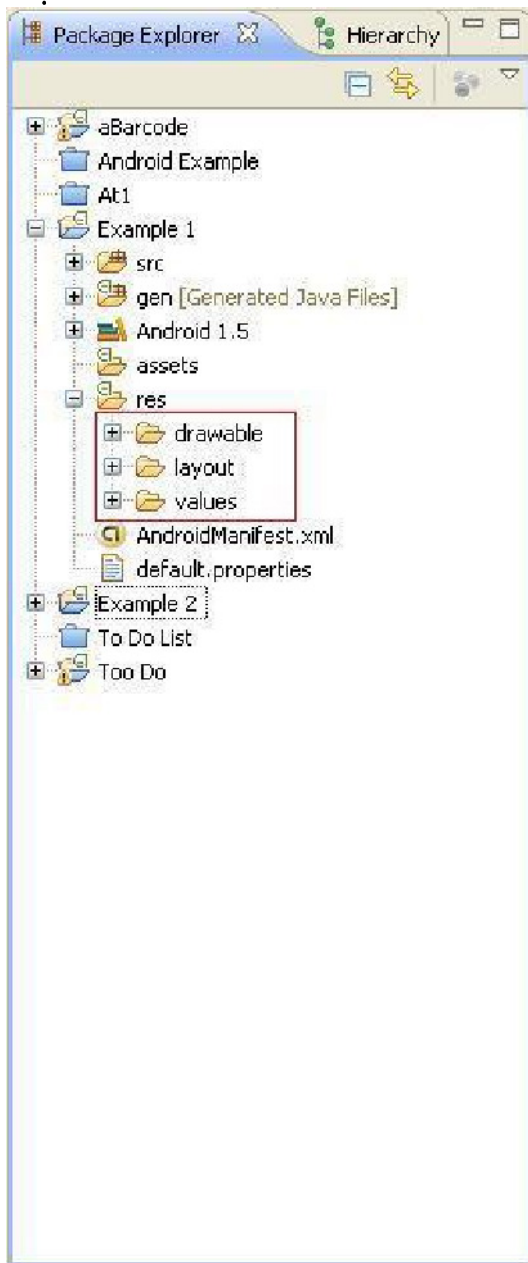


Hướng dẫn lập trình cơ bản với Android

Part 5

B3: Bên khung Package Explorer bên trái đi tới thư mục res, bạn sẽ thấy có 3 thư mục con:



- drawable: thư mục chứa các hình ảnh để làm icon hoặc tài nguyên cho giao diện...
- layout: chứa các file xml để thiết kế giao diện.

- values: chứa các giá trị sử dụng trong ứng dụng được bạn định nghĩa, như các dòng ký tự (string), các màu (color), các themes...

B4: Vào thư mục layout, chọn file main.xml và gõ đoạn code sau vào thay cho toàn bộ nội dung có sẵn (Eclipse hỗ trợ kéo thả cho xml nhưng theo mình không nên sử dụng):

Mã:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/andro
id"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    >
    <EditText
        android:id="@+id/edit_text"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:hint="@string/edit_hint"
    />
    <TextView
        android:id="@+id/text_view"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:textColor="@color/text_color"
        android:textSize="28px"
        android:typeface="monospace"
    />
</LinearLayout>
```

Trong đoạn XML này chúng ta khai báo một Linear Layout với 2 thành phần con của nó là 1 Edit Text (dùng để gõ xâu ký tự) với 1 Text View (hiển thị xâu ký tự). Linear Layout được khai báo với từ khóa **orientation** nhằm chỉ ra chiều sắp xếp của 2 thành phần con là chiều dọc. Còn với **layout_width**, **layout_height** các bạn có thể cho giá trị bằng "fill_parent" hoặc "wrap_content" để thông báo thành phần này sẽ có chiều rộng (dài) phủ đầy thành phần cha hoặc chỉ vừa bao đủ nội dung.

Trong Edit Text và Text View các bạn có thể thấy có từ khóa **id**, từ khóa này cho phép khai báo id của các thành phần để lấy về trong code (sẽ đề cập sau).

Ngoài ra từ khóa **hint** trong Edit Text cho phép hiện ra phần nội dung mờ khi Edit

Text vẫn chưa có ký tự nào. "@string/edit_hint" thông báo lấy trong file strings.xml xâu có tên là edit_hint.

Còn **textColor** của Text View thì thông báo đoạn ký tự sẽ được hiển thị với màu lấy trong file colors.xml, **textSize** chỉ ra cỡ chữ bằng 28 pixel và **typeface** chỉ ra kiểu chữ là monospace

B5: Vẫn trong thư mục res, vào values và chọn file strings.xml. Bổ sung thêm dòng định nghĩa cho edit_hint như sau:

Mã:

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string name="hello">Hello World, Example!</string>
    <string name="app_name">Example 1</string>
    <string name="edit_hint">Enter the work
here</string>
</resources>
```

B6: Trong thư mục values, tạo file colors.xml (chuột phải vào thư mục, chọn New -> Android XML File, và lưu ý chữ s, không phải là color.xml). Gõ nội dung cho file như sau:

Mã:

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <color name="text_color">#ff3300</color>
</resources>
```

OK, vậy là bạn đã tạo một màu mới cho dòng chữ sẽ được hiển thị trong Text View (ff3300 là mã hexa của màu đỏ). Thực chất bạn hoàn toàn có thể gõ thẳng Mã:

```
android:textColor="#ff3300"
```

trong file main.xml mà không cần tạo mới file colors.xml, nhưng mục đích của XML trong Android chính là để hỗ trợ nâng cấp chỉnh sửa dễ dàng. Nếu sau này bạn muốn sửa màu của dòng text thì chỉ cần vào colors.xml thay đổi thay vì mò mẫm trong main.xml (có thể rất dài nếu giao diện phức tạp).

Các thành phần trên mới chỉ là các phần cơ bản của XML. Ngoài ra các bạn có thể

khai báo thêm về Animation, Style và Theme (phức tạp hơn nhiều nên mình không giới thiệu trong phần cơ bản này).

B7: Vậy là chúng ta đã hoàn thiện phần giao diện với XML, giờ đến viết code để xử lý các sự kiện cho các thành phần:

=> vào thư mục src (source code của project) => at.exam => Example.java, gõ nội dung code sau vào:

Mã:

```
package at.exam;

import android.app.Activity;
import android.os.Bundle;
import android.view.KeyEvent;
import android.view.View;
import android.view.View.OnKeyListener;
import android.widget.EditText;
import android.widget.TextView;

public class Example extends Activity {
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        //Thiết lập giao diện lấy từ file main.xml
        setContentView(R.layout.main);

        //Lấy về các thành phần trong main.xml thông
qua id
        final EditText edit = (EditText)
findViewById(R.id.edit_text);
        final TextView text = (TextView)
findViewById(R.id.text_view);

        //Thiết lập xử lý cho sự kiện nhấn nút giữa của
điện thoại
        edit.setOnKeyListener(new OnKeyListener() {
            @Override
```

```
        public boolean onKey(View v, int keyCode,
KeyEvent event) {
            if (event.getAction() ==
KeyEvent.ACTION_DOWN
                        && keyCode ==
KeyEvent.KEYCODE_DPAD_CENTER) {
text.setText(edit.getText().toString());
                edit.setText("");
                return true;
            }
            else {
                return false;
            }
        }
    });
}
```