

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:

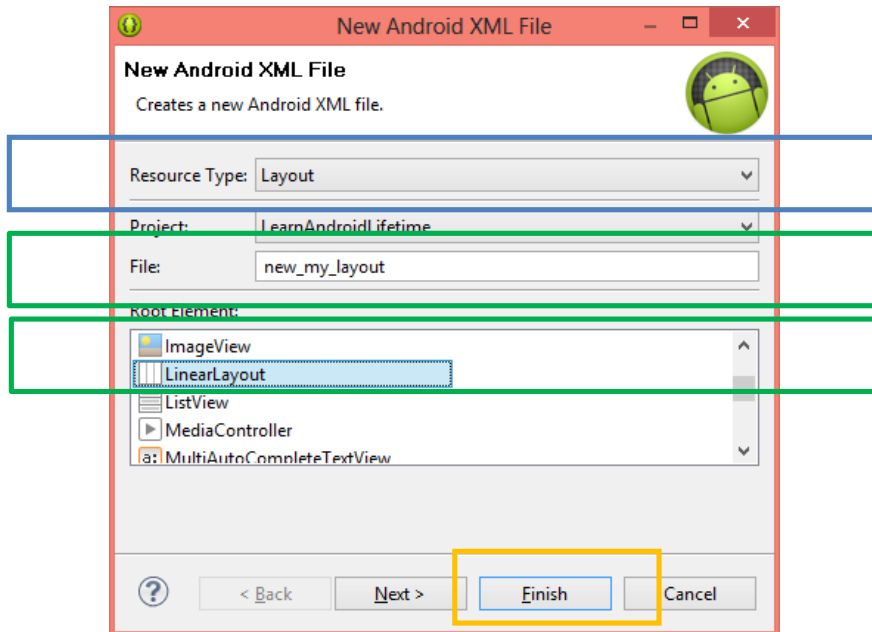
- Tạo và hiệu chỉnh XML Layout
- Sử dụng XML Layout

Yêu cầu:

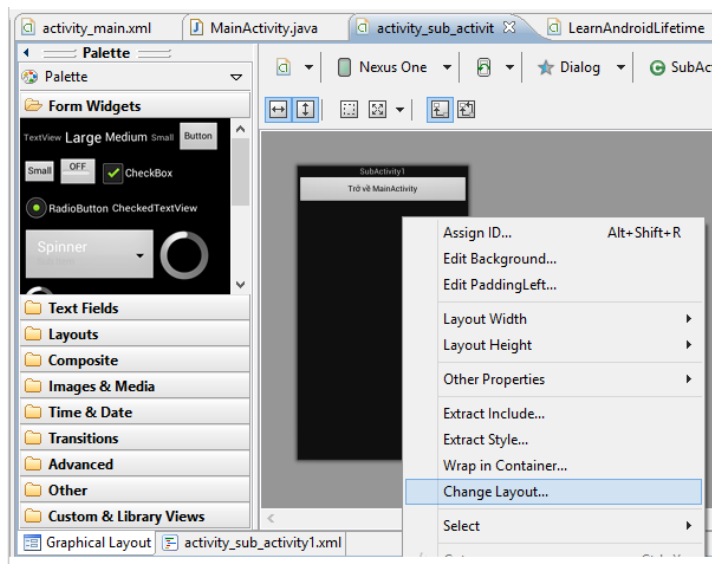
- Tạo một Android Project
- Tạo thêm một XML Layout (tùy ý) đặt tên là **new_my_layout.xml**
- Kéo thả một số control vào layout mới này: Button, Checkbox, Radio...
- Tiến hành thay đổi layout mới này qua các loại: FrameLayout, LinearLayout, TableLayout, RelativeLayout, AbsoluteLayout, ... và quan sát 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:

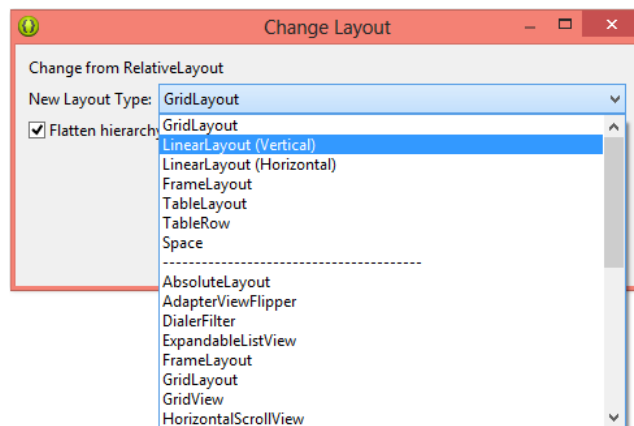
- Để tạo một XML Layout: Bấm chuột phải vào Project/ New/ Android XML File:
 - o Màn hình tạo Android XML File sẽ hiển thị lên, tại màn hình này **Resource Type** chọn là **Layout**, phần **Root Element** chọn loại **Layout** giả sử là **LinearLayout**, mục **File** nhập tên là **new_my_layout** rồi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

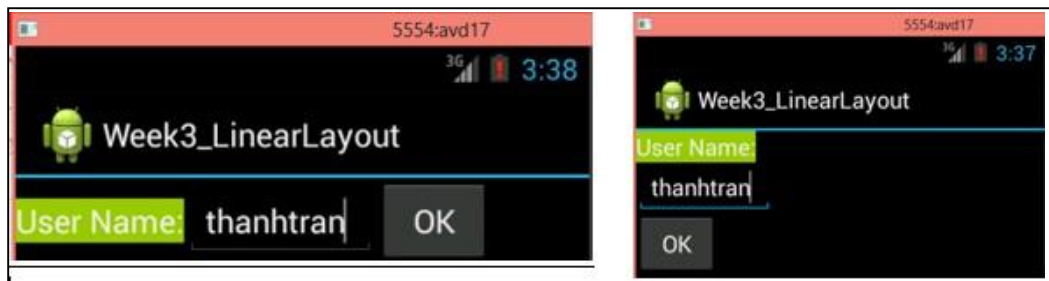
Bài tập 3:

Mục đích:

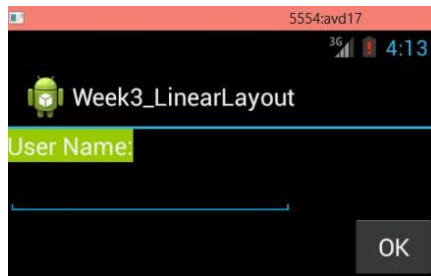
- Thực hành và hiểu về **LinearLayout**:
 - o Cách tạo **LinearLayout** (vertical, horizontal)
 - o Orientation
 - o Fill model
 - o Weight
 - o Gravity
 - o Padding
 - o Margin

Yêu cầu:

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



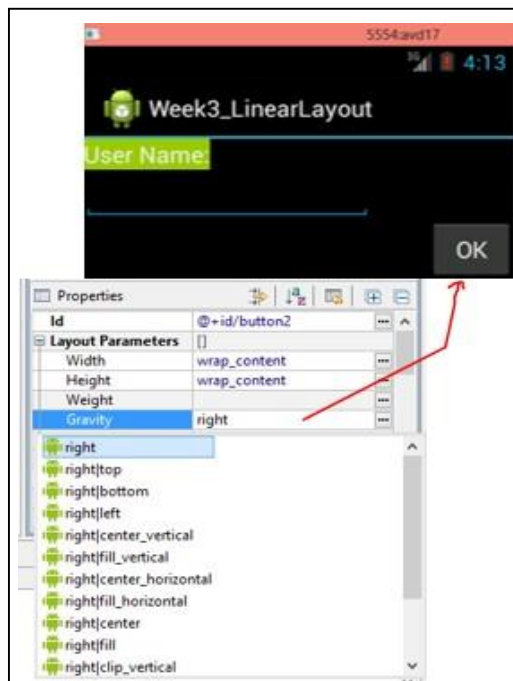
- Thay đổi **android:orientation="horizontal"** và **android:orientation="vertical"** để quan sát các control trên giao diện sắp xếp khác nhau.
- Trong mỗi control thay đổi **android:layout_width** và **android:layout_height**: thiết lập 3 thông số (125dip, wrap_content, fill_parent), quan sát sự thay 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>

```

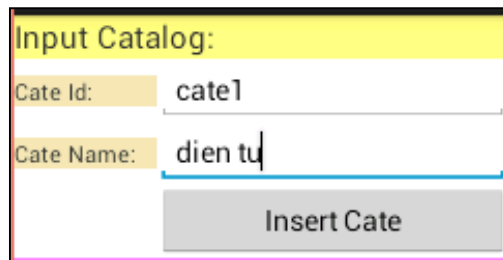
Bài tập 4:

Mục đích:

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

Yêu cầu:

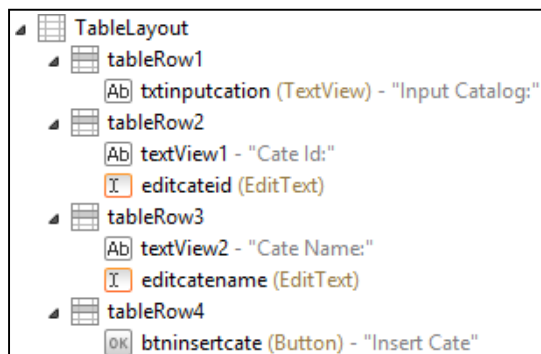
- Tạo Android Project có kiểu layout là TableLayout, thiết kế các control trên giao diện giống như hình bên dưới:



- 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:
 - o **android:stretchColumns="*"** cho TableLayout
 - o **android:layout_span="2"** cho TextView ở dòng 1

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

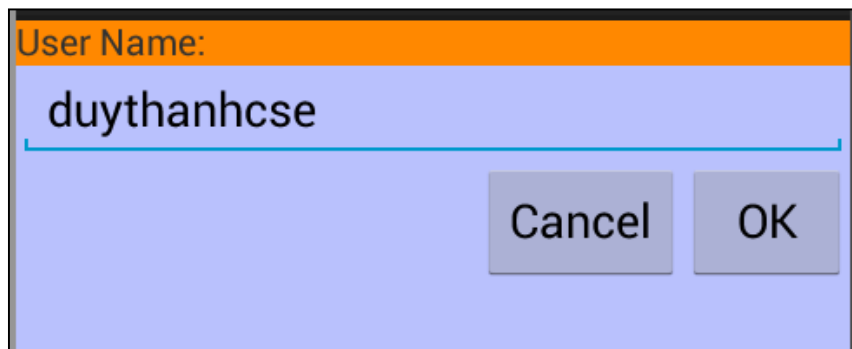
Bài tập 5:

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:

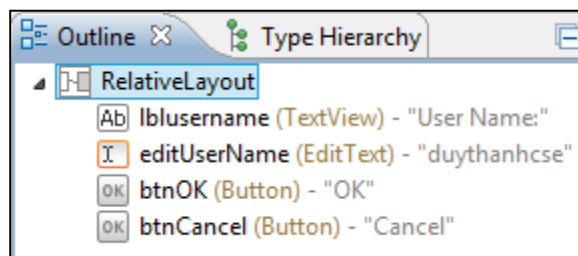
- Tạo Android Project có kiểu Layout là **RelativeLayout**, thiết kết các control như giao diện bên dưới:



- 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**

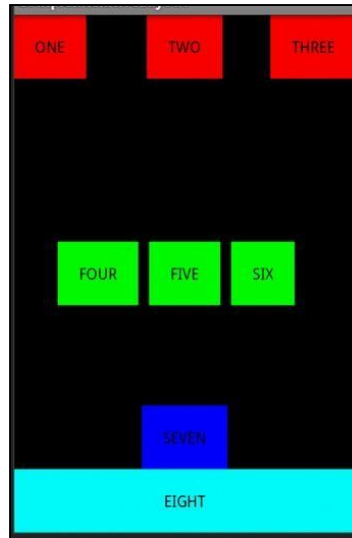
Bài tập 6:

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:

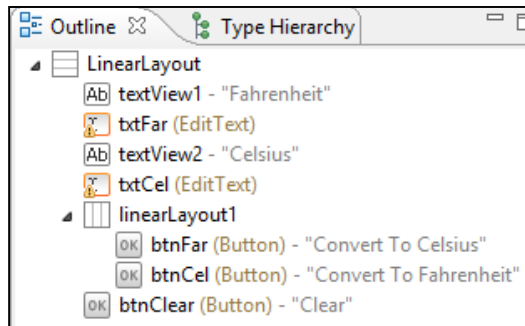
- Viết chương trình chuyển đổi nhiệt độ C ↔ F. Thiết kế giao diện và thực hiện các yêu cầu như bên dưới:

The image shows a mobile application interface for temperature conversion. It consists of two text input fields. The first field is labeled 'Fahrenheit' and the second is labeled 'Celsius'. Below these fields are three buttons. The first button is labeled 'Convert To Celsius', the second is labeled 'Convert To Fahrenheit', and the third is a wider button labeled 'Clear'.

- Khi bấm Convert To Celsius sẽ chuyển đổi từ F → C
- Khi bấm Convert To Fahrenheit sẽ chuyển đổi từ C → 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 : $(98.6^{\circ}\text{F} - 32) \times 5/9 = 37^{\circ}\text{C}$

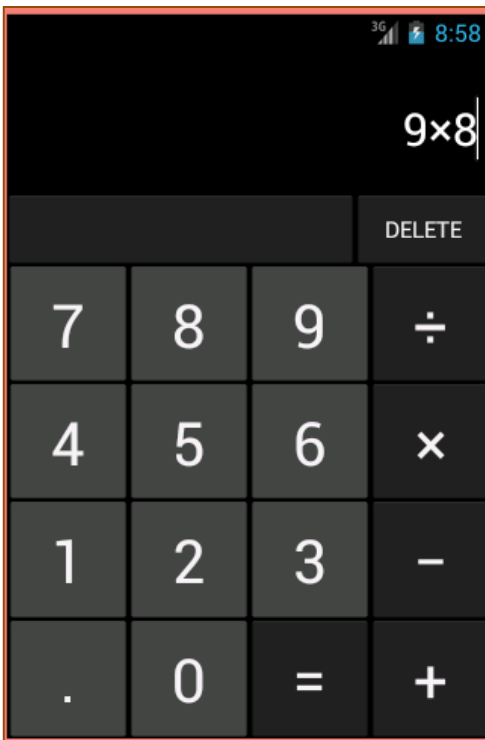
Bài tập 10 (*):

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):

The image shows two screenshots of an Android application. The left screenshot displays a form titled 'Chương trình tính tiền bán sách online'. It has a yellow header 'Thông tin hóa đơn:' and contains fields for 'Tên Khách Hàng' (Đoàn Ái Nương), 'Số lượng sách' (113), a checked 'Khách hàng là VIP' checkbox, and a 'Thành tiền' field showing 2034000. Below these are buttons 'Tính TT', 'Tiếp', and 'Thống kê'. A second yellow header 'Thông tin thống kê:' is followed by fields for 'Tổng số KH' (3), 'Tổng số KH là VIP' (1), and 'Tổng doanh thu' (2184000). A red power button icon is at the bottom right. The right screenshot shows the same form on a mobile screen, with a red arrow pointing to the bottom, indicating the scroll view.

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

This screenshot shows the top portion of the application interface. It includes a dark blue title bar with the text 'Chương trình tính tiền bán sách online'. Below it is a yellow header section titled 'Thông tin hóa đơn:'. The first input field, 'Tên Khách Hàng: Đoàn Ái Nương', is visible.

- 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.

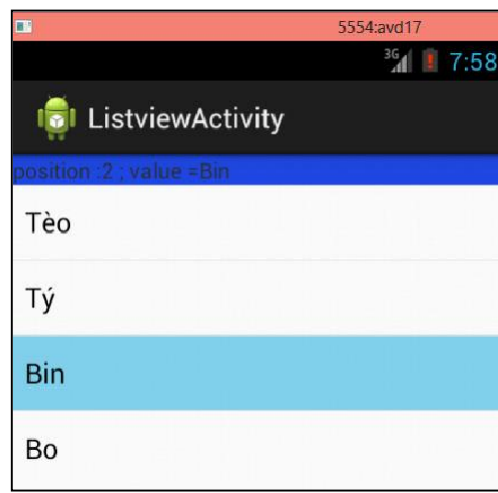
Bài tập 16:

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

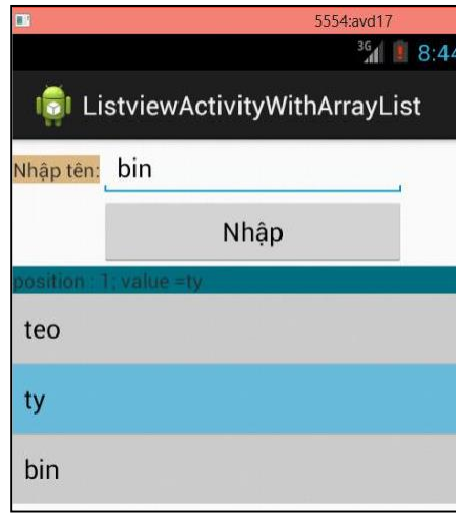
Bài tập 17:

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

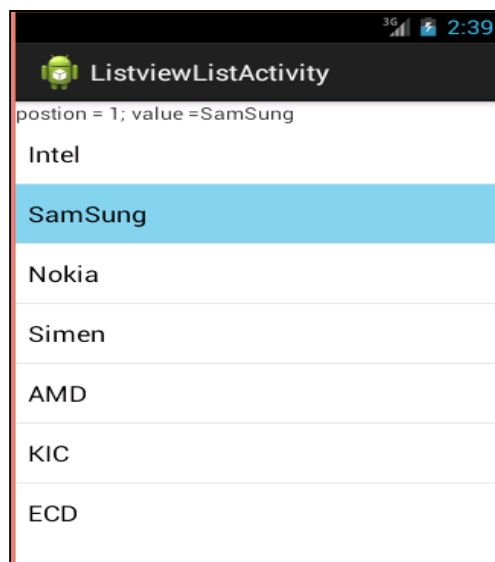
Bài tập 18:

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

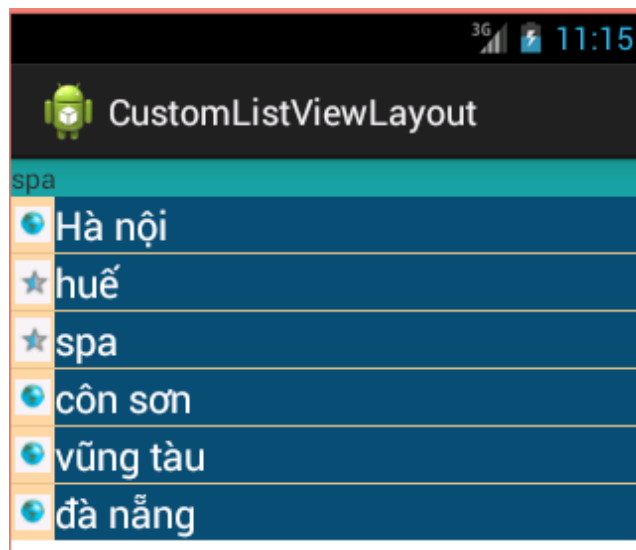
Bài tập 19:

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 hiển 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:

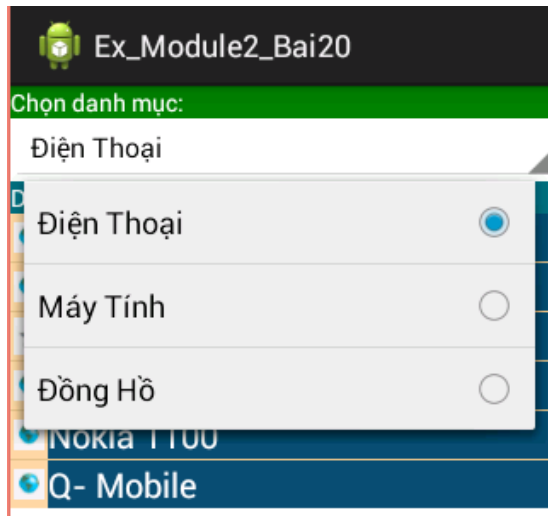
- Thực hành và hiểu về các control nâng cao trong Android:
 - o Spinner
 - o Kết hợp Spinner, ListView

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.

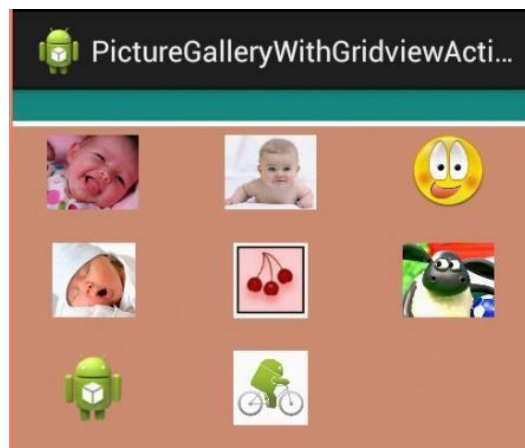
Bài tập 22 (*):

Mục đích:

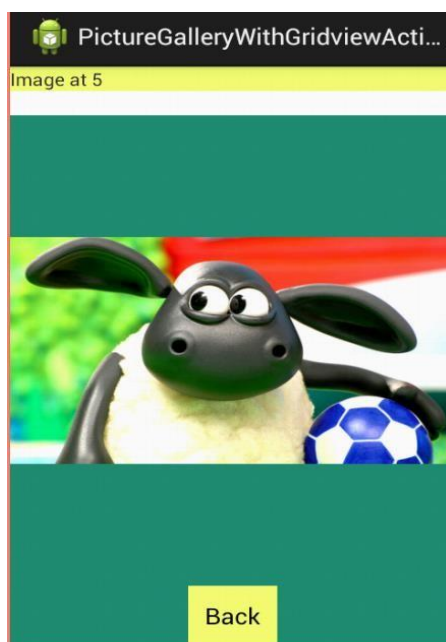
- Thực hành và hiểu về các control nâng cao trong Android:
 - o Hiểu cách đưa hình ảnh vào GridView, biết cách sử dụng **BaseAdapter**

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:

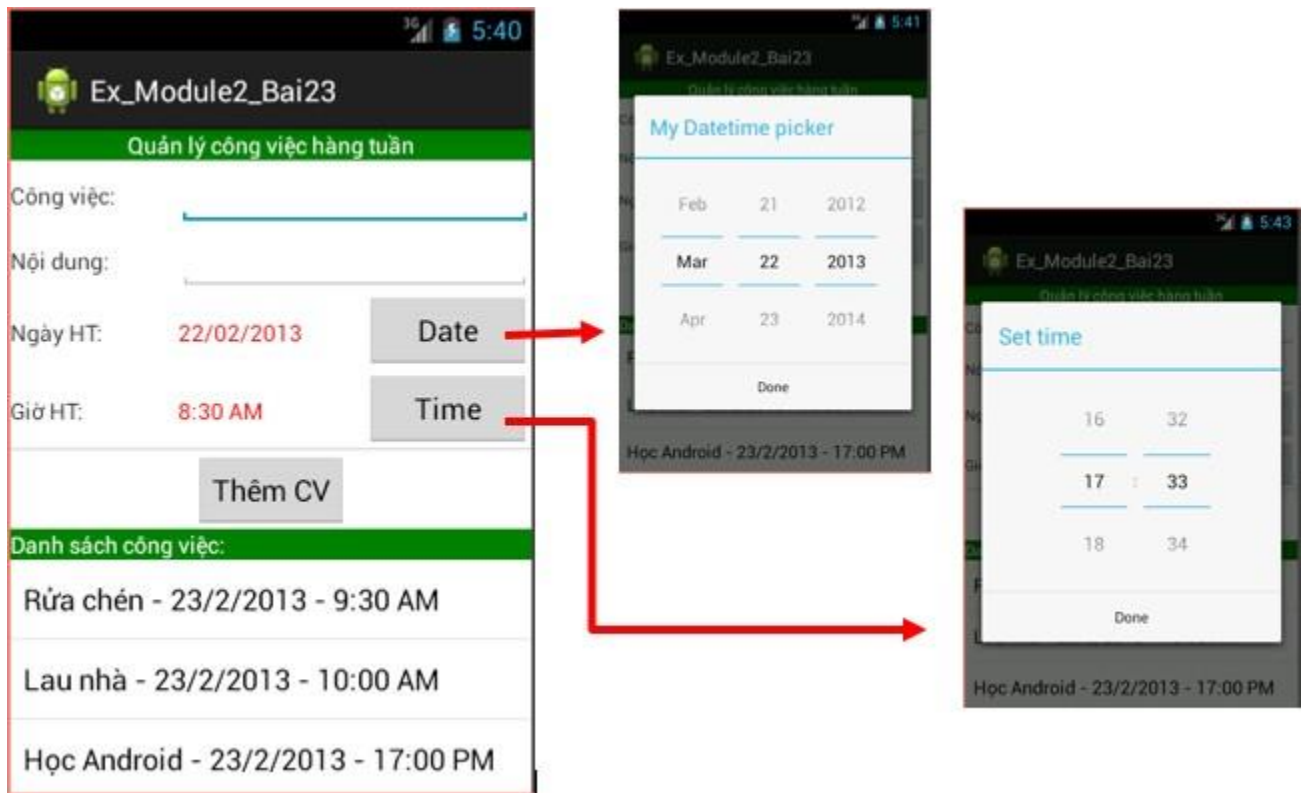
Mục đích:

- 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 (*):

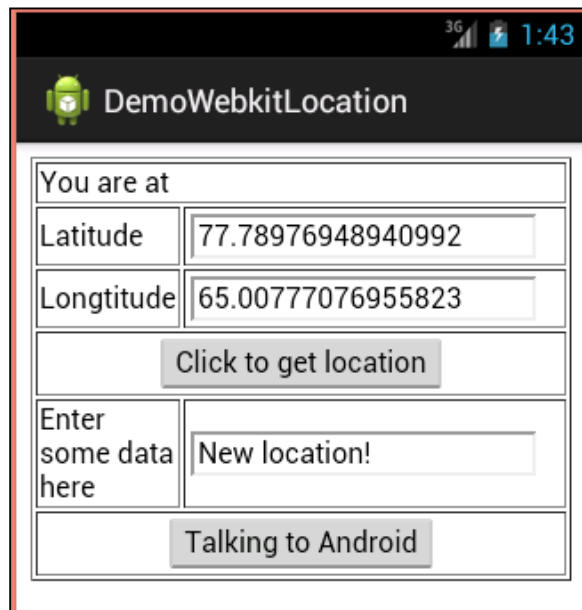
Mục đích:

- 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 phân 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:

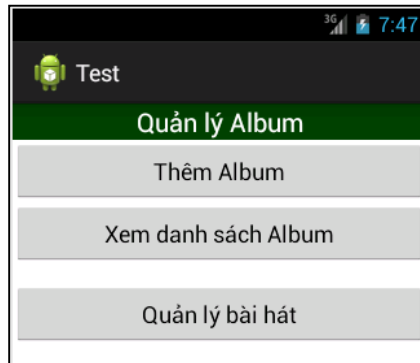
Bài tập: Lập trình Android

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

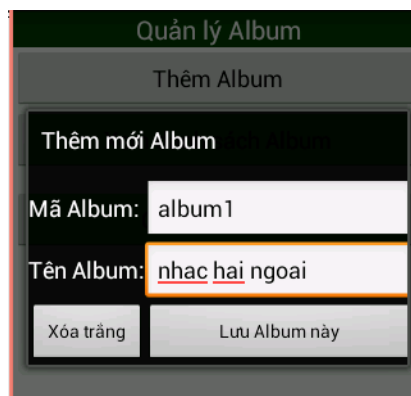
- Explicit Intent
- Implicit Intent
- Cách lấy kết quả qua lại giữa các Activity thông qua Intent

Yêu cầu:

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



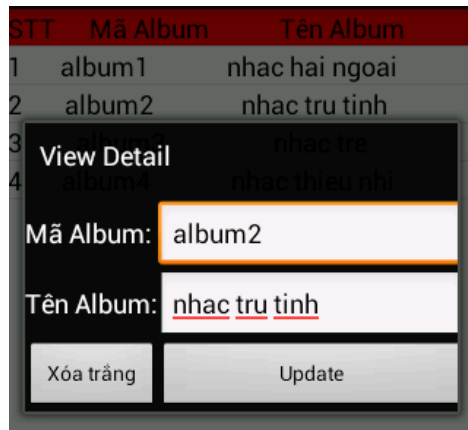
- Phải tạo Album trước khi vào chức năng quản lý bài hát
- Khi chọn chức năng thêm Album, chương trình sẽ xuất hiện màn hình dưới đây (chú ý mở dưới dạng Dialog):



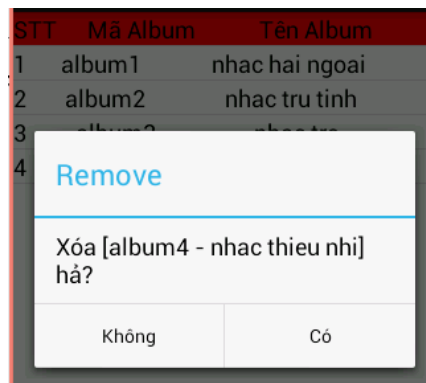
- Chọn “lưu album này” để lưu, chọn xóa trắng để xóa dữ liệu vừa nhập, focus tới mã.
- Khi chọn “xem danh sách album”, chương trình sẽ mở một màn hình mới để hiển thị danh sách Album đã tạo – chú ý dùng CustomLayout cho ListView:

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

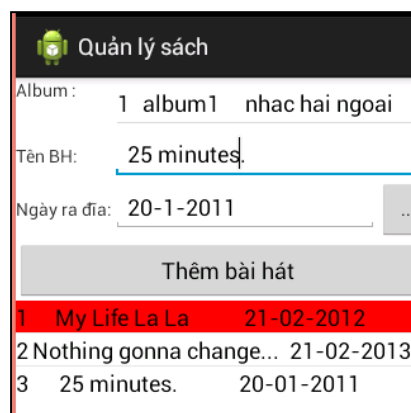
- Khi chọn từng Album trong danh sách, sẽ hiển thị màn hình cho phép chỉnh sửa Album:



- 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.

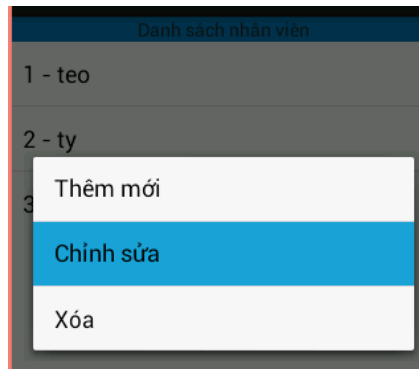
Bài tập 28:

Mục đích:

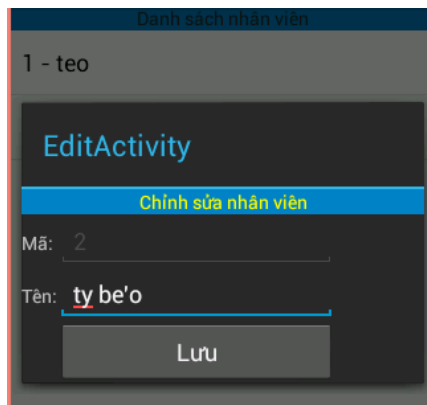
- Thực hành và hiểu được về Intent:
 - o Explicit Intent
 - o Implicit Intent
 - o Cách lấy kết quả qua lại giữa các Activity thông qua Intent

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

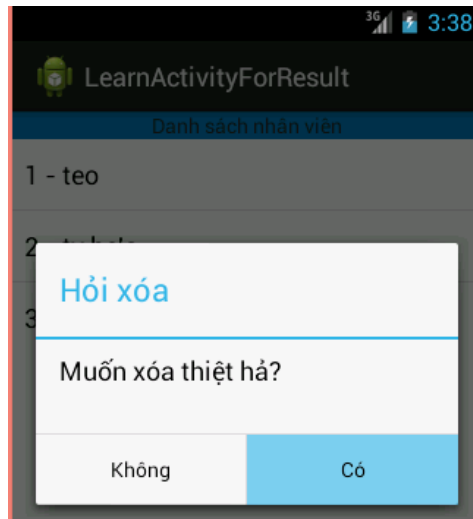


- o Chọn Lưu để cập nhật
- Chọn Thêm mới để thêm nhân viên mới:



- o 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