

Hướng Dẫn Thực Hành

Lập Trình Cho Thiết Bị Di Động

(Android OS)

Khối: Đại Học

Năm 2014



Hướng dẫn:

- Bài tập thực hành được chia theo Module
- Mỗi Module được thiết kế cho thời lượng là 3→12 tiết thực hành tại lớp với sự hướng dẫn của giảng viên.
- Tùy theo số tiết phân bổ, mỗi tuần học có thể thực hiện nhiều Module.
- Sinh viên phải làm tất cả các bài tập trong các Module ở tuần tương ứng. Những sinh viên chưa hoàn tất phần bài tập tại lớp có trách nhiệm tự làm tiếp tục ở nhà.
- Các bài có dấu (*) là các bài tập nâng cao dành cho sinh viên khá giỏi.



MỤC LỤC

Module 1: Giới thiệu môi trường phát triển điện thoại di động.....	3
Module 2: Xử lý giao diện người dùng	16
Module 3: Xử lý tập tin, lưu trạng thái, CSDL, Content Provider	45
Module 4: Xử lý đa tiến trình và dịch vụ.....	56
Module 5: Networking APIs và Multimedia APIs	70
Module 6: Telephony APIs và Location Base Service APIs	78
Module 7: Thao tác với thiết bị cảm ứng.....	82
Module 8: Danh sách đề tài môn học.....	85



Module 1: Giới thiệu môi trường phát triển điện thoại di động

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

Bài tập 1:

Mục đích:

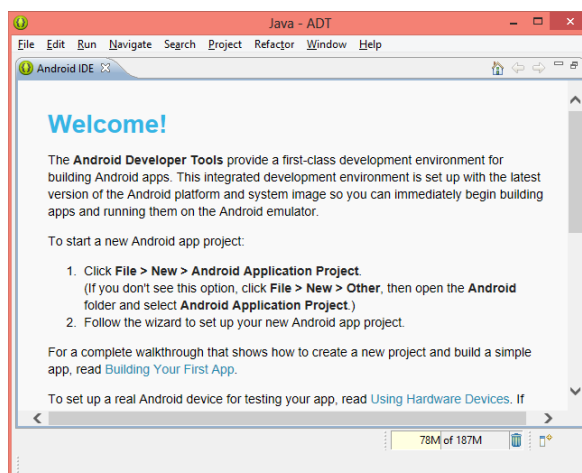
- Cài đặt được Java JDK, Android SDK, ADT plugin for Eclipse, update Android SDK.

Yêu cầu:

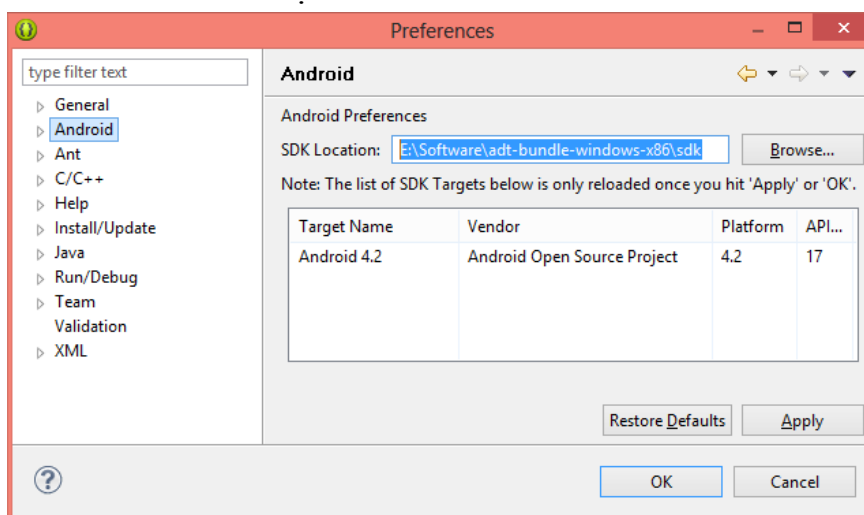
- Tải và cài đặt Java JDK: <http://java.sun.com/javase/downloads/index.jsp>
- Tải và cài đặt Android SDK: <http://developer.android.com/sdk/index.html> , tải gói ADT Bundle for Windows, gói này sẽ chứa các thành phần:
 - o Eclipse + ADT plugin
 - o Android SDK Tools
 - o Android Platform tools
 - o Android API mới nhất
 - o 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.
- Cập nhật Android SDK
- Tham chiếu Android SDK

Hướng dẫn:

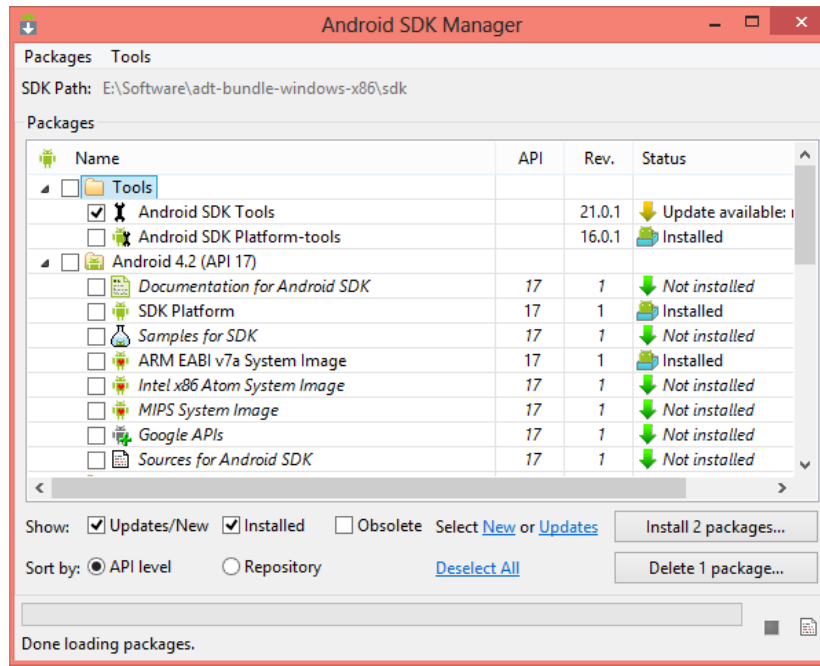
- Sau khi cấu hình thành công và khởi động Eclipse thì ta sẽ có màn hình như dưới đây:



- Đóng màn hình Welcome bằng cách click vào dấu chéo kế bên chữ **Android IDE**
- Để tham chiếu tới Android SDK, ta vào menu Windows/ Preferences/ chọn Android vào Browse tới SDK location mà bạn lưu trữ:



- Để cập nhật Android SDK, ta vào menu Windows/ Android SDK Manager (hoặc bấm vào biểu tượng trên thanh toolbar), chọn các gói cần cập nhật và bấm Install packages...:



Bài tập 2:

Mục đích:

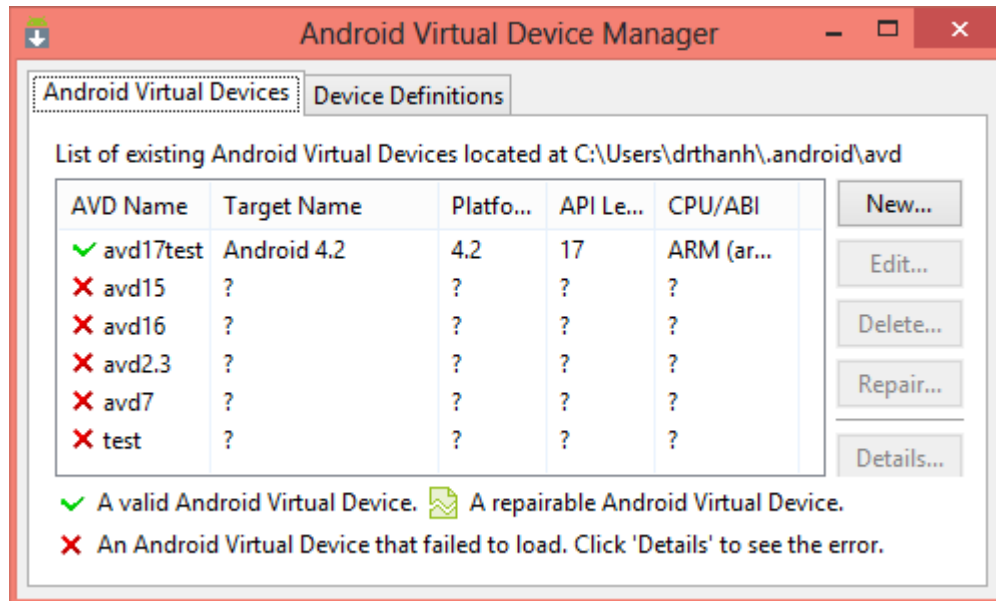
- Tạo và thao tác được với Android Virtual Device
- Sử dụng được DDMS
- Cấu hình thiết bị để debug

Yêu cầu:

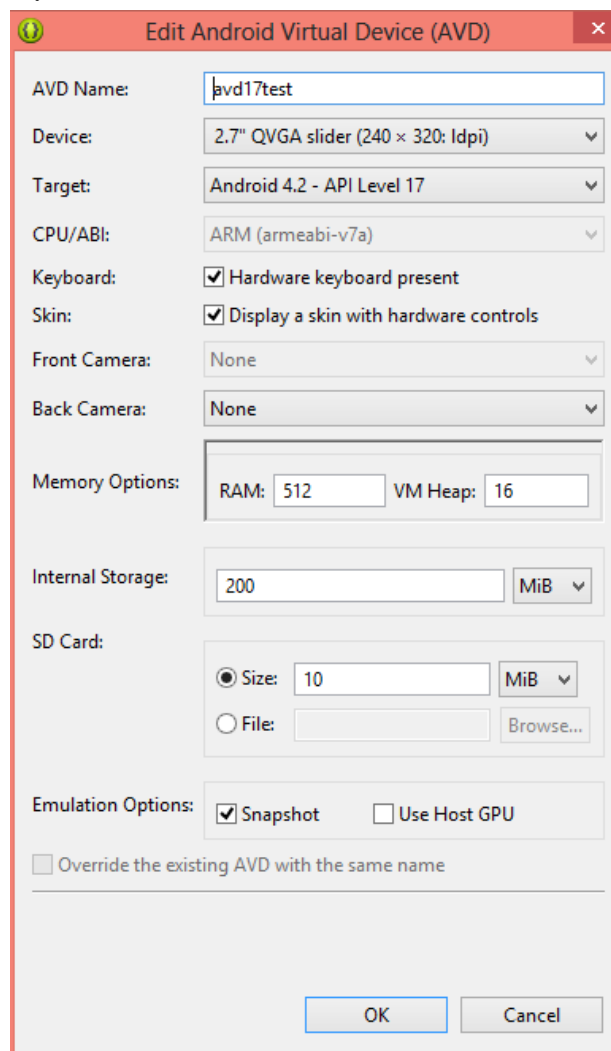
- Tạo Android Virtual Device:
 - o Đặt tên cho thiết bị
 - o Chọn loại thiết bị
 - o Chọn Target API
 - o Cấu hình Keyboard, Skin, Camera
 - o Thiết lập internal storage
 - o Thiết lập SD Card
 - o Chức năng Snapshot
- Khởi động AVD vừa mới tạo, tiến hành thực hiện: quan sát và thử nghiệm các chức năng bên trong AVD, thiết lập USB Debugging
- Thao tác trên DDMS:
 - o Kiểm tra cấu trúc thư mục, kéo thả một số tập tin vào AVD và quan sát hiện tượng
 - o Thực hiện unmount và Mount SD Card
 - o Thực hiện chức năng Gọi điện và gửi tin nhắn trong Emulator Control

Hướng dẫn:

- Từ menu Windows/ chọn Android Virtual Device Manager:



- Tại màn hình này chọn nút New



- Chọn các thông số kỹ thuật cần thiết và bấm OK
- Muốn khởi động AVD nào thì click chọn AVD đó và bấm nút Start, quan sát:



Bài tập 3:

Mục đích:

- Dalvik Virtual machine
- Hiểu được các thành phần bên trong ứng dụng Android:
 - o Cách tạo Android Project
 - o Activity, Intent, View
 - o Auto gen
 - o Android libs
 - o Resource
 - o Layout, menu, values
 - o Manifest XML

Yêu cầu:

- Tạo một Android Project, ứng dụng đơn giản chỉ xuất hiện dòng chữ “Hello World!”
- Kiểm tra tất cả cấu trúc thư mục của ứng dụng, cho nhận xét
- Chạy ứng dụng này trên Emulator/ trên điện thoại thật có hệ điều hành Android.
- Giải thích cơ chế mà Android Project được thực hiện trên thiết bị như thế nào: cách thức biên dịch, cách thức đóng gói, cách thức upload cài đặt vào thiết bị
- Tiến hành xóa thư mục Auto gen (gen), quan sát hiện tượng
- Vào menu Project/ chọn Clean, chọn Build automatically, quan sát hiện tượng
- Bấm chuột phải vào Project/ chọn Build Path/ configure build Path, quan sát màn hình
- Bấm chuột phải vào Project/ chọn Android Tools/ quan sát chức năng Fix project Properties và các chức năng khác



- Tiến hành Uninstall ứng dụng ra khỏi máy

Hướng dẫn:

- Để tạo một Android Project: Vào menu File/ chọn new/ chọn Android Application Project
- Hoặc trong Package Explorer/ bấm chuột phải/ chọn new / chọn Android Application Project
- Trong một số trường hợp không thấy Package Explorer (màn hình chính để lập trình), ta vào menu Windows/ chọn Show View/ chọn Package Explorer
- Có thể hiển thị Logcat trong Windows/ show view/ logcat để quan sát thông tin quá trình ứng dụng thực thi

Bài tập 4:

Mục đích:

- Tìm hiểu vòng đời của ứng dụng Android: các sự kiện xảy ra theo trình tự như thế nào

Yêu cầu:

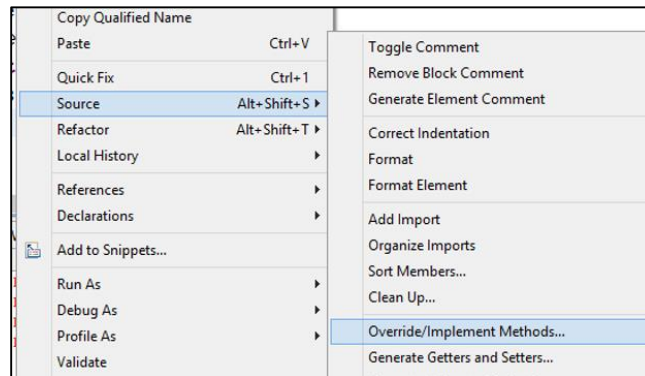
- Tạo một Android Project, đặt tên là **LearnAndroidLifetime**
- Trong **MainActivity**, tiến hành Override các hàm onStart, onResume, onPause, onStop, onDestroy, onRestart; trong mỗi hàm này đơn giản chỉ sử dụng Toast để hiển thị tên hàm được thực thi.
- Hãy tìm cách để cho các hàm Override ở trên xảy ra, cho nhận xét.

Hướng dẫn:

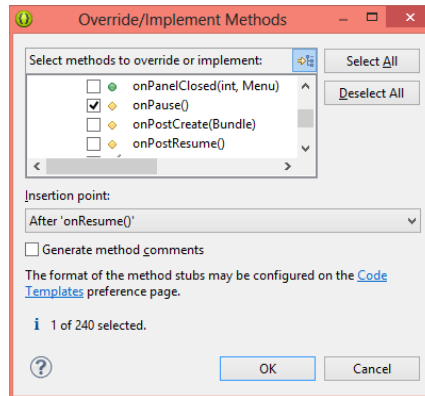
- Cách sử dụng Toast:

```
@Override
protected void onResume() {
    super.onResume();
    Toast.makeText
        (this, "onResume", Toast.LENGTH_LONG)
        .show();
}
```

- Để chương trình tự thêm các Override method, ta làm như sau: Bấm chuột phải vào màn hình soạn thảo/ chọn Source/ chọn Override Implement methods:



- Khi đó cửa sổ Implement sẽ hiển thị lên, ta chọn các hàm mong muốn rồi bấm nút Ok



Bài tập 5:

Mục đích:

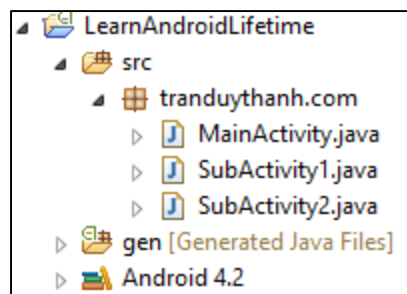
- Hiểu được Activity Stack
- Hiểu sâu hơn về vòng đời của ứng dụng
- Hiểu được Foreground lifetime và Visible lifetime

Yêu cầu:

- Hiệu chỉnh **LearnAndroidLifetime** Project trong bài 4
- Trong Project này tạo thêm 2 Activity lần lượt tên là **SubActivity1** và **SubActivity2** (như vậy Project sẽ có tổng cộng 3 Activity bao gồm 2 cái vừa tạo và 1 **MainActivity**)
- Chú ý rằng cấu hình sao cho **SubActivity1** sẽ 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**.
- Trong **MainActivity** tạo 2 **Button** để cho phép hiển thị **SubActivity1**, và **SubActivity2**. Trong mỗi trường hợp hãy quan sát sự kiện xảy ra trong **MainActivity** và cho nhận xét (chú ý là các sự kiện chắc chắn sẽ xảy ra khác nhau trong 2 trường hợp này).
 - o Foreground Lifetime xảy ra từ onResume() → onPause()
 - o Visible Lifetime xảy ra từ onStart() → onStop()

Hướng dẫn:

- Project sẽ như hình sau:



- Cấu hình để **SubActivity1** hiển thị dưới dạng Dialog: Trong Manifest XML, thêm android:theme= "@android:style/Theme.Dialog". Ta có thể dùng GUI trong Eclipse để thay đổi thông số cho dễ dàng.

```
<activity
    android:name="tranduythanh.com.SubActivity1"
    android:label="@string/title_activity_sub_activity1"
    android:theme="@android:style/Theme.Dialog">
</activity>
```



- Cách mở một Activity trong Android: Ở đây từ MainActivity ta mở SubActivity1 và SubActivity2 khi người sử dụng chọn Button1, Button2

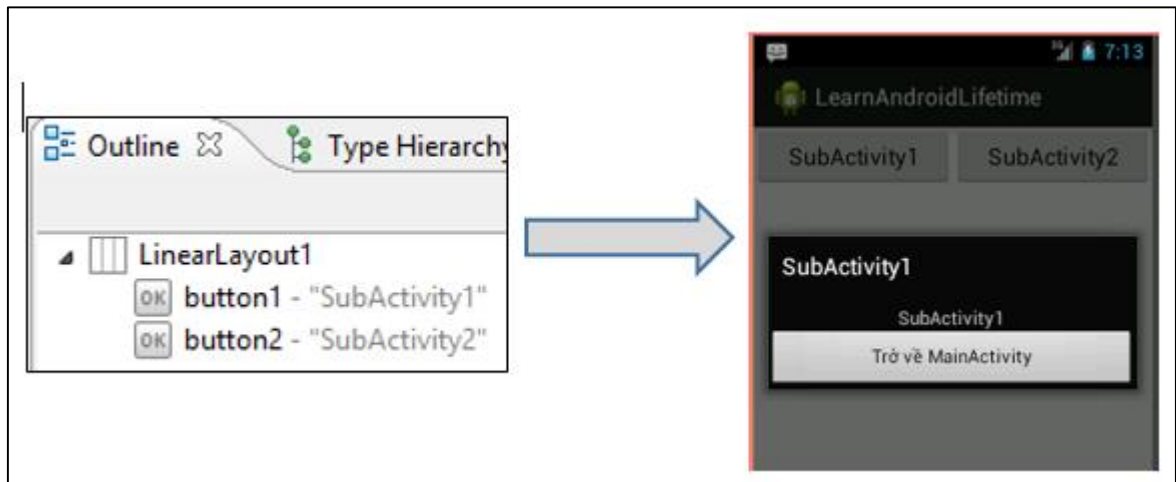


```
public class MainActivity extends Activity {

    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.



Bài tập 6:

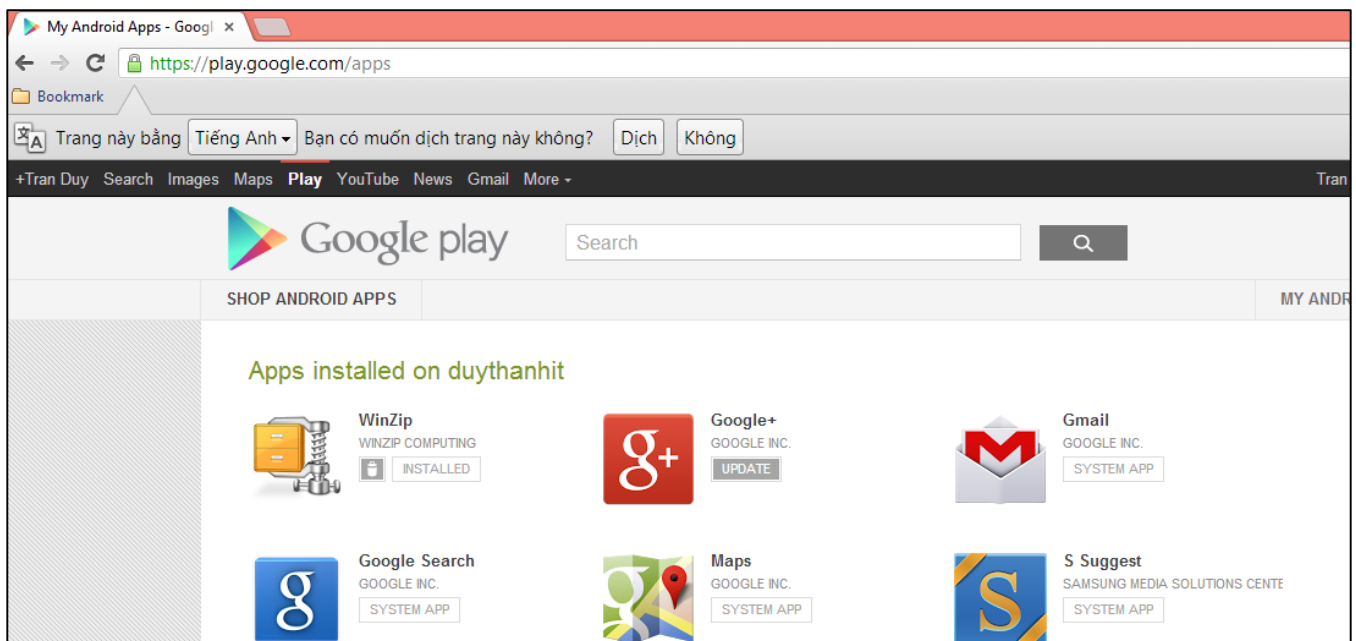
Mục đích:

- Tìm hiểu Google Play

Yêu cầu:

- Vào trong <https://play.google.com/store> tiến hành đăng ký và upload .apk (có đóng phí, hiện tại là 25USD)
- Xem hướng dẫn cụ thể : <http://support.google.com/googleplay/android-developer/answer/113469?hl=en>

Hướng dẫn:



Bài tập này sinh viên nào có nhu cầu thì làm. Khi có ứng dụng tốt muốn chia sẻ với cộng đồng thì nên làm bài tập này.



Bài tập 7 (*):

Mục đích:

- Biết được cách Upgrade Firmware cho một số thiết bị thật sử dụng hệ điều hành Android.

Yêu cầu:

- Tìm hiểu Odin mode
- Một số trang hữu ích : <http://androidmuscle.com/how-to-update-your-galaxy-s2-i9100-to-jelly-bean-4-1-2-xxlsj-firmware-tutorial/>
- <http://www.theandroidsoul.com/?s=Odin>

Hướng dẫn:

- Phải kiểm tra kỹ thông số kỹ thuật của điện thoại xem có tương thích hay không sau đó mới tiến hành Upgrade Firmware
- Nếu không có chắc chắn thì đừng có thực hiện (nhất là đối với những điện thoại mắc tiền)
- Đôi khi các sinh viên phải mạo hiểm với thiết bị của mình để khám phá (khuyến khích)



Module 2: Xử lý giao diện người dùng

Nội dung kiến thức thực hành:

- Thực hành về XML Layout – XML Container
- Các kiểu lập trình sự kiện trong Android
- Thực hành về Toast & Alert Dialog
- Thực hành các control cơ bản: View, TextView, EditText, Button, RadioButton,...
- Thực hành các control nâng cao: ListView, GridView, Spinner, Autocomplete, Menu...
- Cách tạo Custom Layout
- Thực hành về Webkit, Javascript
- Thực hành về Intent: cách mở nhiều màn hình (Activity) trong Android
- Cách lập trình chạm đơn điểm và đa điểm trong Android
- Cách cấu hình đa ngôn ngữ trong Android

Bài tập 1:

Mục đích:

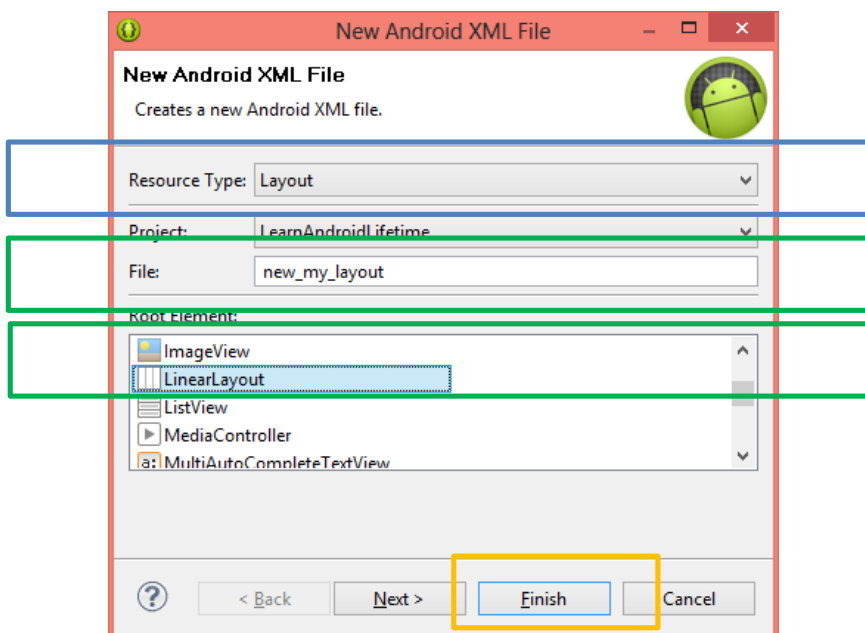
- Tạo và hiệu chỉnh XML Layout
- Sử dụng XML Layout

Yêu cầu:

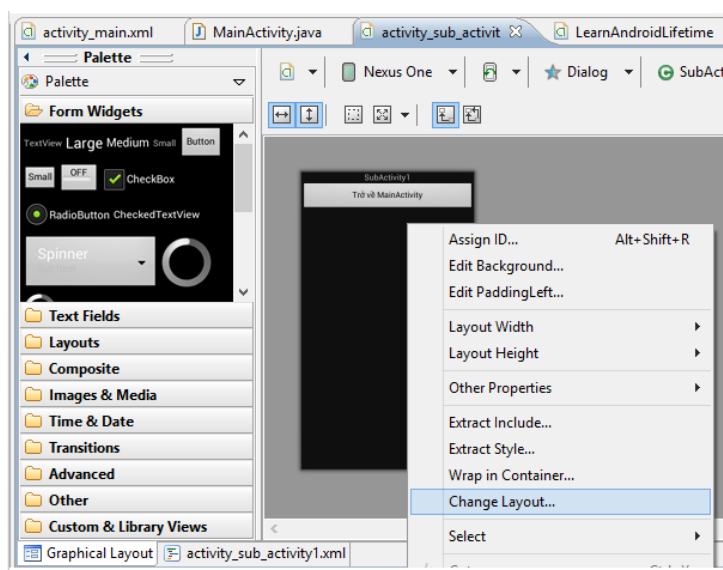
- Tạo một Android Project
- Tạo thêm một XML Layout (tùy ý) đặt tên là **new_my_layout.xml**
- Kéo thả một số control vào layout mới này: Button, Checkbox, Radio...
- Tiến hành thay đổi layout mới này qua các loại: FrameLayout, LinearLayout, TableLayout, RelativeLayout, AbsoluteLayout, ... và quan sát hiện tượng
- Trong **MainActivity**, sửa dòng lệnh **setContentView(R.layout.activity_main);** thành **setContentView(R.layout.new_my_layout);**
- Chạy ứng dụng này và cho nhận xét

Hướng dẫn:

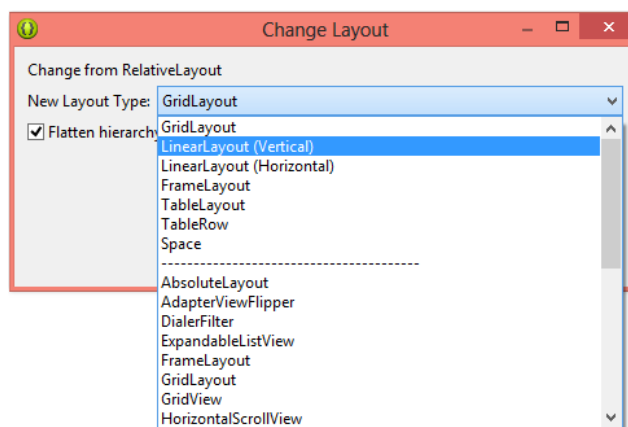
- Để tạo một XML Layout: Bấm chuột phải vào Project/ New/ Android XML File:
 - o Màn hình tạo Android XML File sẽ hiển thị lên, tại màn hình này **Resource Type** chọn là **Layout**, phần **Root Element** chọn loại **Layout** giả sử là **LinearLayout**, mục **File** nhập tên là **new_my_layout** rồi bấm nút Finish (xem hình trang bên).



- Để thay đổi Layout ta làm như sau: trong Graphical Layout/ bấm chuột phải chọn Change Layout



- Chọn loại Layout rồi bấm nút OK để thay đổi Layout



- Để xem cấu trúc XML layout dưới dạng outline: vào menu Windows/ Show View/ Outline

Bài tập 2:

Mục đích:

- Thực hành và hiểu về **FrameLayout**

Yêu cầu:

- Tạo một Android Project, chuyển Layout qua loại **FrameLayout**
- Thêm 2 **ImageView** vào **FrameLayout** này
- Chạy ứng dụng này và cho nhận xét về vị trí xuất hiện của 2 **ImageView** trên

Hướng dẫn:

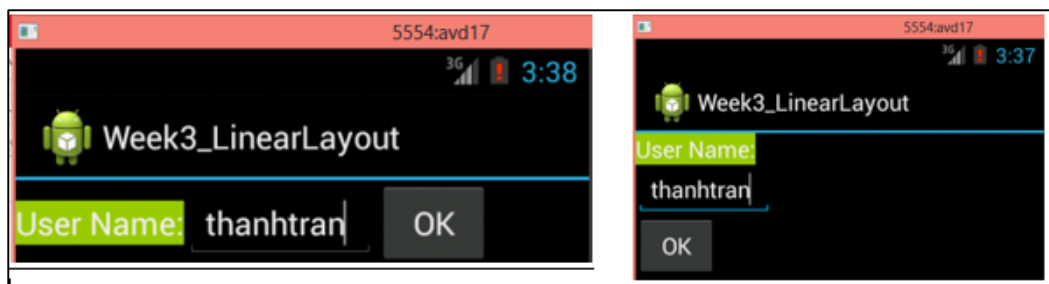
Bài tập 3:

Mục đích:

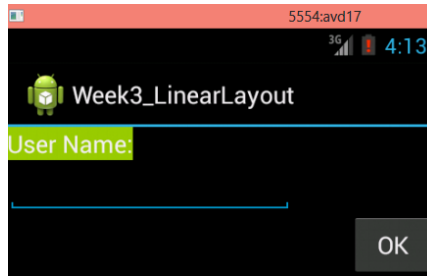
- Thực hành và hiểu về **LinearLayout**:
 - o Cách tạo **LinearLayout** (vertical, horizontal)
 - o Orientation
 - o Fill model
 - o Weight
 - o Gravity
 - o Padding
 - o Margin

Yêu cầu:

- Tạo Android Project có các control như hình bên dưới:



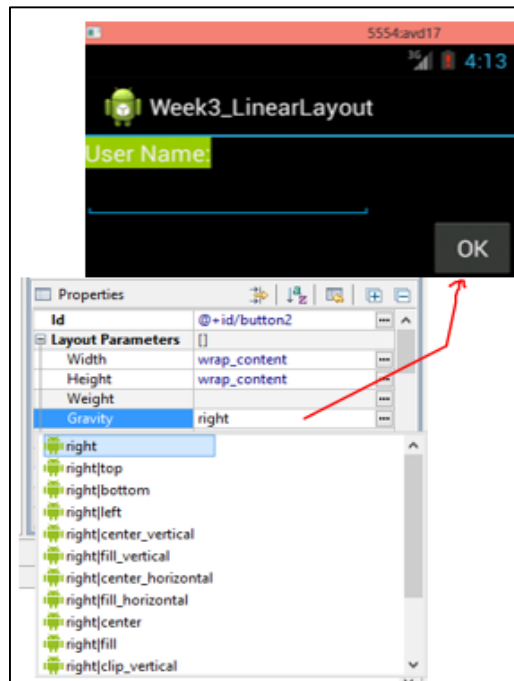
- Thay đổi **android:orientation="horizontal"** và **android:orientation="vertical"** để quan sát các control trên giao diện sắp xếp khác nhau.
- Trong mỗi control thay đổi **android:layout_width** và **android:layout_height**: thiết lập 3 thông số (125dip, wrap_content, fill_parent), quan sát sự thay đổi
- Thiết lập **TextView User Name** có **android:layout_weight="1"**, **EditText thanhtran** có **android:layout_weight="2"**, **Button OK** có **android:layout_weight="1"**, quan sát sự thay đổi của các control trên giao diện
- Thiết lập thông số **android:layout_gravity="right"** cho **Button OK**, quan sát sự thay đổi của các control có giống như màn hình bên dưới hay không:



- Thiết lập thông số **android:gravity="right"** cho Button **OK**, quan sát sự thay đổi. Hãy cho nhận xét sự khác nhau giữa **layout_gravity** và **gravity**
- Thiết lập thông số **android:padding="30dip"** và **android:layout_margin="10dip"** cho **EditText**, hãy cho biết sự khác biệt giữa padding và margin

Hướng dẫn:

- Để thiết lập các thông số cho control trên giao diện, ta sử dụng GUI của eclipse trong chế độ Graphical Layout, hoặc chỉnh sửa trực tiếp trong file XML layout:



Chế độ Graphical Layout

```

6      <TextView
7          android:id="@+id/textView1"
8          android:layout_width="match_parent"
9          android:layout_height="wrap_content"
10         android:background="#EDBD99"
11         android:text="User Name:" />
12     <EditText
13         android:id="@+id/editText1"
14         android:layout_width="match_parent"
15         android:layout_height="wrap_content"
16         android:inputType="text"
17         android:ems="10" >
18         <requestFocus />
19     </EditText>
    
```

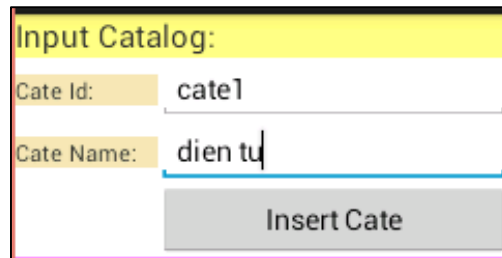
Bài tập 4:

Mục đích:

- Thực hành và hiểu về TableLayout:
 - Tạo và sử dụng được **TableLayout**
 - Hiểu được **TableRow**
 - Hiểu được **android:layout_span**
 - Hiểu được **android:column**
 - Hiểu được **android:stretchColumns**

Yêu cầu:

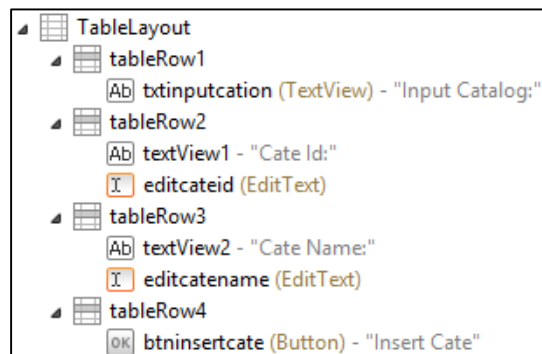
- Tạo Android Project có kiểu layout là TableLayout, thiết kế các control trên giao diện giống như hình bên dưới:



- Dòng 1 chứa TextView **Input Catalog** (có nền màu vàng)
- Dòng 2 chứa Text View Cate Id và EditText
- Dòng 3 chứa Text View Cate Name và EditText
- Dòng 4 chứa Button InsertCate
- Chạy ứng dụng và cho nhận xét.
- Hãy giải thích ý nghĩa của **android:layout_span**, **android:layout_column** và **android:stretchColumns**
- Thêm 2 Button nữa vào dòng số 4, chạy ứng dụng và cho nhận xét

Hướng dẫn:

- Làm theo outline layout sau:



- Thiết lập các thông số:
 - **android:stretchColumns="*"** cho TableLayout
 - **android:layout_span="2"** cho TextView ở dòng 1

- **android:layout_column="1"** cho Button ở dòng 4

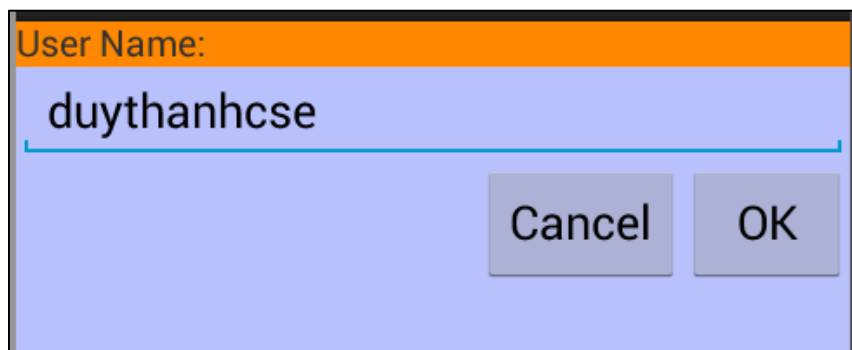
Bài tập 5:

Mục đích:

- Thực hành và hiểu về RelativeLayout
 - Tạo và sử dụng được RelativeLayout
 - Hiểu được các RelativeLayout Parameter **android:layout_** : width, height, below, above, alignTop, alignParentTop, alignBottom, alignParentBottom, toLeftOf, toRightOf, padding, margin.

Yêu cầu:

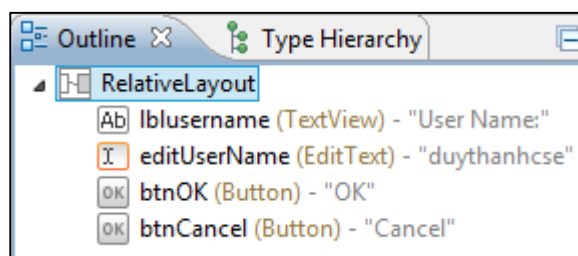
- Tạo Android Project có kiểu Layout là **RelativeLayout**, thiết kết các control như giao diện bên dưới:



- Sau khi thiết kế các control giống như trên, yêu cầu thiết lập lại **Id** cho **EditText** duythanhcse, quan sát hiện tượng xảy ra đối với các control còn lại và cho nhận xét; Tìm cách sửa lại các parameter để giữ được giao diện thiết kế ban đầu.
- Thêm 2 Checkbox lần lượt có id=chk1, id=chk2. Hai Checkbox này phải cùng nằm trên một dòng và chk1 nằm ở bên trái màn hình, chk2 nằm bên phải màn hình

Hướng dẫn:

- Làm theo outline layout sau:



- Thiết lập các thông số:
 - alignParentLeft="true", alignParentTop="true" cho **lblusername**
 - alignParentLeft="true", below="@+id/lblusername" cho **editUserName**
 - alignParentRight="true", below="@+id/editUserName" cho **btnOK**

- `alignBaseline="@+id/btnOK", alignBottom="@+id/btnOK", toLeftOf="@+id/btnOK"` cho **btnCancel**

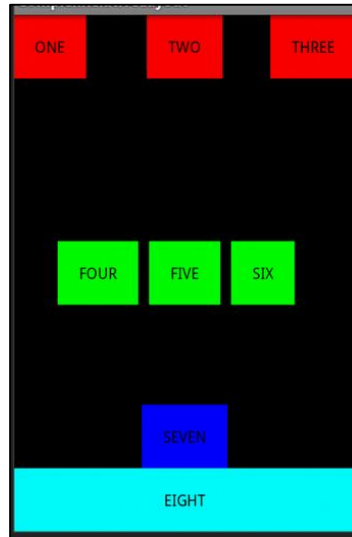
Bài tập 6:

Mục đích:

- Thực hành và hiểu về RelativeLayout (tiếp tục)

Yêu cầu:

- Dùng RelativeLayout, thiết kế giao diện như hình bên dưới:



Hướng dẫn:

Bài tập 7:

Mục đích:

- Kết hợp nhiều loại Layout

Yêu cầu:

- Thiết kế giao diện như hình bên dưới:



Hướng dẫn:

Bài tập 8:

Mục đích:

- Thực hành và hiểu được các kiểu lập trình sự kiện trong Android
 - o Onclick in XML
 - o Inline anonymous listener
 - o Activity is listener
 - o Listener in variable
 - o Explicit listener class
 - o View Subclassing

Yêu cầu:

- Thiết kế giao diện như hình bên dưới và thực hiện các yêu cầu sau:

Các kiểu lập trình sự kiện	
Số a:	80
Số b:	33
113	
Tổng 2 số	
Hiệu 2 số	
Tích 2 số	
Thương 2 số	
Ước số chung lớn nhất	
Thoát chương trình	

- Mỗi lần click chuột vào các Button thì sẽ thực hiện phép toán tương ứng: ví dụ khi click chuột vào Tổng 2 số thì kết quả ở trên là 113.

- Riêng nút Thoát chương trình: cho phép đóng chương trình
- Tổng 2 số viết theo → Onclick In XML
- Hiệu 2 số viết theo → Inline anonymous listener
- Tích 2 số viết theo → Activity is listener
- Thương 2 số viết theo → Listener in variable
- Ước số chung lớn nhất viết theo → Explicit listener class
- Và Thoát chương trình viết theo → View Subclassing

Hướng dẫn:

- Để lấy số a: `EditText edita=(EditText)findViewById(R.id.editsoa);`
`int a=Integer.parseInt(edita.getText()+"");`
- Để thoát chương trình gọi hàm: `finish();`

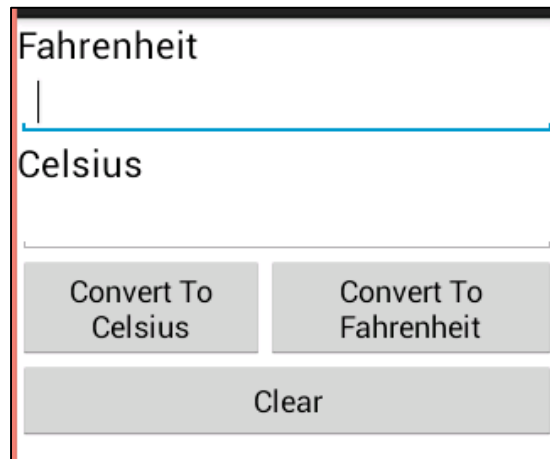
Bài tập 9:

Mục đích:

- Thực hành về các kiểu lập trình sự kiện trong Android (tiếp tục)
- củng cố thêm kỹ thuật lập trình sự kiện trong Android

Yêu cầu:

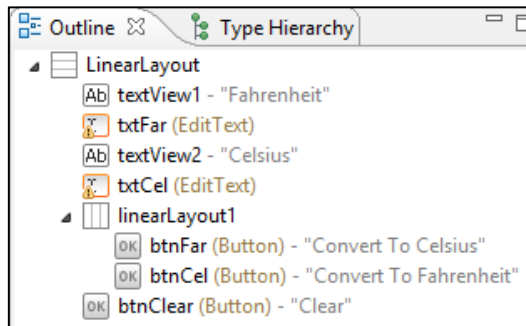
- Viết chương trình chuyển đổi nhiệt độ $C \Leftrightarrow F$. Thiết kế giao diện và thực hiện các yêu cầu như bên dưới:



- Khi bấm Convert To Celsius sẽ chuyển đổi từ $F \rightarrow C$
- Khi bấm Convert To Fahrenheit sẽ chuyển đổi từ $C \rightarrow F$
- Khi bấm nút Clear sẽ xóa trắng toàn bộ dữ liệu trong EditText

Hướng dẫn:

- Kết hợp LinearLayout – vertical và LinearLayout – horizontal để thiết kế giao diện trên



- Chọn kiểu lập trình sự kiện tùy thích
- 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 : $(98.6^{\circ}\text{F} - 32) \times 5/9 = 37^{\circ}\text{C}$

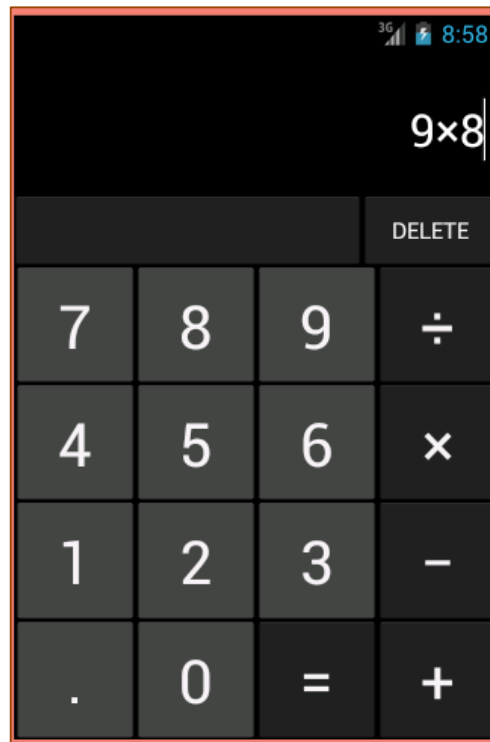
Bài tập 10 (*)

Mục đích:

- Thực hành về các kiểu lập trình sự kiện trong Android (tiếp tục)
- củng cố thêm kỹ thuật lập trình sự kiện trong Android
- củng cố lại Layout

Yêu cầu:

- Viết chương trình máy tính bỏ túi đơn giản, kết hợp một số kiểu Layout để thiết kế giao diện như hình bên dưới:



- Chương trình chỉ cần thực hiện phép toán đơn (chưa cần lồng phép toán)
- Bấm Delete : xóa dữ liệu hiện hữu trên EditText trên cùng
- Bấm vào các con số → hiển thị lên EditText
- Bấm vào phép toán: +, -, *, / → hiển thị lên EditText
- Bấm tiếp các con số → hiển thị lên EditText
- Bấm dấu “=” để thực hiện phép toán

Hướng dẫn:

- Kết hợp TableLayout và LinearLayout (hoặc tùy thích)
- Các Ô trên màn hình chính là các Button, chỉ là thay đổi màu nền và màu chữ.
- Khuyến khích các sinh viên khá giỏi hoàn thành đầy đủ chức năng của ứng dụng này

Bài tập 11:

Mục đích:

- Thực hành và hiểu được Toast Notification

Yêu cầu:

- Viết ứng dụng Android để kiểm tra Toast notification
- Giải thích trong đối số đầu tiên của **Toast.makeText** khi thì sử dụng **MainActivity.this**, khi chỉ sử dụng **this** không?

Hướng dẫn:

- **Toast toast=Toast.makeText(MainActivity.this, "Hello Toast!",**

Toast.LENGTH_SHORT) .show();

- Toast.LENGTH_SHORT → hiển thị 2 giây, Toast.LENGTH_LONG → hiển thị 3.5 giây

Bài tập 12:

Mục đích:

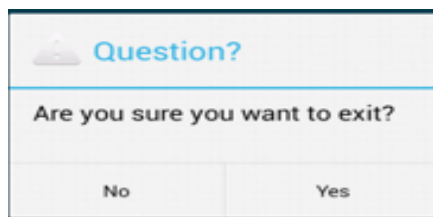
- Thực hành và hiểu được Alert Dialog

Yêu cầu:

- Viết chương trình với giao diện như hình bên dưới và thực hiện các chức năng:



- Khi người sử dụng bấm nút Đăng nhập:
 - o Nếu người sử dụng checked vào Lưu thông tin thì Toast thông báo là “chào mừng bạn đăng nhập hệ thống, bạn đã lưu thông tin”
 - o Nếu người sử dụng unchecked vào Lưu thông tin thì Toast thông báo là “chào mừng bạn đăng nhập hệ thống, bạn không lưu thông tin”
- Khi người sử dụng bấm nút Thoát, chương trình sẽ hiển thị Aler Dialog như bên dưới, chọn Yes sẽ đóng ứng dụng.



Hướng dẫn:

- Xem trong slide bài giảng (Slide 62 – xử lý giao diện người dùng)

Bài tập 13:

Mục đích:

- Thực hành và hiểu được các Control cơ bản trong Android:
 - o TextView, EditText, Button

Yêu cầu:

- Viết chương trình đổi năm Dương lịch qua năm âm lịch:

Chuyển đổi năm dương lịch

Năm dương lịch:

Chuyển đổi

Năm âm lịch: Quý tỵ

- EditText năm dương lịch chỉ được nhập số ≥ 1900
- Khi bấm nút Chuyển đổi, chương trình sẽ chuyển từ năm dương lịch qua năm âm lịch, kết quả sẽ được hiển thị vào TextView màu vàng.

Hướng dẫn:

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

Chi = Năm dương % 12 :

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

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

Bài tập 14:

Mục đích:

- Thực hành và hiểu được các Control cơ bản trong Android:
 - o TextView, EditText, Button, Checkbox, RadioButton

Yêu cầu:

- Thiết kế giao diện như hình bên dưới và thực hiện các chức năng:

- 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):

Hướng dẫn:

- Dùng **RadioGroup** để xử lý bằng cấp.
- Dùng **isChecked()** để kiểm tra RadioButton và CheckBox có được checked hay không.
- Chi tiết xem slide 69, 72 phần Xử lý giao diện người dùng.

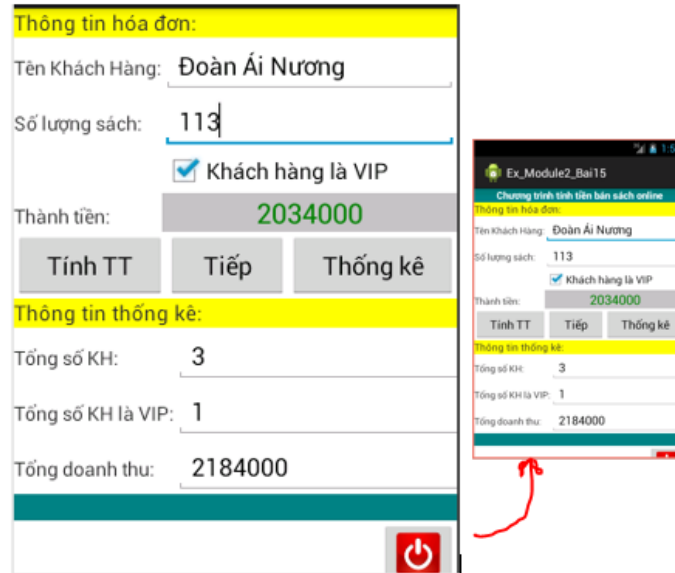
Bài tập 15:

Mục đích:

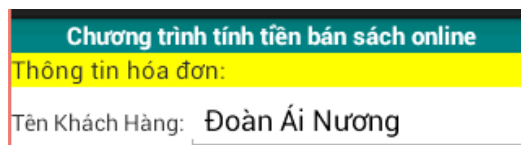
- Thực hành và hiểu được các Control cơ bản trong Android:
 - o TextView, EditText, Button, Checkbox, RadioButton, Image, ScrollView

Yêu cầu:

- Viết chương trình tính tiền bán sách online, thiết kế giao diện như hình bên dưới và thực hiện các chức năng theo yêu cầu (chú ý dùng Scroll View vì giao diện có kích thước dài hơn chiều cao của màn hình):



Dùng chức năng ScrollView lên phía trên để thấy thêm được dòng tiêu đề bị che khuất như bên dưới:



- Yêu cầu thực hiện:
 - o 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%
 - o Khi bấm nút Tiếp, chương trình sẽ lưu thông tin hóa đơn vừa tính Thành Tiền vào danh sách, đồng thời xóa trắng dữ liệu trong hóa đơn và cho focus tới EditText Tên khách hàng
 - o 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
 - o Khi bấm vào nút thoát (dùng ImageButton): hiển thị AlertDialog hỏi xem người sử dụng có chắc chắn muốn thoát hay không?

Hướng dẫn:

- Kết hợp ScrollView (để ở ngoài cùng), LinearLayout và TableLayout

- Dùng List hoặc ArrayList để lưu lại thông tin mỗi lần bấm nút Tính Thành Tiền, mục đích để phục vụ cho chức năng thống kê.

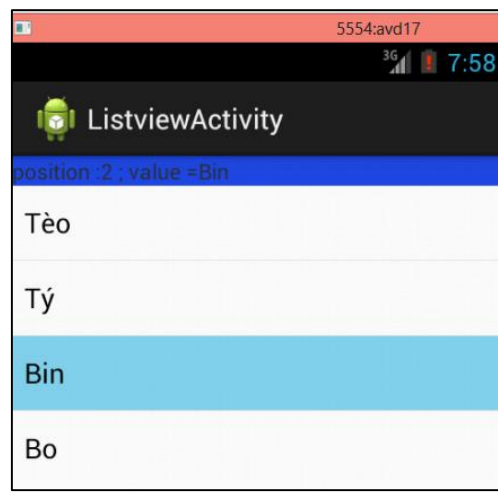
Bài tập 16:

Mục đích:

- Thực hành và hiểu được các control nâng cao trong Android:
 - o ListView
- Hiểu được cách sử dụng String-Array, ArrayAdapter

Yêu cầu:

- Thiết kế giao diện như hình bên dưới:



- Giao diện gồm 1 TextView và 1 ListView
- Trong res/values/strings.xml tạo một String-**Array**: chứa một số các giá trị như hình trên
- Khi ứng dụng được khởi động, nó sẽ tự động đọc các giá trị trong String-Array và hiển thị lên ListView
- Khi click chuột vào từng phần tử trên ListView, sẽ hiển thị vị trí và giá trị phần tử được click vào TextView có màu xanh dương bên trên.

Hướng dẫn:

- Trong hàm onCreate của Activity thiết lập ArrayAdapter cho ListView
- Chi tiết xem slide 78,79,80 phần xử lý giao diện người dùng

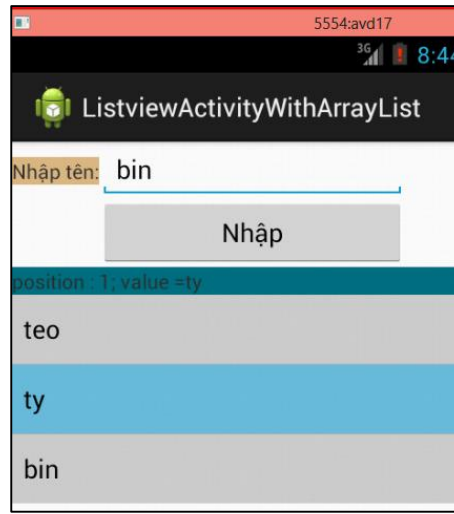
Bài tập 17:

Mục đích:

- Thực hành và hiểu được các control nâng cao trong Android:
 - o ListView
- Hiểu được cách sử dụng ArrayList, ArrayAdapter

Yêu cầu:

- Thiết kế giao diện như yêu cầu:



- Mỗi lần người sử dụng chọn nút Nhập, chương trình sẽ lưu dữ liệu trong mục Nhập tên vào ArrayList đồng thời cập nhật danh sách vào ListView
- Khi chọn từng phần tử trong ListView, chương trình sẽ hiển thị vị trí và giá trị của phần tử đang chọn vào TextView màu xanh

Hướng dẫn:

- Sử dụng hàm **notifyDataSetChanged** của **ArrayAdapter** để tự động cập nhật lại danh sách cho ListView khi ArrayList thay đổi dữ liệu.
- Chi tiết xem slide 81, 82 phần xử lý giao diện người dùng

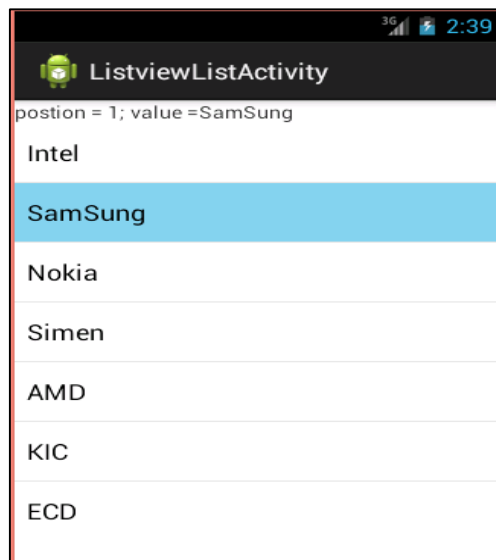
Bài tập 18:

Mục đích:

- Thực hành và hiểu được các control nâng cao trong Android:
 - o ListView
- Hiểu được cách sử dụng ListActivity, ArrayAdapter

Yêu cầu:

- Thiết kế giao diện như yêu cầu bên dưới (kế thừa từ **ListActivity** thay vì **Activity**):



Hướng dẫn:

- Kế thừa từ ListActivity
- Đặt id cho ListView là **@android:id/list** – Chính là tên mặc định trong Android
- Chi tiết xem slide 83, 84, 85, 86 phần Xử lý giao diện

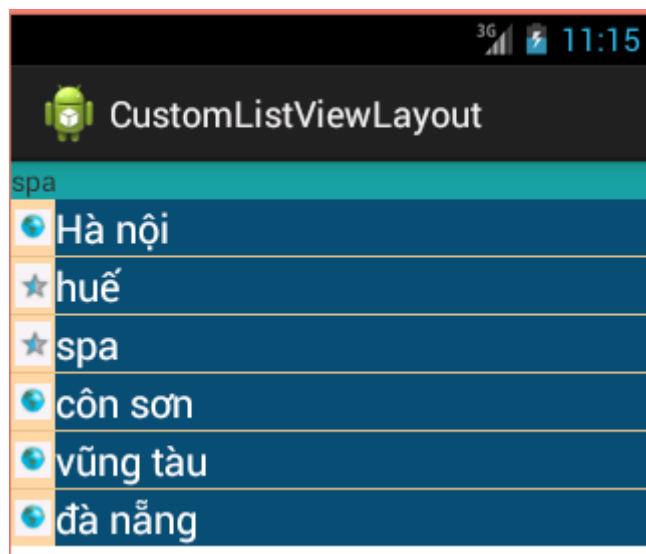
Bài tập 19:

Mục đích:

- Thực hành và hiểu về các control nâng cao trong Android:
 - o Học cách tạo Custom ListView
 - o Cách kế thừa ArrayAdapter

Yêu cầu:

- Hãy xây dựng ứng dụng Android như hình bên dưới, yêu cầu hiệu chỉnh lại ListView để có thể hiện thị theo ý muốn riêng



- Mỗi phần tử trong ListView sẽ có 1 hình ảnh và 1 nội dung, nếu nội dung có số lượng ký tự ≤ 3 thì hình ngôi sao, còn lại hình địa cầu
- Mỗi lần chọn từng phần tử trong ListView sẽ hiển thị thông tin của phần tử đó lên TextView

Hướng dẫn:

- Xem slide 143, 144, 145, 146 phần xử lý giao diện

Bài tập 20:

Mục đích:

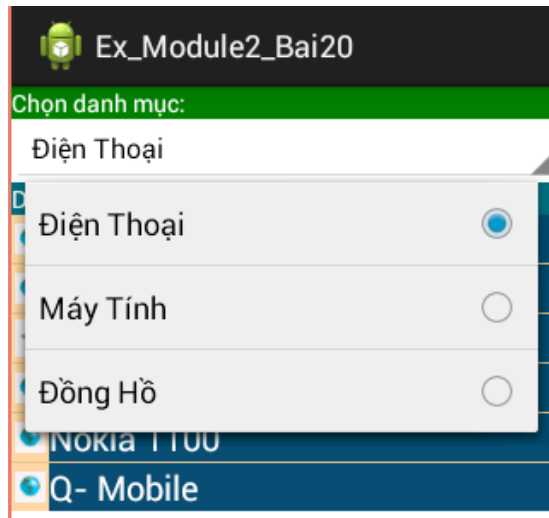
- Thực hành và hiểu về các control nâng cao trong Android:
 - o Spinner
 - o Kết hợp Spinner, ListView

Yêu cầu:

- Thiết kế ứng dụng như màn hình bên dưới:



- Khi ứng dụng khởi động, Danh mục sản phẩm sẽ hiển thị vào trong Spinner như hình
- Mỗi lần chọn vào danh mục trong Spinner, chương trình sẽ hiển thị danh sách sản phẩm vào ListView theo danh mục tương ứng:



- ứng với mỗi sản phẩm là điện thoại sẽ có mỗi hình ảnh riêng
- Khi chọn từng sản phẩm trong ListView, chương trình sẽ dùng Toast để hiển thị chi tiết

Hướng dẫn:

- Áp dụng Custom Layout để thiết kế cho ListView
- Giả lập 2 ArrayList, ArrayList 1 chứa danh sách Danh Mục Sản phẩm, ArrayList 2 chứa danh sách sản phẩm theo danh mục
- ứng dụng để load từ CSDL lên giao diện cho phần học kế tiếp

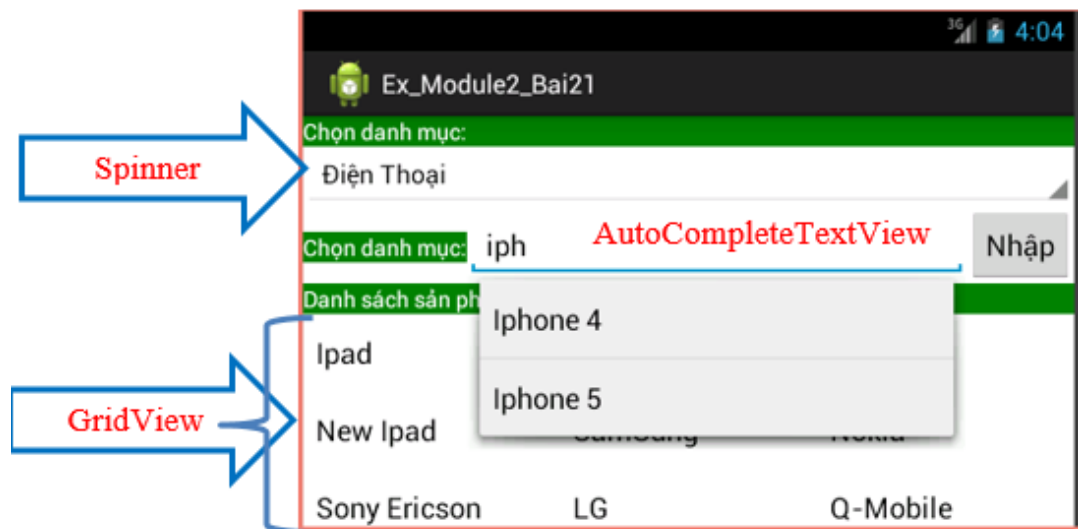
Bài tập 21 (*)

Mục đích:

- Thực hành và hiểu về các control nâng cao trong Android:
 - o Hiểu và sử dụng được AutoCompleteTextView
 - o Hiểu và sử dụng được GridView, kết hợp với Spinner

Yêu cầu:

- Thiết kế giao diện **nằm ngang** như hình bên dưới và thực hiện các yêu cầu:



- Khi khởi động chương trình sẽ load danh sách danh mục sản phẩm vào Spinner
- Mỗi lần chọn Danh mục sản phẩm, chương trình sẽ hiển thị danh sách sản phẩm của danh mục đang chọn vào GridView (xem hình)
- Khi người sử dụng nhập 1 tên sản phẩm mới vào AutoCompleteTextView và chọn nút Nhập thì chương trình sẽ đưa sản phẩm mới này vào đúng danh mục đang chọn; cập nhập vào AutoComplete và đồng thời cập nhập vào GridView (2 chỗ cập nhập)
- Mỗi lần chọn từng phần tử trong GridView, chương trình sẽ dùng Alert Dialog để hiển thị thông tin chi tiết cho từng phần tử.

Hướng dẫn:

- Cả 3 đối tượng: Spinner, AutoCompleteTextView, GridView đều sử dụng ArrayAdapter
- Như vậy mỗi lần có sự thay đổi dữ liệu (List chứa dữ liệu tương ứng thay đổi) thì ta gọi hàm **notifyDataSetChanged** của ArrayAdapter tương ứng cho mỗi đối tượng.
- Thiết lập **android:completionThreshold="1"** cho AutoCompleteTextView. Tùy vào ý đồ của người lập trình mà ta thiết lập các giá trị khác nhau, Số này có ý nghĩa để kích hoạt chức năng AutoComplete, nếu thiết lập là 1 thì gõ 1 ký tự nó sẽ kích hoạt, nếu thiết lập là 3 thì gõ 3 ký tự nó sẽ kích hoạt...
- **setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_SENSOR_LANDSCAPE);** để thiết lập ứng dụng nằm ngang điện thoại.

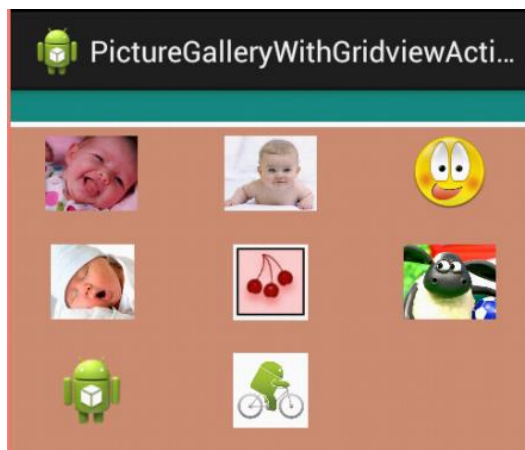
Bài tập 22 (*):

Mục đích:

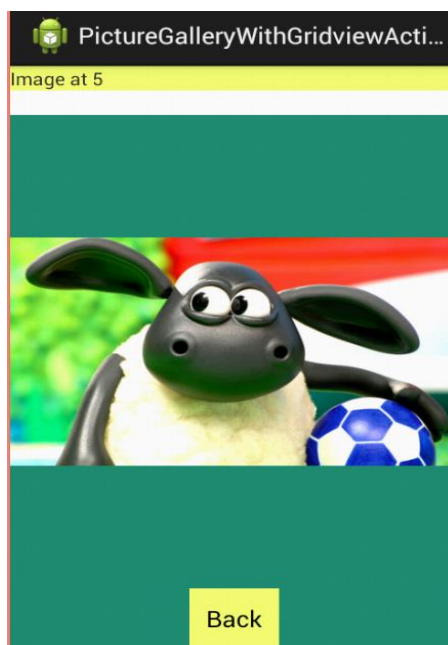
- Thực hành và hiểu về các control nâng cao trong Android:
 - o Hiểu cách đưa hình ảnh vào GridView, biết cách sử dụng **BaseAdapter**

Yêu cầu:

- Viết ứng dụng trình diễn hình ảnh như sau (sử dụng GridView):



- Khi chọn từng hình ảnh trong GridView thì sẽ mở một màn hình mới để hiển thị thông tin chi tiết của hình đang chọn (ví dụ chọn hình chú Cừu):



- Chọn nút Back để trở về màn hình chính

Hướng dẫn:

- Xem hướng dẫn chi tiết slide 120, 121, 122, 123, 124 phần xử lý giao diện người dùng

Bài tập 23:

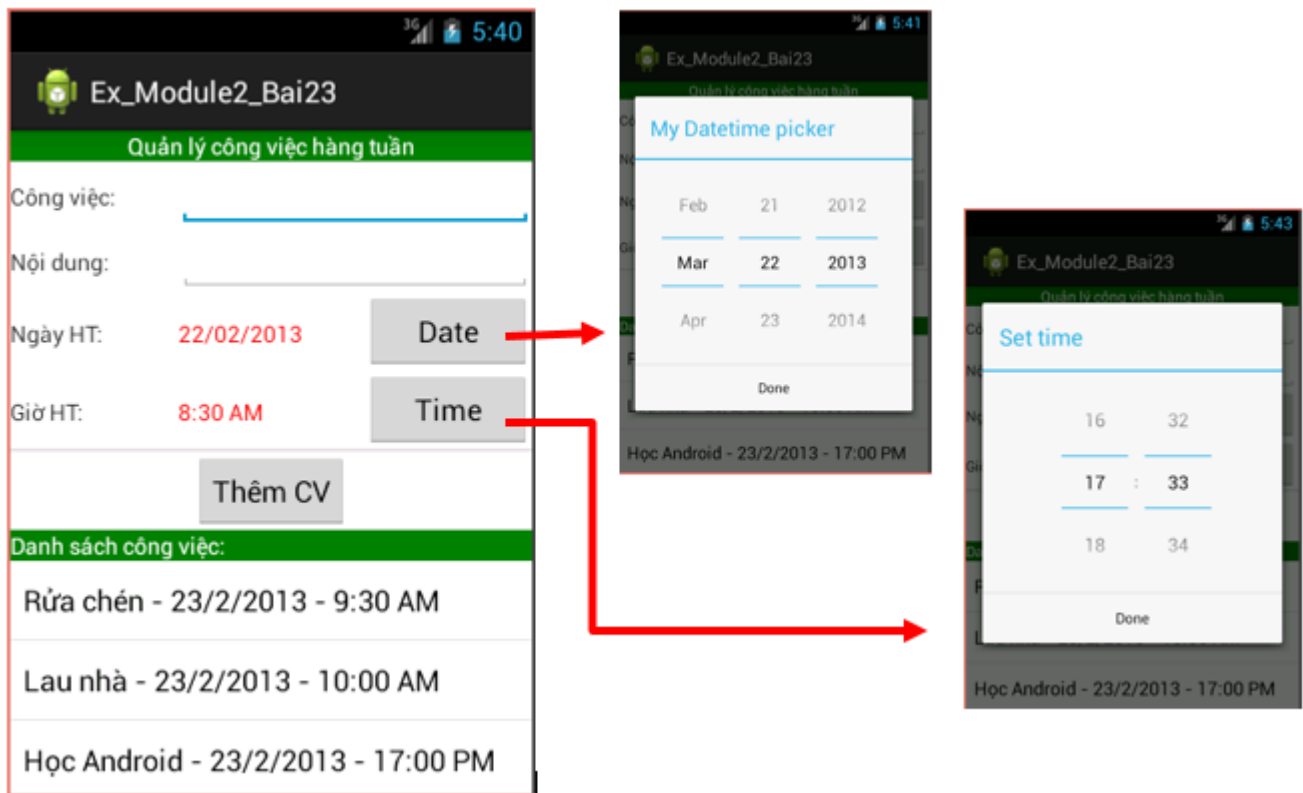
Mục đích:

- Thực hành và hiểu các control nâng cao trong Android:
- Hiểu và sử dụng được Time Selection:
 - o AnalogClock
 - o Chronometer
 - o TimePickerDialog

- DatePickerDialog

Yêu cầu:

- Viết chương trình quản lý mục tiêu hoàn thành công việc hàng tuần: Lý do chúng ta làm việc không có quy củ, không nhất quán, gặp đâu làm đó, đang làm việc này thấy việc kia xảy ra lại nhảy qua việc đó làm để dở công việc cũ chưa xong, không theo quy trình định sẵn dẫn đến kết thúc một ngày mà cuối cùng không hoàn thành được việc gì cả.
- Thiết kế giao diện như hình bên dưới và thực hiện một số chức năng:



- Khi chọn nút Date sẽ hiển thị DatePickerDialog – cập nhật Ngày hoàn thành
- Khi chọn nút Time sẽ hiển thị TimePickerDialog – cập nhật giờ hoàn thành
- Khi chọn nút Thêm công việc (Thêm CV), chương trình sẽ cập nhật vào ListView bên dưới màn hình

Hướng dẫn:

- Xem hướng dẫn cách sử dụng DatePickerDialog và TimePickerDialog ở slide 113, 114 phần Xử lý giao diện người dùng

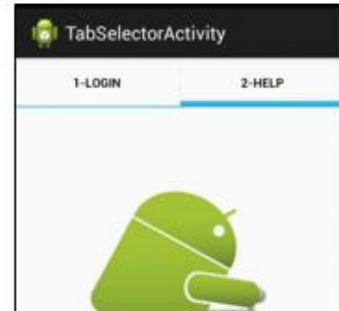
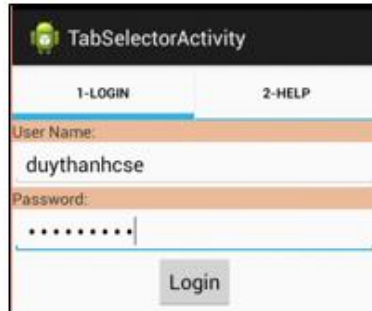
Bài tập 24:

Mục đích:

- Thực hành và hiểu các control nâng cao trong Android:
 - Hiểu và sử dụng được TabSelector

Yêu cầu:

- Thiết kế ứng dụng như yêu cầu bên dưới:



- ứng dụng gồm có 2 Tab: tab Login và tab Help
- ứng với mỗi Tab sẽ có giao diện khác nhau, sinh viên tự tạo giao diện tùy thích để thao tác với TabSelector

Hướng dẫn:

- Xem Slide 128, 129, 130 phần Xử lý giao diện người dùng

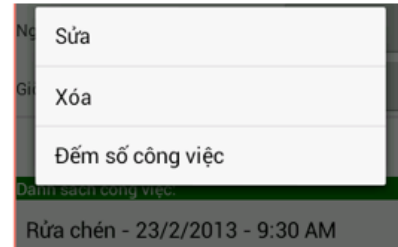
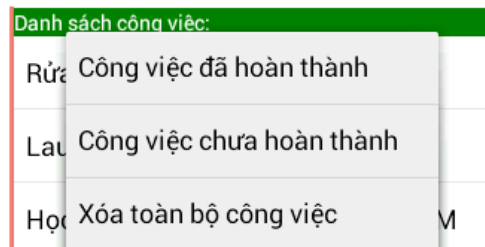
Bài tập 25:

Mục đích:

- Thực hành và hiểu các control nâng cao trong Android:
- Hiểu và sử dụng được Menu, ContextMenu

Yêu cầu:

- Hiệu chỉnh bài tập 23, thêm 1 Menu và 1 ContextMenu như sau:



- Thiết lập ContextMenu cho ListView (bấm long time để hiển thị ContextMenu): chọn sửa cho phép sửa công việc, chọn Xóa cho phép xóa công việc, và đếm số CV
- Thiết lập Menu cho ứng dụng (bấm vào nút Menu của emulator sẽ hiển thị Menu): Xem công việc đã hoàn thành, chưa hoàn thành và xóa toàn bộ công việc

Hướng dẫn:

- Xem hướng dẫn ở slide 133→140 phần xử lý giao diện người dùng

Bài tập 26 (*):

Mục đích:

- Thực hành và hiểu về Webkit:
 - o Thao tác được với Webkit Browser

- Các Browser Command
- Cách đăng ký đối tượng trong Android vào JavaScript
- Tương tác Javascript và Android

Yêu cầu:

- Xây dựng ứng dụng tìm vị trí (dùng dữ liệu giả), cho phép tương tác giữa Android với HTML (Java Script). Giao diện màn hình như sau:



- Khi chọn nút “Click to get location”, chương trình sẽ lấy thông số từ Android và hiển thị lên giao diện HTML
- Khi chọn nút “Talking to Android”, chương trình sẽ gửi thông số trên HTML tới Android và Android dùng Toast để hiển thị những thông số đã lấy được từ HTML này

Hướng dẫn:

- Dùng Microsoft Express Web hoặc trình soạn thảo bất kỳ để thiết kế HTML như trên
- Chú ý nội dung tập tin HTML chỉ có chứa tag html ngoài cùng, tuyệt đối không chứa các Tag lạ khác.
- Kéo thả tập tin HTML vào thư mục assets của ứng dụng Android
- Chi tiết xem slide hướng dẫn 161→167 phần xử lý giao diện người dùng
- Nhớ thiết lập 3 thông số này vào Manifest XML

```
<uses-permission android:name="android.permission.INTERNET"/>
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION"/>
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"/>
```

Bài tập 27:

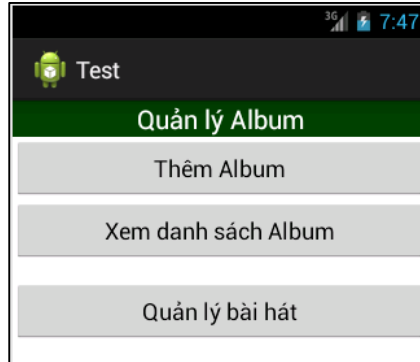
Mục đích:

- Thực hành và hiểu được về Intent (phải làm hết bài tập này, rất quan trọng):

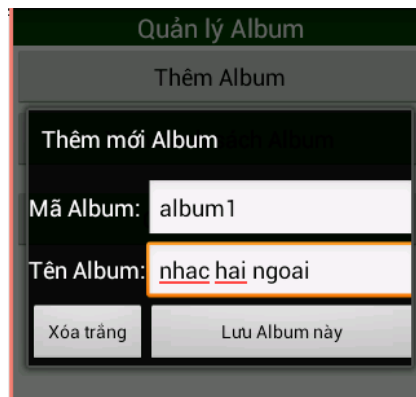
- Explicit Intent
- Implicit Intent
- Cách lấy kết quả qua lại giữa các Activity thông qua Intent

Yêu cầu:

- Viết chương trình quản lý bài hát, thiết kế giao diện như hình bên dưới:



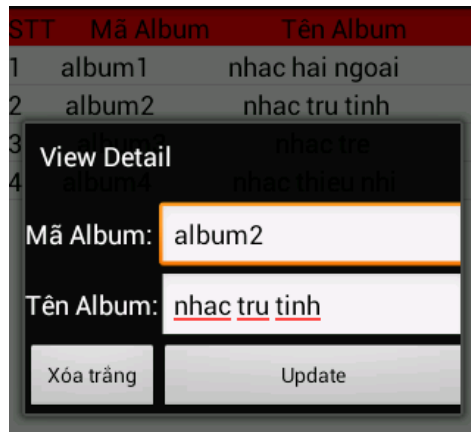
- Phải tạo Album trước khi vào chức năng quản lý bài hát
- Khi chọn chức năng thêm Album, chương trình sẽ xuất hiện màn hình dưới đây (chú ý mở dưới dạng Dialog):



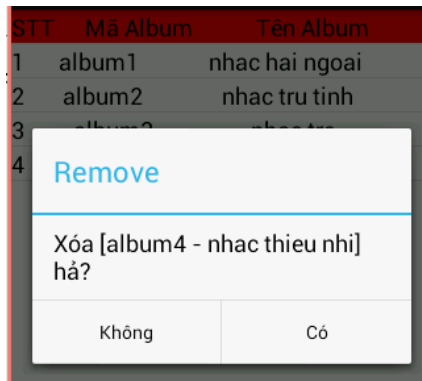
- Chọn “lưu album này” để lưu, chọn xóa trắng để xóa dữ liệu vừa nhập, focus tới mã.
- Khi chọn “xem danh sách album”, chương trình sẽ mở một màn hình mới để hiển thị danh sách Album đã tạo – chú ý dùng CustomLayout cho ListView:

STT	Mã Album	Tên Album
1	album1	nhạc hai ngoai
2	album2	nhac tru tinh
3	album3	nhac tre
4	album4	nhac thieu nhi

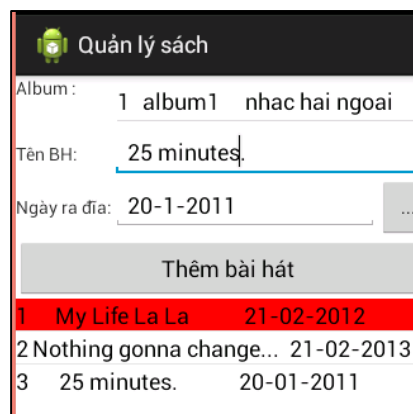
- Khi chọn từng Album trong danh sách, sẽ hiển thị màn hình cho phép chỉnh sửa Album:



- Chọn Update để cập nhật lại Album
- Khi nhấn Thật Lâu (Long time) vào từng Album trong danh sách, chương trình sẽ hiển thị Alert Dialog hỏi xem có muốn xóa Album này hay không?



- Chọn Có để xóa album muốn xóa, chọn Không để trở về màn hình xem danh sách
- Khi chọn chức năng “quản lý bài hát” ở màn hình chính chương trình sẽ hiển thị:



- Load danh sách album vào Spinner
- Sử dụng DatePickerDialog để chọn ngày ra đĩa
- Chọn “Thêm bài hát” để đưa bài hát mới vào ListView bên dưới.

Hướng dẫn:

- Xem hướng dẫn sử dụng Intent, get result trong slide 170→180; 191→205 phần xử lý giao diện người dùng.

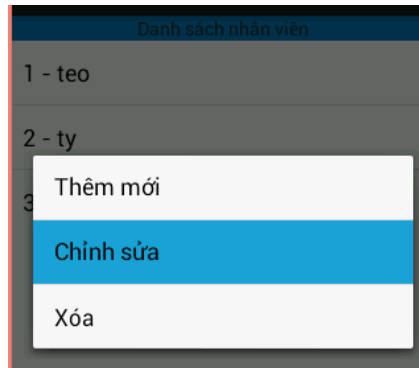
Bài tập 28:

Mục đích:

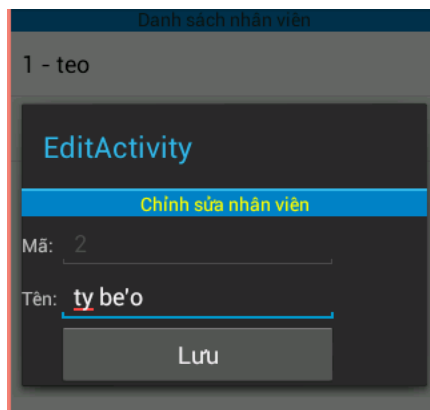
- Thực hành và hiểu được về Intent:
 - Explicit Intent
 - Implicit Intent
 - Cách lấy kết quả qua lại giữa các Activity thông qua Intent

Yêu cầu:

- Viết chương trình quản lý nhân sự như bên dưới (sử dụng Context Menu hoặc Menu):



- Chọn Chỉnh sửa để hiệu chỉnh nhân viên đang chọn trong ListView

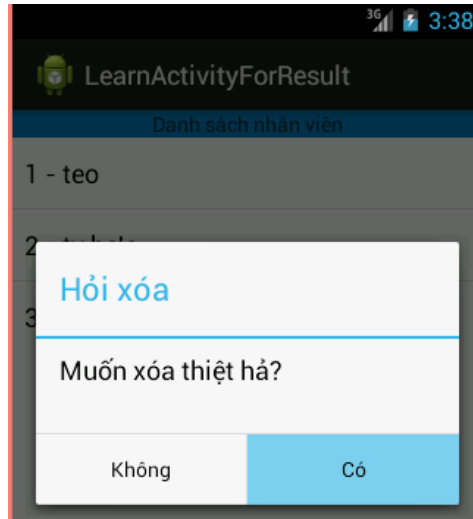


- Chọn Lưu để cập nhật
- Chọn Thêm mới để thêm nhân viên mới:



- Chọn Lưu để thêm mới

- Khi chọn Xóa, chương trình sẽ hiển thị Alert Dialog xác nhận xem có xóa hay không?



Hướng dẫn:

- Xem hướng dẫn sử dụng Intent, get result trong slide 170→180; 191→205 phần xử lý giao diện người dùng.

Bài tập 29:

Mục đích:

- Thực hành và hiểu về chạm đa điểm trong Android

Yêu cầu:

- Viết một ứng dụng kiểm tra nếu như có 3 ngón tay của người sử dụng lướt trên màn hình thì chuyển màu nền sang Màu đỏ
- Nếu có 5 ngón tay của người sử dụng lướt trên màn hình thì mở một bài nhạc bất kỳ nào đó (dùng **Implicit Intent**).

Hướng dẫn:

- Xem slide hướng dẫn Multitouch 207→212 phần xử lý giao diện người dùng

Bài tập 30:

Mục đích:

- Thực hành và hiểu về đa ngôn ngữ trong Android

Yêu cầu:

- Viết một ứng dụng cho phép xử lý đa ngôn ngữ trong Android (ứng dụng bất kỳ)

Hướng dẫn:

- Xem hướng dẫn đa ngôn ngữ 215→218 phần xử lý giao diện người dùng

Module 3: Xử lý tập tin, lưu trạng thái, CSDL, Content Provider

Nội dung kiến thức thực hành:

- Thao tác với các tập tin trong Android: Internal Storage, External Storage và Cache files
- Thực hành về XML Parser: DOM, SAX
- Thực hành về lưu trạng thái ứng dụng: Share Preferences
- Thực hành về Sqlite: cách tạo/ xóa CSDL, tạo bảng, truy vấn, thêm, sửa, xóa dữ liệu...
- Thực hành về Content Provider: thao tác với các Content Provider có sẵn, cách tạo mới các Content Provider

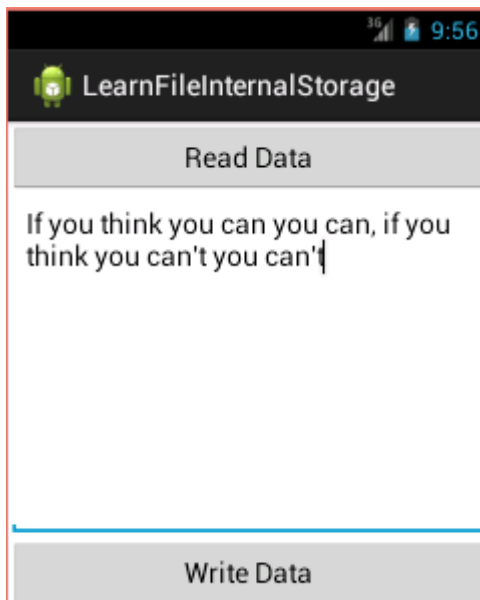
Bài tập 1:

Mục đích:

- Thực hành và hiểu được cơ chế lưu tập tin trong Android:
 - o Internal Storage
 - o External Storage
 - o Cache files

Yêu cầu:

- Viết chương trình cho phép đọc / ghi một TextFile bất kỳ từ ứng dụng:



- Khi bấm Write Data, chương trình sẽ lưu toàn bộ thông tin trong EditText xuống tập tin
- Khi bấm Read Data, chương trình sẽ đọc thông tin từ tập tin và hiển thị lên EditText
- Chú ý thử trên 2 trường hợp: Internal và SD card, quan sát nơi lưu trữ tập tin trong File Explorer của DDMS

Hướng dẫn:

- Xem slide 8 , 9 phần xử lý tập tin

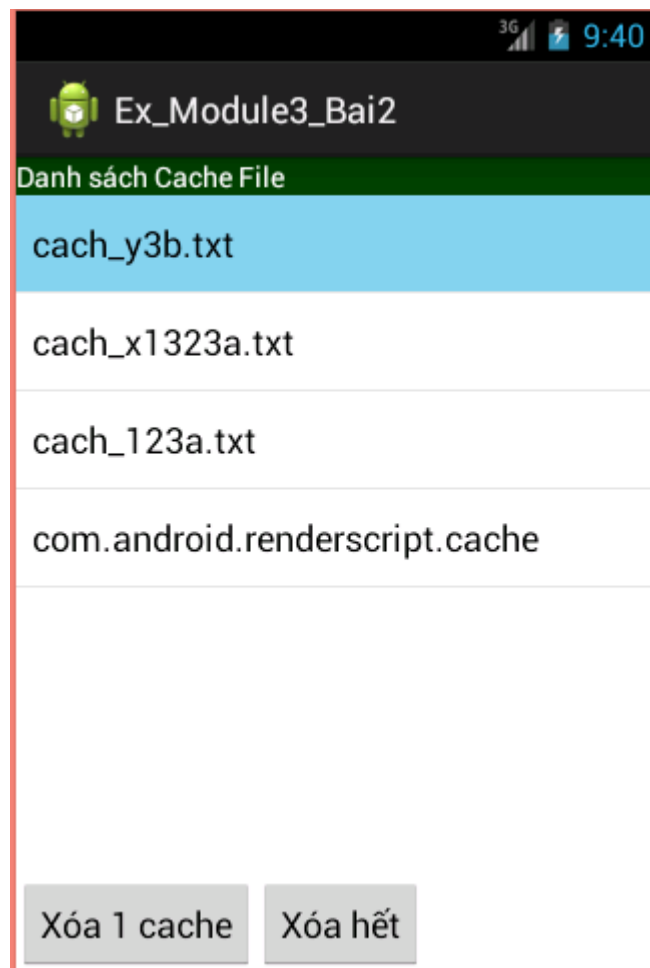
Bài tập 2:

Mục đích:

- Thực hành và hiểu được cơ chế lưu tập tin trong Android:
 - o Internal Storage
 - o External Storage
 - o Cache files

Yêu cầu:

- Viết chương trình cho phép load toàn bộ danh sách Cache files lên giao diện, mỗi lần chọn vào từng Cache sẽ cho phép đọc thông tin từ cache. Chương trình còn hỗ trợ chức năng Xóa từng Cache và xóa toàn bộ Cache



Hướng dẫn:

- Sử dụng Intent để mở Activity xem chi tiết nội dung của từng cache chọn trên ListView
- Sử dụng lệnh sau để lấy toàn bộ Cache:
 - File** cachDir=getCacheDir();
 - File** listCache[]=cachDir.listFiles();

Bài tập 3:

Mục đích:

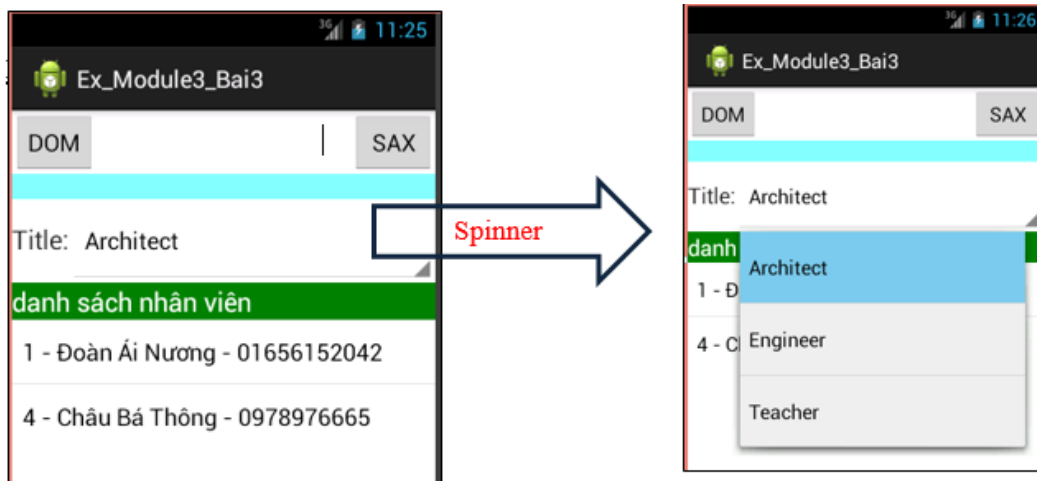
- Thực hành và hiểu được XML Parser:
 - o Kỹ thuật dùng DOM
 - o Kỹ thuật dùng SAX

Yêu cầu:

- Cho cấu trúc XML sau:

```
<?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 Thuận</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>
```

- Hãy dùng 2 kỹ thuật DOM và SAX (viết 2 cách) để thực hiện đọc file XML trên và đưa lên giao diện như hình bên dưới:



- Chương trình sẽ đọc danh sách Title vào Spinner (chú ý không được trùng lặp)
- Chọn Title nào thì sẽ hiển thị danh sách nhân viên tương ứng vào ListView theo Title đó.

Hướng dẫn:

- Xem kỹ thuật DOM, SAX trong slide 27→33 phần xử lý tập tin

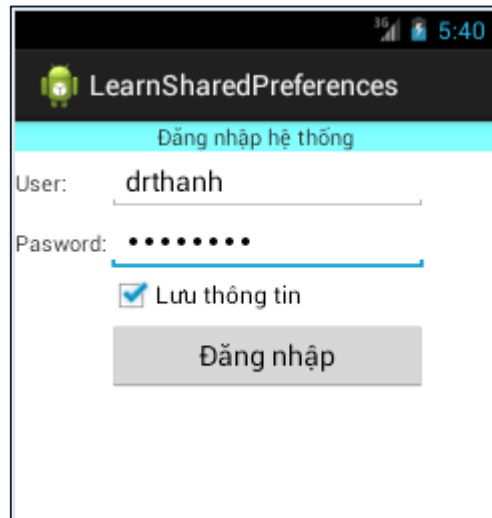
Bài tập 4:

Mục đích:

- Thực hành và hiểu được Share Preferences:
 - o Cách tạo và sử dụng Share Preferences
 - o Cách sử dụng Share Preferences Change listener (tạo các Settings Activity)

Yêu cầu:

- Viết ứng dụng với giao diện giống hình dưới đây:



- Yêu cầu thực hiện:
 - o Khi người sử dụng chọn Đăng nhập thì đóng ứng dụng này lại. Chú ý nếu checkbox Lưu thông tin được checked thì chương trình sẽ lưu toàn bộ thông tin (user, password). Nếu checked =false thì xóa toàn bộ thông tin lưu trước đó.
 - o Khi khởi động lại ứng dụng, chương trình sẽ tìm xem thông tin có được lưu hay không, nếu có lưu thì hiển thị lại lên màn hình cho người sử dụng

Hướng dẫn:

- Xem slide hướng dẫn 38 , 39 phần xử lý tập tin

```

SharedPreferences pre=getSharedPreferences
                                (prefname, MODE_PRIVATE);
SharedPreferences.Editor editor=pre.edit();
editor.putString("user", user);
editor.commit();

String user=pre.getString("user", "");
    
```

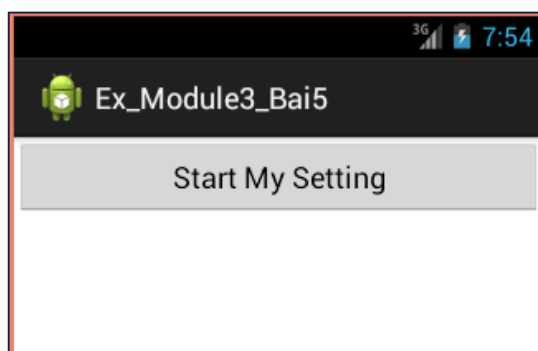

Bài tập 5:

Mục đích:

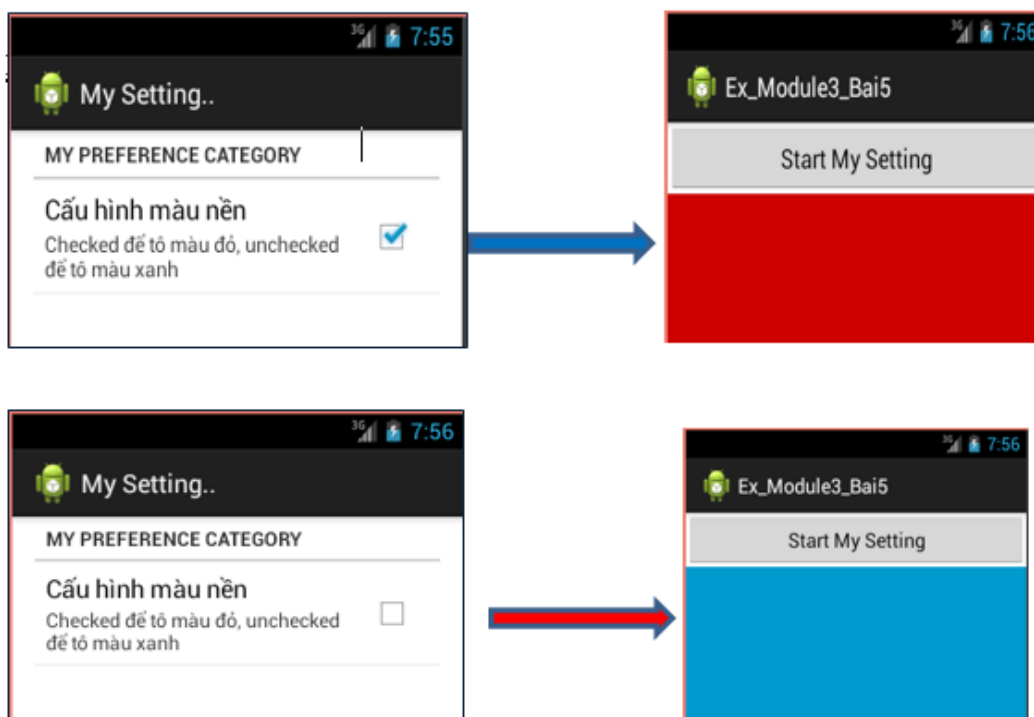
- Thực hành và hiểu được Share Preferences:
 - Cách tạo và sử dụng Share Preferences
 - Cách sử dụng Share Preferences Change listener (tạo các Settings Activity)

Yêu cầu:

- Viết ứng dụng dạng PreferenceActivity có giao diện như bên dưới:



- Khi bấm Start my Setting sẽ hiển thị màn hình Setting:



- Nếu checked màn hình chính sẽ có màu đỏ
- Nếu unchecked màn hình chính sẽ có màu xanh



- Chú ý là màn hình chính phải đăng ký Share Preferences Listener thì mới tự động bắt được những sự thay đổi trong Setting Activity

Hướng dẫn:

- Xem hướng dẫn trong slide 48→53 phần xử lý tập tin.

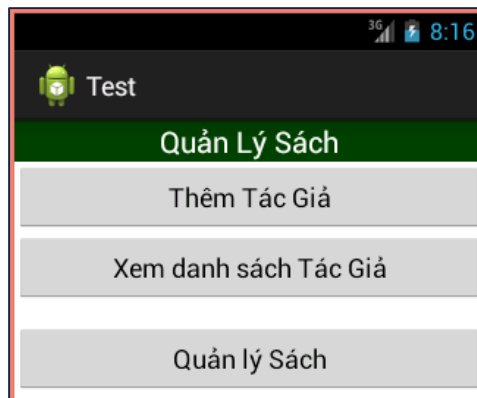
Bài tập 6:

Mục đích:

- Thực hành và hiểu về SQLite
 - o Tạo được CSDL
 - o Thao tác với bảng dữ liệu: thêm, sửa , xóa
 - o Và một số tính năng khác được cung cấp bởi SQLite

Yêu cầu:

- Viết chương trình quản lý sách được mô tả như sau:
 - o Một tác giả sẽ có nhiều cuốn sách, thông tin tác giả gồm: mã , tên
 - o Mỗi cuốn sách thuộc về một nhà tác giả nào đó, thông tin mỗi cuốn sách gồm: mã sách, tên sách, ngày xuất bản
 - o 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):



- o 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:

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

View Detail

Mã tác giả:

Tên tác giả:

Xóa trắng Update

- o 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 nhật vào ListView bên dưới.

Hướng dẫn:

- Xem hướng dẫn cách sử dụng SQLite trong slide 54→68 phần xử lý tập tin

Bài tập 7:

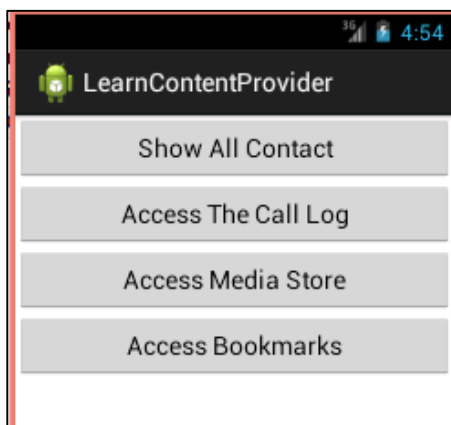
Mục đích:

- Thực hành và hiểu về Content Provider:

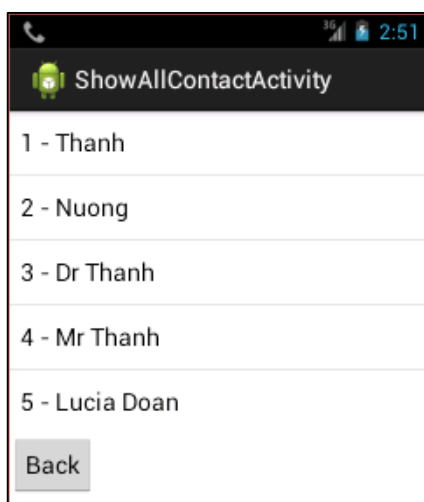
- Sử dụng được Built – in Content Provier
- Tự xây dựng được Content Provider riêng

Yêu cầu:

- Viết chương trình để sử dụng một số Content Provider phổ biến, thiết kế giao diện giống như hình bên dưới:



- Khi chọn Show All Contact, chương trình sẽ hiển thị danh sách các Contact được lưu trong điện thoại:



- Tương tự như vậy viết chương trình sử dụng Content Provider có sẵn để sử dụng Call Log, Media Store, Book Marks

Hướng dẫn:

- Xem hướng dẫn chi tiết slide 73→80 phần xử lý tập tin

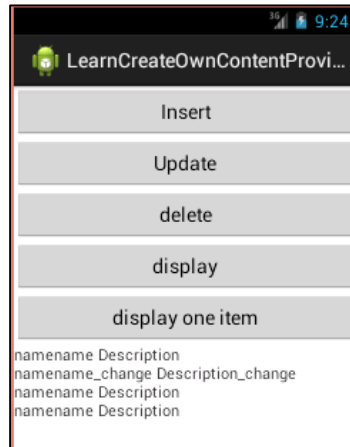
Bài tập 8 (*):**

Mục đích:

- Thực hành và hiểu về Content Provider:
 - Sử dụng được Built – in Content Provier
 - Tự xây dựng được Content Provider riêng

Yêu cầu:

- Tự xây dựng một Content Provider cho phép tương tác với Movie Object, thiết kế giao diện như bên dưới:



Hướng dẫn:

- Làm lại slide hướng dẫn từ 83→97, dành cho sinh viên khá giỏi

Module 4: Xử lý đa tiến trình và dịch vụ

Nội dung kiến thức thực hành:

- Thực hành về đa tiến trình: Cách thức lập trình với Handler class và AsyncTask
- Thực hành về Intent Filter
- Thực hành về Broadcast Receiver: Cách đăng ký broadcast Receiver trong coding và trong Manifest XML, thao tác với một số chức năng trong Broadcast Receiver
- Thực hành về Android Service: cách thức giao tiếp với Service thông qua Broadcast, Binder, Messenger.
- Thực hành về System Service: Pending Intent, Alarm Manager, NotificationManager
- Thực hành về .Net Webservice: Cách tạo .net webservice, cách cấu hình IIS, cách thức tương tác dữ liệu giữa Android và .Net Webservice: KSOAP Api

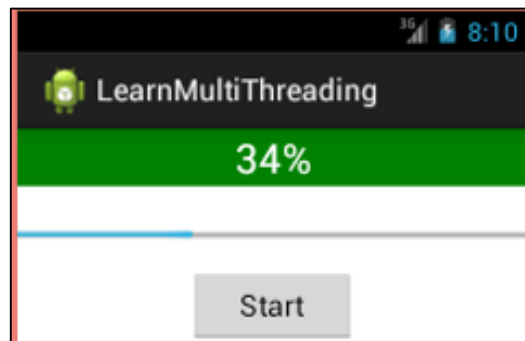
Bài 1:

Mục đích:

- Thực hành và hiểu được kỹ thuật lập trình Đa tiến trình
 - o Cách tiếp cận bằng Handler
 - o Cách tiếp cận bằng AsyncTask

Yêu cầu:

- Viết chương trình sau dưới dạng đa tiến trình:



- Yêu cầu thực hiện với nhiều cách (thực hiện bài này với 3 cách khác nhau):
 - o Handler → using sendMessage
 - o Handler → using post
 - o Dùng AsyncTask
- Khi bấm Start chương trình sẽ cập nhật thông số từ 1% → 100% đồng thời ProgressBar cũng được thực thi theo (xem hình trên)
- Khi đạt 100% chương trình tự động đóng.
- Thử cố gắng chọn nút Start thật nhiều lần, quan sát và giải thích hiện tượng xảy ra.

Hướng dẫn:

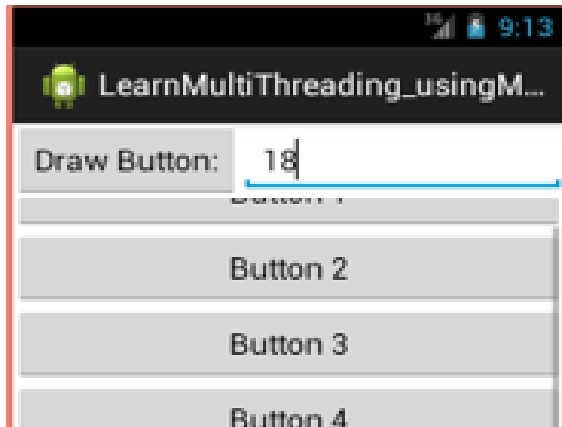
Bài tập 2:

Mục đích:

- Thực hành và hiểu được kỹ thuật lập trình Đa tiến trình
 - o Cách tiếp cận bằng Handler
 - o Cách tiếp cận bằng AsyncTask

Yêu cầu:

- Viết chương trình cho phép vẽ các Button tự động lúc Runtime:



- Yêu cầu thực hiện với nhiều cách (thực hiện bài này với 3 cách khác nhau):
 - o Handler → using sendMessage
 - o Handler → using post
 - o Dùng AsyncTask
- Khi người sử dụng nhập số lượng các Button cần vẽ vào EditText và chọn chức năng Draw Button thì cứ 100 miliseconds mỗi Button sẽ được vẽ ra (xem hình), đồng thời viết sự kiện động cho các Button này. Mỗi lần chọn các Button mới vẽ thì dùng Toast Thông báo là chẵn hay lẻ.
- Chú ý nên dùng ScrollView để chứa các Button vì có thể số lượng Button lớn, vượt quá chiều dài của màn hình

Hướng dẫn:

```
public void drawButton(Message msg) {
    Button btn=new Button(this);
    btn.setText("Button "+msg.arg1);
    btn.setWidth(100);
    btn.setHeight(30);
    btn.setId(113+msg.arg1);

    //gán sự kiện ở đây nữa

    llbutton.addView(btn);
}
```

- **llbutton** là một View nào đó (do logic lập trình của mỗi người chọn)
- gán sự kiện giống như cách viết bình thường

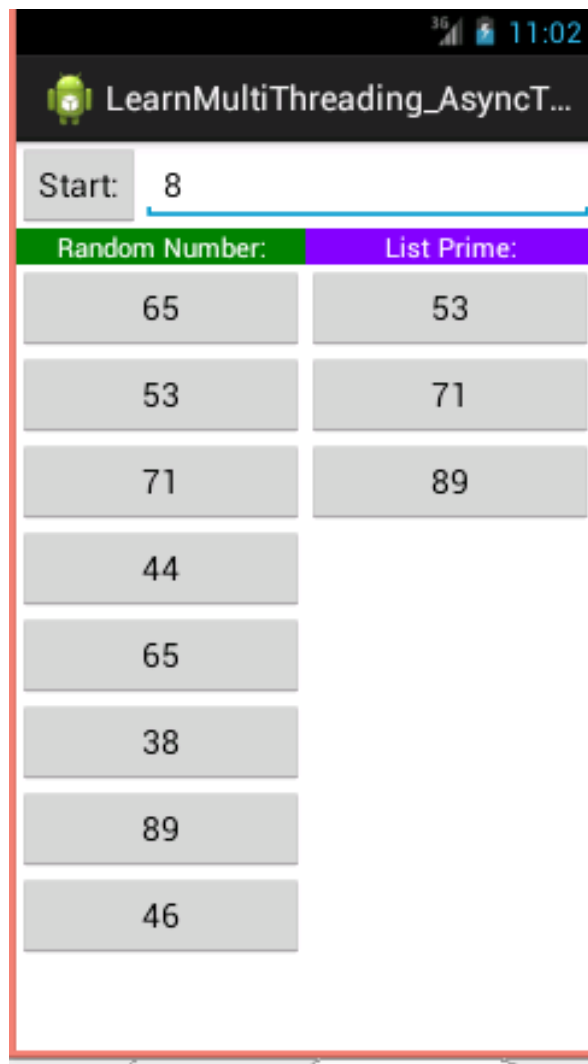
Bài tập 3 (*):

Mục đích:

- Thực hành và hiểu được kỹ thuật lập trình Đa tiến trình
 - o Cách tiếp cận bằng Handler
 - o Cách tiếp cận bằng AsyncTask
 - o Kết hợp nhuần nhuyễn giữa Handler và AsyncTask

Yêu cầu:

- Kết hợp Handler và AsyncTask, viết chương trình như hình dưới đây:
 - o Màn hình được chia ra làm 2 Layout (suy nghĩ cách dùng Layout cho tối ưu),
 - o Layout 1 vẽ các Button với các số ngẫu nhiên
 - o Layout 2 vẽ các Button là các số nguyên tố (chú ý rằng các số nguyên tố này là kết quả sau khi AsyncTask thực hiện công việc vẽ các Button ngẫu nhiên ở Layout 1)



Hướng dẫn:

- Xem hướng dẫn ở slide 31 → 37 phân xử lý đa tiến trình và dịch vụ

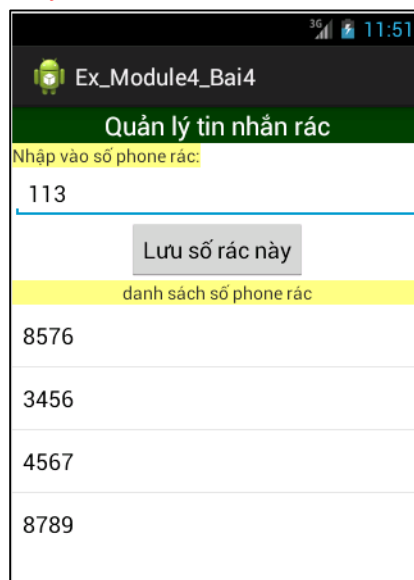
Bài tập 4:

Mục đích:

- Thực hành và hiểu về Intent filter, Broadcast Receiver
 - o Cách khai báo và sử dụng Intent filter
 - o Cách đăng ký/ gỡ bỏ Broadcast Receiver
 - o Cách xử lý Broadcast Receiver trong coding
 - o Cách xử lý Broadcast Receiver trong Manifest XML

Yêu cầu:

- Viết ứng dụng Android xử lý tin nhắn rác.
- Hiện nay rất nhiều tin nhắn quảng cáo gửi vô tội vạ, cái Alô của bạn phải nhận rất nhiều tin nhắn như thế gây phiền hà và mất thời gian cho bạn.
- Hãy viết chương trình để giải quyết vấn đề này, yêu cầu như sau:
 - o Chương trình có giao diện thêm, sửa, xóa các đầu số mà bạn cho là đầu số rác. Như vậy nên dùng SQLite hoặc XML để lưu các đầu số rác.
 - o Chương trình phải tự động nhận tin nhắn và phát hiện ra được tin nhắn nào là gửi đi từ các đầu số rác đó.
 - o Nếu phát hiện ra đó là đầu số rác thì xóa nó khỏi inbox của messaging đồng thời phát ra Audio là “đây là tin nhắn rác, đã bị xử” để người sử dụng không phải mở điện thoại lên xem.
- Yêu cầu viết theo 2 cách (tức là bài tập này làm thành 2 Project khác nhau):
 - o Đăng ký Broadcast Receiver để nhận tin nhắn tới trong coding
 - o Đăng ký Broadcast Receiver để nhận tin nhắn tới trong Manifest XML
- Sau khi làm xong theo 2 cách trên, hãy so sánh sự khác biệt giữa chúng. Và kết luận chúng ta nên chọn cách nào?
- **Chương trình phải cài đặt và chạy được trên điện thoại Android thật.**



- Khi nhấn thật lâu vào 1 đầu số trong ListView thì số này sẽ tự động Remove khỏi CSDL



Hướng dẫn:

- Phải sử dụng 2 permission sau để thao tác với SMS, Audio sẽ xem trong Multimedia
`<uses-permission android:name="android.permission.RECEIVE_SMS" />`
`<uses-permission android:name="android.permission.WRITE_SMS" />`
- Để đăng ký Broadcast Receiver trong Manifest:
`<receiver android:name="MySmsReceiver" >`
`<intent-filter>`
`<action android:name="`
`android.provider.Telephony.SMS_RECEIVED" />`
`</intent-filter>`
`</receiver>`
- Với **MySmsReceiver** là class ta tạo ra kế thừa từ **BroadcastReceiver**
- Cách broadcast một SMS đã hướng dẫn trong Slide 52→60 phần xử lý đa tiến trình
- Để mở một tập tin Audio, xem hướng dẫn trong Multimedia APIs (phần kế tiếp)
- Cách xóa một tin nhắn trong điện thoại như sau (dựa vào đây để truy vấn danh sách đầu số rác trong CSDL để xử lý), dưới đây có 2 ví dụ về xóa tin nhắn:
 - Xóa toàn bộ tin nhắn do 1 đầu số nào đó gửi tới (tức là xóa luôn trong inbox cũ)

```
private void XoaToanBoSms(Context con,
                           SmsMessage[] msgs,
                           String phoneNumber)
{
    try
    {
        for (int i=0; i<msgs.length; i++)
        {
            if(msgs[i].getOriginatingAddress()
                .equals(phoneNumber)==false)
                continue;
            con.getContentResolver()
                .delete(
                    Uri.parse("content://sms"),
                    "address=?",
                    new String[]{phoneNumber}
                );
            Toast.makeText(con, "Xóa rùi đó "+
                msgs[i].getOriginatingAddress(),
                Toast.LENGTH_SHORT).show();
        }
    }
    catch (Exception ex)
    {
        Toast.makeText(con, "Error: " + ex, Toast.LENGTH_SHORT).show();
    }
}
```



}

- Xóa tin nhắn mới gửi tới tức thì do 1 đầu số nào đó gửi tới (chỉ xóa tin mới nhất)

```
private void XoaSmsMoiNhat(Context con,
                           SmsMessage[] msgs,
                           String phoneNumber)
{
    try
    {
        for (int i=0; i<msgs.length; i++)
        {
            if(msgs[i].getOriginatingAddress()
                .equals(phoneNumber)==false)
                continue;
            con.getContentResolver()
                .delete(
                    Uri.parse("content://sms"),
                    "address=? and date=?",
                    new String[]
                    { phoneNumber,
                      String.valueOf(
                        msgs[i].getTimestampMillis()
                      )
                    }
                );
            Toast.makeText(con, " Xóa rùi đó "+
                           msgs[i].getOriginatingAddress(),
                           Toast.LENGTH_SHORT).show();
        }
    }
    catch (Exception ex)
    {
        Toast.makeText(con, "Error: " + ex, Toast.LENGTH_SHORT).show();
    }
}
```

- Chú ý là SmsMessage[] **msgs** , và Context **con** được lấy từ hàm onReceive của class **MySmsReceiver**
- Sinh viên dựa vào 2 hàm trên để ứng dụng vào chương trình, trong quá trình vòng lặp duyệt mảng các tin nhắn gửi đến, đem nó so sánh với các số phone Rác trong cơ sở dữ liệu. Nếu như tồn tại giống nhau thì tiến hành Xóa, sau khi xóa thành công thì dùng **Implicit Intent** để mở một tập tin Audio như yêu cầu.

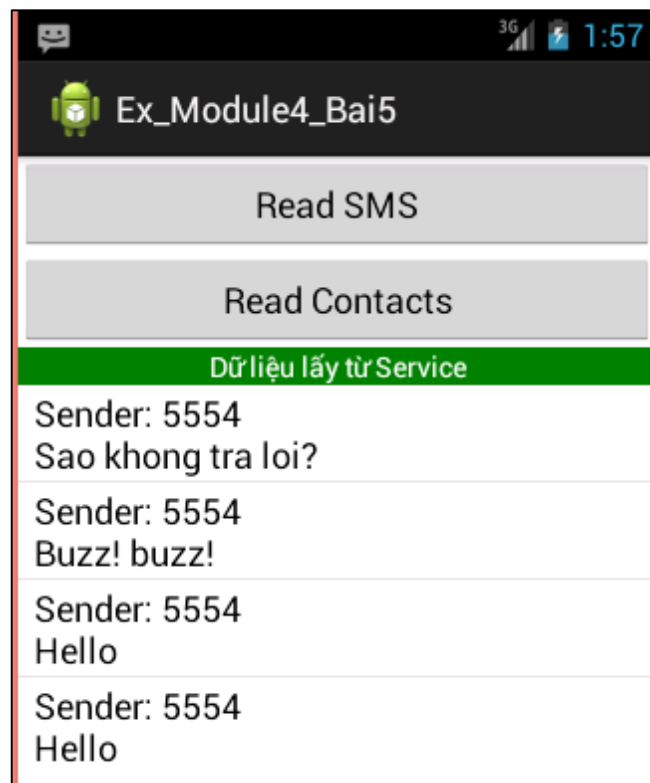
Bài tập 5:

Mục đích:

- Thực hành và hiểu được Android Service
 - Hiểu được cách tạo và khởi động Service
 - Hiểu được cách giao tiếp với service thông qua:
 - Broadcast
 - Binder
 - Messenger

Yêu cầu:

- Bài tập này viết theo 3 cách: Broadcast, Binder, Messenger (nên tạo 3 Project riêng)
- Hãy viết một dịch vụ cho phép thực hiện 2 công việc:
 - Trả về danh sách các tin nhắn trong inbox (cứ 100 miliseconds trả về 1 tin nhắn)
 - Trả về danh sách các Contacts trong điện thoại (trả về 1 lần hết các danh bạ)
- Như vậy service phải kết hợp với đa tiến trình để gửi thông tin cho ứng dụng (yêu cầu dùng AsyncTask)
- Cài đặt và chạy được trên điện thoại Android thật
- Thiết kế một ứng dụng để triệu gọi dịch vụ trên, xem giao diện bên dưới:



Hướng dẫn:

- Xem slide hướng dẫn 76→81 phân xử lý đa tiến trình và dịch vụ
- Uri uri=Uri.parse("content://sms/inbox"); → uri để đọc tin nhắn SMS trong inbox



- Uri uri=Uri.parse("<content://contacts/people>"); → uri để đọc danh bạ trong điện thoại

Bài tập 6:

Mục đích:

- Thực hành và hiểu được Android Service
 - o Hiểu được cách tạo và khởi động Service
 - o Hiểu được cách giao tiếp với service thông qua:
 - Broadcast
 - Binder
 - Messenger

Yêu cầu:

- Nâng cấp bài tập 5 ở trên:
- Yêu cầu Service cung cấp thêm 2 chức năng
 - o Trả về danh sách các tin nhắn trong inbox (cứ 100 miliseconds trả về 1 tin nhắn)
 - o Trả về danh sách các Contacts trong điện thoại (trả về 1 lần hết các danh bạ)
 - o **Xóa một tin nhắn trong inbox**
 - o **Xóa một contact trong danh bạ**
- Yêu cầu đối với giao diện:
 - o Khi bấm thật lâu vào một tin nhắn thì yêu cầu Service xóa tin nhắn này
 - o Khi bấm thật lâu vào 1 contact thì yêu cầu Service xóa contact này

Hướng dẫn:

- Làm được bài 5 thì phải làm được bài 6.

Bài tập 7:

Mục đích:

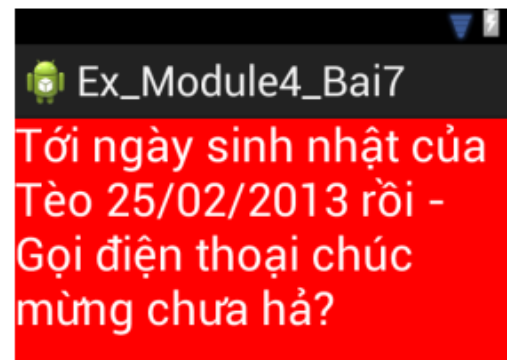
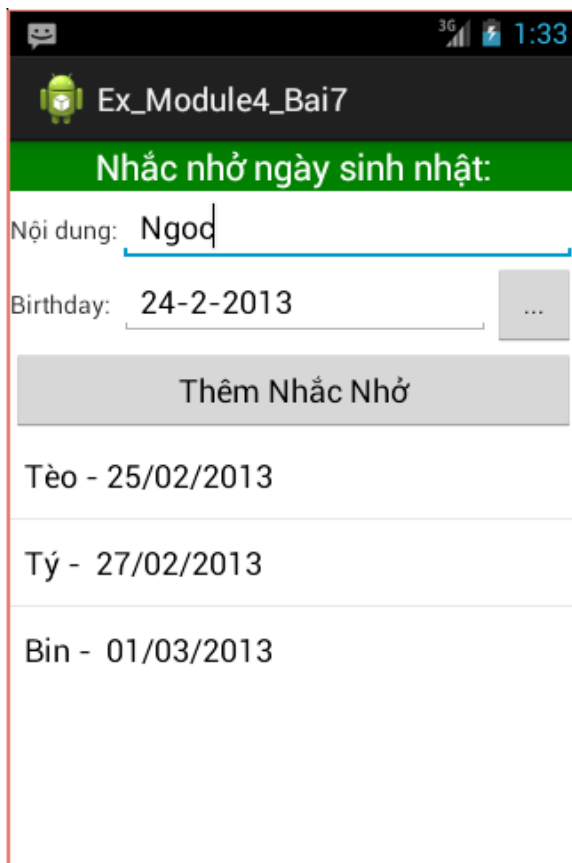
- Thực hành và hiểu được System Android Service
 - o Hiểu được PendingIntent
 - o Tạo và sử dụng được AlarmManager (dùng getBroadcast và getService)
 - o Tạo và sử dụng được NotificationManager

Yêu cầu:

- Dùng **AlarmManager** để viết chương trình nhắc nhở ngày sinh nhật.
 - o Hiện nay vì phải vật lộn với mọi lo toan cuộc sống, làm cho con người chúng ta lúc nào cũng bận rộn, chúng ta lúc nào cũng chỉ có công việc và công việc. Chúng ta quên đi những ngày kỷ niệm quan trọng của bố mẹ, anh em, bạn bè, đối tác.... Điện thoại thì sẵn trong người những ít gửi bắt máy lên hỏi han một ai đó, hay là chỉ nhắn những lời chúc sinh nhật đầm ấm tới người thân cũng không nhớ. Nó

vô tình làm cho chúng ta tách biệt với cuộc sống êm đẹp hàng ngày, nó làm cho chúng ta trở lên khô khan tới mức vô tình.

- Vì lý do này, yêu cầu các sinh viên viết phần mềm nhắc nhở ngày sinh nhật. Chương trình phải có giao diện cho phép nhập vào danh sách các Nội dung nhắc nhở với những ngày nhắc quy định trên giao diện.
- Khi đến đúng ngày đó, giờ đó thì điện thoại phải tự động thông báo. Hình thức thông báo tùy sinh viên chọn lựa, có thể:
 - Mở một audio nói tới ngày sinh nhật
 - Hoặc tự động mở một Activity lên màn hình thông báo là tới ngày sinh nhật
 - Hoặc hình thức nào đó tùy
- Chương trình phải cài đặt và chạy được trên điện thoại Android thiết
- Xem giao diện bên dưới:



- Khi đúng tới ngày sinh nhật, hiển thị Activity với màn hình màu đỏ như trên, sinh viên có thể tùy chọn hình thức Nhắc Nhở khác
- Sử dụng DatePickerDialog để nhập Birthday
- Khi chọn Thêm nhắc nhở sẽ đưa thông tin xuống ListView
- Khi nhấn rất lâu vào 1 nhắc nhở nào đó trong ListView thì xóa nó ra khỏi ListView

Hướng dẫn:

- Xem slide hướng dẫn sử dụng AlarmManager : slide 83→90

- Sử dụng phương thức set của class Calendar để thiết lập thời gian active
`cal.set(2013, 2, 25, 0, 0, 0);`//năm, tháng, ngày, giờ, phút, giây

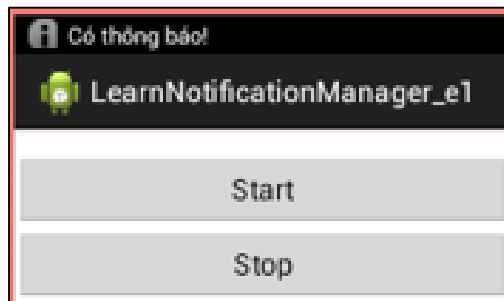
Bài tập 8:

Mục đích:

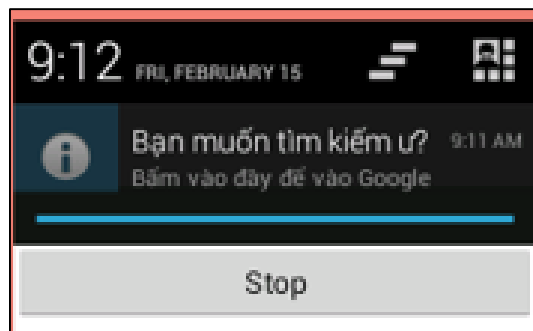
- Thực hành và hiểu được System Android Service
 - o Hiểu được PendingIntent
 - o Tạo và sử dụng được AlarmManager (dùng getBroadcast và getService)
 - o Tạo và sử dụng được NotificationManager

Yêu cầu:

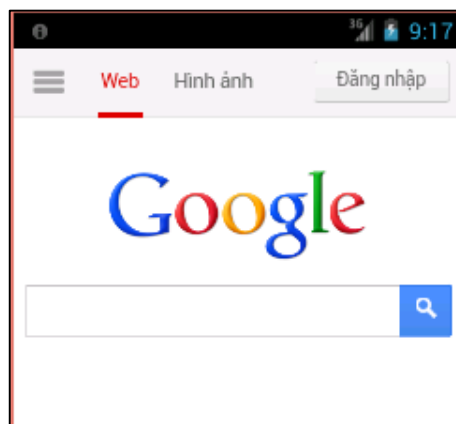
- Dùng **NotificationManager** để viết ứng dụng:
- Hãy viết ứng dụng đơn giản có giao diện như bên dưới:



- Khi chọn Start, ứng dụng sẽ hiển thị Notification thông báo như bên dưới



- Kéo Notification xuống, và chọn nó, màn hình tìm kiếm Google sẽ xuất hiện như dưới:



- Chọn Stop để ngừng Notification

Hướng dẫn:

- Xem hướng dẫn slide 97→99

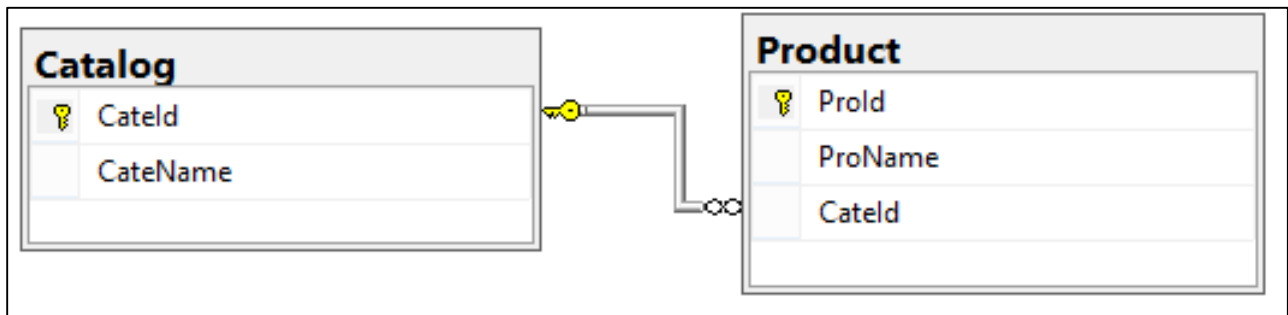
Bài tập 9 (*):

Mục đích:

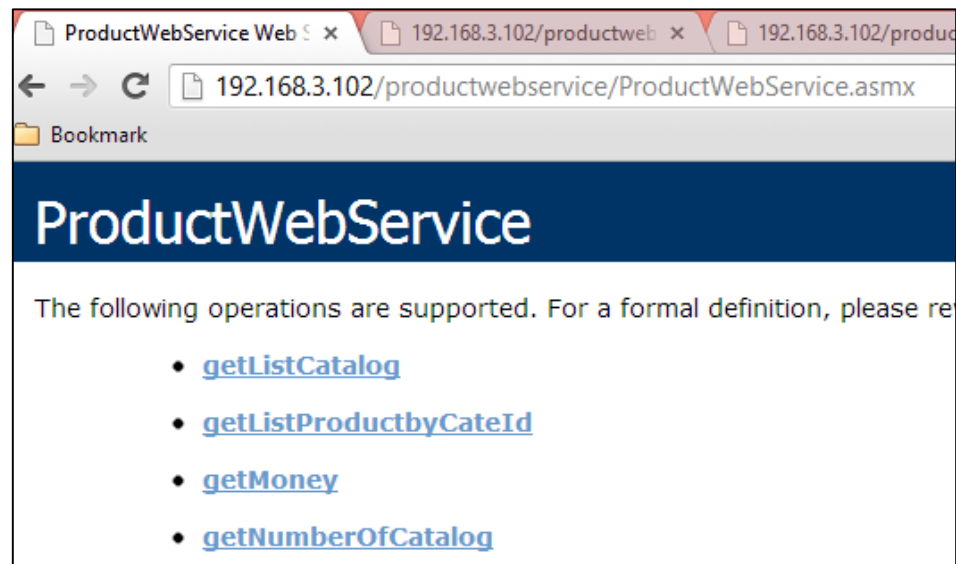
- Thực hành và hiểu được cách tương tác dữ liệu giữa Android và .Net Webservice
 - o Biết tạo .Net Webservice
 - o Biết cách cấu hình IIS
 - o Tương tác Android với .Net Webservice thông qua KSOAP API

Yêu cầu:

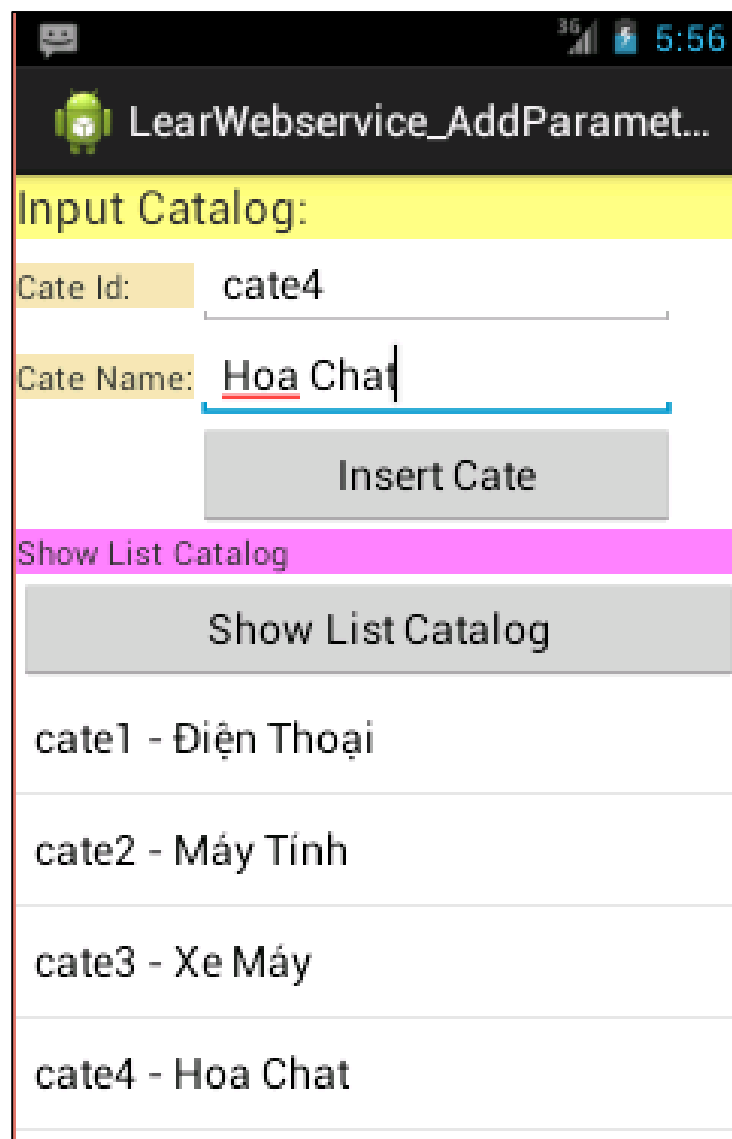
- Hãy xây dựng .Net Webservice với mô tả sau (CSDL là Sqlserver):



- Webservice này sẽ cung cấp các dịch vụ:
 - o Đếm tổng số danh mục
 - o Trả về danh sách các danh mục
 - o Thêm danh mục vào CSDL
 - o Xóa danh mục khỏi CSDL
 - o Trả về danh sách các sản phẩm theo một mã danh mục nào đó
 - o Tính thành tiền, với 2 đối số truyền vào là số lượng và đơn giá
 - o Và một số dịch vụ phát sinh khác khi làm ...
- Cấu hình IIS để chạy Webservice này



- Xây dựng phần mềm Android để tương tác với Webservice trên thông qua KSOAP API:



- Khi chọn Insert Cate: chương trình sẽ gọi dịch vụ thêm 1 danh mục vào CSDL (chú ý chỉ được nhập không quá 10 danh mục)
- Khi chọn Show List Catalog: chương trình sẽ hiển thị toàn bộ danh sách danh mục vào ListView như hình.
- Khi chọn thật lâu một danh mục nào đó trên ListView thì cho phép xóa danh mục này, chú ý là nếu kiểm tra thấy danh mục đã có sản phẩm rồi thì dùng AlertDialog hỏi xem người sử dụng biết và hỏi xem họ có chắc chắn muốn xóa hay không.

Hướng dẫn:

- Xem slide hướng dẫn 107→114
- Bài tập này rất quan trọng, khi đi làm thực tế chúng ta hay phải tương tác tới một Webservice nào đó
-

Bài tập 10:

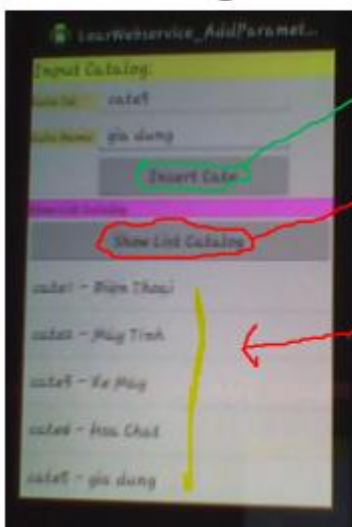
Mục đích:

- Thực hành và biết được cách tương tác giữa thiết bị Android với Máy chủ (Laptop)
 - o Thiết lập được kết nối giữa điện thoại và Laptop
 - o Tương tác Android với .Net Webservice thông qua KSOAP API

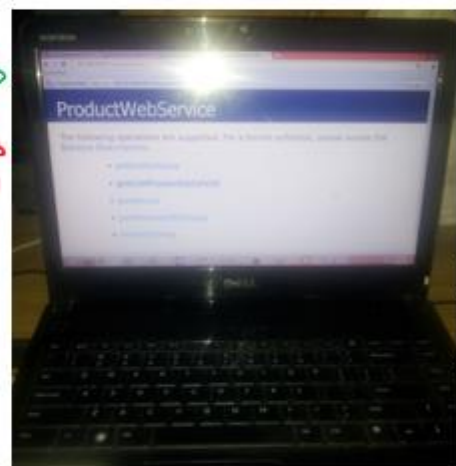
Yêu cầu:

- Yêu cầu tương tác được giữa thiết bị Android thiết với Laptop

Here is my real Phone:
SamSung S2



Database + webservice
in My Laptop



Insert Cate

Get list Cate



Hướng dẫn:

- Tự triển khai:



- Gợi ý dùng chức năng **Portable wifi hotspot** của thiết bị
- Hoặc một số phần mềm giúp Laptop phát WIFI (Win7, Win8)
- Chỉ cần xác định được IP của máy Laptop, kết nối theo IP này

Module 5: Networking APIs và Multimedia APIs

Nội dung kiến thức thực hành:

- Thực hành cách xử lý Strict Mode: Dùng permitAll() và AsyncTask
- Thực hành truy suất dữ liệu thông qua giao thức HTTP: đọc Text File, đọc Binary File, cách lấy Network Status...
- Thực hành về Camera: cách xử lý hình ảnh từ Camera, thao tác với các Parameters, cách thức Zoomin, cách thức chia sẻ hình ảnh, thiết lập màn hình nền từ Camera...
- Thực hành về Video: Cách thức lưu trữ Video, cách play Video, làm việc với nhận dạng khuôn mặt...
- Thực hành về Audio: Cách thức lưu trữ audio, cách play audio, cách tìm kiếm, chia sẻ audio và thiết lập nhạc chuông điện thoại.
- **Nếu máy không có internet thì tự cấu hình IIS local để sử dụng website bình thường**

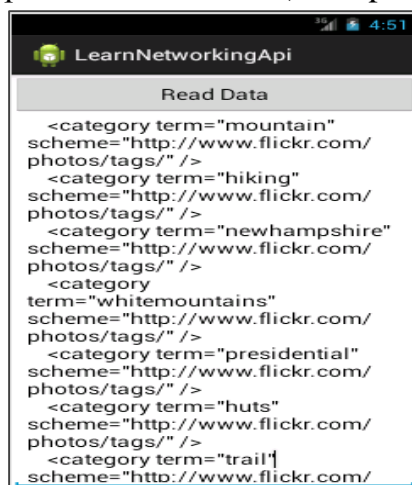
Bài 1:

Mục đích:

- Thực hành và hiểu được cách đọc 1 Text file từ internet
 - o Biết xử lý Strict mode
 - permitAll()
 - AsyncTask
 - Phân biệt được sự khác biệt giữa 2 cách này, nên chọn cách nào
 - o Sử dụng được một số class hỗ trợ đọc Text file từ internet

Yêu cầu:

- Viết chương trình cho phép đọc một Text file (URL path), giao diện như hình bên dưới:



- Cách 1- Kết nối trong Main Thread dùng permitAll()
- Cách 2- Kết nối ở một Thread khác (dùng AsyncTask)
- Giải thích sự khác biệt giữa 2 cách và kết luận nên sử dụng cách nào

Hướng dẫn:

- Xem slide hướng dẫn 10→12 phần Networking APIs và Multimedia APIs

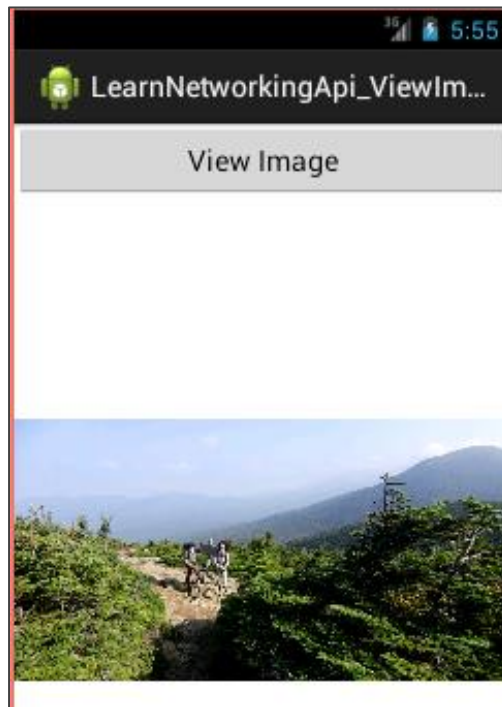
Bài tập 2:

Mục đích:

- Thực hành và hiểu được cách đọc 1 Binary file từ internet
 - o Biết xử lý Strict mode
 - `permitAll()`
 - `AsyncTask`
 - Phân biệt được sự khác biệt giữa 2 cách này, nên chọn cách nào
 - o Sử dụng được một số class hỗ trợ đọc Binary file từ internet

Yêu cầu:

- Viết ứng dụng cho phép hiển thị một hình ảnh bất kỳ từ internet như sau:



- Yêu cầu dùng `AsyncTask` để load hình ảnh về điện thoại

Hướng dẫn:

- Xem slide hướng dẫn từ 14→16 phần Networking APIs và Multimedia APIs

Bài tập 3:

Mục đích:

- Hiểu thêm cách đọc 1 Binary file từ internet

Yêu cầu:

- Cải tiến bài tập 2 bằng cách thêm 1 Progressbar để load hình ảnh về, vì bài tập 2 thường gây ra tình trạng giống như “Đơ máy” khi mà tập tin quá nặng hoặc internet chậm

Hướng dẫn:

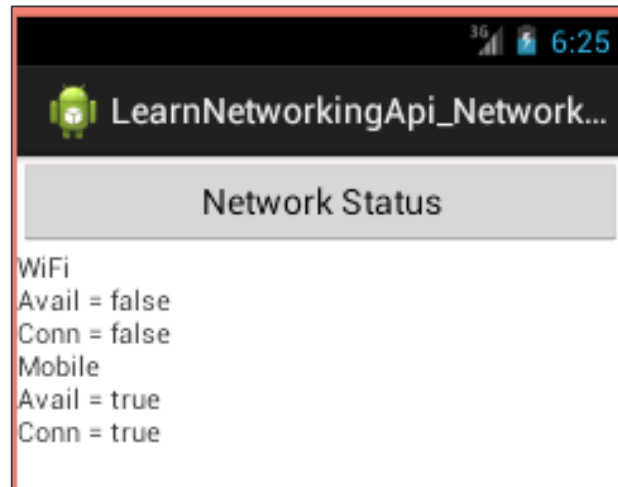
Bài tập 4:

Mục đích:

- Thực hành và hiểu được cách kiểm tra trạng thái mạng của thiết bị Android:
 - o Hiểu và sử dụng được ConnectivityManager

Yêu cầu:

- Viết chương trình cho phép kiểm tra tình trạng mạng của thiết bị di động Android, giao diện như hình bên dưới:



- Yêu cầu dùng AsyncTask

Hướng dẫn:

- Xem slide hướng dẫn 18→21 phần Networking APIs và Multimedia APIs
- Ứng dụng bài tập này để sửa lại bài tập 1, 2, 3, . Kiểm tra nếu máy chưa có Wifi hoặc 3G thì yêu cầu mở Wifi hoặc đăng ký 3G trước khi thực hiện kết nối internet.

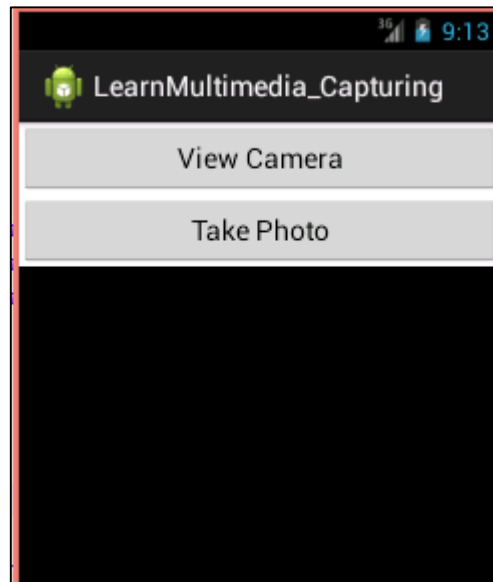
Bài tập 5:

Mục đích:

- Thực hành và hiểu về Camera:
 - o Cách chụp hình ảnh từ Camera
 - o Cách sử dụng Camera Parameter
 - o Chia sẻ hình ảnh
 - o Thiết lập màn hình điện thoại từ hình ảnh trong Camera
- Không thử được trên emulator, phải thử trên điện thoại Android thật

Yêu cầu:

- Viết chương trình cho phép kích hoạt Camera và chụp một hình ảnh từ Camera rồi lưu vào SD Card, xem giao diện minh họa:



- Khi chọn View Camera, sẽ cho phép kích hoạt Camera
- Khi chọn take Photo sẽ cho phép chụp hình ảnh hiện tại trong Camera và lưu vào thẻ.

Hướng dẫn:

- Xem slide hướng dẫn 27→34 phần Networking APIs và Multimedia APIs

Bài tập 6 (*):

Mục đích:

- Thực hành và hiểu về Camera:
 - o Cách chụp hình ảnh từ Camera
 - o Cách sử dụng Camera Parameter
 - o Chia sẻ hình ảnh
 - o Thiết lập màn hình điện thoại từ hình ảnh trong Camera
- Không thử được trên emulator, phải thử trên điện thoại Android thật

Yêu cầu:

- Cải tiến bài tập 5, thêm các chức năng :
 - o Zoom – in, Zoom – out
 - o Xoay Camera
 - o Chia sẻ hình ảnh
 - o Thiết lập màn hình nền
 - o Chức năng bật / tắt đèn Flash (thay cho đèn Pin khi buổi tối Cúp Điện)
- Giao diện sinh viên tự sáng tạo sao cho dễ sử dụng

Hướng dẫn:

- Xem cách tạo đèn flash trong Android, dựa vào đây để đưa vào bài tập 6.



- Trong Manifest XML cấu hình permission:

```
<uses-permission android:name="android.permission.CAMERA" />
```

```
<uses-feature android:name="android.hardware.camera" />
```

- Coding trong Activity:

```
public class MainActivity extends Activity {
    //Button để bật đèn Flash
    Button btnflashlight;
    //Biến để chờ kiểm tra đèn Flash đang bật hay tắt
    private boolean isFlashOn = false;
    //tạo đối tượng camera để truy xuất Flash light
    private Camera camera;
    //Parameter để thiết lập cho camera
    private Parameters params;
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        btnflashlight=(Button)
            findViewById(R.id.btnturnonoroffflash);
        btnflashlight.setOnClickListener(new
            View.OnClickListener() {
                public void onClick(View arg0) {
                    doTouchtoOn();
                }
            });
    }
    public void doTouchtoOn()
    {
        //Nếu đèn flash đang mở
        if (isFlashOn) {
            //Thiết lập flashmode off cho parameter
            params.setFlashMode(Parameters.FLASH_MODE_OFF);
            //thiết lập parameter cho camera
            camera.setParameters(params);
        }
    }
}
```



```

        //gán cờ kiểm tra bằng flase
        isFlashOn = false;
        //Thiết lập chữ cho button là Bật đèn
        btnflashlight.setText("Bật đèn");
    }
    //Nếu đèn flash đang đó
    else {
        //thiết lập flashmode on cho parameter
        params.setFlashMode(Parameters.FLASH_MODE_TORCH);
        //thiết lập parameter cho camera
        camera.setParameters(params);
        //gán cờ kiểm tra bằng true
        isFlashOn = true;
        //Thiết lập chữ cho Button là Tắt đèn
        btnflashlight.setText("Tắt đèn");
    }
}

protected void onResume() {
    super.onResume();
    PackageManager pm = getPackageManager();
    if(!pm.hasSystemFeature
        (PackageManager.FEATURE_CAMERA)) {
        Toast.makeText(getApplicationContext(),
            "Máy không có Camera!",
            Toast.LENGTH_SHORT).show();
        btnflashlight.setEnabled(false);
        return;
    }
    camera = Camera.open();
    if(camera!=null)
        params = camera.getParameters();
}

protected void onStop() {
    super.onStop();
    if (camera != null) {
        camera.release();
    }
}
}

```

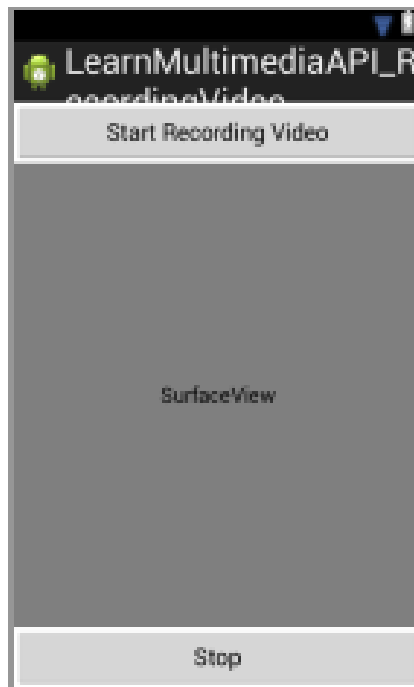
Bài tập 7:

Mục đích:

- Thực hành và hiểu về Video
 - o Cách ghi Video
 - o Cách chơi Video
 - o Biết chức năng nhận diện khuôn mặt trong Android

Yêu cầu:

- Viết ứng dụng cho phép ghi và mở Video, xem giao diện:



- Khi chọn Start Recording Video, chương trình khởi động MediaRecord lên SurfaceView và tự động quay phim lưu vào SD Card
- Khi chọn Stop thì sẽ ngừng quay phim và mở lại phim vừa được ghi theo 2 cách:
 - o Dùng Implicit Intent để mở
 - o Dùng VideoView, MediaController để mở

Hướng dẫn:

- Xem slide hướng dẫn 41→48 phần Networking APIs và Multimedia APIs
- Trong Manifest thêm các permission:

```
<uses-permission android:name=
    "android.permission.WRITE_EXTERNAL_STORAGE"/>
<uses-permission android:name=
    "android.permission.READ_EXTERNAL_STORAGE"/>
<uses-permission android:name=
    "android.permission.CAMERA"/>
```



Bài tập 8:

Mục đích:

- Thực hành và hiểu về Audio
 - o Cách ghi Audio
 - o Cách mở Audio
 - o Cách tìm kiếm Audio
 - o Cách thiết lập nhạc chuông điện thoại

Yêu cầu:

- Hãy viết chương trình thực hiện các chức năng sau:
 - o Cho phép ghi âm cuộc nói chuyện (lưu vào SD Card)
 - o Cho phép mở các tập tin ghi âm
 - o Hỗ trợ chức năng tìm kiếm theo ngày, theo tên tập tin
 - o Cho phép lấy 1 tập tin ghi âm nào đó làm nhạc chuông điện thoại
- Giao diện sinh viên tự thiết kế sao cho dễ sử dụng
- Yêu cầu cài đặt và chạy được trên điện thoại Android thiết

Hướng dẫn:

- Xem slide hướng dẫn 51→60 phần Networking APIs và MultiMedia APIs

Bài tập 9 (*):

Mục đích:

- Tổng hợp chức năng trong Networking và MultiMedia

Yêu cầu:

- Hãy giả lập chương trình Music Player của Android
 - o Cho phép thực thi được các tập tin Audio, Video
 - o Thực thi được các tập tin Audio, Video qua mạng internet
 - o Và một số tính năng cần thiết khác

Hướng dẫn:

- Xem cách thiết kế chương trình Music Player của Android
- Tìm kiếm trên mạng
- Dành cho các sinh viên khá giỏi



Module 6: Telephony APIs và Location Base Service APIs

Nội dung kiến thức thực hành:

- Thực hành về các tiện ích của Telephony APIs: Kiểm tra Phone State, Service State, làm việc với Phone Number...
- Thực hành về SMS: Cách cấp quyền để nhận và gửi SMS, cách gửi/nhận SMS, cách đọc các tin nhắn trong Inbox, đăng ký BroadCast Receiver để thao tác với các SMS
- Thực hành về cách tạo cuộc gọi, nhận cuộc gọi: cách cấp quyền, sử dụng ACTION_CALL, ACTION_DIAL, ACTION_ANSWER, đăng ký BroadCast Receiver để tương tác với các cuộc gọi đến.
- Thực hành về Call logs: Xem lịch sử cuộc gọi, xóa lịch sử cuộc gọi...
- Thực hành về LBS APIs: Tìm hiểu GPS, Geocoding và Mapping location, thực hành Mapping Intent và Mapping View

Bài 1:

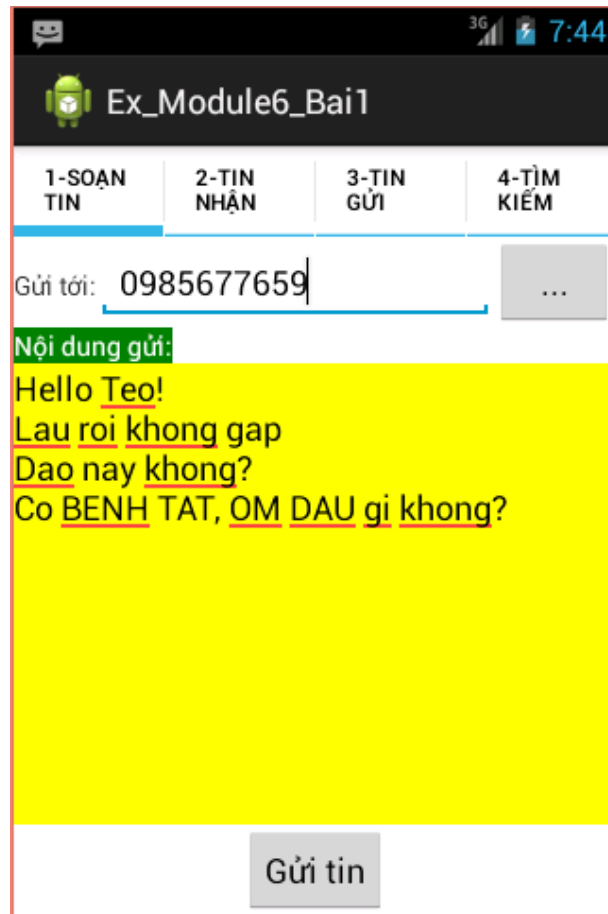
Mục đích:

- Thực hành và hiểu về Telephony APIs:
 - o Cách gửi một tin nhắn SMS
 - o Cách nhận một tin nhắn SMS
 - o Cách thao tác với tin nhắn

Yêu cầu:

- Hãy viết ứng dụng thay thế cho chương trình Messaging của Android:
 - o Cho phép soạn thảo và gửi tin nhắn
 - o Cho phép nhận tin nhắn
 - o Cho phép thao tác tin nhắn:
 - Xóa tin nhắn
 - Tìm kiếm tin nhắn
 - Sắp xếp tin nhắn theo số điện thoại, theo ngày gửi, theo ngày nhận
- Chú ý chương trình chỉ cần lưu các thông số sau của mỗi tin nhắn:
 - o Địa chỉ gửi đến
 - o Nội dung gửi đến
 - o Thời gian gửi đến
 - o Nội dung gửi đi
 - o Thời gian gửi đi
 - o (các thông số khác không cần lưu)
- Dùng SQLite để lưu trữ thông tin của một tin nhắn như mô tả ở trên:
 - o Bảng tin gửi đi (mã tự động, số gửi đi, nội dung gửi đi, thời gian gửi đi)
 - o Bảng tin nhận được (mã tự động, số gửi đến, nội dung gửi đến, thời gian gửi đến)

- Chương trình nên đăng ký BroadCast Receiver trong Manifest XML để tự động nhận các tin nhắn gửi đến và lưu vào cơ sở dữ liệu (cho dù ứng dụng đã bị tắt)
- Xem giao diện minh họa (dùng Tabselector và include các Layout vào đúng vị trí):



- o Có 4 tab như ở trên
- o Chọn nút “...” để lấy danh bạ hoặc tự nhập số vào EditText
- o Khi chọn tab nào thì xử lý theo tab đó, sinh viên hãy thiết kế Layout phù hợp.

Hướng dẫn:

- Học cách gửi 1 tin nhắn slide 13→15 phần Telephony APIs và LBS APIs
- Học cách nhận 1 tin nhắn slide 16→19 chú ý đăng ký BroadCast Receiver trong Manifest
- Cách xóa : xem lại Sqlite cách xóa như thế nào
- Cách tìm kiếm: nghĩ cách tìm kiếm hợp lý
- Cách sắp xếp: xem Sqlite sắp xếp như thế nào, nên dùng ContextMenu để gọi lệnh.
- Chú ý là CSDL này độc lập với Messaging của Android, ta thao tác SMS (xóa, tìm kiếm, sắp xếp) đều nằm trên Sqlite do ta tạo, không được đụng chạm tới SMS của Android. Tức là khi có 1 tin nhắn bất kỳ nào đó gửi tới thì dĩ nhiên Tin nhắn này sẽ được tự động lưu vào 2 nơi:
 - o Đó là Inbox thực tế của Android (Ta không liên quan, tự Android làm)
 - o và Sqlite mà chương trình chúng ta tạo ra→ta thao tác trên cái này

- Các tin gửi đi chương trình chỉ lưu khi người sử dụng dùng phần mềm này để gửi tin nhắn, còn gửi bằng Messaging của Android thì không liên quan.

Bài tập 2:

Mục đích:

- Thực hành và hiểu về Telephony APIs:
 - o Xử lý các trạng thái của điện thoại (idle, in use, ringing)
 - o Cách tạo một cuộc gọi đi
 - o Cách nhận một cuộc gọi tới

Yêu cầu:

- Viết chương trình quản lý Blacklist
- Thiết kế giao diện phù hợp: cho phép thêm, xóa, sửa danh Blacklist
- Dùng Sqlite để lưu danh sách Blacklist
- Khi nào có cuộc gọi đến nếu kiểm tra số gọi đến có tồn tại trong black list thì tiến hành hủy cuộc gọi (nếu được), còn không thì xuất hiện một Activity thông báo.

Hướng dẫn:

- Xem hướng dẫn xử lý cuộc gọi slide 22→25 phần Telephony APIs và LBS APIs

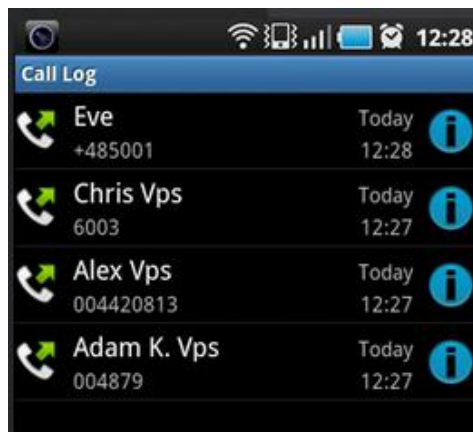
Bài tập 3:

Mục đích:

- Thực hành và hiểu về Telephony APIs:
 - o Thao tác với Call log : xem, xóa.

Yêu cầu:

- Viết chương trình cho phép quản lý Call log
- Thiết kế giao diện cho phép xem danh sách Call log, cho phép xóa call log:



- Khi mở chương trình sẽ tự động đọc danh sách call log vào ListView như trên



- ContextMenu có 2 item: Xóa call log đang chọn và Xóa toàn bộ Call log, mọi thao tác đều dùng AlertDialog để xác nhận xem người dùng có muốn xóa hay không.

Hướng dẫn:

- Tự làm

Bài tập 4 (*):

Mục đích:

- Thực hành và hiểu về Location Base Service APIs
 - o Biết Android Google API
 - o Hiểu và sử dụng Geocoding location
 - o Hiểu và sử dụng Mapping Location: Mapping Intent, Mapping View

Yêu cầu:

- Công ty Bitis cần viết một ứng dụng Android như sau:
- Chương trình cho phép xem danh sách sản phẩm của công ty, cũng như các chính sách khuyến mãi, các thông tin về sản phẩm mới. (Công ty sẽ cung cấp 1 .Net Webservice để cho phép ứng dụng Android xem danh sách sản phẩm, xem thông tin khuyến mãi, sản phẩm mới, ở đây là ta tự làm .Net Webservice thay cho công ty)
- Chương trình đồng thời cho phép cập nhật vị trí các đại lý của công ty và cho phép tìm kiếm các đại lý lân cận của một đại lý nào đó mà ta cần (hoặc tìm các đại lý gần vị trí mà người sử dụng đang đứng)
- Sinh viên có thể dùng Mapping Intent hoặc Mapping View để xử lý tìm địa điểm.

Hướng dẫn:

- Dành cho sinh viên khá giỏi
- Xem hướng dẫn slide 28→44 phần Telephony APIs và LBS APIs
- Chủ yếu tham khảo trên mạng internet



Module 7: Thao tác với thiết bị cảm ứng

Nội dung kiến thức thực hành:

- Thực hành về SensorManager, Sensor, SensorEventListener, SensorEvent: Cách thức đăng ký Sensor, cách thức gỡ bỏ đăng ký, cách lấy Sensor mặc định, cách lấy danh sách Sensors, cách lấy các giá trị trong SensorEvent
- Thực hành về các thiết bị cảm biến cụ thể: cảm biến nhiệt độ, cảm biến ánh sáng, cảm biến áp suất, cảm biến khoảng cách, cảm biến gia tốc, cảm biến con quay hồi chuyển, cảm biến định hướng...
- Thực hành về Pin điện thoại: Cách đăng ký Broadcast Receiver để đọc thông số Pin
- **Phải thử trên điện thoại Android thật**

Bài 1:

Mục đích:

- Thực hành và hiểu về Sensor Temperature:
 - o Cách tạo Sensor
 - o Cách đăng ký Sensor
 - o Cách gỡ bỏ đăng ký
 - o Lấy các thông tin trong SensorEvent

Yêu cầu:

- Viết ứng dụng báo động cháy nổ.
- Chương trình có giao diện để quy định nhiệt độ cao nhất cho phép tại môi trường, khi nhiệt độ môi trường mà vượt qua nhiệt độ cho phép đó thì điện thoại sẽ mở một Audio cảnh báo “Có nguy cơ cháy nổ”

Hướng dẫn:

- Xem slide hướng dẫn 11, 12
- Xem trên internet

Bài tập 2:

Mục đích:

- Thực hành và hiểu về Sensor Light:
 - o Cách tạo Sensor
 - o Cách đăng ký Sensor
 - o Cách gỡ bỏ đăng ký
 - o Lấy các thông tin trong SensorEvent



Yêu cầu:

- Viết chương trình cảnh báo phòng thiếu ánh sáng
- Ứng dụng cho phép thiết lập ánh sáng tối thiểu trong phòng, khi ánh sáng trong phòng mà nhỏ hơn ánh sáng tối thiểu cho phép thì điện thoại sẽ mở Audio “Phòng quá tối, yêu cầu bật đèn”

Hướng dẫn:

- Xem slide hướng dẫn 13, 14
- Xem trên internet

Bài tập 3:

Mục đích:

- Thực hành và hiểu về Sensor Pressure:
 - o Cách tạo Sensor
 - o Cách đăng ký Sensor
 - o Cách gỡ bỏ đăng ký
 - o Lấy các thông tin trong SensorEvent

Yêu cầu:

- Viết chương trình cảnh báo Áp suất môi trường quá lớn
- Ứng dụng cho phép thiết lập Áp suất tối đa có thể chấp nhận, khi môi trường có áp suất vượt quá thông số cho phép thì điện thoại sẽ mở Audio “áp suất quá lớn, nguy hiểm!”

Hướng dẫn:

- Xem slide hướng dẫn 15, 16
- Xem trên internet

Bài tập 4:

Mục đích:

- Thực hành và hiểu về Sensor Proximity:
 - o Cách tạo Sensor
 - o Cách đăng ký Sensor
 - o Cách gỡ bỏ đăng ký
 - o Lấy các thông tin trong SensorEvent

Yêu cầu:

- Khi gọi điện thoại, ta thường áp sát điện thoại vào tai để nghe; nên đôi khi màn hình cảm ứng sẽ chạm vào tai làm hủy cuộc gọi mà ta không mong muốn.

- Do đó cần viết ứng dụng cho phép thiết lập khoảng cách tối thiểu giữa điện thoại và đối tượng, khi nào vượt quá giới hạn tối thiểu thì tự động tắt màn hình điện thoại.

Hướng dẫn:

- Xem slide hướng dẫn 17→19
- Xem trên internet

Bài tập 5:

Mục đích:

- Thực hành và hiểu về Sensor Accelerometer:
 - o Cách tạo Sensor
 - o Cách đăng ký Sensor
 - o Cách gỡ bỏ đăng ký
 - o Lấy các thông tin trong SensorEvent

Yêu cầu:

- Viết ứng dụng nghe nhạc có hỗ trợ chức năng Lắc điện thoại để chuyển bài hát:
 - o Nếu điện thoại lắc qua trái thì mở bài hát đằng trước
 - o Nếu điện thoại lắc qua phải thì mở bài hát đằng sau

Hướng dẫn:

- Xem slide hướng dẫn 20→23
- Xem trên internet

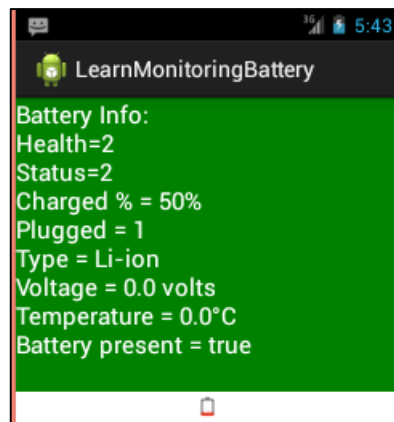
Bài tập 6:

Mục đích:

- Thực hành và hiểu về Pin điện thoại:
 - o Lấy các thông tin trong Pin

Yêu cầu:

- Viết chương trình để lấy các thông số của Pin điện thoại



Hướng dẫn:



- Xem slide hướng dẫn 30→34
- Xem trên internet

Module 8: Danh sách đề tài môn học

Đề tài 1:

- Viết chương trình tra cứu từ điển Anh Việt – Việt Anh

Đề tài 2:

- Viết chương trình đăng ký học phần tín chỉ bằng SMS

Đề tài 3:

- Viết chương trình đăng ký học chứng chỉ tin học A, B bằng SMS

Đề tài 4:

- Viết game bắn bóng sử dụng Sensor

Đề tài 5:

- Viết chương trình quản lý Album ảnh

Đề tài 6:

- Viết chương trình gõ tiếng việt trong Android

Đề tài 7:

- Viết chương trình tìm đường đi

Đề tài 8:

- Viết game caro (chơi với máy)

Đề tài 9:

- Viết game xếp gạch



Đề tài 10:

- Viết chương trình chỉnh sửa hình ảnh trong Android

Đề tài 11:

- Viết chương trình quản lý tài chính cá nhân

Đề tài 12:

- Viết game phi thuyền bằng Sensor

Đề tài 13:

- Mô phỏng trò chơi Line 98 trong Android

Đề tài 14:

- Viết chương trình nhận dạng khuôn mặt và dự đoán trong vòng 20 năm nữa thì khuôn mặt đó sẽ thay đổi như thế nào

Đề tài 15:

- Viết chương trình vẽ đồ thị toán học trong Android