THIẾT KẾ GIAO DIỆN TRÊN ANDROID

Tổng quan

- Trong Android, dùng Activity để hiển thị màn hình.
- Mỗi activity sẽ chứa các View theo dạng cấu trúc cây, nghĩa là một Layout gốc chứa các view/layout con bên trong hoặc chỉ có 1 view duy nhất. (lưu ý Layout cũng là một view)
- Có thể thiết kế giao diện trong code java hoặc trong file xml trong thư mục res/layout.

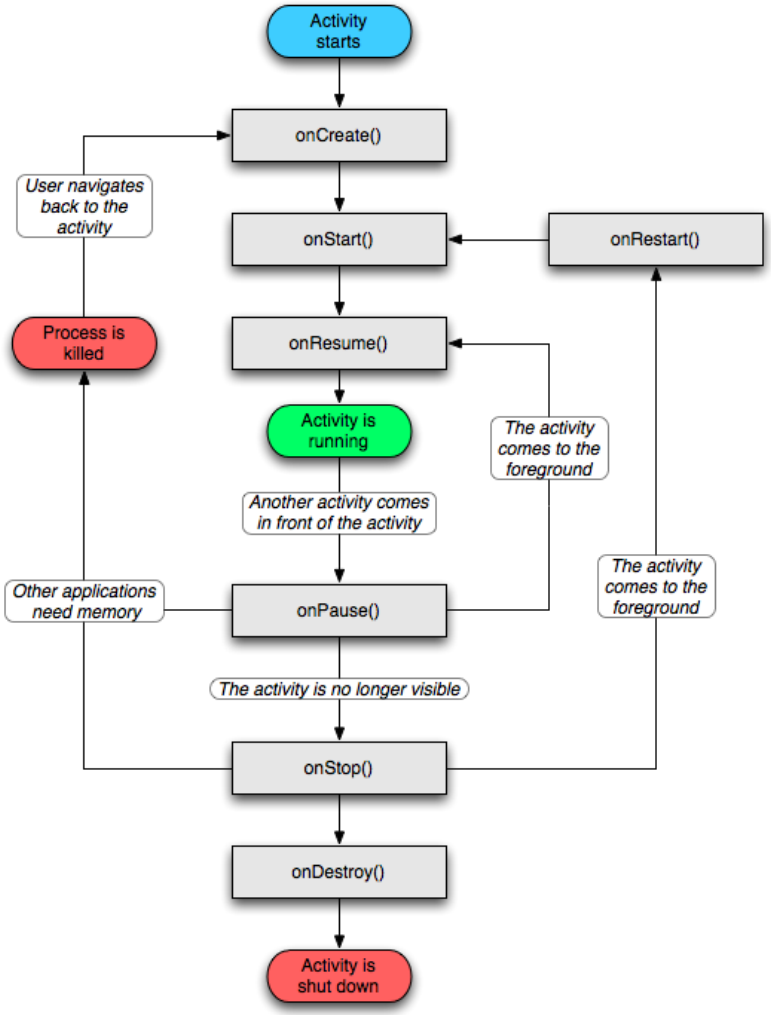
Nội dung

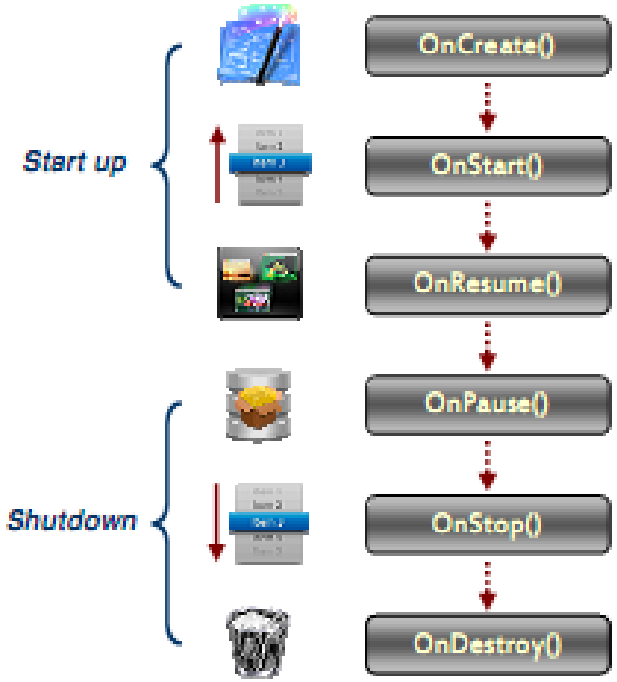
- Activity
- View class
- XML layout
- Android layout
- Connect layout to Java Code

Activity?

- Activity là một trong 4 thành phần chính của một ứng dụng Android (gồm Activity, BroadcastReceiver, Service, ContentProvider). **Tham khảo:**
  http://www.scribd.com/doc/90023242/22/Ch%C6%A3%C6%A1ng-2-Cac-thanh-ph%E1%BA%A7n-c%C6%A1-b%E1%BA%A3n-c%E1%BB%A7a-m%E1%BB%99t-%E1%BB%A9ng-d%E1%BB%A5ng-tren-Android
- Activity được dùng để hiện thị một màn hình.
- Khi làm việc với activity cần bắt đầu với một số kiến thức cơ bản sau:
  - Lifecycle của activity
  - Khởi động một activity, liên lạc giữa 2 activity
  - Task
  - Tạo menu, dialog

Lifecycle của Activity





Khởi động một activity

- Dùng Intent:
  - Khai báo tường minh: cung cấp chính xác thông tin của activity cần gọi (nếu cùng ứng dụng chỉ cần cung cấp tên class, nếu ứng dụng khác nhau thì cung cấp tên package, tên class)
  - Khai báo không tường minh: cung cấp thao tác cần làm gì, với loại dữ liệu nào, thao tác thuộc nhóm nào… hệ thống sẽ tìm activity tương ứng để khởi động.

Khởi động một activity

- Tường minh: đoạn code bên dưới sẽ tạo khởi động Activity tên là TargetActivity

```
Intent  intent = new Intent(getApplicationContext(),
        TargetActivity.class);

startActivity(intent);
```

Khởi động một activity

- Không tường minh: đoạn code bên dưới sẽ khởi động một activity nào đó đăng có khả năng xem ảnh.

```
Intent intent = new Intent(Intent.ACTION_VIEW);
intent.setData(MediaStore.Images.Media.EXTERNAL_CONTENT_URI);
startActivity(intent);
```

Khởi động một activity

- Với cách khởi động activity không tường minh, cần biết một chút về Intent-filter.
- Intent-filter sẽ giúp một activity (chung hơn là một thành phần ứng dụng) đăng ký với hệ thống mình có thể làm được thao tác gì, trong nhóm nào, với loại dữ liệu nào.
- Như vậy khi intent và intent-filter khớp nhau, activity sẽ được hệ thống khởi động.

Liên lạc giữa 2 activity

- Khi khởi động một activity, ta có thể gửi kèm dữ liệu trong intent như ví dụ sau:

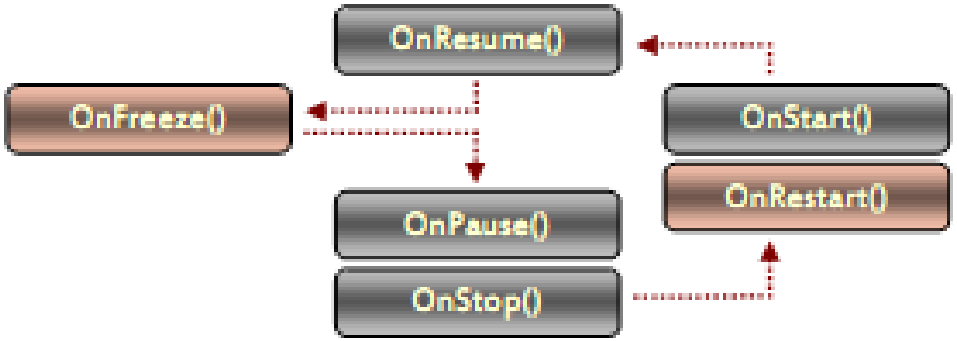
```
intent.putExtra("value1", new String("Hello"));
intent.putExtra(“value2", new Long(100));
```

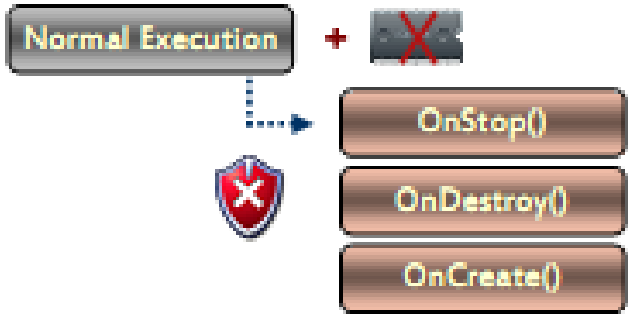
- Bên phía activity được khởi động, có thể lấy dữ liệu được gửi như sau:

```
getIntent().getExtras().getString("value1");
getIntent().getExtras().getLong("value2");
```

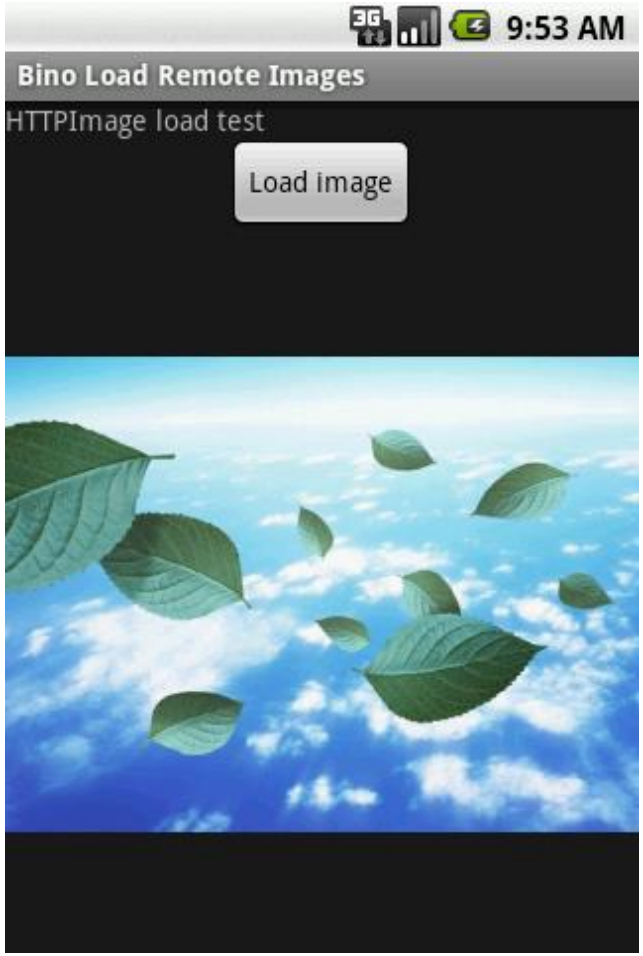
Liên lạc giữa 2 activity

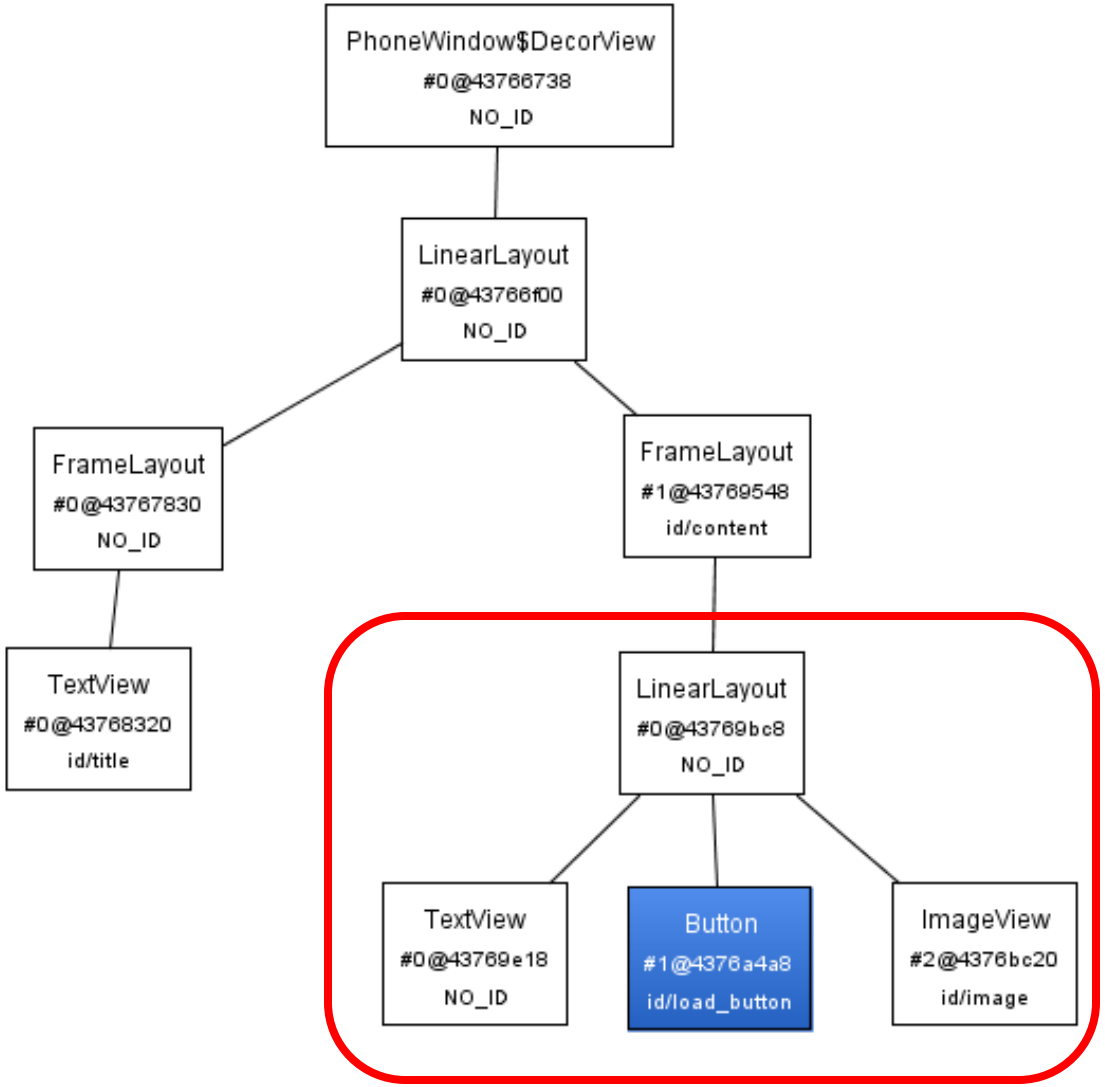
- Có thể khởi động một activity với một yêu cầu nào đó và activity kia khi làm xong công việc sẽ trả lại kết quả cho activity trước
- Ví dụ activity A yêu cầu một activity làm giúp việc chụp ảnh, activity B đáp ứng được việc này, sau khi user chụp ảnh xong sẽ trả lại file ảnh cho activity A.
- Như thế sẽ đỡ tốn nhiều công sức làm một việc mà người khác đã làm rồi.





Tree view


9:53 AM
Bino Load Remote Images
HTTPImage load test
Load image


PhoneWindow$DecorView
#0@43766738
NO_ID
LinearLayout
#0@43766f00
NO_ID
FrameLayout
#0@43767830
NO_ID
FrameLayout
#1@43769548
id/content
TextView
#0@43768320
id/title
LinearLayout
#0@43769bc8
NO_ID
TextView
#0@43769e18
NO_ID
Button
#1@4376a4a8
id/load_button
ImageView
#2@4376bc20
id/image

Layout mẫu của helloworld

```
<?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:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text=“@string/hello“ />

</LinearLayout>
```

Một số thuộc tính cơ bản

- Layout_width, layout_height: chiều rộng của view (fill_parent là to bằng kích thước của layout chứa view này, wrap_content là vừa đủ nội dung cần hiển thị của view)
- Orientation: với LinearLayout, việc sắp xếp các view là nằm kề nhau theo hàng ngang hoặc hàng dọc, ta khai báo orientation để chọn sắp theo kiểu nào (horizontal/vertical)

Một số thuộc tính cơ bản

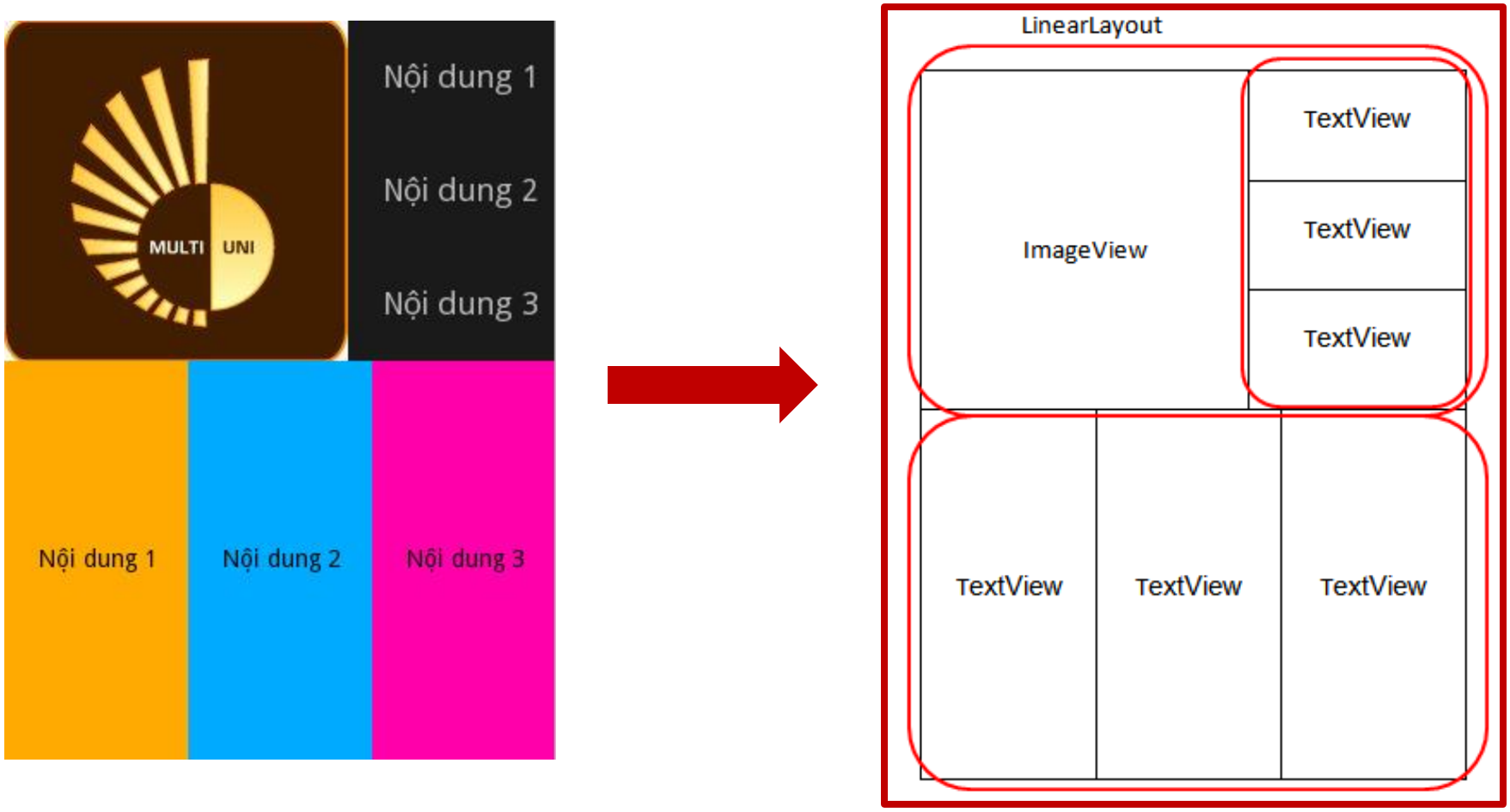
- Gravity: thuộc tính này qui định các view nằm bên trong layout sẽ đặt theo vị trí nào so với layout(trung tâm, trái , phải, trên dưới…)
- Weight: để các view phan chia tỉ lệ diện tích hiển thị trên màn hình (tỉ lệ tính theo weight của từng view trên tổng số weight, các view ko khai báo weight thì sẽ xem qua width và height)

Ví dụ cơ bản

- Vào link này lấy code cho vào ứng dụng, lưu ý đọc thêm các dòng giải thích tiếng Anh http://developer.android.com/resources/tutorials/views/hello-linearlayout.html

Giao diện với LinearLayout

- Giả sử cần thiết kế một màn hình như sau:

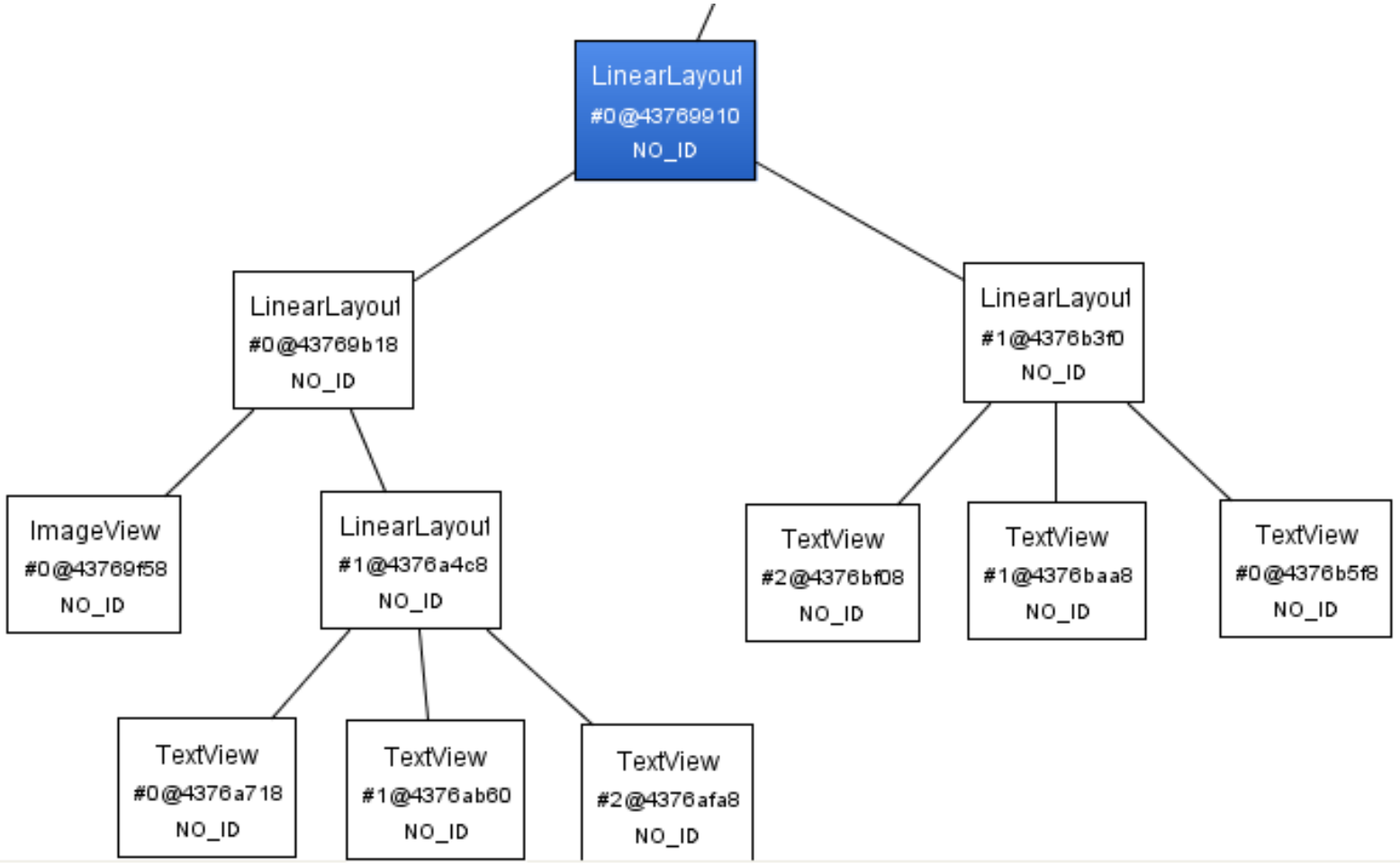


Giao diện với LinearLayout

- Với LinearLayout (LL), các view bên trong nó được đặt kề nhau theo hàng ngang hoặc hàng dọc (cần lưu ý đặc điểm này)
- Với ví dự vừa rồi, ta thấy cách phân tích như sau:
  - Nguyên tắc chủ yếu là phân nhóm các View liên tiếp kề nhau (như 3 TextView kề nhau theo hàng dọc hoặc hàng ngang trong ví dụ trên) vào trong một LL, phân rã từ lớn đến nhỏ.
  - Như vậy màn hình gồm 1 LL lớn bao bên ngoài, nhìn thấy bên trong chia thanh 2 phần trên dưới rõ ràng vậy thuộc tính của LL này là dạng dọc, sau đó chia đôi ra và phân tích tiếp.
  - Phần bên trên lại chia thành 2 nữa theo hàng ngang → là một LL dạng ngang, lại chia đôi: một bên là 1 ImageView (vì chỉ có 1 view nên ko cần bỏ vào trong LL), một bên lại là 1 LL chứa 3 TextView theo hàng dọc.
  - Nửa bên dưới ta thấy rõ ràng chứa 3 TextView kề nhau theo hàng ngang → cho vào 1 LL dạng ngang là xong.

Giao diện với LinearLayout

- Xem cây:



LinearLayout

- Làm một layout hiển thị như trong hình:

- Các bước như sau:
  - Phân tích thành phần layout trên giấy (thảo luận)
  - Phân tích đặc điểm các view
  - Add thêm resource ảnh
  - Thử trước với với tab layout (khi view file xml trong eclipse)
  - Đưa vào thực thi trên máy, đánh giá.

Một số loại layout khác

- FrameLayout: các view bên trong được qui định vị trí bằng khoảng cách so với biên trái và trên so với layout, các view có thể đè lên nhau.
- RelativeLayout: các view được thiết kế dựa trên quan hệ giữa chúng với nhau và với layout chứa chúng.
- AbsoluteLayout.

Lưu ý khi thiết kế giao diện

- Hạn chế độ sâu của cây
- Với các Layout phức tạp, đừng dùng RelativeLayout
- Nên chèn vào dữ liệu tạm để xem trước layout hiển thị ra sao bên tab layout (trong eclipse), nhưng xong rồi thì nhớ xóa dữ liệu tạm đi.
- FrameLayout có vấn đề với background
- Muốn tìm thuộc tính gì, bấm "android:" rồi đợi suggestion sổ ra xem.

Một số ví dụ

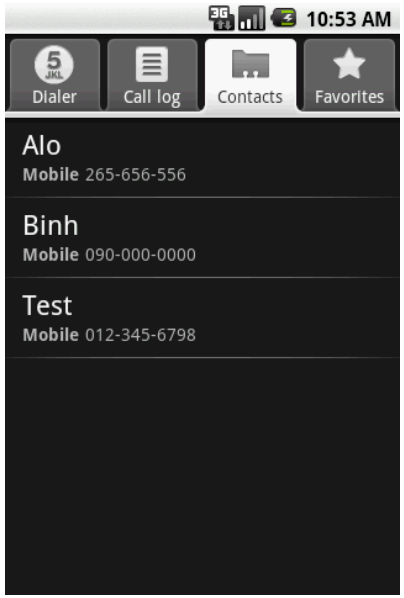
- Tìm hiểu một số ví dụ sau:

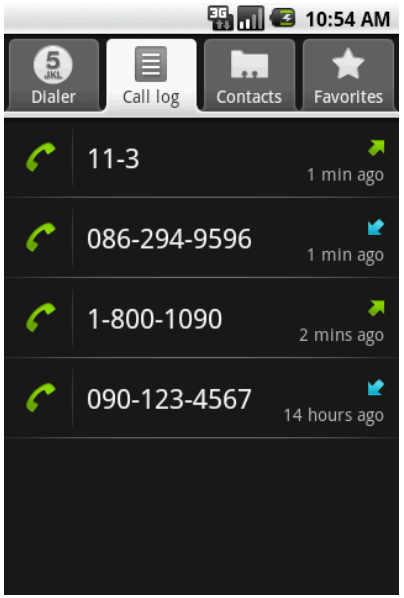
http://developer.android.com/resources/tutorials/views/hello-formstuff.html

http://developer.android.com/resources/tutorials/views/hello-webview.html

List

- Rất hay dùng trong Android. Đặc biệt các ứng dụng cần lưu trữ và hiển thị nhiều dữ liệu.
- List là một danh sách các view thông thường có cùng dạng layout đặt liền nhau.





ApiDemos

- Mở ứng dụng ApiDemos đã có sẵn:
  - New android project → Create project from existing source → Browse → mở thư mục SDK → platforms → android-3 → samples → ApiDemos → OK → Finish.
  - Mở thư mục project trong eclipse → src → mở com.example.android.apis.view