

ADB_TUTOR_2

Full name	ADB_Tutor2_hoang.pt_v1.0.doc
Class	Android beginner
Date	20-06-2013

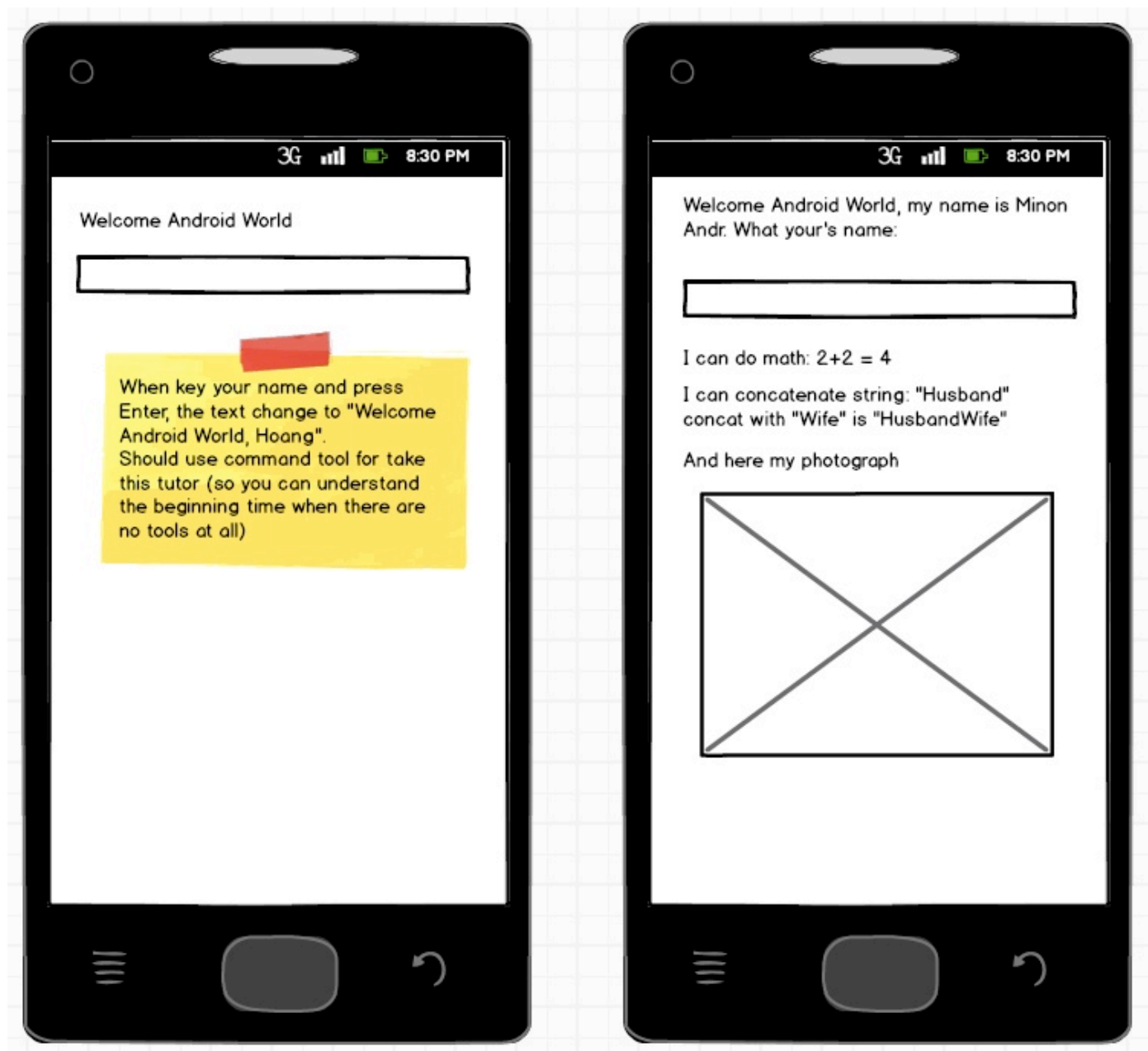
Who	Name	Role	Date	Signature
CREATOR	Phan Tích Hoàng	Lecturer	20 June 2013	
REVIEWER				
APPROVER				

MONSTER 1: Hello_world

Đề bài: Xây dựng ứng dụng Hello World.

Mục tiêu: hiểu các bước xây dựng một ứng dụng Android sử dụng command_line.

1. Cài đặt java_sdk, android_sdk, ant và eclipse.
2. Sử dụng command line tạo code skeleton cho ứng dụng Hello world.
3. Code ứng dụng gồm một TextView và một EditText (sử dụng java) trên LinearLayout. Khi nhập tên vào EditText, dòng chào của TextView sẽ hiển thị "Welcome Android World, {tên_nhập}"
4. Dùng ant để build ra file apk và dùng sdk để đẩy lên máy ảo android.
5. Sửa code nếu gặp lỗi.
6. Mở rộng ứng dụng như hình 2. Sau dòng nhập tên là các phép tính cơ bản như +, nối chuỗi, ... và hiển thị một ảnh nào đó làm avatar.



MONSTER 2: Flower_sell

Đề bài: Hoa hậu Mai Xuân sau một thời gian chinh chiến xô-bít, quyết định về nhà mở cửa hàng bán hoa. Lượng khách quá đông nên Mai Xuân quyết định thuê bạn viết một app android cho phép khách hàng đặt hàng trên điện thoại.

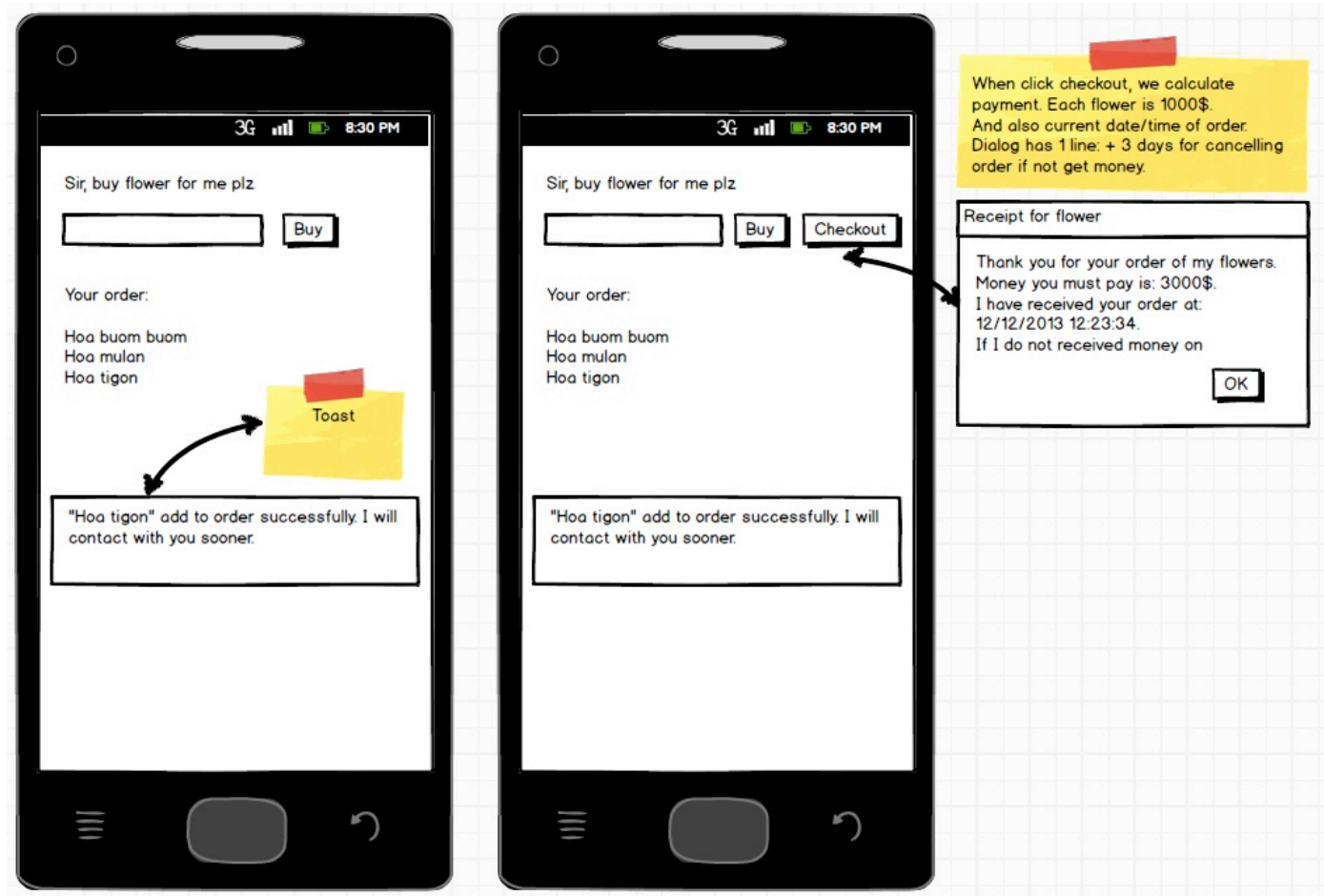
Mục đích: tìm hiểu cách sử dụng elipse và xml layout để build 1 app của android.

1- Tạo mới một project android bằng eclipse.

2- Sử dụng build giao diện bằng file xml như hình vẽ.

3- Khi người dùng nhập hoa và click vào nút Buy, tên của hoa sẽ hiển thị ở phần order, đồng thời một Toaster sẽ hiển thị cho người dùng biết đã thêm vào thành công. Chú ý: nếu nhập hoa rỗng thì không click được.

4- Thêm một nút Checkout, để khi click vào nút đó hiển thị một Dialog thông báo giá tiền, ngày giờ của order và ngày giờ cần phải trả tiền.



Bài làm:

MONSTER 3: Todo_list

Đề bài: Bà Tung từ hồi tham gia Xôbit nhận được rất nhiều hợp đồng làm việc. Việc quá nhiều khiến cô ấy không thể nhớ hết được và muốn bạn viết một ứng dụng quản lý công việc giúp cô ấy.

1- Tạo giao diện như hình 1.

2- Khi Tung nhập công việc và thời gian ok, add vào ListView như hình vẽ. Nếu thiếu 1 trong các trường cần có, hiển thị Warning Dialog giúp bà Tung nhập đúng.

3- Mở rộng ứng dụng cho phép sắp xếp công việc theo thời gian.

4- Khi bà Tung thực hiện xong một task, cô ấy có thể gạch nhiệm vụ đó.



Bài làm:

Trước khi vào làm, chúng ta cùng tìm hiểu qua về **Android Manifest**. Trong cấu trúc của phần mềm, thư mục gốc có chứa một file tên gọi là **AndroidManifest.xml**. Đây là file mô tả những thông tin quan trọng của hệ thống (trong Netbean, bạn sẽ tìm thấy ở phía Important File, còn Eclipse là file root và hỗ trợ cho bạn tool để edit file XML này một cách nhanh nhất. Bạn nên hiểu cách vận hành chứ không nên phụ thuộc vào tool. Hãy làm việc smart)

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.me"
    android:versionCode="1"
    android:versionName="1.0">
    <application android:label="@string/app_name">
        <activity android:name=".ToDoActivity"
            android:label="@string/app_name">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
</manifest>
```

Để dàng suy đoán những công việc mà file này thực hiện:

- Đặt tên cho Java package của ứng dụng.
- Mô tả các thành phần (component) của ứng dụng: activity, service, broadcast receiver hoặc content provider.
- Thông báo những permission mà ứng dụng cần có để truy nhập các protected API và tương tác với các ứng dụng khác.
- Thông báo những permission mà các ứng dụng khác cần có để tương tác với ứng dụng hiện thời.
- Thông báo level thấp nhất của Android API mà ứng dụng cần để chạy. (Android 1.0 là level 1, 1.1 là level 2, 1.5 level 3, 1.6 level 4 và 2.0 là level 5)

Cùng xem file AndroidManifest.xml của chương trình đang xây:

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.me"
    android:versionCode="1"
    android:versionName="1.0">
    <application android:label="@string/app_name">
        <activity android:name=".ToDoActivity"
            android:screenOrientation="landscape"
            android:theme="@android:style/Theme.NoTitleBar.Fullscreen"
            android:label="@string/app_name">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity android:name=".WorkEnter">
        </activity>
        <receiver android:name=".AlarmReceiver">
        </receiver>
    </application>
    <uses-sdk android:minSdkVersion="3" />
    <uses-permission android:name="android.permission.VIBRATE"/>
</manifest>
```

Thiết lập vào màn hình ở chế độ ngang, fullscreen và không có title bar phía trên.

Thêm một Activity # để nhập thời gian và công việc. 1 Broadcast Receiver để nhận alarm gửi tới trong intent, khi có alarm sẽ có âm thanh và rung

Truy cập rung của máy.

Ngoài ra, Intent filter là bộ lọc dùng để giới hạn các intent được sử dụng trong activity hay receiver... (sẽ được nói ở bài về Intent)

```

        <intent-filter>
            <action android:name="android.intent.action.MAIN" />
            <action android:name="android.intent.action.VIEW"/>
            <category android:name="android.intent.category.LAUNCHER" />
            <category android:name="android.intent.category.DEFAULT"/>
            <category android:name="android.intent.category.BROWSABLE"/>
            <data android:scheme="http" android:host="www.google.com"
android:path="/m/products/scan"/>
        </intent-filter>

```

Bước 1: Vẫn bắt đầu bằng cách khởi tạo một Project mới như bài trước, đặt tên project là TooDo, có main activity là TooDoActivity.

Bước 2: Đi tới res/main.xml để xây dựng giao diện cho chương trình (có thể sử dụng tool kéo thả kết hợp với copy paste cho nhanh nha)

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical" android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    >
    <TextView android:id="@+id/welcome_text"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:textColor="#ff3300"
        android:textSize="28px"
        android:typeface="monospace"
        android:text="Welcome baby" />

    <EditText android:id="@+id/work_text"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:hint="@string/edit_text_hint"
        android:lines="1"
        android:textSize="24px" />

    <LinearLayout android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:orientation="horizontal">
        <TextView android:layout_width="50px"
            android:layout_height="wrap_content"
            android:text="@string/hour_edit"
            android:typeface="normal"
            android:textSize="15px"
            android:textStyle="bold" android:padding="5px" />
        <EditText android:id="@+id/hour_edit"
            android:layout_width="45px"
            android:layout_height="wrap_content"
            android:textColorHint="@color/hint_color"
            android:textSize="20px"
            android:gravity="center"
            android:padding="5px"
            android:numeric="integer"
            android:maxLength="2"
            android:hint="12"/>

        <TextView android:layout_width="65px"
            android:layout_height="wrap_content"
            android:text="@string/minute_edit"
            android:typeface="normal"
            android:textSize="15px"
            android:textStyle="bold"
            android:padding="5px" />
        <EditText android:id="@+id/minute_edit"
            android:layout_width="45px"
            android:layout_height="wrap_content"
            android:textColorHint="@color/hint_color"
            android:textSize="20px"

```

Chỉ cho phép nhập 1 dòng, nên dùng kết hợp với wrap_content vì rất có thể chúng ta nhập dài quá sẽ làm hỏng giao diện.

Sử dụng padding để các control cách nhau ra (giống với HTML). Có thể sử dụng thêm paddingLeft, paddingRight

```

<EditText android:id="@+id/minute_edit"
    android:layout_width="45px"
    android:layout_height="wrap_content"
    android:textColorHint="@color/hint_color"
    android:textSize="20px"
    android:gravity="center"
    android:padding="5px"
    android:numeric="integer"
    android:maxLength="2"
    android:hint="00"/>
</LinearLayout>
<Button
    android:id="@+id/button"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:gravity="center"
    android:text="@string/button_content"
    />
<ListView
    android:id="@+id/list"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    />

```

Từ khóa gravity thông báo các thành phần sẽ được căn chỉnh thế nào, ở đây mình dùng center tức là sẽ căn vào giữa. Thử thêm 1 EditText, bạn sẽ thấy dòng chữ nhập vào sẽ bắt đầu từ giữa của EditText chứ không phải là trái như mặc định. Từ khóa numeric chỉ cho phép nhập số (nghĩa là khi vào ô này, bạn sẽ thấy hiện lên bàn phím ảo chỉ có số). Maxlength=2 vì giờ chỉ có thể lớn nhất là 24.

Bước 3: Tối values/strings.xml chỉnh sửa như sau:

```

<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string name="app_name">ToDo</string>
    <string name="edit_text_hint">Enter work to do here</string>
    <string name="hour_edit">Hour</string>
    <string name="minute_edit">Minute</string>
    <string name="button_content">Add work</string>
</resources>

```

Bước 4: Tạo mới colors.xml trong values với nội dung:

```

<?xml version="1.0" encoding="UTF-8"?>
<resources>
    <color name="hint_color">#cccccc</color>
</resources>

```

OK, vậy là đã hoàn thiện phần giao diện. Các bạn có thể cho chạy thử ngay để kiểm tra xem giao diện đã như ý muốn chưa chứ không cần đợi hoàn thành cả code (Run as -> Android Application).

Bước 5: Code code code, tìm thức ăn nào

```

package com.me;

import android.app.Activity;
import android.app.AlertDialog;
import android.content.DialogInterface;
import android.os.Bundle;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.EditText;

```

```

import android.widget.Button;
import android.widget.EditText;
import android.widget.ListView;
import java.util.ArrayList;

public class ToDoActivity extends Activity {

    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        //call super to render view
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);

        //bind all control to variable so you can handle of this
        final ArrayList<String> listWorks = new ArrayList<String>();
        final ArrayAdapter<String> adapterWorks = new ArrayAdapter<String>(this,
        android.R.layout.simple_list_item_1, listWorks);

        final EditText workEnter = (EditText) findViewById(R.id.work_text);
        final EditText hourEdit = (EditText) findViewById(R.id.hour_edit);
        final EditText minuteEdit = (EditText) findViewById(R.id.minute_edit);

        final Button button = (Button) findViewById(R.id.button);

        final ListView list = (ListView) findViewById(R.id.list);
        list.setAdapter(adapterWorks);

        //add listener to button
        OnClickListener addListener = new OnClickListener() {

            public void onClick(View view) {
                //if 1 of 3 is null, display alert
                if (workEnter.getText().toString().equals("")
                    || hourEdit.getText().toString().equals("")
                    || minuteEdit.getText().toString().equals("")) {
                    AlertDialog.Builder builder = new
        AlertDialog.Builder(ToDoActivity.this);
                    builder.setTitle("Info missing");
                    builder.setMessage("Please enter all information of the work");

                    //build 1 continue button in alert, you can set default. When click
        Continue, nothing.
                    builder.setPositiveButton("Continue", new
        DialogInterface.OnClickListener() {

                        public void onClick(DialogInterface dialog, int which) {
                            }
                        });
                    builder.show();
                } else {
                    String str = workEnter.getText().toString() + " - "
                        + hourEdit.getText().toString() + ":"
                        + minuteEdit.getText().toString();
                    //add to header, so if you want to sort?
                    listWorks.add(0, str);
                    adapterWorks.notifyDataSetChanged();

                    //reset form
                    workEnter.setText("");
                    hourEdit.setText("");
                    minuteEdit.setText("");
                }
            }
        };
        button.setOnClickListener(addListener);
    }
}

```


Mình đã chú thích đầy đủ và đoạn code cũng khá dễ hiểu. Tuy nhiên cần lưu ý 2 vấn đề ở đây.

- Khởi tạo đối tượng ArrayAdapter: Các bạn thấy đối số truyền vào là **(this, android.R.layout.simple_list_item_1, arrayWork)**. This là đối số của lớp Context (ở đây chính là activity Example). Bạn sẽ gặp Context trong rất nhiều khởi tạo các lớp và nên hiểu Context có ý nghĩa gì.

Context thuộc **android.content (android.content.Context)**: Là một Interface (lớp giao tiếp) chứa hầu hết thông tin về môi trường ứng dụng của android, có nghĩa là mọi thao tác, tương tác với hệ điều hành đều phải qua lớp này. Nó là một lớp abstract (trừu tượng) cung cấp cho những lớp khác các phương thức để tương tác với hệ thống Android. Nó cho phép truy cập tới các nguồn tài nguyên (resources) đã được định nghĩa và các lớp khác. Ví dụ như nó có thể khởi tạo và chạy các activities, các broadcast và các intents,... Chúng ta coi như Context là một lớp ở mức ứng dụng (Application level- liên quan tới hệ thống).

Tóm lại context giúp chúng ta dễ dàng truy cập và tương tác tới các tài nguyên của hệ thống, các thông tin, các dịch vụ (services), các thông số cấu hình, database, wallpaper, danh bạ, cuộc gọi, kết nối, chế độ rung (vibrator),... Sở dĩ hầu hết các lớp có liên quan tới UI (layout, button, textview, imageview, listview,...) đều phải super tới Context vì bản thân nó đảm nhiệm việc truy cập resource (R.id, R.layout,...). Nếu chúng ta không tham chiếu tới Context class thì đương nhiên không thể dùng tới các resources mà chúng ta đã tạo ra.

Tiếp theo là android.R.layout.simple_list_item_1, đối này định nghĩa cách thể hiện item (ở đây là String) trong List View. Các bạn hãy ghi nhớ android.R.* là các tài nguyên (resource) có sẵn của Android cho phép bạn truy cập và sử dụng. Sau này khi hướng dẫn tạo custom View cho List View mình sẽ đề cập lại vấn đề này. Cuối cùng arrayWork chính là mảng cần được bind của adapter.

- AlertDialog là lớp cho phép đưa ra 1 hộp thoại, thường dùng để đưa ra thông tin hoặc cảnh báo đơn giản. Trong code mình tạo 1 builder, tạo tiêu đề (title) cho nó, đưa ra thông báo (message) và cuối cùng là tạo 1 positive button (nhưng không định nghĩa xử lý khi nhấn nút này, vì vậy nếu bạn nhấn nút thì dialog sẽ chỉ đơn giản thực hiện việc đóng lại).

Bước 6: chạy thử chương trình và tận hưởng thành quả của do it yourself.

Chú ý: Mở rộng ứng dụng cho phép sắp xếp các task theo thời gian. Đồng thời cho phép gạch một task khi thực hiện xong task đó.