, values);
171 database.delete("tblAuthors", "ma=?", new String[]{"x"});
172 //Khi nào hàm này được gọi thì các thao tác bên trên mới thực
173 hiện được
174 //Nếu nó không được gọi thì mọi thao tác bên trên đều bị hủy
175 database.setTransactionSuccessful();
176 }
177 catch(Exception ex)
178 {
179 Toast.makeText(MainActivity.this, ex.getMessage(),
180 Toast.LENGTH_LONG).show();
181 }
182 finally
183 {
184 database.endTransaction();
185 }
186 }
187 /**
188 * hàm xử lý kết quả trả về
189 */
190 @Override
191 protected void onActivityResult(int requestCode, int resultCode,
192 Intent data) {

```

```

193 // TODO Auto-generated method stub
194 super.onActivityResult(requestCode, resultCode, data);
195 if(resultCode==SEND_DATA_FROM_AUTHOR_ACTIVITY)
196 {
197 Bundle bundle= data.getBundleExtra("DATA_AUTHOR");
198 String firstname=bundle.getString("firstname");
199 String lastname=bundle.getString("lastname");
200 ContentValues content=new ContentValues();
201 content.put("firstname", firstname);
202 content.put("lastname", lastname);
203 if(database!=null)
204 {
205 long authorid=database.insert("tblAuthors", null, content);
206 if(authorid==-1)
207 {
208 Toast.makeText(MainActivity.this,authorid+" - "+firstname +"-"
209 "+lastname +" ==> insert error!", Toast.LENGTH_LONG).show();
210 }
211 else
212 {
213 Toast.makeText(MainActivity.this, authorid+" - "+firstname +"-"
214 "+lastname +" ==>insert OK!", Toast.LENGTH_LONG).show();
215 }
216 }
217 }
218 }
219 /**
220 * class xử lý sự kiện
221 * @author drthanh
222 *
223 */
224 private class MyEvent implements OnClickListener
225 {
226
227 @Override
228 public void onClick(View v) {
229 // TODO Auto-generated method stub
230 if(v.getId()==R.id.btnInsertAuthor)
231 {
232 showInsertAuthorDialog();
233 }
234 else if(v.getId()==R.id.buttonShowAuthorList)
235 {
236 showAuthorList1();
237 }
238
239 else if(v.getId()==R.id.buttonInsertBook)
240 {
241 Intent intent=new Intent(MainActivity.this,
242 InsertBookActivity.class);
243 startActivity(intent);
244 }
245 }

```

```

    }
    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        getMenuInflater().inflate(R.menu.activity_simple_database_main,
        menu);
        return true;
    }
}

```

- XML source thêm tác giả **activity_create_author.xml**:

```

1 <LinearLayout
2     xmlns:android="http://schemas.android.com/apk/res/android"
3     xmlns:tools="http://schemas.android.com/tools"
4     android:id="@+id/LinearLayout1"
5     android:layout_width="match_parent"
6     android:layout_height="match_parent"
7     android:orientation="vertical" >
8
9     <TableLayout
10        android:layout_width="match_parent"
11        android:layout_height="wrap_content" >
12
13     <TableRow
14         android:id="@+id/tableRow1"
15         android:layout_width="wrap_content"
16         android:layout_height="wrap_content" >
17
18     <TextView
19         android:id="@+id/textView1"
20         android:layout_width="wrap_content"
21         android:layout_height="wrap_content"
22         android:text="Mã tác giả:"
23         android:textAppearance="?android:attr/textAppearanceMedium" />
24     <EditText
25         android:id="@+id/editTextFirstName"
26         android:layout_width="wrap_content"
27         android:layout_height="wrap_content"
28         android:ems="10" >
29
30     <requestFocus />
31     </EditText>
32
33 </TableRow>
34
35     <TableRow
36         android:id="@+id/tableRow2"
37         android:layout_width="wrap_content"
38         android:layout_height="wrap_content" >
39
40     <TextView
41         android:id="@+id/textView2"
42         android:layout_width="wrap_content"
43         android:layout_height="wrap_content"

```

```

44     android:text="Tên tác giả:"
45     android:textAppearance="?android:attr/textAppearanceMedium" />
46     <EditText
47         android:id="@+id/editTextLastName"
48         android:layout_width="wrap_content"
49         android:layout_height="wrap_content"
50         android:ems="10" />
51
52     </TableRow>
53
54     <TableRow
55         android:id="@+id/tableRow3"
56         android:layout_width="wrap_content"
57         android:layout_height="wrap_content" >
58         <Button
59             android:id="@+id/buttonClear"
60             android:layout_width="wrap_content"
61             android:layout_height="wrap_content"
62             android:text="Xóa trắng" />
63
64         <Button
65             android:id="@+id/buttonInsert"
66             android:layout_width="wrap_content"
67             android:layout_height="wrap_content"
68             android:text="Lưu tác giả" />
69
70     </TableRow>
71
72     <TableRow
73         android:id="@+id/tableRow4"
74         android:layout_width="wrap_content"
75         android:layout_height="wrap_content" >
76         </TableRow>
77     </TableLayout>
78 </LinearLayout>

```

- Source code xử lý thêm tác giả – **CreateAuthorActivity.java**:

```

1  package tranduythanh.com;
2
3  import android.os.Bundle;
4  import android.app.Activity;
5  import android.content.Intent;
6  import android.view.Menu;
7  import android.view.View;
8  import android.view.View.OnClickListener;
9  import android.widget.Button;
10 import android.widget.EditText;
11 /**
12  * class nhập thông tin tác giả
13  * Mọi thay đổi đều gửi thông tin về MainActivity để xử lý
14  * @author drthanh
15 */

```

```

16     */
17     public class CreateAuthorActivity extends Activity {
18
19     @Override
20     public void onCreate(Bundle savedInstanceState) {
21         super.onCreate(savedInstanceState);
22         setContentView(R.layout.activity_create_author);
23         final Button btnInsert = (Button)
24             findViewById(R.id.buttonInsert);
25         final EditText txtFirstname=(EditText)
26             findViewById(R.id.editTextFirstName);
27         final EditText txtLastname=(EditText)
28             findViewById(R.id.editTextLastName);
29         final Intent intent= getIntent();
30         btnInsert.setOnClickListener(new OnClickListener() {
31
32             @Override
33             public void onClick(View v) {
34                 // TODO Auto-generated method stub
35                 Intent intent=new Intent();
36                 Bundle bundle=new Bundle();
37                 bundle.putString("firstname",
38                     txtFirstname.getText().toString());
39                 bundle.putString("lastname",
40                     txtLastname.getText().toString());
41                 intent.putExtra("DATA_AUTHOR", bundle);
42                 setResult(MainActivity.SEND_DATA_FROM_AUTHOR_ACTIVITY,
43                 intent);
44                 CreateAuthorActivity.this.finish();
45             }
46         });
47         final Button btnClear=(Button) findViewById(R.id.buttonClear);
48         btnClear.setOnClickListener(new OnClickListener() {
49
50             @Override
51             public void onClick(View v) {
52                 // TODO Auto-generated method stub
53                 txtFirstname.setText("");
54                 txtLastname.setText("");
55                 txtFirstname.requestFocus();
56             }
57         });
58
59         Bundle bundle= intent.getBundleExtra("DATA");
60         if(bundle!=null && bundle.getInt("KEY")==1)
61         {
62             String f2=bundle.getString("getField2");
63             String f3=bundle.getString("getField3");
64             txtFirstname.setText(f2);
65             txtLastname.setText(f3);
66             btnInsert.setText("Update");
67             this.setTitle("View Detail");
68             /*TableRow row=(TableRow) findViewById(R.id.tableRow3);
69             row.removeViewAt(0);
70             row.setGravity(Gravity.RIGHT);*/

```

```

70     }
71 }
72
73 @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        getMenuInflater().inflate(R.menu.activity_create_author,
menu);
        return true;
    }
}

```

- XML source thêm Sách – **activity_insert_book.xml**:

```

1 <TableLayout
2     xmlns:android="http://schemas.android.com/apk/res/android"
3     xmlns:tools="http://schemas.android.com/tools"
4     android:id="@+id/TableLayout1"
5     android:layout_width="match_parent"
6     android:layout_height="match_parent"
7     android:stretchColumns="*"
8     >
9
10 <TableRow
11     android:id="@+id/tableRow1"
12     android:layout_width="wrap_content"
13     android:layout_height="wrap_content" >
14
15 <TextView
16     android:id="@+id/textView1"
17     android:layout_width="wrap_content"
18     android:layout_height="wrap_content"
19     android:text="Tác giả:" />
20
21 <Spinner
22     android:id="@+id/spinner1"
23     android:layout_width="wrap_content"
24     android:layout_height="wrap_content"
25     android:layout_span="2" />
26
27 </TableRow>
28
29 <TableRow
30     android:id="@+id/tableRow2"
31     android:layout_width="wrap_content"
32     android:layout_height="wrap_content" >
33
34 <TextView
35     android:id="@+id/textView2"
36     android:layout_width="wrap_content"
37     android:layout_height="wrap_content"
38     android:text="Tên sách:" />
39
40 <EditText
41     android:id="@+id/editTextTitle"

```

```
42     android:layout_width="wrap_content"
43     android:layout_height="wrap_content"
44     android:layout_span="2"
45     android:ems="10" >
46
47     <requestFocus />
48   </EditText>
49
50 </TableRow>
51
52 <TableRow
53   android:id="@+id/tableRow3"
54   android:layout_width="wrap_content"
55   android:layout_height="wrap_content" >
56
57   <TextView
58     android:id="@+id/textView3"
59     android:layout_width="wrap_content"
60     android:layout_height="wrap_content"
61     android:text="Ngày XB:" />
62
63   <EditText
64     android:id="@+id/editTextDate"
65     android:layout_width="wrap_content"
66     android:layout_height="wrap_content"
67     android:ems="10"
68     android:inputType="date" />
69
70   <Button
71     android:id="@+id/buttonDate"
72     style="?android:attr/buttonStyleSmall"
73     android:layout_width="wrap_content"
74     android:layout_height="wrap_content"
75     android:text="..." />
76
77 </TableRow>
78
79 <TableRow
80   android:id="@+id/tableRow4"
81   android:layout_width="wrap_content"
82   android:layout_height="wrap_content" >
83
84   <Button
85     android:id="@+id/buttonInsertBook"
86     android:layout_width="wrap_content"
87     android:layout_height="wrap_content"
88     android:layout_span="3"
89     android:text="Thêm Sách" />
90
91 </TableRow>
92
93 <TableRow
94   android:id="@+id/tableRow5"
95   android:layout_width="wrap_content"
96   android:layout_height="wrap_content" >
```

```

96      </TableRow>
97
98      <ListView
99          android:id="@+id/listViewBook"
100         android:layout_width="match_parent"
101         android:layout_height="wrap_content" >
102     </ListView>
103
104 </TableLayout>

```

- Source code xử lý thêm sách – **InsertBookActivity.java**:

```

1  package tranduythanh.com;
2  import java.text.SimpleDateFormat;
3  import java.util.ArrayList;
4  import java.util.Calendar;
5  import java.util.List;
6  import android.os.Bundle;
7  import android.app.Activity;
8  import android.app.DatePickerDialog;
9  import android.app.DatePickerDialog.OnDateSetListener;
10 import android.app.Dialog;
11 import android.content.ContentValues;
12 import android.database.Cursor;
13 import android.database.sqlite.SQLiteDatabase;
14 import android.view.Menu;
15 import android.view.View;
16 import android.view.View.OnClickListener;
17 import android.widget.AdapterView.OnItemSelectedListener;
18 import android.widget.AdapterView;
19 import android.widget.Button;
20 import android.widget.DatePicker;
21 import android.widget.EditText;
22 import android.widget.ListView;
23 import android.widget.Spinner;
24 import android.widget.Toast;
25 /**
26  * class hiển thị thông tin Tác giả và Spinner
27  * và hiển thị thông tin sách vào ListView
28  * đồng thời cho phép thao tác với sách
29  * Class này là khó hiểu nhất, nhưng chỉ là tổng hợp của
30  * các kiến thức đã học trước đó
31  * @author drthanh
32  *
33  */
34 public class InsertBookActivity extends Activity {
35
36     SQLiteDatabase database=null;
37     List<InforData>listBook=null;
38     List<InforData>listAuthor=null;
39     InforData authorData=null;
40     MySimpleArrayAdapter adapter=null;
41     int day,month,year;
42     @Override

```

```

43     public void onCreate(Bundle savedInstanceState) {
44         super.onCreate(savedInstanceState);
45         setContentView(R.layout.activity_insert_book);
46         Spinner pinner=(Spinner) findViewById(R.id.spinner1);
47         listAuthor=new ArrayList<InforData>();
48         InforData d1=new InforData();
49         d1.setField1(" ");
50         d1.setField2("Show All");
51         d1.setField3(" ");
52         listAuthor.add(d1);
53         //Lệnh xử lý đưa dữ liệu là Tác giả và Spinner
54         database=openOrCreateDatabase("mydata.db",
55             SQLiteDatabase.CREATE_IF_NECESSARY, null);
56         if(database!=null)
57         {
58
59             Cursor cursor=database.query("tblAuthors", null, null, null,
60             null, null, null);
61             cursor.moveToFirst();
62             while(cursor.isAfterLast()==false)
63             {
64                 InforData d=new InforData();
65                 d.setField1(cursor.getInt(0));
66                 d.setField2(cursor.getString(1));
67                 d.setField3(cursor.getString(2));
68                 listAuthor.add(d);
69                 cursor.moveToNext();
70             }
71             cursor.close();
72         }
73         adapter=new MySimpleArrayAdapter(InsertBookActivity.this,
74             R.layout.my_layout_for_show_list_data,listAuthor);
75         adapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
76         pinner.setAdapter(adapter);
77         //Xử lý sự kiện chọn trong Spinner
78         //chọn tác giả nào thì hiển thị toàn bộ sách của tác giả đó
79         mà thôi
80         //Nếu chọn All thì hiển thị toàn bộ không phân hiệt tác giả
81         pinner.setOnItemSelectedListener(new
82         OnItemSelectedListener() {
83             @Override
84             public void onItemClick(AdapterView<?> arg0, View arg1,
85             int arg2, long arg3) {
86                 // TODO Auto-generated method stub
87                 if(arg2==0)
88                 {
89                     //Hiển thị mọi sách trong CSDL
90                     authorData=null;
91                     loadAllListBook();
92                 }
93                 else
94                 {
95                     //Hiển thị sách theo tác giả chọn trong Spinner
96                     authorData=listAuthor.get(arg2);

```

```

97     loadListBookByAuthor(authorData.getField1().toString());
98 }
99 }
100 @Override
101 public void onNothingSelected(AdapterView<?> arg0) {
102     // TODO Auto-generated method stub
103     authorData=null;
104 }
105 );
106
107 setCurrentDateOnView();
108 //lệnh xử lý DatePickerDialog
109 Button bChangeDate=(Button) findViewById(R.id.buttonDate);
110 bChangeDate.setOnClickListener(new OnClickListener() {
111
112     @Override
113     public void onClick(View v) {
114         // TODO Auto-generated method stub
115         showDialog(113);
116     }
117 });
118 //Lệnh xử lý thêm mới một sản phẩm theo tác giả đang chọn
119 Button btnInsertBook =(Button)
120 findViewById(R.id.buttonInsertBook);
121 btnInsertBook.setOnClickListener(new OnClickListener() {
122
123     @Override
124     public void onClick(View v) {
125         // TODO Auto-generated method stub
126         if(authorData==null)
127         {
128             Toast.makeText(InsertBookActivity.this, "Please choose an
129             author to insert", Toast.LENGTH_LONG).show();
130             return;
131         }
132         EditText txtTitle=(EditText)
133         findViewById(R.id.editTextTitle);
134         ContentValues values=new ContentValues();
135         values.put("title", txtTitle.getText().toString());
136         Calendar c=Calendar.getInstance();
137         c.set(year, month, day);
138         SimpleDateFormat dfmt=new SimpleDateFormat("dd-MM-yyyy");
139         values.put("dateadded", dfmt.format(c.getTime()));
140         values.put("authorid", authorData.getField1().toString());
141         long bId=database.insert("tblBooks", null, values);
142         if(bId>0)
143         {
144             Toast.makeText(InsertBookActivity.this, "Insert Book OK",
145             Toast.LENGTH_LONG).show();
146             loadListBookByAuthor(authorData.getField1().toString());
147         }
148     }
149     Toast.makeText(InsertBookActivity.this, "Insert Book
150 Failed", Toast.LENGTH_LONG).show();

```

```

151     }
152     }
153   });
154 }
155 /*
156 * Hàm hiển thị mọi sách trong CSDL
157 */
158 public void loadAllListBook()
159 {
160   Cursor cur=database.query("tblBooks", null, null, null,
161   null, null, null);
162   cur.moveToFirst();
163   listBook=new ArrayList<InforData>();
164   while(cur.isAfterLast()==false)
165   {
166     InforData d=new InforData();
167     d.setField1(cur.getInt(0));
168     d.setField2(cur.getString(1));
169     d.setField3(cur.getString(2));
170     listBook.add(d);
171     cur.moveToNext();
172   }
173   cur.close();
174   adapter=new MySimpleArrayAdapter(InsertBookActivity.this,
175   R.layout.my_layout_for_show_list_data, listBook);
176   ListView lv=(ListView) findViewById(R.id.listViewBook);
177   lv.setAdapter(adapter);
178 }
179 /**
180 * hàm hiển thị sách theo tác giả
181 * @param authorid
182 */
183 public void loadListBookByAuthor(String authorid)
184 {
185   Cursor cur=database.query("tblBooks", null, "authorid=?", 
186   new String[]{authorid}, null, null, null);
187   cur.moveToFirst();
188   listBook=new ArrayList<InforData>();
189   while(cur.isAfterLast()==false)
190   {
191     InforData d=new InforData();
192     d.setField1(cur.getInt(0));
193     d.setField2(cur.getString(1));
194     d.setField3(cur.getString(2));
195     listBook.add(d);
196     cur.moveToNext();
197   }
198   cur.close();
199   adapter=new MySimpleArrayAdapter(InsertBookActivity.this,
200   R.layout.my_layout_for_show_list_data, listBook);
201   ListView lv=(ListView) findViewById(R.id.listViewBook);
202   lv.setAdapter(adapter);
203 }
204 @Override
protected Dialog onCreateDialog(int id) {

```

```

205 // TODO Auto-generated method stub
206 if(id==113)
207 {
208     return new DatePickerDialog(this, dateChange, year, month,
209 day);
210 }
211 return null;
212 }
213 /**
214 * xử lý DatePickerDialog
215 */
216 private DatePickerDialog.OnDateSetListener dateChange= new
217 OnDateSetListener() {
218
219     @Override
220     public void onDateSet(DatePicker view, int year1, int
221 monthOfYear,
222     int dayOfMonth) {
223         // TODO Auto-generated method stub
224         year=year1;
225         month=monthOfYear;
226         day=dayOfMonth;
227         EditText eDate=(EditText) findViewById(R.id.editTextDate);
228         eDate.setText(day+"-"+(month+1)+"-"+year);
229     }
230 };
231 /**
232 * thiết lập ngày tháng năm hiện tại
233 */
234 public void setCurrentDateOnView()
235 {
236     EditText eDate=(EditText) findViewById(R.id.editTextDate);
237     Calendar cal=Calendar.getInstance();
238     day=cal.get(Calendar.DAY_OF_MONTH);
239     month=cal.get(Calendar.MONTH);
240     year=cal.get(Calendar.YEAR);
241     eDate.setText(day+"-"+(month+1)+"-"+year);
242 }
243     @Override
244     public boolean onCreateOptionsMenu(Menu menu) {
245         getMenuInflater().inflate(R.menu.activity_insert_book,
246 menu);
247         return true;
248     }
249 }

```

- XML Source cho custom layout:

```

1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout
3     xmlns:android="http://schemas.android.com/apk/res/android"
4     android:layout_width="match_parent"
5     android:layout_height="match_parent"
6     android:orientation="horizontal"
7     android:gravity="left"

```

```

7      >
8
9  <TextView
10     android:id="@+id/textView1"
11     android:layout_width="wrap_content"
12     android:layout_height="wrap_content"
13     android:layout_weight="0.2"
14     android:gravity="left"
15     android:text="Medium Text"
16     android:textAppearance="?android:attr/textAppearanceMedium" />
17
18 <TextView
19     android:id="@+id/textView2"
20     android:layout_width="wrap_content"
21     android:layout_height="wrap_content"
22     android:layout_gravity="left"
23     android:layout_weight="0.4"
24     android:gravity="left"
25     android:text="Medium Text"
26     android:textAppearance="?android:attr/textAppearanceMedium" />
27
28 &nbsp;
29
30 <TextView
31     android:id="@+id/textView3"
32     android:layout_width="wrap_content"
33     android:layout_height="wrap_content"
34     android:layout_gravity="left"
35     android:layout_weight="0.4"
36     android:text="Medium Text"
37     android:gravity="left"
38     android:textAppearance="?android:attr/textAppearanceMedium" />
39
40 </LinearLayout>

```

- Source code xử lý Custom layout – **InforData.java**:

```

1  package tranduythanh.com;
2 /**
3  * class dùng chung để đọc dữ liệu hiển thị lên ListView Sách
4  và tác giả
5  * bạn có thể bỏ class này viết class khác
6  * Ở đây Tôi hơi làm biếng 1 chút là Tôi muốn viết 1 Class có
7  các kiểu Object
8  * để nó tự hiểu mọi kiểu dữ liệu đỡ phải viết lại nên bạn đọc
9  có vẻ khó hiểu
10 * Nhưng thôi ---> ráng lên
11 * @author drthanh
12 *
13 */
14 public class InforData {
15     private Object field1;
16     private Object field2;
17     private Object field3;

```

```

18     public Object getField1() {
19         return field1;
20     }
21     public void setField1(Object field1) {
22         this.field1 = field1;
23     }
24     public Object getField2() {
25         return field2;
26     }
27     public void setField2(Object field2) {
28         this.field2 = field2;
29     }
30     public Object getField3() {
31         return field3;
32     }
33     public void setField3(Object field3) {
34         this.field3 = field3;
35     }
36     @Override
37     public String toString() {
38         // TODO Auto-generated method stub
39         return this.field1 + " - " +this.field2 + " - "+this.field3;
40     }
41 }
```

- Source code xử lý Custom layout - **MySimpleArrayAdapter.java**:

```

1     package tranduythanh.com;
2
3     import java.util.List;
4
5     import android.app.Activity;
6     import android.content.Context;
7     import android.graphics.Color;
8     import android.view.Gravity;
9     import android.view.LayoutInflater;
10    import android.view.View;
11    import android.view.ViewGroup;
12    import android.widget.ArrayAdapter;
13    import android.widget.TextView;
14    /**
15     * class dùng để custom layout
16     * dùng chung cho hiển thị Sách và Tác giả
17     * @author drthanh
18     *
19     */
20    public class MySimpleArrayAdapter extends
21    ArrayAdapter<InforData> {
22
23        private Activity context;
24        private int layout;
25        private List<InforData>list;
26        public MySimpleArrayAdapter(Context context, int
27        textViewResourceId,
```

```

28     List<InforData> objects) {
29     super(context, textViewResourceId, objects);
30     // TODO Auto-generated constructor stub
31     this.context=(Activity) context;
32     this.layout=textViewResourceId;
33     this.list=objects;
34   }
35   @Override
36   public View getView(int position, View convertView, ViewGroup
37   parent) {
38     // TODO Auto-generated method stub
39     LayoutInflater flater=context.getLayoutInflater();
40     View row=flater.inflate(layout, parent, false);
41     TextView txt1=(TextView) row.findViewById(R.id.textView1);
42     TextView txt2=(TextView) row.findViewById(R.id.textView2);
43     TextView txt3=(TextView) row.findViewById(R.id.textView3);
44     /*txt1.setTextAlignment(Gravity.LEFT);
45     txt2.setTextAlignment(Gravity.LEFT);
46     txt3.setTextAlignment(Gravity.LEFT);*/
47     InforData data=list.get(position);
48     txt1.setText(data.getField1()==null?"":data.getField1().toStri
49     ng());
50     txt2.setText(data.getField2()==null?"":data.getField2().toStri
51     ng());
52     txt3.setText(data.getField3()==null?"":data.getField3().toStri
53     ng());
54     if(position==0)
55     {
56       row.setBackgroundColor(Color.RED);
57     }
58     return row;
59   }
60 }
```

- XML source xem danh sách tác giả – **layout_show_data.xml**:

```

1  <?xml version="1.0" encoding="utf-8"?>
2  <LinearLayout
3  xmlns:android="http://schemas.android.com/apk/res/android"
4  android:layout_width="match_parent"
5  android:layout_height="match_parent"
6  android:orientation="vertical" >
7
8  <LinearLayout
9  android:layout_width="match_parent"
10 android:layout_height="wrap_content"
11 android:orientation="vertical" >
12
13 <ListView
14 android:id="@+id/listViewShowData"
15 android:layout_width="match_parent"
16 android:layout_height="405dp" >
17 </ListView>
18 </LinearLayout>
```

```

19    <Button
20        android:id="@+id/buttonBack"
21        android:layout_width="wrap_content"
22        android:layout_height="wrap_content"
23        android:text="Back" />
24
</LinearLayout>
```

- Source code xử lý xem danh sách tác giả:

```

1 package tranduythanh.com;
2
3 import java.util.ArrayList;
4 import java.util.List;
5 import android.app.Activity;
6 import android.app.AlertDialog;
7 import android.app.AlertDialog.Builder;
8 import android.content.ContentValues;
9 import android.content.DialogInterface;
10 import android.content.Intent;
11 import android.database.Cursor;
12 import android.database.sqlite.SQLiteDatabase;
13 import android.os.Bundle;
14 import android.view.View;
15 import android.view.View.OnClickListener;
16 import android.widget.AdapterView.OnItemClickListener;
17 import android.widget.AdapterView;
18 import android.widget.AdapterView.OnItemLongClickListener;
19 import android.widget.Button;
20 import android.widget.ListView;
21 import android.widget.Toast;
22 /**
 * class xem danh sách tác giả
 * @author drthanh
 *
 */
23 public class ShowListAuthorActivity extends Activity{
24     @Override
25     protected void onCreate(Bundle savedInstanceState) {
26         // TODO Auto-generated method stub
27         super.onCreate(savedInstanceState);
28         setContentView(R.layout.layout_show_data);
29         updateUI();
30         Button btn=(Button) findViewById(R.id.buttonBack);
31         btn.setOnClickListener(new OnClickListener() {
32
33             @Override
34             public void onClick(View v) {
35                 // TODO Auto-generated method stub
36                 ShowListAuthorActivity.this.finish();
37             }
38         });
39     }
40     List<InforData>list=new ArrayList<InforData>();
```

```

45     InforData dataClick=null;
46     SQLiteDatabase database=null;
47     MySimpleArrayAdapter adapter=null;
48     public void updateUI()
49     {
50         database=openOrCreateDatabase("mydata.db",
51             SQLiteDatabase.CREATE_IF_NECESSARY, null);
52         if(database!=null)
53         {
54
55             Cursor cursor=database.query("tblAuthors", null, null, null,
56             null, null, null);
57             startManagingCursor(cursor);
58             InforData header=new InforData();
59             header.setField1("STT");
60             header.setField2("Mã tác giả");
61             header.setField3("Tên tác giả");
62             list.add(header);
63             cursor.moveToFirst();
64             while(!cursor.isAfterLast())
65             {
66                 InforData data=new InforData();
67                 data.setField1(cursor.getInt(0));
68                 data.setField2(cursor.getString(1));
69                 data.setField3(cursor.getString(2));
70                 list.add(data);
71                 cursor.moveToNext();
72             }
73             cursor.close();
74             adapter=new
75             MySimpleArrayAdapter(ShowListAuthorActivity.this,
76             R.layout.my_layout_for_show_list_data, list);
77             final ListView lv= (ListView)
78             findViewById(R.id.listViewShowData);
79             lv.setAdapter(adapter);
80             lv.setOnItemClickListener(new OnItemClickListener() {
81                 @Override
82                 public void onItemClick(AdapterView<?> arg0, View arg1,
83                 int arg2, long arg3) {
84                     // TODO Auto-generated method stub
85                     Toast.makeText(ShowListAuthorActivity.this,"View -->" +
86                     list.get(arg2).toString(), Toast.LENGTH_LONG).show();
87                     Intent intent=new Intent(ShowListAuthorActivity.this,
88                     CreateAuthorActivity.class);
89                     Bundle bundle=new Bundle();
90                     bundle.putInt("KEY", 1);
91                     bundle.putString("getField1",
92                     list.get(arg2).getField1().toString());
93                     bundle.putString("getField2",
94                     list.get(arg2).getField2().toString());
95                     bundle.putString("getField3",
96                     list.get(arg2).getField3().toString());
97                     intent.putExtra("DATA", bundle);
98                     dataClick=list.get(arg2);
startActivityForResult(intent,

```

```

99     MainActivity.OPEN_AUTHOR_DIALOG);
100    }
101   });
102   lv.setOnItemLongClickListener(new OnItemLongClickListener()
103 {
104   @Override
105   public boolean onItemLongClick(AdapterView<?> arg0, View
106 arg1,
107   int arg2, long arg3) {
108   // TODO Auto-generated method stub
109   final InforData data=list.get(arg2);
110   final int pos=arg2;
111   Toast.makeText(ShowListAuthorActivity.this, "Edit--"
112 >"+data.toString(), Toast.LENGTH_LONG).show();
113   AlertDialog.Builder b=new
114   Builder(ShowListAuthorActivity.this);
115   b.setTitle("Remove");
116   b.setMessage("Xóa ["+data.getField2() +" - "
117 "+data.getField3() +"] हả?");
118   b.setPositiveButton("Có", new
119   DialogInterface.OnClickListener() {
120
121   @Override
122   public void onClick(DialogInterface dialog, int which) {
123   // TODO Auto-generated method stub
124   int n=database.delete("tblAuthors", "id=?", new
125   String[]{data.getField1().toString()});
126   if(n>0)
127   {
128   Toast.makeText(ShowListAuthorActivity.this, "Remove ok",
129   Toast.LENGTH_LONG).show();
130   list.remove(pos);
131   adapter.notifyDataSetChanged();
132   }
133   else
134   {
135   Toast.makeText(ShowListAuthorActivity.this, "Remove not ok",
136   Toast.LENGTH_LONG).show();
137   }
138   });
139   b.setNegativeButton("Không", new
140   DialogInterface.OnClickListener() {
141
142   @Override
143   public void onClick(DialogInterface dialog, int which) {
144   // TODO Auto-generated method stub
145   dialog.cancel();
146   }
147   });
148   b.show();
149   return false;
150   }
151   );
152 }

```

```

153     }
154     @Override
155     protected void onActivityResult(int requestCode, int
156     resultCode, Intent data) {
157         // TODO Auto-generated method stub
158         super.onActivityResult(requestCode, resultCode, data);
159         if(resultCode==MainActivity.SEND_DATA_FROM_AUTHOR_ACTIVITY)
160         {
161             Bundle bundle=data.getBundleExtra("DATA_AUTHOR");
162             String f2=bundle.getString("firstname");
163             String f3=bundle.getString("lastname");
164             String f1=dataClick.getField1().toString();
165             ContentValues values=new ContentValues();
166             values.put("firstname", f2);
167             values.put("lastname", f3);
168             if(database!=null)
169             {
170                 int n=database.update("tblAuthors", values, "id=?", new
171                 String[]{f1});
172                 if(n>0)
173                 {
174                     Toast.makeText(ShowListAuthorActivity.this, "update ok ok ok
175 ", Toast.LENGTH_LONG).show();
176                     dataClick.setField2(f2);
177                     dataClick.setField3(f3);
178                     if(adapter!=null)
179                         adapter.notifyDataSetChanged();
180                 }
181             }
182         }
183     }

```

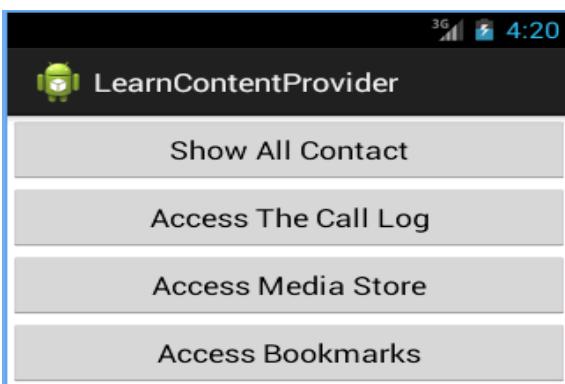
- Bạn có thể tải source code đầy đủ ở

đây: http://www.mediafire.com/download/nf4ghdbr44igpa6/Ex_Module3_Bai6.rar

- Bài tập kế tiếp Tôi sẽ hướng dẫn các bạn cách sử dụng Content Provider để đọc danh bạ trong điện thoại, cách đọc lịch sử cuộc gọi, cách đọc Media trong điện thoại cũng như bookmark. Bạn chú ý theo dõi

Bài tập 33: Sử dụng ContentProvider trong Android

- Bài tập này Tôi sẽ hướng dẫn các bạn cách sử dụng **ContentProvider**. Cụ thể là cách đọc danh bạ, cách đọc lịch sử cuộc gọi, cách đọc Media và bookmark.
- Phần này rất nhiều và phong phú, bạn cần khám phá nó trên mạng nhiều hơn.
- Thời gian không cho phép do đó Tôi chỉ hướng dẫn những tính năng mà ta thường xuyên sử dụng nhất.
- Tôi có giao diện chính sau:



- Ta có cú pháp tổng quát:

<standard_prefix>://<authority>/<data_path>/<id>

- Ví dụ để lấy tất cả các bookmark trong trình duyệt ta dùng cú pháp:

content://browser/bookmarks

- Để lấy toàn bộ danh bạ trong điện thoại ta dùng cú pháp:

content://contacts/people

- Để lấy 1 contact theo 1 định danh nào đó:

content://contacts/people/3

- Để lấy các kết quả trả về ta cũng dùng Cursor để quản lý.

- Có 2 cách sử dụng hàm lấy kết quả ở đây:

Cách 1:

CursorLoader loader=new CursorLoader(context, uri, null, null, null, null);

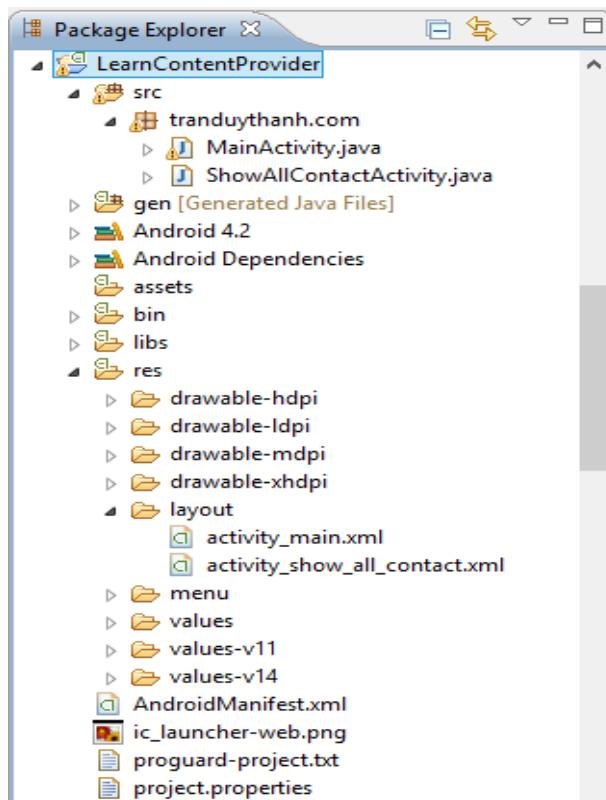
```
Cursor c=loader.loadInBackground();
```

cách 2:

```
Cursor c = getContentResolver().query(uri, null, null, null, null);
```

- Ta sẽ làm cụ thể từng chức năng trong ví dụ trên

- Bạn xem cấu trúc của bài tập này:



- Bạn xem XML Resource của màn hình chính (**activity_main.xml**):

```
1 <LinearLayout  
2     xmlns:android="http://schemas.android.com/apk/res/android"  
3     xmlns:tools="http://schemas.android.com/tools"  
4     android:id="@+id/LinearLayout1"  
5     android:layout_width="match_parent"  
6     android:layout_height="match_parent"  
7     android:orientation="vertical"  
8     tools:context=".MainActivity" >  
9  
10    <Button  
11        android:id="@+id/btnshowallcontact"  
12        android:layout_width="match_parent"  
13        android:layout_height="wrap_content"  
14        android:text="Show All Contact" />  
15  
16    <Button  
17        android:id="@+id/btnaccesscalllog"  
18        android:layout_width="match_parent"
```

```

19    android:layout_height="wrap_content"
20    android:text="Access The Call Log" />
21
22 <Button
23     android:id="@+id/btnmediastore"
24     android:layout_width="match_parent"
25     android:layout_height="wrap_content"
26     android:text="Access Media Store" />
27
28 <Button
29     android:id="@+id/btnaccessbookmarks"
30     android:layout_width="match_parent"
31     android:layout_height="wrap_content"
32     android:text="Access Bookmarks" />
33
34 </LinearLayout>

```

- Source code xử lý MainActivity.java:

```

1 package tranduythanh.com;
2
3 import android.os.Bundle;
4 import android.provider.Browser;
5 import android.provider.CallLog;
6 import android.provider.CallLog.Calls;
7 import android.provider.MediaStore;
8 import android.provider.MediaStore.Audio.Media;
9 import android.app.Activity;
10 import android.content.CursorLoader;
11 import android.content.Intent;
12 import android.database.Cursor;
13 import android.view.Menu;
14 import android.view.View;
15 import android.view.View.OnClickListener;
16 import android.widget.Button;
17 import android.widget.Toast;
18
19 public class MainActivity extends Activity implements
20 OnClickListener{
21
22     Button btnshowallcontact;
23     Button btnaccesscalllog;
24     Button btnaccessmediastore;
25     Button btnaccessbookmarks;
26     @Override
27     protected void onCreate(Bundle savedInstanceState) {
28         super.onCreate(savedInstanceState);
29         setContentView(R.layout.activity_main);
30         btnshowallcontact=(Button)
31         findViewById(R.id.btnshowallcontact);
32         btnshowallcontact.setOnClickListener(this);
33         btnaccesscalllog=(Button)
34         findViewById(R.id.btnaccesscalllog);
35         btnaccesscalllog.setOnClickListener(this);

```

```

36     btnaccessmediastore=(Button)
37     findViewById(R.id.btnmediastore);
38     btnaccessmediastore.setOnClickListener(this);
39     btnaccessbookmarks=(Button)
40     findViewById(R.id.btnaccessbookmarks);
41     btnaccessbookmarks.setOnClickListener(this);
42     }
43
44 @Override
45     public boolean onCreateOptionsMenu(Menu menu) {
46     // Inflate the menu; this adds items to the action bar if it
47     is present.
48     getMenuInflater().inflate(R.menu.activity_main, menu);
49     return true;
50     }
51
52 @Override
53     public void onClick(View v) {
54     Intent intent=null;
55     if(v==btnshowallcontact)
56     {
57     intent=new Intent(this, ShowAllContactActivity.class);
58     startActivity(intent);
59     }
60     else if(v==btncalllog)
61     {
62     accessTheCallLog();
63     }
64     else if(v==btnaccessmediastore)
65     {
66     accessMediaStore();
67     }
68     else if(v==btncalllog)
69     {
70     accessBookmarks();
71     }
72     }
73 /**
74 * hàm lấy danh sách lịch sử cuộc gọi
75 * với thời gian nhỏ hơn 30 giây và sắp xếp theo ngày gọi
76 */
77 public void accessTheCallLog()
78 {
79     String [] projection=new String[]{}
80     Calls.DATE,
81     Calls.NUMBER,
82     Calls.DURATION
83     };
84     Cursor c=getContentResolver().query(
85     CallLog.Calls.CONTENT_URI,
86     projection,
87     Calls.DURATION+"<?", new String[]{"30"},
88     Calls.DATE +" Asc");
89     c.moveToFirst();
90     String s="";

```

```

90     while(c.isAfterLast()==false) {
91         for(int i=0;i<c.getColumnCount();i++) {
92             s+=c.getString(i)+" - ";
93         }
94         s+="\n";
95         c.moveToNext();
96     }
97     c.close();
98     Toast.makeText(this, s, Toast.LENGTH_LONG).show();
99 }
100 /**
101 * hàm đọc danh sách các Media trong SD CARD
102 */
103 public void accessMediaStore()
104 {
105     String []projection={
106         MediaStore.MediaColumns.DISPLAY_NAME,
107         MediaStore.MediaColumns.DATE_ADDED,
108         MediaStore.MediaColumns.MIME_TYPE
109     };
110     CursorLoader loader=new CursorLoader
111         (this, Media.EXTERNAL_CONTENT_URI,
112         projection, null, null, null);
113     Cursor c=loader.loadInBackground();
114     c.moveToFirst();
115     String s="";
116     while(!c.isAfterLast()){
117         for(int i=0;i<c.getColumnCount();i++) {
118             s+=c.getString(i)+" - ";
119         }
120         s+="\n";
121         c.moveToNext();
122     }
123     Toast.makeText(this, s, Toast.LENGTH_LONG).show();
124     c.close();
125 }
126 /**
127 * hàm đọc danh sách Bookmark trong trình duyệt
128 */
129 public void accessBookmarks()
130 {
131     String []projection={
132         Browser.BookmarkColumns.TITLE,
133         Browser.BookmarkColumns.URL,
134     };
135     Cursor c=getContentResolver()
136         .query(Browser.BOOKMARKS_URI, projection,
137         null, null, null);
138     c.moveToFirst();
139     String s="";
140     int titleIndex=c.getColumnIndex
141         (Browser.BookmarkColumns.TITLE);
142     int urlIndex=c.getColumnIndex
143         (Browser.BookmarkColumns.URL);
144     while(!c.isAfterLast())

```

```

144    {
145        s+=c.getString(titleIndex)+" - "+
146        c.getString(urlIndex);
147        c.moveToNext();
148    }
149    c.close();
150    Toast.makeText(this, s, Toast.LENGTH_LONG).show();
151 }

```

- Source XML xử lý xem danh bạ (**activity_show_all_contact.xml**):

```

1 <LinearLayout
2     xmlns:android="http://schemas.android.com/apk/res/android"
3     xmlns:tools="http://schemas.android.com/tools"
4     android:id="@+id/LinearLayout1"
5     android:layout_width="match_parent"
6     android:layout_height="match_parent"
7     android:orientation="vertical"
8     tools:context=".ShowAllContactActivity" >
9
10 <ListView
11     android:id="@+id/listView1"
12     android:layout_width="match_parent"
13     android:layout_height="wrap_content" >
14     </ListView>
15
16     <Button
17         android:id="@+id/btnback"
18         android:layout_width="wrap_content"
19         android:layout_height="wrap_content"
20         android:text="Back" />
21 </LinearLayout>

```

- Source code xử lý xem danh bạ (**ShowAllContactActivity.java**):

- Tôi viết theo 2 cách : Dùng **CursorLoader** và **getContentResolver**

```

1 package tranduythanh.com;
2 import java.util.ArrayList;
3
4 import android.net.Uri;
5 import android.os.Bundle;
6 import android.provider.ContactsContract;
7 import android.app.Activity;
8 import android.content.CursorLoader;
9 import android.database.Cursor;
10 import android.view.Menu;
11 import android.view.View;
12 import android.view.View.OnClickListener;
13 import android.widget.ArrayAdapter;
14 import android.widget.Button;

```

```

15 import android.widget.ListView;
16
17 public class ShowAllContactActivity extends Activity {
18     Button btnback;
19     @Override
20     protected void onCreate(Bundle savedInstanceState) {
21         super.onCreate(savedInstanceState);
22         setContentView(R.layout.activity_show_all_contact);
23         btnback=(Button) findViewById(R.id.btnback);
24         btnback.setOnClickListener(new OnClickListener() {
25
26             @Override
27             public void onClick(View v) {
28                 // TODO Auto-generated method stub
29                 finish();
30             }
31         });
32         showAllContacts1();
33     }
34     /**
35      * hàm danh toàn bộ danh bạ
36      * dùng CursorLoader
37      */
38     public void showAllContacts1()
39     {
40         Uri uri=Uri.parse("content://contacts/people");
41         ArrayList<String> list=new ArrayList<String>();
42         CursorLoader loader=new
43         CursorLoader(this, uri, null, null, null, null);
44         Cursor c1=loader.loadInBackground();
45         c1.moveToFirst();
46         while(c1.isAfterLast()==false){
47             String s="";
48             String idColumnName=ContactsContract.Contacts._ID;
49             int idIndex=c1.getColumnIndex(idColumnName);
50             s+=c1.getString(idIndex)+" - ";
51             String
52             nameColumnName=ContactsContract.Contacts.DISPLAY_NAME;
53             int nameIndex=c1.getColumnIndex(nameColumnName);
54             s+=c1.getString(nameIndex);
55             c1.moveToNext();
56             list.add(s);
57         }
58         c1.close();
59         ListView lv=(ListView) findViewById(R.id.listView1);
60         ArrayAdapter<String>adapter=new ArrayAdapter<String>(this,
61             android.R.layout.simple_list_item_1, list);
62         lv.setAdapter(adapter);
63     }
64     /**
65      * hàm danh toàn bộ danh bạ
66      * dùng getContentResolver
67      */
68     public void showAllContacts2()
{

```

```

69 Uri uri=Uri.parse("content://contacts/people");
70 ArrayList<String> list=new ArrayList<String>();
71 Cursor c1=getContentResolver()
72 .query(uri, null, null, null, null);
73 c1.moveToFirst();
74 while(c1.isAfterLast()==false)
75 {
76     String s="";
77     String idColumnName=ContactsContract.Contacts._ID;
78     int idIndex=c1.getColumnIndex(idColumnName);
79     s=c1.getString(idIndex)+" - ";
80     String
81     nameColumnName=ContactsContract.Contacts.DISPLAY_NAME;
82     int nameIndex=c1.getColumnIndex(nameColumnName);
83     s+=c1.getString(nameIndex);
84     c1.moveToNext();
85     list.add(s);
86 }
87 c1.close();
88 ListView lv=(ListView) findViewById(R.id.listView1);
89 ArrayAdapter<String>adapter=new ArrayAdapter<String>(this,
90 android.R.layout.simple_list_item_1, list);
91 lv.setAdapter(adapter);
92 }
93 @Override
94 public boolean onCreateOptionsMenu(Menu menu) {
95     // Inflate the menu; this adds items to the action bar if
96     it is present.
97     getMenuInflater().inflate(R.menu.activity_show_all_contact,
98     menu);
99     return true;
}

```

}

- Lưu ý cấp quyền cho ứng dụng:

AndroidManifest.xml:

```

1 <?xml version="1.0" encoding="utf-8"?>
2 <manifest
3     xmlns:android="http://schemas.android.com/apk/res/android"
4         package="tranduythanh.com"
5         android:versionCode="1"
6         android:versionName="1.0" >
7
8     <uses-sdk
9         android:minSdkVersion="14"
10        android:targetSdkVersion="17" />
11
12     <uses-permission android:name="android.permission.READ_CONTACTS"
13     />
14     <uses-permission
15        android:name="android.permission.READ_CALL_LOG" />

```

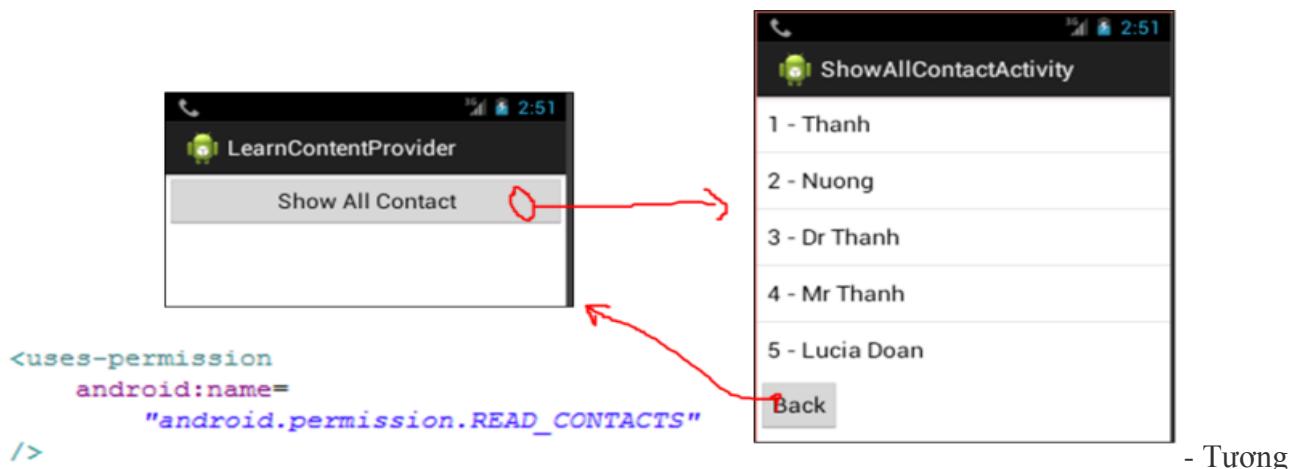
```

16    <uses-permission
17        android:name="android.permission.READ_EXTERNAL_STORAGE" />
18    <uses-permission
19        android:name="com.android.browser.permission.READ_HISTORY_BOOKMARKS" />
20
21
22    <application
23        android:allowBackup="true"
24        android:icon="@drawable/ic_launcher"
25        android:label="@string/app_name"
26        android:theme="@style/AppTheme" >
27        <activity
28            android:name="tranduythanh.com.MainActivity"
29            android:label="@string/app_name" >
30            <intent-filter>
31                <action android:name="android.intent.action.MAIN" />
32
33            <category android:name="android.intent.category.LAUNCHER" />
34            </intent-filter>
35        </activity>
36        <activity
37            android:name="tranduythanh.com.ShowAllContactActivity"
38            android:label="@string/title_activity_show_all_contact" >
39        </activity>
40    </application>
41
42    </manifest>

```

- Dòng 11, 12, 13, 14 là cấp quyền cho ứng dụng có thể truy suất các chức năng mong muốn.

- Bây giờ bạn thực hiện ứng dụng và quan sát, giờ Tôi chạy lên và dùng chức năng xem toàn bộ danh bạ:



tự như vậy bản thử các chức năng còn lại trong ứng dụng.

- Bạn có thể tải coding mẫu đầy đủ ở
đây: <http://www.mediafire.com/download/kaac7d36ocvr8ba/LearnContentProvider.rar>

- Bài tập kế tiếp các bạn sẽ được học về một kỹ thuật hoàn toàn mới trong Android và rất khó và rất hữu dụng, đó là kỹ thuật xử lý **Đa Tiến Trình** trong Android. Các bạn phải chú ý theo dõi vì nó vô cùng quan trọng, nó cũng giống như Intent là huyết mạch trong các ứng dụng của Android.

Bài tập 34: đa tiến trình trong Android (Multi-Threading)

- Chúc mừng các bạn đã vượt qua được 33 cửa ải bài tập Android.

- Tôi thật sự vui mừng xúc động khi các bạn đã bền bỉ theo dõi từng bài, để tới được đây thì Tôi cam đoan rằng các bạn đã phải tốn rất nhiều công sức và thời gian, các bạn phải thức ngày đêm để làm được những bài tập trước đó.

- Để đáp lại những cố gắng của các bạn, trong bài tập này Tôi sẽ cung cấp cho các bạn một kiến thức hay, kiến thức mới và cực kỳ khó để các bạn quen với nỗi khổ đau đẻ về sau có bị khổ nữa thì cũng quen rồi sẽ không còn thấy khổ (Tôi viết theo phạm trù triết học, chỉ có lập trình quá khổ mới hiểu).

- Như Tôi đã từng nói **Intent** là linh hồn của Android, trong mọi ngõ nghách hẻm hay mặt tiền của Android thì **Intent** vẫn tồn tại như chưa từng được tồn tại.

- Còn đa tiến trình (Multi – Threading)? nó cũng vậy, nó cũng giống như kỹ thuật truyền huyết mạch của từng ứng dụng Android, đặc biệt là những ứng dụng Vô Đối. Ví dụ như bạn cần cập nhật giao diện lúc thời gian thực, bạn cần kết nối internet hay làm những giao tác nào đó mà phải phân ra nhiều tiểu trình để chạy. Để xử lý được đa tiến trình thì bạn phải có một tư duy lập trình logic thật tốt, nếu không tốt thì phải (Lấy Cần Cù Bù Thông Minh).

- Kỹ thuật đa tiến trình rất khó mà không khó (nếu bạn hiểu).

- Trong bài này Tôi sẽ hướng dẫn các bạn xử lý đa tiến trình với **Handler** class và **AsyncTask** class.

Tài liệu tham khảo: <http://duythanhcse.wordpress.com>