

PHÁT TRIỂN ỨNG DỤNG TRÊN THIẾT BỊ DI ĐỘNG

Hồ Văn Tú

Bộ môn Tin học ứng dụng
Khoa CNTT và truyền thông

hvtu@ctu.edu.vn

Chương 3

Xây dựng giao diện ứng dụng

NỘI DUNG

- Những nội dung cơ bản
- Các Layout thông dụng
- Các điều khiển thông dụng
- Điều khiển dạng danh sách
- Tạo và sử dụng thực đơn
- Tạo Activity dạng Tab (Tabbed Activity)
- Tạo Activity điều hướng (Navigation Drawer Activity)
- Thiết kế giao diện ứng dụng cho nhiều loại màn hình

Những nội dung cơ bản _ Giới thiệu

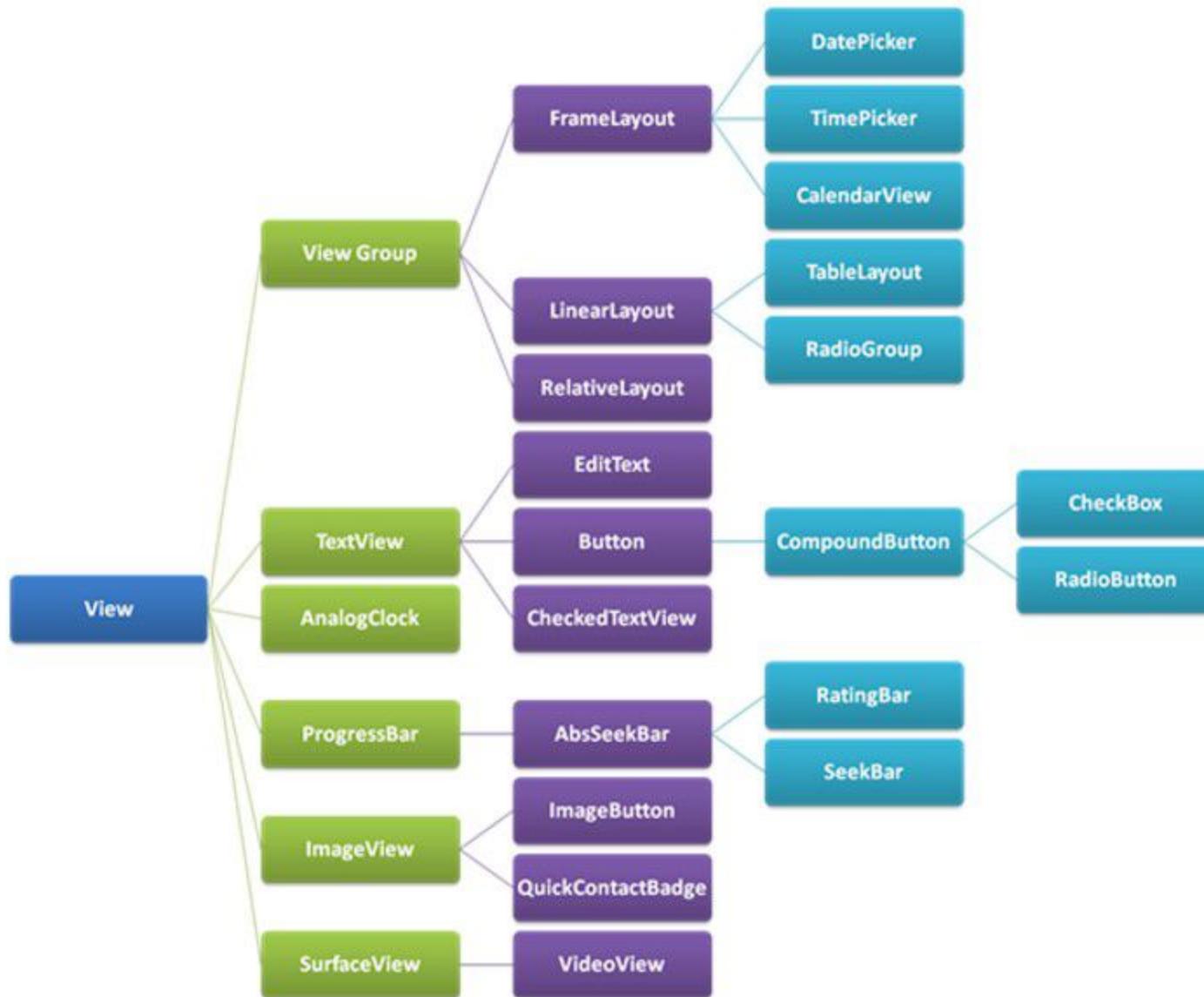
- Android dùng Activity để hiển thị màn hình
- Giao diện Activity được xây dựng từ các đối tượng Widget (gọi chung là View) và ViewGroup theo dạng cấu trúc cây
- Có thể thiết kế giao diện bằng mã lệnh java hoặc dạng xml

VD: Activity KhachHang có tập tin giao diện khachhang.xml trong thư mục res/layout

Trong sự kiện onCreate() của lớp KhachHang

```
setContentView(R.layout.khachhang);
```

View và ViewGroup



Giao diện ứng dụng (1)

UI Layout

UI Elements

UI Elements

ViewGroup

ViewGroup

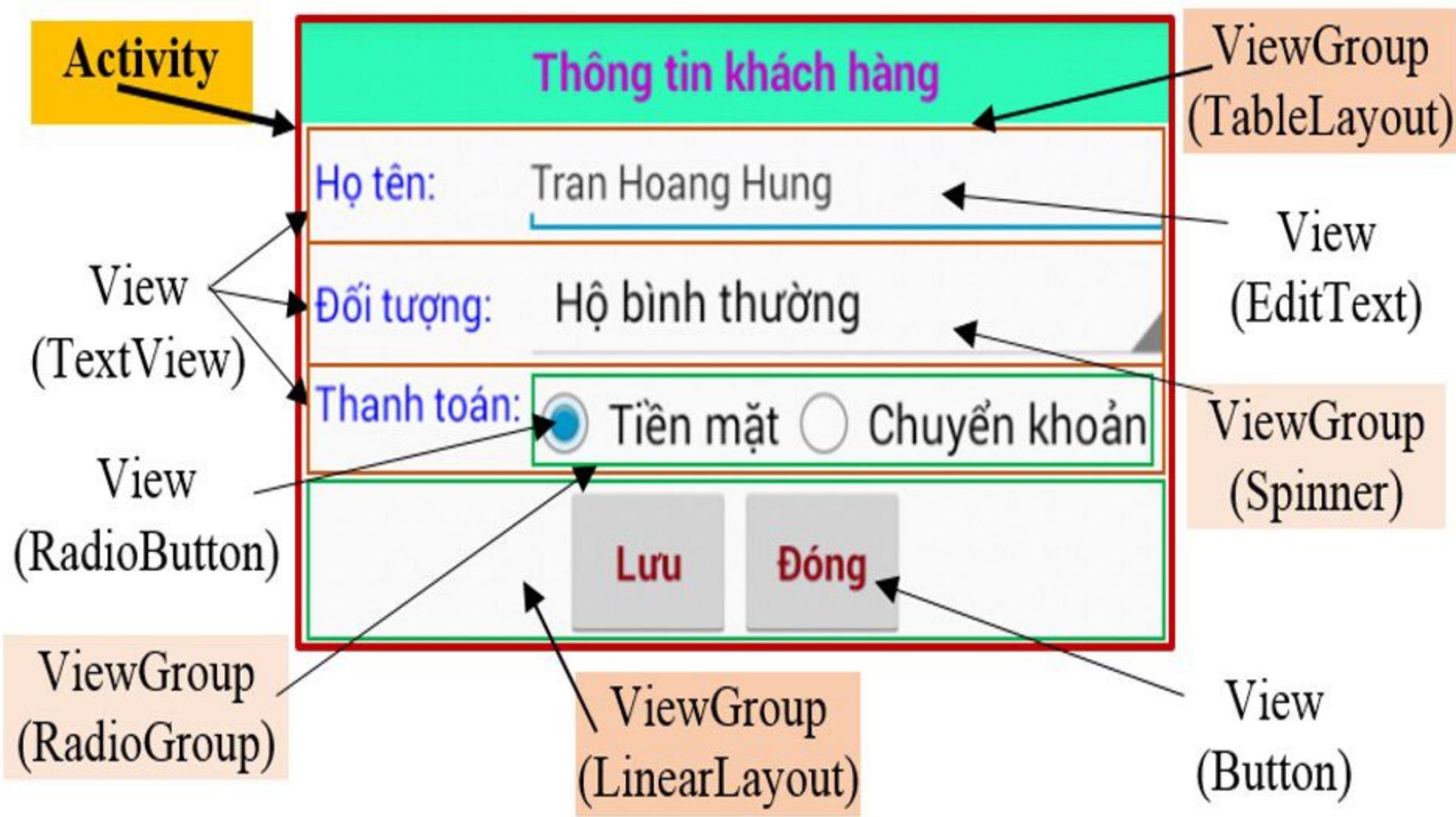
View

View

View

View

Giao diện ứng dụng (2)



Thuộc tính chung

Thuộc tính	Mô tả
id	Tên của đối tượng
layout_width	Chiều rộng/ cao: match_parent (bằng parent),
layout_height	wrap_content (vừa nội dung)
background	Hình hoặc màu nền
foreground	Màu nội dung văn bản
onClick	Khai báo tên phương thức nhận sự kiện Click
paddingLeft	Khoảng trống giữa nội dung văn bản và đường viền bên trái View. Tương tự cho paddingRight, paddingTop, paddingBottom, và padding.

Xử lý sự kiện tương tác (1)

- Sự kiện (Event): tương tác của người dùng lên đối tượng
VD: onClick, onLongClick, onFocusChange, onKey, onTouch, ...
- Có nhiều cách xử lý sự kiện (Event handling): tùy sự kiện và điều khiển sử dụng

```
<Button  
    android:id="@+id/btn_thucthi"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:onClick="ThucThiCongViec"  
    android:text="@string/thuc_thi" />
```

Xử lý sự kiện tương tác (2)

- Cách 1: khai báo phương thức trong Activity khi có thuộc tính onClick trong giao diện

```
<Button  
    android:id="@+id/btn_thucthi"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:onClick="ThucThiCongViec"  
    android:text="@string/thuc_thi" />
```

```
public void ThucThiCongViec(View view) {  
    // Mã lệnh xử lý công việc  
}
```

Xử lý sự kiện tương tác (3)

- Cách 2: khai báo yêu cầu lắng nghe và xử lý sự kiện onClick cho nút lệnh trong sự kiện onCreate của Activity

```
<Button  
    android:id="@+id/btn_thucthi"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:onClick="ThucThiCongViec"  
    android:text="@string/thuc_thi" />
```

```
Button btnThucThi = (Button) findViewById(R.id.btn_thucthi);  
btnThucThi.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View view) {  
        // Mã lệnh xử lý công việc  
    }  
});
```

Xử lý sự kiện tương tác (4)

- **Cách 3:** khai báo yêu cầu lắng nghe và xử lý sự kiện onClick cho nút lệnh thông qua lớp lắng nghe cục bộ
 - Trong sự kiện onCreate của Activity

```
Button btnThucThi = (Button) findViewById(R.id.btn_thucthi);  
btnThucThi.setOnClickListener(new ThucThi());
```

- Trong Activity: khai báo lớp lắng nghe cục bộ xử lý sự kiện
- ```
private class ThucThi implements View.OnClickListener{
 @Override
 public void onClick(View view) {
 // Mã lệnh xử lý công việc
 }
}
```

# Xử lý sự kiện tương tác (5)

- **Cách 4:** thực thi giao diện OnClickListener
  - Trong khai báo lớp: thực thi giao diện OnClickListener

```
public class MainActivity extends AppCompatActivity
 implements View.OnClickListener {
 ...
}
```

- Trong sự kiện onCreate của Activity

```
Button btnThucThi = (Button) findViewById(R.id.btn_thucthi);
btnThucThi.setOnClickListener(this);
```

- Trong Activity: ghi đè phương thức xử lý sự kiện onClick

```
@Override
public void onClick(View view) {
 // Mã lệnh xử lý công việc
}
```

# Đơn vị đo thông dụng

| Đơn vị | Ý nghĩa                                                                                                 |
|--------|---------------------------------------------------------------------------------------------------------|
| dp     | Density-independent Pixels: với màn hình 160dpi thì 1dp = 1px, thường sử dụng cho chiều dài/ chiều rộng |
| sp     | Scale-independent Pixels: sử dụng cho kích thước Font                                                   |
| dpi    | Dots per inch: số điểm ảnh (pixels) trên 1 inch màn hình                                                |
| pt     | Points = 1/72 inch: phụ thuộc vào kích thước màn hình                                                   |
| px     | Pixels: pixel thực trên màn hình → điều khiển hiển thị thay đổi trên các màn hình khác nhau             |
| in     | Inchs: phụ thuộc vào kích thước màn hình                                                                |
| mm     | Millimeters: phụ thuộc vào kích thước màn hình                                                          |
| ems    | Đơn vị đo tương đối theo kích thước Font đang chọn. Khi chọn Font Size là 12 thì 1ems = 12pt            |

# Kích thước, độ phân giải, và mật độ màn hình

- Kích thước màn hình (Screen Size): là độ dài đường chéo (*small, normal, large, extra-large*)
- Độ phân giải màn hình (Screen Resolution): là số điểm ảnh (pixels) như HD (1280x720px), Full HD (1920x1080px), và Ultra HD hay 4K (3840x2160px)
- Mật độ màn hình (Screen density): là số điểm ảnh (pixels) trên 1 inch màn hình

ldpi\_120dpi

mdpi\_160dpi

hdpi\_240dpi,

xhdpi\_320dpi

xxhdpi\_480dpi

xxxhdpi\_640dpi

# Tạo và sử dụng Style (1)

- Gán các thuộc tính giống nhau cho các thành phần
- Đặt trong thẻ resources của tập tin styles.xml (res\values)

```
<style name="TitleStyle" parent="@android:style/TextAppearance.Small">
 <item name="android:layout_width">match_parent</item>
 <item name="android:layout_height">30dp</item>
 <item name="android:gravity">center</item>
 <item name="android:background">#33FFBB</item>
 <item name="android:textColor">#CC00CC</item>
 <item name="android:textStyle">bold</item>
 <item name="android:textSize">15sp</item>
</style>

<style name="LabelStyle" parent="@android:style/TextAppearance.Small">
 <item name="android:layout_width">wrap_content</item>
 <item name="android:layout_height">wrap_content</item>
 <item name="android:paddingLeft">5dp</item>
 <item name="android:textColor">#0000FF</item>
</style>
```

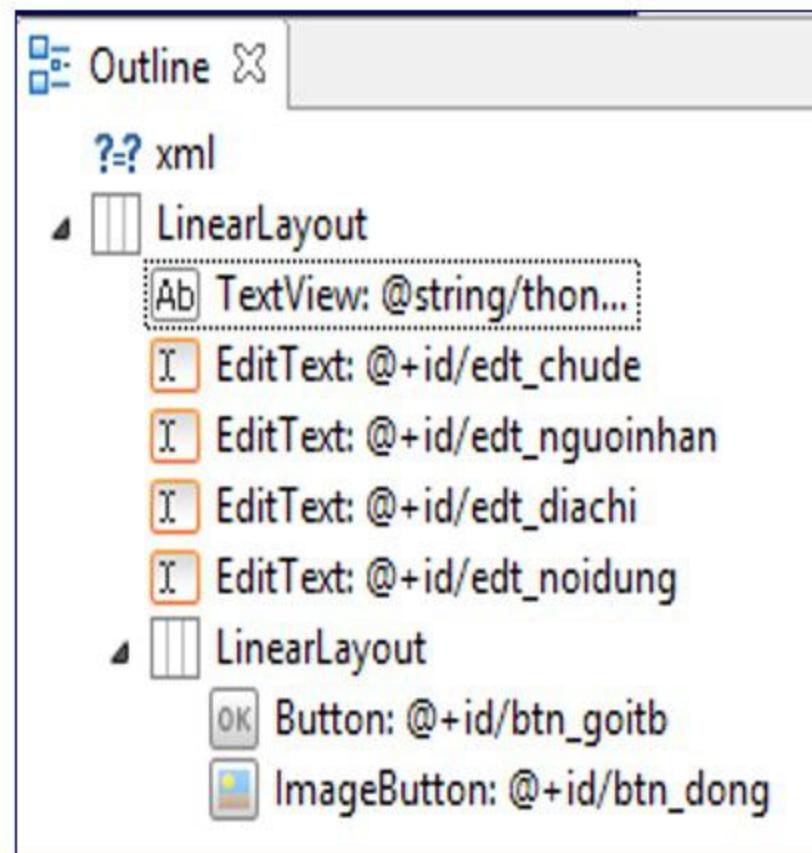
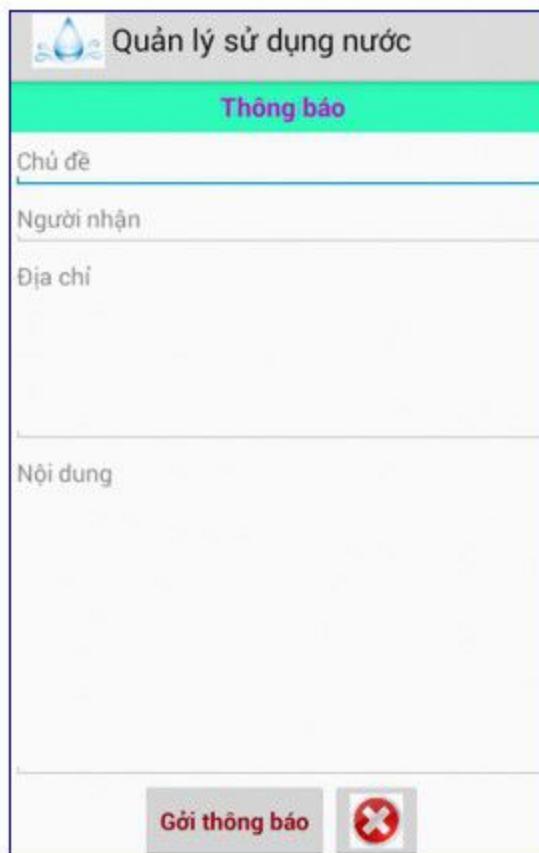
## Tạo và sử dụng Style (2)

```
8 <TextView
9 android:layout_width="match_parent"
10 android:layout_height="30dp"
11 android:background="#33FFBB"
12 android:gravity="center"
13 android:text="@string/thong_bao"
14 android:textColor="#CC00CC"
15 android:textSize="15sp"
16 android:textStyle="bold" />
```

```
8 <TextView
9 style="@style/TitleStyle"
10 android:text="@string/thong_bao" />
```

# Linear layout (1)

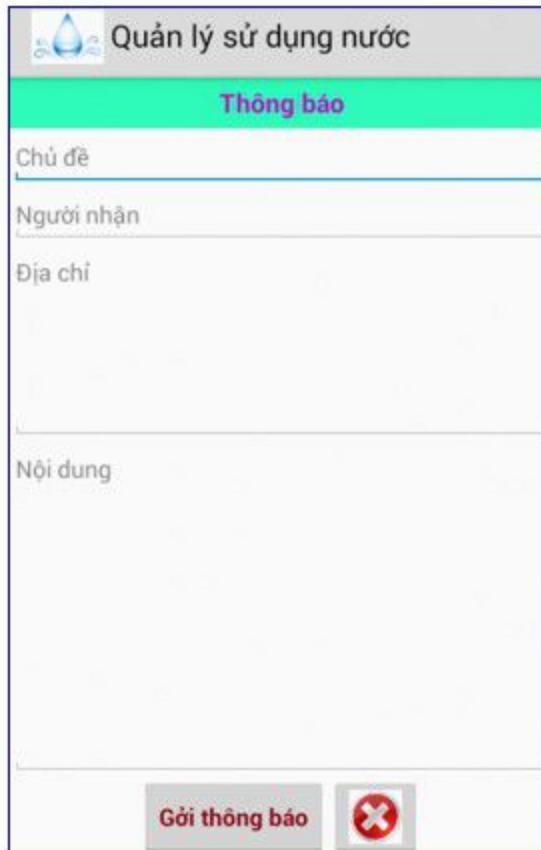
- Bố trí các điều khiển tuần tự theo hàng ngang/ hàng dọc
- Các LinearLayout có thể lồng nhau



# Linear layout (2)

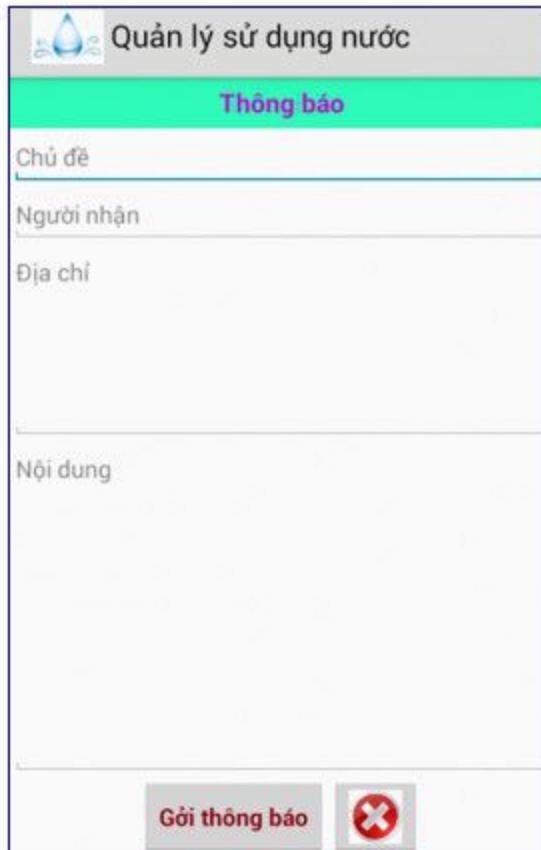
| Thuộc tính     | Ý nghĩa                                                                                                        |
|----------------|----------------------------------------------------------------------------------------------------------------|
| orientation    | Hướng bố trí: vertical (dọc), horizontal (ngang)                                                               |
| layout_width   | Chiều rộng/ cao: match_parent (bằng parent),                                                                   |
| layout_height  | wrap_content (vừa nội dung)                                                                                    |
| layout_weight  | Tỷ lệ diện tích hiển thị trên màn hình: 1, 2, ...                                                              |
| layout_gravity | Canh điều khiển nằm trong thành phần: top, bottom, left, center, right, center_vertical, center_horizontal, .. |
| gravity        | Canh nội dung nằm trong thành phần: top, bottom, left, center, right, center_vertical, center_horizontal, ..   |

# Linear layout (3)



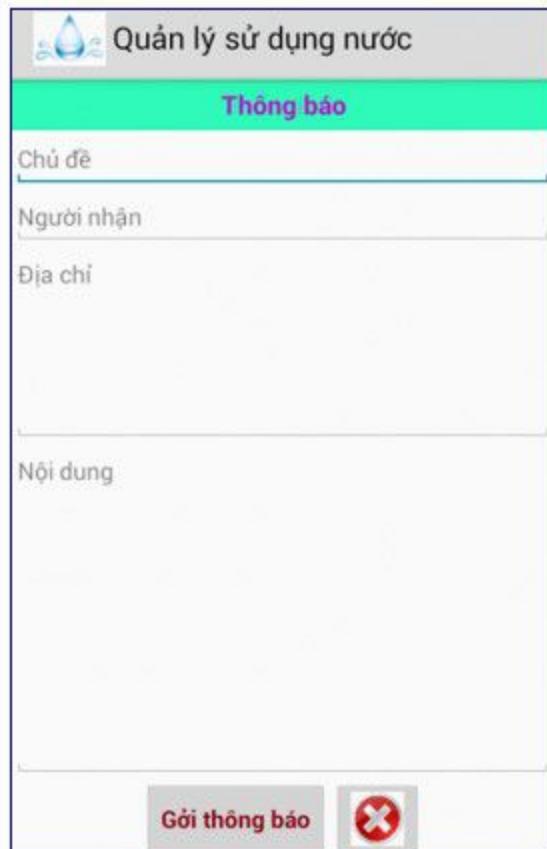
```
1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3 android:layout_width="match_parent"
4 android:layout_height="match_parent"
5 android:background="@color/bgr_color"
6 android:orientation="vertical" >
7
8 <TextView
9 android:layout_width="match_parent"
10 android:layout_height="30dp"
11 android:background="#33FFBB"
12 android:gravity="center"
13 android:text="@string/thong_bao"
14 android:textColor="#CC00CC"
15 android:textSize="15sp"
16 android:textStyle="bold" />
17
18 <EditText
19 android:id="@+id/edt_chude"
20 android:layout_width="match_parent"
21 android:layout_height="wrap_content"
22 android:hint="@string/chu_de"
23 android:paddingLeft="3dp" />
```

# Linear layout (4)



```
25 <EditText
26 android:id="@+id/edt_nguoinhhan"
27 android:layout_width="match_parent"
28 android:layout_height="wrap_content"
29 android:hint="@string/nguoinhhan"
30 android:paddingLeft="3dp" />
31
32 <EditText
33 android:id="@+id/edt_diachi"
34 android:layout_width="match_parent"
35 android:layout_height="wrap_content"
36 android:layout_weight="1"
37 android:gravity="top"
38 android:hint="@string/dia_chi"
39 android:paddingLeft="3dp" />
40
41 <EditText
42 android:id="@+id/edt_noidung"
43 android:layout_width="match_parent"
44 android:layout_height="wrap_content"
45 android:layout_weight="2"
46 android:gravity="top"
47 android:hint="@string/noi_dung"
48 android:paddingLeft="3dp" />
```

# Linear layout (5)



```
50 <LinearLayout
51 android:layout_width="match_parent"
52 android:layout_height="wrap_content"
53 android:background="@color/bgr_color"
54 android:gravity="center"
55 android:orientation="horizontal" >
56
57 <Button
58 android:id="@+id	btn_goitb"
59 android:layout_width="wrap_content"
60 android:layout_height="wrap_content"
61 android:onClick="HienThiThongBao"
62 android:text="@string/goi_tb"
63 android:textColor="#880011"
64 android:textStyle="bold" />
65
66 <ImageButton
67 android:id="@+id	btn_dong"
68 android:layout_width="55dp"
69 android:layout_height="48dp"
70 android:contentDescription="@string/dong"
71 android:onClick="DongActivity"
72 android:scaleType="fitXY"
73 android:src="@drawable/close01" />
74 </LinearLayout>
75 </LinearLayout>
```

# AlertDialog (1)

- Thuộc lớp android.app.AlertDialog  
(hoặc android.support.v7.app.AlertDialog),  
kế thừa android.app.Dialog,  
thực thi giao diện android.content.DialogInterface
- Dùng để hiển thị thông tin (dạng đơn giản)



# AlertDialog (2)

- Tạo và sử dụng AlertDialog
  - Khởi tạo đối tượng AlertDialog.Builder
  - Gán các thuộc tính: tiêu đề, Icon, chuỗi nội dung, ...
  - Gán thuộc tính và mã lệnh cho nút lệnh lựa chọn xử lý
  - Thực thi tạo (create) và hiển thị (show) AlertDialog

# AlertDialog (3)

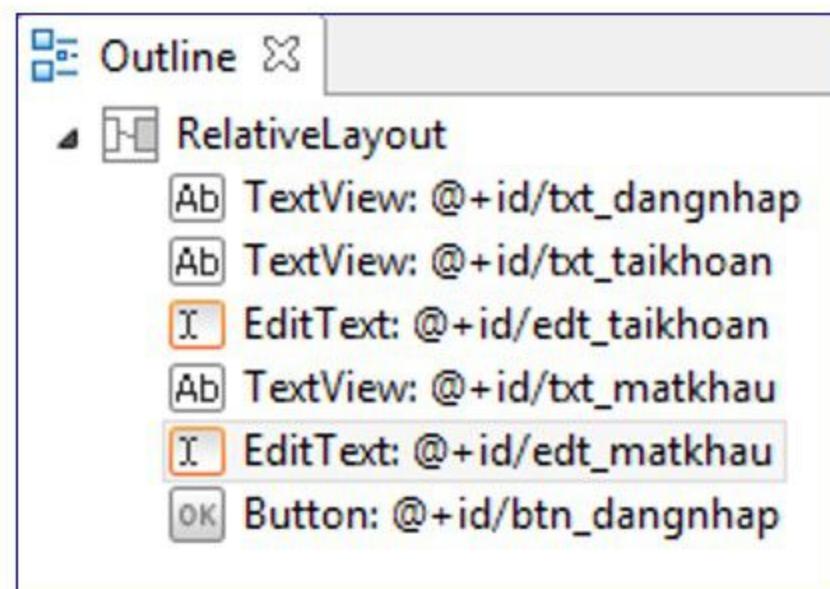
```
AlertDialog.Builder bdrThongbao =
 new AlertDialog.Builder(this);
bdrThongbao.setTitle("Xác nhận");
bdrThongbao.setIcon(R.mipmap.send);
bdrThongbao.setPositiveButton("Đồng ý",
 new DialogInterface.OnClickListener() {
 @Override
 public void onClick(DialogInterface dialog, int which) {
 // Xử lý khi đồng ý thực hiện
 Toast.makeText(ThongBao.this,
 "Đồng ý", Toast.LENGTH_SHORT).show();
 }
 });
```

## AlertDialog (4)

```
bdrThongbao.setNegativeButton("Hủy bỏ",
 new DialogInterface.OnClickListener() {
 @Override
 public void onClick(DialogInterface dialog, int which) {
 Toast.makeText(ThongBao.this,
 "Hủy bỏ", Toast.LENGTH_SHORT).show();
 dialog.cancel();
 }
 });
String strNoiDung = "Đồng ý gửi dữ liệu?";
bdrThongbao.setMessage(strNoiDung);
bdrThongbao.create().show();
```

# Relative layout (1)

- Cho phép thiết lập mối liên hệ hiển thị giữa các thành phần (phân tử) với nhau
- Phân tử hiện tại: so sánh vị trí hiện tại với phân tử chứa nó hoặc các phân tử xung quanh



## Relative layout (2)

| Thuộc tính                                        | Ý nghĩa                                                      |
|---------------------------------------------------|--------------------------------------------------------------|
| layout_above<br>layout_below                      | Phần tử (PT) hiện tại nằm trên (dưới) PT có id được chỉ ra   |
| layout_alignBaseline                              | PT hiện tại nằm cùng dòng PT có id chỉ ra                    |
| layout_alignLeft<br>layout_alignRight             | Canh lè cạnh trái (phải) PT hiện tại và PT có id được chỉ ra |
| layout_alignParentLeft<br>layout_alignParentRight | true: PT hiện tại canh lè trái (phải) so với PT chứa nó      |
| layout_toLeftOf<br>layout_toRightOf               | PT hiện tại nằm bên trái (phải) PT có id được chỉ ra         |

# Relative layout (3)

```
6 <TextView
7 android:id="@+id/txt_dangnhap"
8 style="@style/TitleStyle"
9 android:text="@string/dang_nhap_ht" />
10
11 <TextView
12 android:id="@+id/txt_taikhoan"
13 android:layout_width="wrap_content"
14 android:layout_height="wrap_content"
15 android:layout_alignParentLeft="true"
16 android:layout_below="@+id/txt_dangnhap"
17 android:paddingLeft="5sp"
18 android:paddingTop="10sp"
19 android:text="@string/tai_khoan"
20 android:textColor="#0000FF" />
21
22 <EditText
23 android:id="@+id/edt_taikhoan"
24 android:layout_width="wrap_content"
25 android:layout_height="wrap_content"
26 android:layout_toRightOf="@+id/txt_taikhoan"
27 android:layout_alignBaseline="@+id/txt_taikhoan"
28 android:layout_alignParentRight="true"
29 android:layout_below="@+id/txt_dangnhap" />
```

# Relative layout (4)

```
31 <TextView
32 android:id="@+id/txt_matkhau"
33 android:layout_width="wrap_content"
34 android:layout_height="wrap_content"
35 android:layout_alignParentLeft="true"
36 android:layout_alignBaseline="@+id(edt_matkhau"
37 android:layout_below="@+id/txt_taikhoan"
38 android:paddingLeft="5sp"
39 android:text="@string/mat_khau"
40 android:textColor="#0000FF" />

41
42 <EditText
43 android:id="@+id(edt_matkhau"
44 android:layout_width="wrap_content"
45 android:layout_height="wrap_content"
46 android:layout_alignParentRight="true"
47 android:layout_below="@+id/txt_taikhoan"
48 android:layout_toRightOf="@+id/txt_matkhau"
49 android:layout_alignLeft="@+id/txt_taikhoan"
50 android:inputType="textPassword" />

51
52 <Button
53 android:id="@+id/btn_dangnhap"
54 style="@style/ButtonStyle"
55 android:layout_below="@+id/txt_matkhau"
56 android:layout_centerHorizontal="true"
57 android:text="@string/dang_nhap" />
```

# Table layout (1)

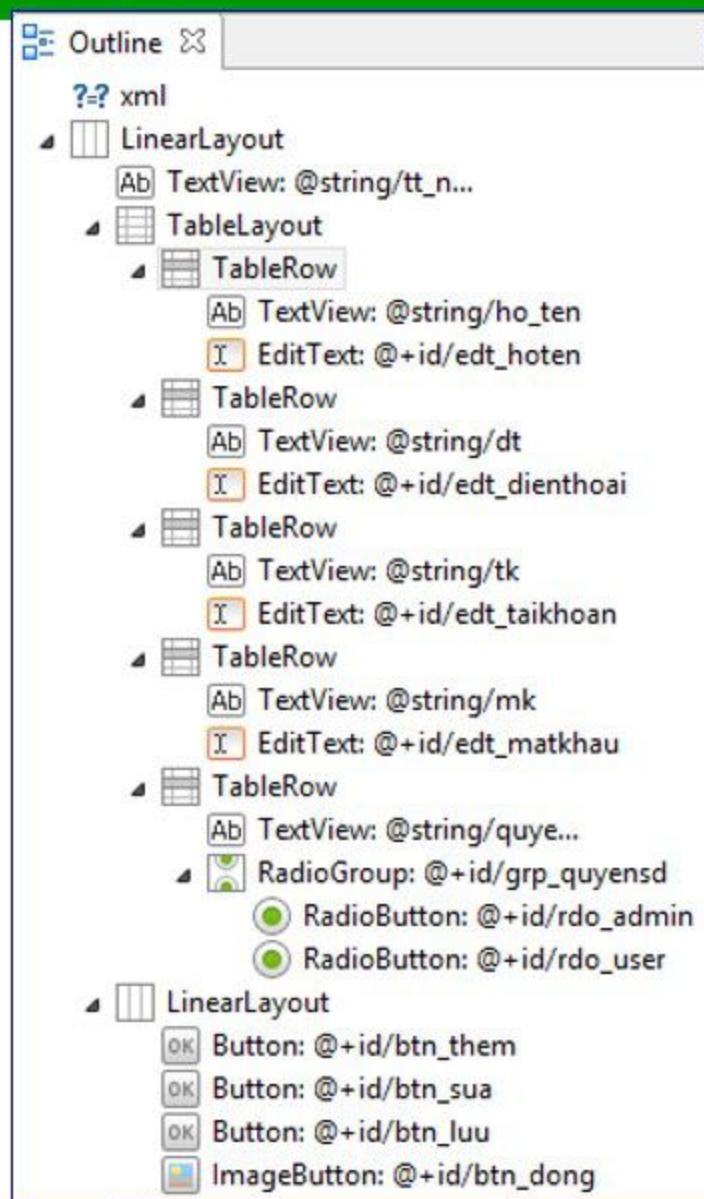
- Thể hiện các thành phần dưới dạng bảng, các ô được tạo có thể chứa các View và Layout khác
- Thẻ <TableRow>: tạo ra các hàng
- Có thể kết hợp với các Layout khác



## Table layout (2)

| Thuộc tính      | Ý nghĩa                                      |
|-----------------|----------------------------------------------|
| stretchColumns  | Xác định cột cần tăng độ rộng: 0, 1, ..., *  |
| shrinkColumns   | Xác định cột cần giảm độ rộng: 0, 1, ..., *  |
| collapseColumns | Xác định cột không hiển thị: 0, 1, 2, ...    |
| layout_column   | Gán vị trí cột cho điều khiển: 0, 1, 2, ...  |
| layout_span     | Xác định số cột điều khiển sử dụng (mở rộng) |

# Table layout (3)



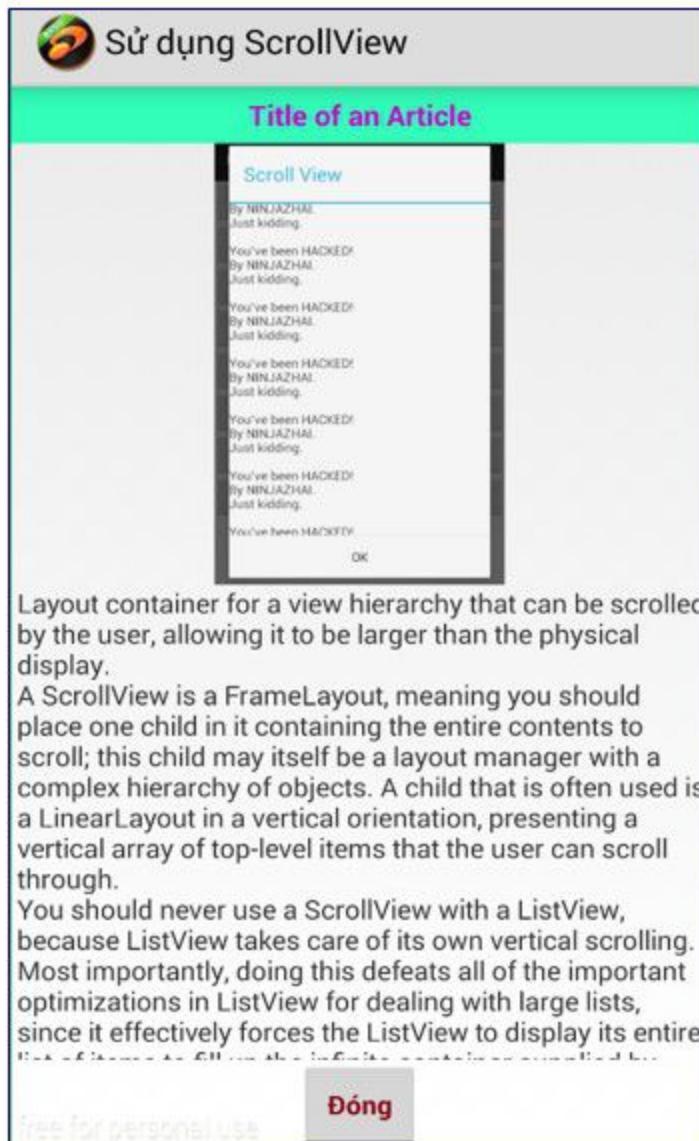
# Table layout (4)

```
8 <TextView
9 style="@style/Titlestyle"
10 android:text="@string/tt_nhan_vien" />
11
12 <TableLayout
13 android:layout_width="match_parent"
14 android:layout_height="wrap_content" >
15
16 <TableRow
17 android:layout_width="match_parent"
18 android:layout_height="wrap_content" >
19
20 <TextView
21 style="@style/Labelstyle"
22 android:text="@string/ho_ten" />
23
24 <EditText
25 android:id="@+id/edt_hoten"
26 style="@style/TextStyle"
27 android:ems="15" />
28 </TableRow>
```

# ScrollView (1)

- Thanh cuộn đứng: cho phép hiển thị thành phần có nội dung dài
- Chỉ chứa 1 thành phần con → kết hợp Layout khác
- HorizontalScrollView: thanh cuộn ngang

# ScrollView (2)



Layout container for a view hierarchy that can be scrolled by the user, allowing it to be larger than the physical display.

A ScrollView is a FrameLayout, meaning you should place one child in it containing the entire contents to scroll; this child may itself be a layout manager with a complex hierarchy of objects. A child that is often used is a LinearLayout in a vertical orientation, presenting a vertical array of top-level items that the user can scroll through.

You should never use a ScrollView with a ListView, because ListView takes care of its own vertical scrolling. Most importantly, doing this defeats all of the important optimizations in ListView for dealing with large lists, since it effectively forces the ListView to display its entire

Đóng

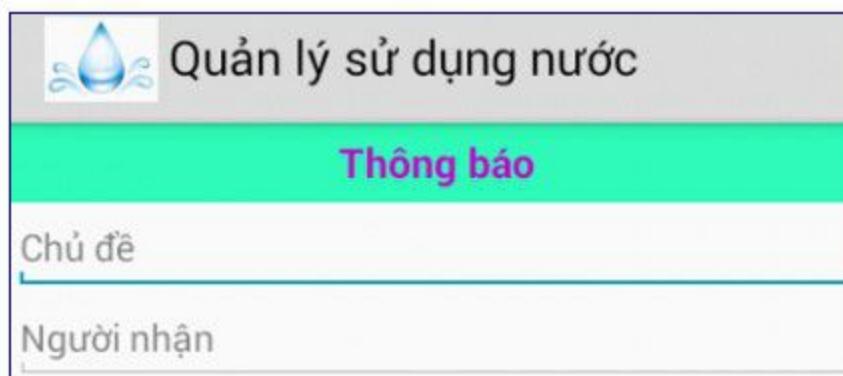
# ScrollView (3)

```
8 <TextView
9 style="@style/TitleStyle"
10 android:text="@string/tieu_de" />
11
12 <ScrollView
13 android:layout_width="match_parent"
14 android:layout_height="wrap_content"
15 android:layout_weight="1" >
16
17 <LinearLayout
18 android:layout_width="match_parent"
19 android:layout_height="wrap_content"
20 android:orientation="vertical" >
21
22 <ImageView
23 android:layout_width="150dp"
24 android:layout_height="250dp"
25 android:layout_gravity="center"
26 android:contentDescription="@string/scrollview"
27 android:scaleType="fitCenter"
28 android:src="@drawable/scrollview" />
```

# TextView (1)

- Nhận hiển thị thông tin, thuộc android.widget.TextView

```
<TextView
 android:id="@+id/txt_tieude"
 style="@style/TitleStyle"
 android:text="@string/thong_bao" />
```



## TextView (2)

- Nhận hiển thị thông tin, thuộc android.widget.TextView

| Thuộc tính | Ý nghĩa                                      |
|------------|----------------------------------------------|
| text       | Nội dung văn bản hiển thị                    |
| textSize   | Kích thước Font                              |
| textStyle  | Kiểu Font: normal, bold, italic, bold italic |
| textColor  | Màu văn bản                                  |

# TextView (3)

- **Các lệnh cơ bản**

- Lấy control theo Id

```
TextView txtTieuDe = (TextView) findViewById(R.id.txt_tieude);
```

- Gán giá trị cho EditText

```
txtTieuDe.setText("Gán giá trị cho TextView");
```

- Lấy dữ liệu bên trong EditText

```
String strTieuDe = txtTieuDe.getText().toString();
```

# EditText và TextInputLayout (1)

- EditText thuộc lớp android.widget.EditText, kế thừa từ TextView, cho phép nhập/ chỉnh sửa dữ liệu
- TextInputLayout thuộc lớp android.support.design.widget.TextInputLayout, kế thừa từ View\ ViewGroup\ LinearLayout



Android EditText

Nội dung

Text

Lỗi nhập nội dung!

# EditText và TextInputLayout (2)

- Cần khai báo hỗ trợ sử dụng TextInputLayout

```
dependencies {
 ...
 compile 'com.android.support:appcompat-v7:26.1.0'
 compile 'com.android.support:design:26.1.0'
}
```

# EditText và TextInputLayout (3)

- Thuộc tính thông dụng

```
<EditText
 android:id="@+id/edt_chude"
 android:layout_width="match_parent"
 android:layout_height="wrap_content"
 android:paddingLeft="3dp"
 android:hint="@string/chu_de" />
```

| Thuộc tính | Ý nghĩa                                                                                    |
|------------|--------------------------------------------------------------------------------------------|
| enable     | Cho phép thao tác (nhập) hay không: true, false                                            |
| hint       | Lời gợi ý                                                                                  |
| inputType  | Kiểu dữ liệu nhập: text, number, numberDecimal, phone, textPassword, textEmailAddress, ... |
| maxLength  | Giới hạn độ dài chuỗi nhập                                                                 |

# EditText và TextInputLayout (4)

```
<android.support.design.widget.TextInputLayout
 android:id="@+id/layout_noidung"
 android:layout_width="match_parent"
 android:layout_height="wrap_content"
 android:layout_marginTop="7dp">

 <EditText
 android:id="@+id/edt_noidung"
 android:layout_width="match_parent"
 android:layout_height="wrap_content"
 android:hint="@string/noi_dung"
 android:textColor="#0000FF" />

</android.support.design.widget.TextInputLayout>
```

# EditText và TextInputLayout (5)

- **Các lệnh cơ bản**

- + Nhận điều khiển theo Id

```
EditText edtNoiDung = (EditText)
 findViewById(R.id.edt_noidung);

TextInputLayout layoutNoiDung = (TextInputLayout)
 findViewById(R.id.layout_noidung);
```

- + Gán giá trị cho EditText

```
edtNoiDung.setText("Gán giá trị cho EditText");
```

- + Nhận nội dung trong EditText

```
String strNoiDung =
 edtNoiDung.getText().toString().trim();
```

# EditText và TextInputLayout (6)

- **Các lệnh cơ bản**

- + Cho phép (hoặc không cho phép) thao tác (nhập) trong EditText

```
edtNoiDung.setEnabled(true);
```

- + Gán tiêu điểm cho EditText

```
edtNoiDung.requestFocus();
```

- + Chọn nội dung trong EditText

```
edtNoiDung.selectAll();
```

# EditText và TextInputLayout (7)

- Các lệnh cơ bản

- + Gán thông báo lỗi cho TextInputLayout

```
String strNoiDung = edtNoiDung.getText().toString().trim();
if(strNoiDung.length() == 0) {
 layoutNoiDung.setError("Lỗi chưa nhập nội dung!");
 edtNoiDung.requestFocus();
 return;
}
else
 layoutNoiDung.setError(null);
```

# Button và ImageButton (1)

- Button thuộc lớp android.widget.Button, kế thừa từ TextView
- ImageButton thuộc lớp android.widget.ImageButton, kế thừa từ ImageView
- Được thể hiện ở dạng văn bản hoặc hình ảnh
- Cho phép nhận và phản hồi tương tác nhấn (Click hay Touch) từ người dùng



## Button và ImageButton (2)

```
<Button
 android:id="@+id	btn_goithb"
 android:layout_width="wrap_content"
 android:layout_height="wrap_content"
 android:textColor="#880011"
 android:textStyle="bold"
 android:text="@string/goi_tb"
 android:onClick="GoiThongBao" />

<ImageButton
 android:id="@+id	btn_dong"
 android:layout_width="70dp"
 android:layout_height="50dp"
 android:scaleType="fitXY"
 android:contentDescription="@string/dong"
 android:src="@mipmap/close"
 android:onClick="DongActivity" />
```

# Button và ImageButton (3)

- **Các lệnh cơ bản**

- + Nhận điều khiển theo Id

```
Button btnGoiTB =
 (Button) findViewById(R.id.btn_goitb);

ImageButton btnDong =
 (ImageButton) findViewById(R.id.btn_dong);
```

- + Khai báo phương thức xử lý sự kiện onClick

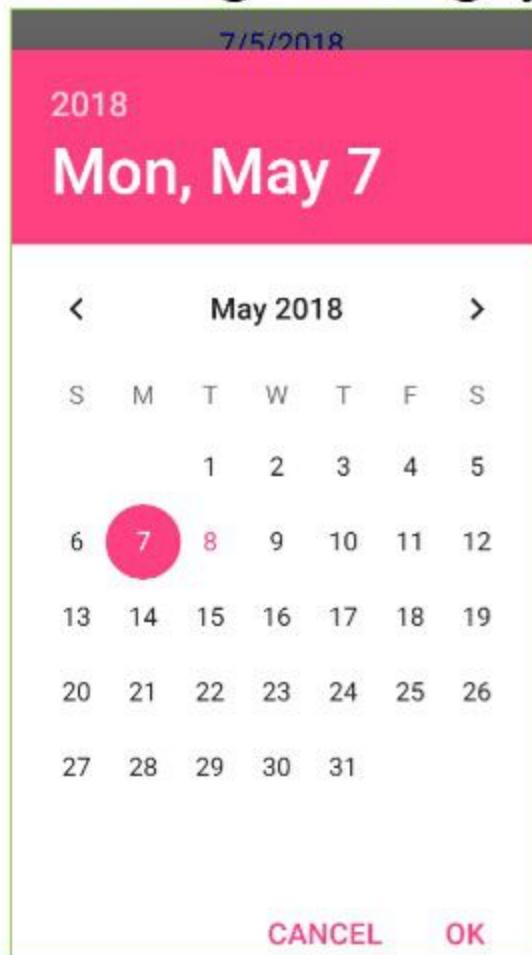
```
public void GoiThongBao(View view) {
 // Mã lệnh xử lý công việc
}
```

- + Cho phép (hoặc không cho phép) tương tác với Button

```
btnDong.setEnabled(true);
```

# DatePickerDialog (1)

- Thuộc lớp android.app.DatePickerDialog, kế thừa AlertDialog, hỗ trợ chọn giá trị ngày



# DatePickerDialog (2)

- **Tạo và sử dụng**
  - Giao diện: tạo TextView hiển thị ngày,
  - Khai báo biến thuộc lớp: biến TextView và biến lắng nghe sự kiện chọn ngày (DatePickerDialog.OnDateSetListener)
  - Trong sự kiện onCreate(): nhận và gán TextView hiển thị ngày hiện tại,
  - Trong sự kiện onClick của TextView:
    - Khai báo biến lắng nghe sự kiện chọn ngày, cập nhật cho TextView hiển thị ngày được chọn,
    - Khai báo và cho hiển thị hộp thoại chọn ngày.

# DatePickerDialog (3)

- **Giao diện**

```
<TextView
 android:id="@+id/txt_date"
 style="@style/LabelStyle"
 android:layout_gravity="center"
 android:onClick="ChonNgay"/>
```

# DatePickerDialog (4)

- **Mã lệnh**

```
public class SuDungDialog extends AppCompatActivity {
 private TextView txtDate;
 private DatePickerDialog.OnDateSetListener myDateSetListener;

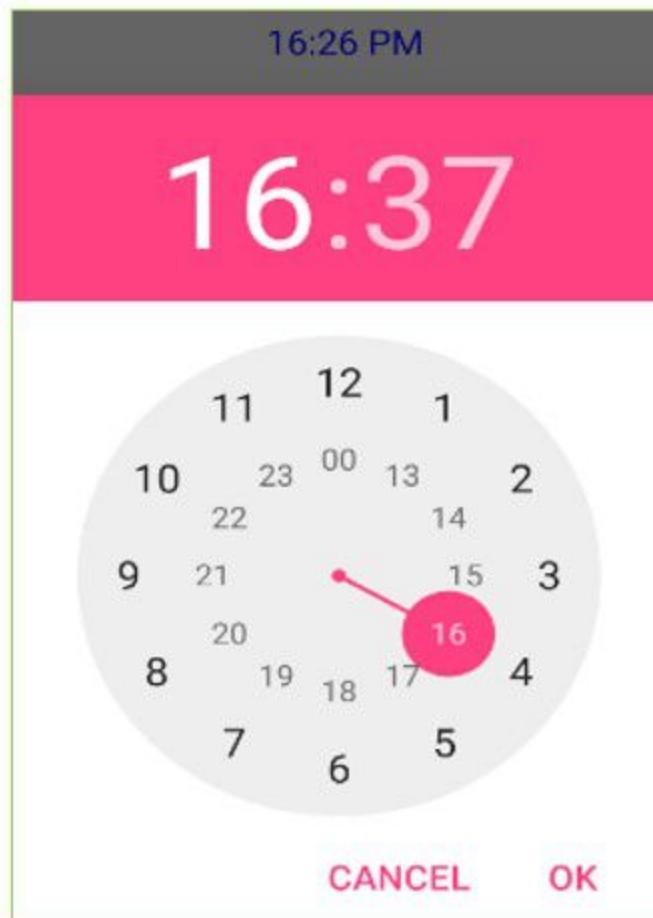
 @Override
 protected void onCreate(Bundle savedInstanceState) {
 super.onCreate(savedInstanceState);
 setContentView(R.layout.using_dialog);
 txtDate = (TextView) findViewById(R.id.txt_date);
 Calendar calendar = Calendar.getInstance();
 SimpleDateFormat dateFormat =
 new SimpleDateFormat("dd/MM/yyyy");
 txtDate.setText(dateFormat.format(calendar.getTime()));
 }
}
```

# DatePickerDialog (5)

```
public void ChonNgay(View view) {
 Calendar calendar = Calendar.getInstance();
 int year = calendar.get(Calendar.YEAR);
 int month = calendar.get(Calendar.MONTH);
 int day = calendar.get(Calendar.DAY_OF_MONTH);
 myDateSetListener =
 new DatePickerDialog.OnDateSetListener() {
 @Override
 public void onDateSet(DatePicker datePicker,
 int year, int month, int day) {
 month = month + 1;
 String date = day + "/" + month + "/" + year;
 txtDate.setText(date);
 }
 };
 DatePickerDialog dialog = new DatePickerDialog(
 this, myDateSetListener, year, month, day);
 dialog.setTitle("Chọn ngày thực hiện");
 dialog.show();
}
```

# TimePickerDialog (1)

- Thuộc lớp android.app.TimePickerDialog, kế thừa AlertDialog, hỗ trợ chọn giá trị giờ



# TimePickerDialog (2)

- **Tạo và sử dụng**

- Giao diện: tạo TextView hiển thị giờ,
- Khai báo biến thuộc lớp: biến TextView và biến lắng nghe sự kiện chọn giờ (TimePickerDialog.OnTimeSetListener)
- Trong sự kiện onCreate(): nhận và gán TextView hiển thị giờ hiện tại,
- Trong sự kiện onClick của TextView:
  - Khai báo biến lắng nghe sự kiện chọn giờ, cập nhật cho TextView hiển thị giờ được chọn,
  - Khai báo và cho hiển thị hộp thoại chọn giờ.

# TimePickerDialog (3)

- Giao diện

```
<TextView
 android:id="@+id/txt_time"
 style="@style/LabelStyle"
 android:layout_gravity="center"
 android:onClick="ChonGio"/>
```

# TimePickerDialog (4)

- Mã lệnh

```
public class SuDungDialog extends AppCompatActivity {

 private TextView txtTime;
 private TimePickerDialog.OnTimeSetListener myTimeSetListener;

 @Override
 protected void onCreate(Bundle savedInstanceState) {
 super.onCreate(savedInstanceState);
 setContentView(R.layout.using_dialog);
 Calendar calendar = Calendar.getInstance();
 SimpleDateFormat timeFormat =
 new SimpleDateFormat("HH:mm");
 txtTime = (TextView) findViewById(R.id.txt_time);
 txtTime.setText(timeFormat.format(calendar.getTime()));
 }
}
```

# TimePickerDialog (5)

```
public void ChonGio(View view) {
 Calendar calendar = Calendar.getInstance();
 int hour = calendar.get(Calendar.HOUR_OF_DAY);
 int minute = calendar.get(Calendar.MINUTE);
 myTimeSetListener =
 new TimePickerDialog.OnTimeSetListener() {
 @Override
 public void onTimeSet(TimePicker timePicker,
 int hour, int minute) {
 String time = hour + ":" + minute;
 txtTime.setText(time);
 }
 };
 TimePickerDialog dialog = new TimePickerDialog(
 this, myTimeSetListener, hour, minute, true);
 dialog.setTitle("Chọn giờ thực hiện");
 dialog.show();
}
```

# CheckBox (1)

- Thuộc lớp android.widget.CheckBox, kế thừa TextView\Button\ CompoundButton, tùy chọn không bắt buộc

```
<CheckBox
 android:id="@+id/chk_hd"
 android:layout_height="wrap_content"
 android:checked="false"
 android:text="@string/hd" />

<CheckBox
 android:id="@+id/chk_3d"
 android:layout_height="wrap_content"
 android:checked="false"
 android:text="@string/threeD" />

<CheckBox
 android:id="@+id/chk_smart"
 android:layout_height="wrap_content"
 android:checked="false"
 android:text="@string/smart" />
```



## CheckBox (2)

- checked: thuộc tính trạng thái chọn lựa (true/ false)
- isChecked(): kiểm tra điều khiển có được checked hay không → true/ false
- setChecked(boolean): thiết lập checked là true/ false
- getText(): nhận chuỗi (nhãn) của điều khiển

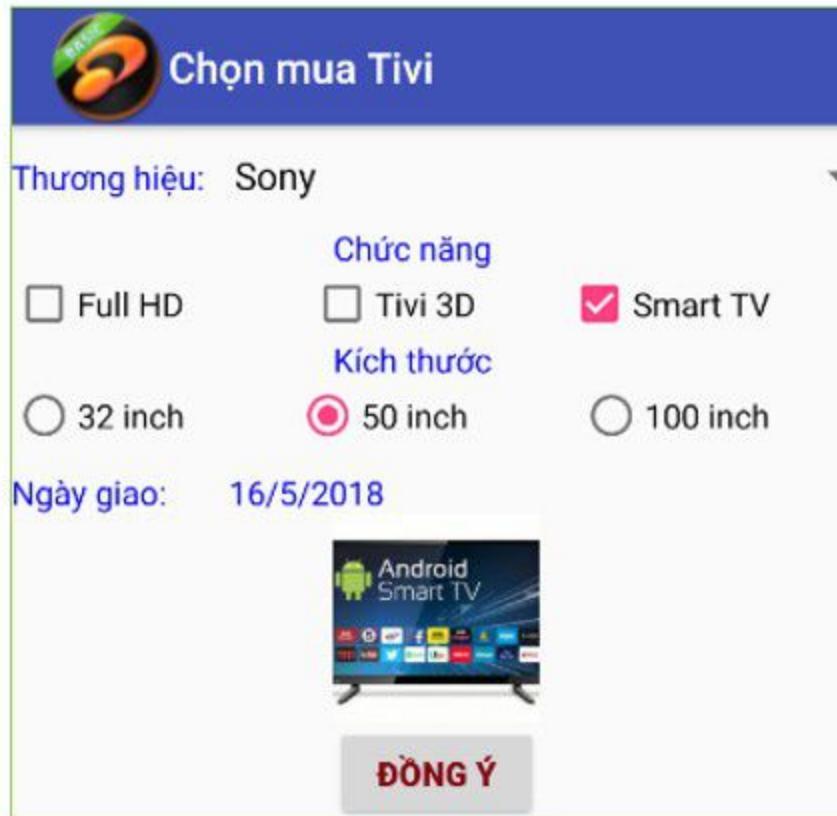
```
String strFunction;

chkSmart=(CheckBox) findViewById(R.id.chk_smart);

if(chkSmart.isChecked()){
 strFunction += chkSmart.getText() + "\n";
 chkSmart.setChecked(false);
}
}
```

# RadioGroup và RadioButton (1)

- RadioGroup kế thừa từ LinearLayout: tạo nhóm lựa chọn
- RadioButton kế thừa từ TextView\ Button\ CompoundButton, tùy chọn bắt buộc



# RadioGroup và RadioButton (2)

```
67 <RadioGroup
68 android:id="@+id/grp_size"
69 android:layout_width="match_parent"
70 android:layout_height="wrap_content"
71 android:layout_marginTop="7dp"
72 android:orientation="horizontal" >
73
74 <RadioButton
75 android:id="@+id/rdo_small"
76 android:layout_height="wrap_content"
77 android:layout_width="0dp"
78 android:checked="true"
79 android:layout_weight="1"
80 android:text="@string/small" />
81
82 <RadioButton
83 android:id="@+id/rdo_med"
84 android:layout_height="wrap_content"
85 android:layout_width="0dp"
86 android:checked="false"
87 android:layout_weight="1"
88 android:text="@string/med" />
89
```

# RadioGroup và RadioButton (3)

- getCheckedRadioButtonId(): trả về Id của điều khiển được checked trong nhóm
- Các phương thức của RadioButton tương tự của CheckBox

```
RadioGroup grpSize;
grpSize = (RadioGroup) findViewById(R.id.grp_size);
int id = grpSize.getCheckedRadioButtonId();
RadioButton rad = findViewById(id);
String strSize = rad.getText().toString().trim();
```

# ImageView (1)

- Hiển thị hình ảnh, thuộc lớp android.widget.ImageView

```
<ImageView
 android:id="@+id/img_Image"
 android:layout_width="90dp"
 android:layout_height="90dp"
 android:scaleType="centerCrop"
 android:layout_gravity="center"
 android:layout_marginTop="5dp"
 android:layout_marginRight="3dp"
 android:src="@drawable/ic_launcher"
 android:onClick="ThucThi" >
</ImageView>
```

# ImageView (2)

- Phương thức thông dụng
  - Gán tập tin hình hiển thị: setImageResource(int)
  - Gán kiểu hiển thị: setScaleType(ImageView.ScaleType.  )
- Sự kiện thông dụng: onClick

```
ImageView imageView = (ImageView) findViewById(R.id.img_Image);
imageView.setImageResource(R.drawable.khachhang);
imageView.setScaleType(ImageView.ScaleType.CENTER_CROP);
```

# Điều khiển dạng danh sách

- Cho phép chọn phần tử (item) trong danh sách  
VD: Spinner, ListView, GridView, RecyclerView
- Nguồn dữ liệu: mảng, tập tin XML, truy xuất từ CSDL
- Adapter (bộ tiếp nối): quản lý nguồn dữ liệu và ghép nối với từng dòng trong danh sách



- + Trong CSDL
- + Tập tin XML
- + Mảng dữ liệu

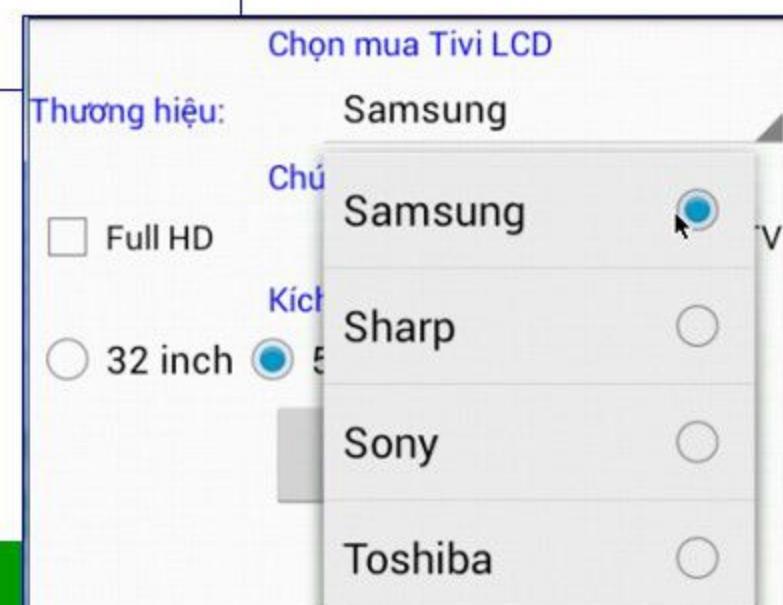
- + ListView
- + GridView
- + Spinner

# Spinner (1)

- Kế thừa từ View\ ViewGroup\ AdapterView\ AbsSpinner
- Cho phép chọn 1 phần tử (item) trong danh sách

```
<Spinner
```

```
 android:id="@+id/spn_name"
 android:layout_width="wrap_content"
 android:layout_height="wrap_content"
 android:layout_weight="2" />
```



## Spinner (2)

- Khai báo nguồn dữ liệu và biến lưu giá trị được chọn

```
String arrNames[]={"Samsung", "Sharp", "Sony", "Toshiba"};
String strName;
```

# Spinner (3)

- Gán nguồn dữ liệu và sử dụng biến Spinner

```
Spinner spnName = (Spinner) findViewById(R.id.spn_name);
ArrayAdapter<String> adapter =
 new ArrayAdapter<String>(this,
 android.R.layout.simple_spinner_dropdown_item,
 arrNames);
adapter.setDropDownViewResource(
 android.R.layout.simple_list_item_single_choice);
spnName.setAdapter(adapter);
spnName.setOnItemSelectedListener(new ChonPhanTu());
```

# Spinner (4)

- Xử lý sự kiện chọn phần tử trong danh sách

```
private class ChonPhanTu implements
 AdapterView.OnItemSelectedListener {

 public void onItemSelected(AdapterView<?> adp,
 View v, int pos, long id) {
 strName = arrNames[pos];
 }

 public void onNothingSelected(AdapterView<?> adp) {
 strName = "";
 }
}
```

# ListView (1)

- Thuộc lớp android.widget.ListView, kế thừa từ android.widget.AbsListView
- Cho phép lưu trữ và hiển thị danh sách
- Giao diện

Đà Lạt

Long Hải

Nha Trang

Vịnh Hạ Long

```
<ListView
 xmlns:android="http://schemas.android.com/apk/res/android"
 android:id="@+id/listview_diadiem"
 android:layout_width="match_parent"
 android:layout_height="wrap_content" />
```

## ListView (2)

- Khai báo biến lớp, sử dụng List<String> để lưu nguồn dữ liệu

```
List<String> listDiaDiem;
String strDiaDiem = "";
ListView listviewDiaDiem;
ArrayAdapter<String> adapter;
```

# ListView (3)

- Nhận điều khiển, khai báo nguồn dữ liệu, khai báo và gán Adapter, khai báo yêu cầu lắng nghe sự kiện chọn phần tử

```
listviewDiaDiem = (ListView)
 findViewById(R.id.listview_diadiem);
listDiaDiem = new ArrayList<String>();
listDiaDiem.add("Đà Lạt");
listDiaDiem.add("Long Hải");
listDiaDiem.add("Nha Trang");
listDiaDiem.add("Vịnh Hạ Long");
adapter = new ArrayAdapter<String>(this,
 android.R.layout.simple_list_item_1, listDiaDiem);
listviewDiaDiem.setAdapter(adapter);
listviewDiaDiem.setOnItemClickListener(
 new ChonDiaDiem());
```

## ListView (4)

- Xử lý sự kiện chọn phần tử trong danh sách

```
private class ChonDiaDiem implements
 AdapterView.OnItemClickListener{
 public void onItemClick(AdapterView<?> adp,
 View v, int pos, long id) {
 strDiaDiem = listDiaDiem.get(pos);
 }
}
```

# Custom ListView (1)

- Thiết kế danh sách với các cột theo nhu cầu sử dụng

| Danh sách khách hàng |                  |                |      |
|----------------------|------------------|----------------|------|
| STT                  | Họ tên           | Đối tượng      | HTTT |
| 1                    | Nguyen Hong Minh | Hộ nghèo       |      |
| 2                    | Le Nam Phuong    | Cơ quan        |      |
| 3                    | Nguyen An Tung   | Hộ bình thường |      |
| 4                    | Vu Thanh Lan     | Hộ bình thường |      |
| 5                    | Phan Ngoc Minh   | Sản xuất       |      |

Layout  
chính

Layout  
hiển thị  
danh sách

# Custom ListView (2)

- Thực hiện
  - Tạo Layout chính: dòng tiêu đề, ListView, ...
  - Tạo Layout hiển thị danh sách: thay cho **simple\_list\_item\_1**
  - Tạo lớp Adapter mới kế thừa ArrayAdapter<...> cho phép hiển thị nội dung danh sách
  - Gán nguồn dữ liệu cho ListView
  - Thêm sự kiện khi chọn phần tử (nếu có)
- **Ghi chú:** hướng dẫn chi tiết trong phần Bài tập

# GridView (1)

- Thuộc lớp android.widget.GridView, kế thừa từ android.widget.AbsListView
- Cho phép lưu trữ và hiển thị các phần tử theo dạng lưới

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <GridView xmlns:android="http://schemas
3 android:id="@+id/gdv_menu"
4 android:layout_width="fill_parent"
5 android:layout_height="fill_parent"
6 android:numColumns="2"
7 android:verticalSpacing="10dp"
8 android:horizontalSpacing="10dp"
9 android:stretchMode="columnWidth"
10 android:gravity="center" />
```



# GridView (2)

- Tạo lớp ImageAdapter: kế thừa BaseAdapter, hỗ trợ hiển thị nội dung trong GridView

```
public class ImageAdapter extends BaseAdapter {

 private Context mContext;
 private Integer[] Images;

 public ImageAdapter(Context context , Integer[] images) {
 mContext = context;
 Images = images;
 }

 public void setImages(Integer[] images) {
 Images = images;
 notifyDataSetChanged();
 }

 @Override
 public int getCount() {
 return Images.length;
 }
```

# GridView (3)

- Tạo lớp ImageAdapter: kế thừa BaseAdapter

```
@Override
public View getView(int i, View convertView,
 ViewGroup parent) {
 ImageView imageView = new ImageView(mContext);
 imageView.setImageResource(Images[i]);
 imageView.setScaleType(ImageView.ScaleType.CENTER_CROP);
 // Khai báo kích thước ô: chiều rộng, chiều cao
 imageView.setLayoutParams(
 new GridView.LayoutParams(400, 450));
 // Khoảng cách với viền: left, top, right, bottom
 imageView.setPadding(10, 40, 10, 10);
 return imageView;
}
}
```

# GridView (4)

- Nhận GridView, khai báo và gán nguồn dữ liệu

```
private Integer[] Images = {R.mipmap.khachhang,
 R.mipmap.goidulieu, R.mipmap.thongtinchung};

@Override
protected void onCreate(Bundle savedInstanceState) {
 super.onCreate(savedInstanceState);
 setContentView(R.layout.activity_main);
 GridView gdvMenu =
 (GridView) findViewById(R.id.gdv_menu);
 ImageAdapter adapter =
 new ImageAdapter(this, Images);
 gdvMenu.setAdapter(adapter);
 gdvMenu.setOnItemClickListener(new ChonCongViec());
}
```

# GridView (5)

- Xử lý sự kiện chọn phần tử trong lưới

```
private class ChonCongViec implements AdapterView.OnItemClickListener {
 public void onItemClick(AdapterView<?> adp, View v, int pos, long id){
 Intent intent = null;
 switch (pos) {
 case 0:
 intent = new Intent(MainActivity.this, KhachHang.class);
 break;
 case 1:
 intent = new Intent(MainActivity.this, GoiDuLieu.class);
 break;
 case 2:
 intent = new Intent(MainActivity.this, ThongTinChung.class);
 break;
 }
 startActivity(intent);
 }
}
```

# Custom GridView (1)

- Thiết kế lưới với các nội dung theo nhu cầu sử dụng



# Custom GridView (2)

- Thực hiện
  - Tạo Layout chính: dạng LinearLayout gồm 1 GridView
  - Tạo Layout hiển thị từng phần tử trong Custom GridView
  - Tạo lớp Adapter mới kế thừa BaseAdapter hỗ trợ hiển thị nội dung trong GridView
  - Gán nguồn dữ liệu cho GridView
  - Thêm sự kiện khi chọn phần tử (nếu có)
- **Ghi chú:** hướng dẫn chi tiết trong phần Bài tập

# SwipeRefreshLayout (1)

- Thuộc lớp android.support.v4.widget.SwipeRefreshLayout, kế thừa từ ViewGroup
- Hỗ trợ cập nhật (Refresh) nguồn dữ liệu cho các điều khiển dạng danh sách
- Thực hiện
  - Thêm SwipeRefreshLayout vào tập tin giao diện có chứa danh sách
  - Thêm mã lệnh cho Activity thực thi giao diện SwipeRefreshLayout.OnRefreshListener với phương thức onRefresh cho phép cập nhật nguồn dữ liệu của danh sách.

# SwipeRefreshLayout (2)

- Giao diện

```
<android.support.v4.widget.SwipeRefreshLayout
 xmlns:android="http://schemas.android.com/apk/res/android"
 android:id="@+id/swipe_refresh"
 android:layout_width="match_parent"
 android:layout_height="match_parent">

 <ListView
 android:id="@+id/listview_diadiem"
 android:layout_width="match_parent"
 android:layout_height="wrap_content" />

</android.support.v4.widget.SwipeRefreshLayout>
```

# SwipeRefreshLayout (3)

- Mã lệnh
  - Khai báo Activity thực thi SwipeRefreshLayout.OnRefreshListener

```
public class QuanLyDuLich extends AppCompatActivity
 implements SwipeRefreshLayout.OnRefreshListener{
```

- Khai báo biến lớp cho đối tượng SwipeRefreshLayout
- ```
SwipeRefreshLayout swipeRefresh;
```
- Trong phương thức onCreate, nhận và yêu cầu điều khiển SwipeRefreshLayout lắng nghe sự kiện

```
swipeRefresh = (SwipeRefreshLayout)  
    findViewById(R.id.swipe_refresh);  
swipeRefresh.setOnRefreshListener(this);
```

SwipeRefreshLayout (4)

- Mã lệnh
 - Viết mã lệnh cập nhật nguồn dữ liệu của danh sách trong sự kiện onRefresh (Giả sử thêm hai địa điểm mới vào danh sách).

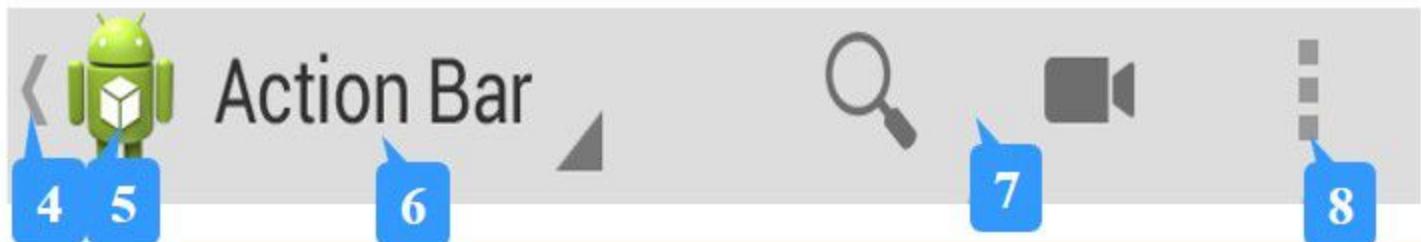
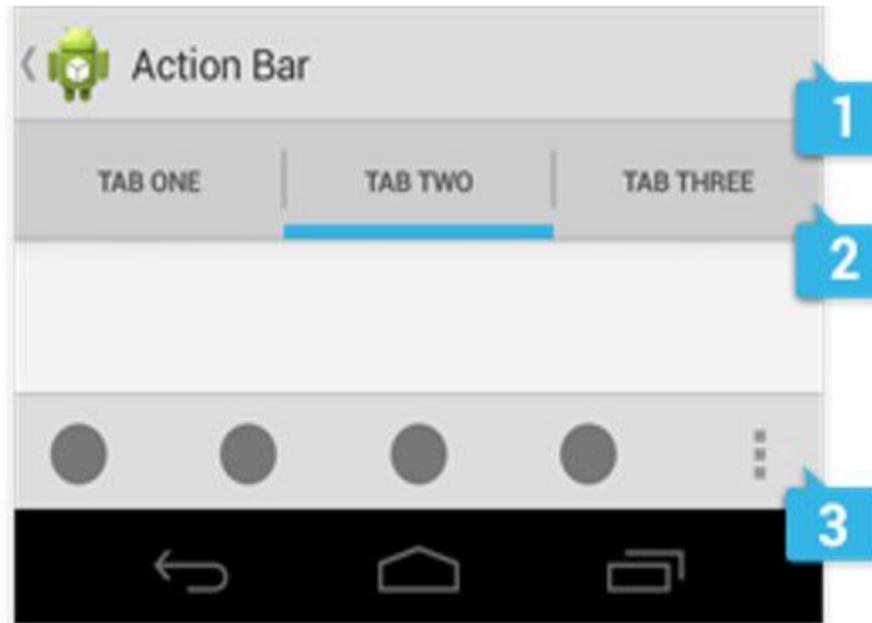
```
@Override  
public void onRefresh() {  
    listDiaDiem.add("Phú Quốc");  
    listDiaDiem.add("Sa Pa");  
    adapter.notifyDataSetChanged();  
    swipeRefreshLayout.setRefreshing(false);  
}
```

Tạo và sử dụng thực đơn _Menu

- Cho phép hiển thị và chọn công việc cần thực hiện, có 3 loại thông dụng
 - Thực đơn tùy chọn và thanh hành động (Options menu và Action bar)
 - Thực đơn theo ngữ cảnh (Context menu)
 - Thực đơn bật lên (Popup menu)

ActionBar (1)

- ActionBar (Thanh hành động), còn gọi là AppBar (Thanh ứng dụng), thuộc lớp android.support.v7.app.ActionBar



ActionBar (2)

- Nhận và sử dụng ActionBar: sử dụng Theme thông thường (không sử dụng các Theme dạng NoActionBar)
 - + Trong sự kiện onCreate: hỗ trợ sử dụng ActionBar, hiển thị Icon

```
ActionBar myActionBar = getSupportActionBar();  
myActionBar.setDisplayShowHomeEnabled(true);  
myActionBar.setIcon(R.mipmap.globeicon);
```

- + Có thể sử dụng Logo (có thể chiếm không gian lớn hơn) thay vì Icon

```
myActionBar.setDisplayShowHomeEnabled(true);  
myActionBar.setLogo(R.mipmap.globelogo);  
myActionBar.setDisplayUseLogoEnabled(true);
```

ActionBar (3)

- Nhận và sử dụng ActionBar
 - + Thêm nút trở về Activity cha (Up Action)
 - Trong sự kiện onCreate: hiển thị và sử dụng nút trở về Activity cha

```
myActionBar.setDisplayHomeAsUpEnabled(true);
```

- Khai báo Activity cha trong AndroidManifest
 - VD: Khai báo MainActivity là Activity cha của Activity KhachHang

```
<activity  
    android:name=".KhachHang"  
    android:parentActivityName=".MainActivity" />
```

ToolBar (1)

- Toolbar (Thanh công cụ) thuộc lớp android.support.v7.widget.Toolbar, kế thừa từ ViewGroup
- Được giới thiệu từ API 21, như là mở rộng của ActionBar, giúp tùy biến dễ dàng hơn.
- Khác nhau giữa Toolbar và ActionBar: Toolbar là một ViewGroup → có thể khai báo trong Layout, có thể chứa các View ở nhiều vị trí → tùy biến dễ dàng hơn
- Activity (hay Application) sử dụng Theme không có ActionBar

ToolBar (2)

- Khai báo và sử dụng ActionBar
 - + Khai báo trong tập tin giao diện: thêm Toolbar trước các điều khiển khác

```
<android.support.v7.widget.Toolbar  
    android:id="@+id/my_toolbar"  
    android:layout_width="match_parent"  
    android:layout_height="?attr/actionBarSize"  
    android:background="?attr/colorPrimary"  
    android:elevation="4dp"  
    android:theme="@style/ThemeOverlay.AppCompat.ActionBar"  
    app:popupTheme="@style/ThemeOverlay.AppCompat.Light"/>
```

ToolBar (3)

- Khai báo và sử dụng ActionBar

- + Mã lệnh trong sự kiện onCreate: nhận và hỗ trợ sử dụng Toolbar như một ActionBar, hiển thị Logo

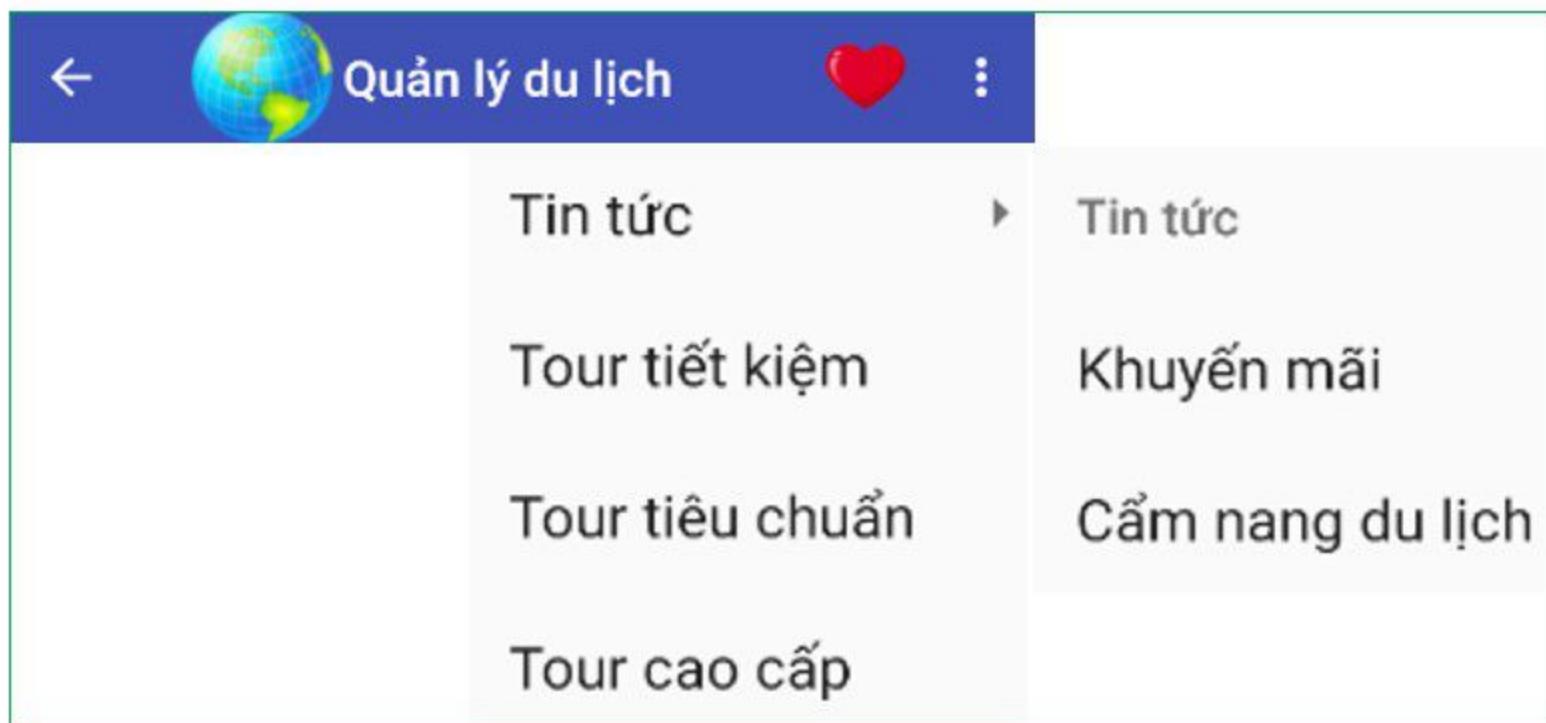
```
Toolbar myToolbar = (Toolbar) findViewById(R.id.my_toolbar);  
setSupportActionBar(myToolbar);  
myToolbar.setLogo(R.mipmap.globeLogo);
```

- + Để hiển thị Icon trên Toolbar, tạo và sử dụng nút lệnh trở về Activity cha, ta thực hiện khai báo nhận ActionBar và viết mã lệnh tương tự như đối với ActionBar ở phần trước.

```
ActionBar myActionBar = getSupportActionBar();  
myActionBar.setIcon(R.mipmap.globeIcon);  
...
```

Tạo giao diện thực đơn (1)

- VD



Tạo giao diện thực đơn (2)

```
<?xml version="1.0" encoding="utf-8"?>
<menu
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto">
    <item
        android:id="@+id/mnu_thich"
        android:icon="@mipmap/love"
        android:onClick="DiaDiemYeuThich"
        android:title="@string/thich_diadiem"
        app:showAsAction="ifRoom| collapseActionView" />
```

Tạo giao diện thực đơn (3)

```
<item
    android:id="@+id/mnu_tintuc"
    android:title="@string/tin_tuc">
    <menu>
        <item
            android:id="@+id/mnu_khuyenmai"
            android:title="@string/khuyen_mai" />
        <item
            android:id="@+id/mnu_camnangdl"
            android:title="@string/cam_nang_dl" />
    </menu>
</item>
```

Tạo giao diện thực đơn (4)

```
<group android:id="@+id/grp_loaitour">
    <item
        android:id="@+id/mnu_tietkiem"
        android:onClick="ChonLoaiTour"
        android:title="@string/tiet_kiem" />
    <item
        android:id="@+id/mnu_tieuchuan"
        android:onClick="ChonLoaiTour"
        android:title="@string/tieu_chuan" />
    <item
        android:id="@+id/mnu_caocap"
        android:onClick="ChonLoaiTour"
        android:title="@string/cao_cap" />
</group>
```

Options Menu (1)

- Tạo thực đơn tùy chọn dạng Text hoặc Icon
- Hiển thị và thực hiện các hành động liên quan ứng dụng hoặc Activity hiện tại trên thanh tiêu đề
 - Trong res\menu, tạo danh sách nội dung thực đơn
 - Mã lệnh
 - Khởi tạo thực đơn: ghi đè phương thức onCreateOptionsMenu
 - Xử lý sự kiện Click:
 - Khai báo phương thức nhận sự kiện Click của mục đơn có thuộc tính onClick, hoặc
 - Ghi đè phương thức onOptionsItemSelected

Options Menu (2)

```
@Override  
public boolean onCreateOptionsMenu(Menu menu) {  
    MenuInflater inflater = getMenuInflater();  
    inflater.inflate(R.menu.dulich_optionsmenu, menu);  
    return true;  
}  
  
public void DiaDiemYeuThich(MenuItem menu) {  
    // Mã lệnh xử lý công việc  
}
```

Options Menu (3)

```
@Override  
public boolean onOptionsItemSelected(MenuItem item) {  
    switch (item.getItemId()) {  
        case R.id.mnu_khuyenmai:  
            // Mã lệnh xử lý công việc  
            return true;  
        case R.id.mnu_cavnangdl:  
            // Mã lệnh xử lý công việc  
            return true;  
        default:  
            return super.onOptionsItemSelected(item);  
    }  
}
```

Context Menus (1)

- Thuộc lớp android.view.ContextMenu, thực đơn cho phép thực thi công việc đáp ứng sự kiện LongClick lên phần tử



Context Menus (2)

- Thực hiện
 - Trong res\menu, tạo danh sách nội dung thực đơn
 - Mã lệnh
 - Khởi tạo thực đơn: ghi đè phương thức onCreateContextMenu
 - Xử lý sự kiện LongClick:
 - Khai báo phương thức nhận sự kiện Click của mục đơn có thuộc tính onClick, hoặc
 - Ghi đè phương thức onContextItemSelected.
 - Trong sự kiện onCreate: đăng ký yêu cầu áp dụng thực đơn cho đối tượng liên quan.

Context Menus (3)

- Giao diện danh sách nội dung thực đơn

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android" >
    <item
        android:id="@+id/mnu_chitiet"
        android:title="@string/chi_tiet"
        android:onClick="ThongTinDiaDiem" />
    <item
        android:id="@+id/mnu_dattour"
        android:title="@string/dat_tour" />
    <item
        android:id="@+id/mnu_xoadd"
        android:title="@string/xoa_dd" />
</menu>
```

Context Menus (4)

- Khởi tạo và xử lý sự kiện chọn thực đơn

```
@Override  
public void onCreateContextMenu(ContextMenu menu,  
View v, ContextMenu.ContextMenuItemInfo menuInfo) {  
    super.onCreateContextMenu(menu, v, menuInfo);  
    MenuInflater inflater = getMenuInflater();  
    inflater.inflate(R.menu.dulich_contextmenu, menu);  
}  
  
public void ThongTinDiaDiem(MenuItem menu) {  
    // Mã lệnh xử lý công việc  
}
```

Context Menus (5)

- Khởi tạo và xử lý sự kiện chọn thực đơn

```
@Override
```

```
public boolean onContextItemSelected(MenuItem item) {  
    switch (item.getItemId()) {  
        case R.id.mnu_dattour:  
            // Mã lệnh xử lý công việc  
            return true;  
        case R.id.mnu_xoadd:  
            // Mã lệnh xử lý công việc  
            return true;  
        default:  
            return super.onContextItemSelected(item);  
    }  
}
```

Context Menus (6)

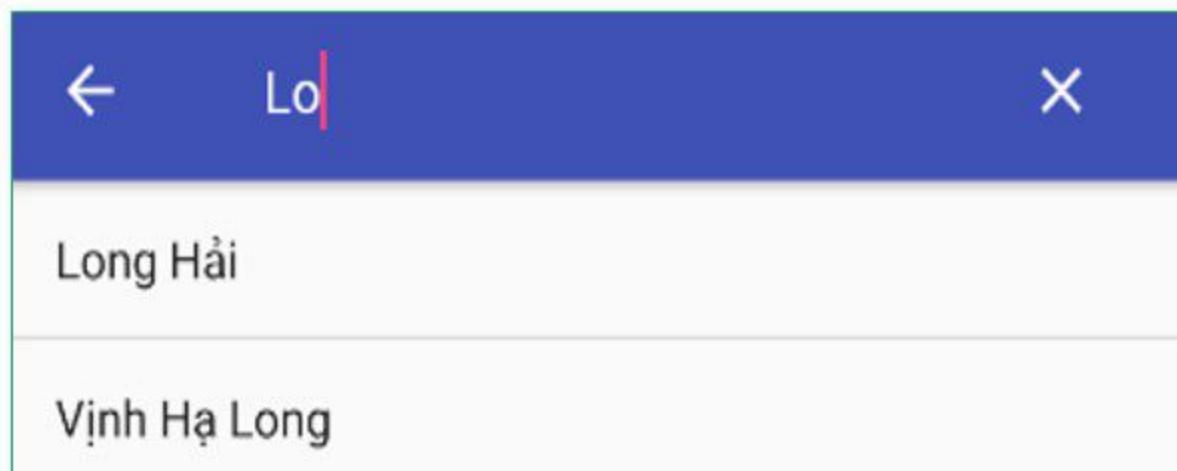
- Trong sự kiện onCreate: đăng ký yêu cầu áp dụng thực đơn cho đối tượng

VD: đăng ký cho ListView listviewDiaDiem

```
registerForContextMenu(listviewDiaDiem);
```

Tìm kiếm trên danh sách với SearchView (1)

- Thuộc lớp android.widget.SearchView, kế thừa LinearLayout
- Hỗ trợ tìm kiếm các phần tử trong danh sách
- Phần này hướng dẫn tạo chức năng tìm kiếm đặt trực tiếp trên thanh tiêu đề (sử dụng OptionsMenu)



Tìm kiếm trên danh sách với SearchView (2)

- Khai báo thực đơn tùy chọn sử dụng SearchView
- Activity: thực thi giao diện SearchView.OnQueryTextListener
- Điều chỉnh phương thức ghi đè sự kiện nhận thực đơn tùy chọn với thực đơn sử dụng SearchView
- Viết mã lệnh các phương thức ghi đè onQueryTextSubmit và onQueryTextChange

Tìm kiếm trên danh sách với SearchView (3)

- Khai báo thực đơn tùy chọn sử dụng SearchView

```
<menu  
    xmlns:android="http://schemas.android.com/apk/res/android"  
    xmlns:app="http://schemas.android.com/apk/res-auto">  
    <item android:id="@+id/menu_search"  
        android:title="@string/tim_dia_diem"  
        app:actionViewClass=  
            "android.support.v7.widget.SearchView"  
        android:icon="@android:drawable/ic_menu_search"  
        app:showAsAction="always|collapseActionView" />  
</menu>
```

Tìm kiếm trên danh sách với SearchView (4)

- Activity: thực thi giao diện SearchView.OnQueryTextListener

```
public class QuanLyDuLich extends AppCompatActivity  
    implements SearchView.OnQueryTextListener {  
    ...  
}
```

Tìm kiếm trên danh sách với SearchView (5)

- Phương thức ghi đè sự kiện nhận thực đơn tùy chọn với thực đơn sử dụng SearchView

```
@Override  
public boolean onCreateOptionsMenu(Menu menu) {  
    MenuInflater inflater = getMenuInflater();  
    inflater.inflate(R.menu.dulich_searchmenu, menu);  
    SearchView searchView = (SearchView)  
        menu.findItem(R.id.menu_search).getActionView();  
    searchView.setOnQueryTextListener(this);  
    return true;  
}
```

Tìm kiếm trên danh sách với SearchView (6)

- Phương thức ghi đè onQueryTextSubmit và onQueryTextChange

```
@Override  
public boolean onQueryTextSubmit(String query) {  
    return true;  
}  
  
@Override  
public boolean onQueryTextChange(String newText) {  
    if (TextUtils.isEmpty(newText)) {  
        adapter.getFilter().filter("");  
        listViewDiaDiem.clearTextFilter();  
    } else {  
        adapter.getFilter().filter(newText);  
    }  
    return true;  
}
```

Fragment (1)

- Thuộc lớp android.support.v4.app.Fragment, biểu diễn một hành vi hay một phần giao diện trong một Activity

Du lịch Đà Lạt

GIỚI THIỆU

HÌNH ẢNH

Đà Lạt từ lâu đã trở thành một trong những thành phố du lịch nổi tiếng nhất của Việt Nam, với nhiều cảnh quan thiên nhiên đẹp và thơ mộng. Có ai đó đã từng ví Đà Lạt cũng giống một thành phố trong rừng, nơi khí trời se lạnh, thiên nhiên và con người gần gũi như luôn hoà quyện, đan xen vào nhau.

Ai đến Đà Lạt đều phải thốt lên rằng: Ôi! Sao Đà Lạt đẹp đến thế! Đẹp mê hồn người lữ khách, đẹp cả trong tim người chưa đến Đà Lạt bao giờ. Phải chăng người ta yêu Đà Lạt bởi những dòng thác cuộn trào, ảo ảnh sương khói của hương hoa, của nước Hồ Xuân Hương dịu mát.

Núi Lang Biang nằm trên địa bàn huyện Lạc Dương, cách trung tâm thành phố Đà Lạt 12km về phía bắc. Núi Lang Biang còn được gọi là Núi Mẹ, gồm 2 ngọn, có độ cao 2.167m.

DÓNG ACTIVITY

g

LÀNG BIANG

Fragment (2)

- Hỗ trợ tạo Activity với giao diện có nhiều thành phần (Fragment), mỗi thành phần là một Sub Activity
- Một Activity có thể có nhiều Fragment, một Fragment có thể sử dụng trong nhiều Activity
- Fragment có vòng đời riêng, bị ảnh hưởng (phụ thuộc) trực tiếp bởi vòng đời của Activity chủ
- Để nhận đối tượng Activity chủ, sử dụng phương thức `getActivity()`.

Fragment (3)

- Thực hiện
 - Tạo Fragment mới, tạo giao diện tương tự giao diện Activity
 - Mã lệnh
 - Kế thừa lớp Fragment
 - Phương thức onCreateView: khai báo Layout sử dụng và trả về View hiển thị trên Fragment
 - Phương thức onActivityCreated: khởi tạo giá trị và thực hiện công việc theo yêu cầu
 - Xử lý sự kiện cho nút lệnh: khai báo trong Activity chủ hoặc khai báo trực tiếp yêu cầu lắng nghe và xử lý sự kiện

Fragment (4)

- Giao diện

Tập tin fragment_gioithieu.xml

```
<LinearLayout  
    xmlns:android="http://schemas.android.com/apk/res/android"  
        xmlns:tools="http://schemas.android.com/tools"  
        android:layout_width="match_parent"  
        android:layout_height="match_parent"  
        android:orientation="vertical" >  
  
    <TextView  
        android:id="@+id/txt_gioithieu"  
        android:layout_width="match_parent"  
        android:layout_height="0dp"  
        android:layout_weight="1"  
        android:gravity="top"  
        android:layout_marginTop="10dp"  
        android:paddingLeft="5dp" />  
  
    <Button  
        android:id="@+id/btn_dong"  
        style="@style/ButtonStyle"  
        android:layout_gravity="center"  
        android:text="@string/dong_activity"  
        android:onClick="DongActivity"  
        android:layout_marginBottom="10dp"/>  
  
</LinearLayout>
```

Fragment (5)

- Mã lệnh

```
public class GioiThieuFragment extends Fragment {  
  
    View viewGioiThieu;  
  
    @Override  
    public View onCreateView(LayoutInflater inflater,  
        ViewGroup container, Bundle savedInstanceState) {  
        viewGioiThieu = inflater.inflate(  
            R.layout.fragment_gioithieu, container, false);  
        return viewGioiThieu;  
    }  
}
```

Fragment (6)

- Mã lệnh

```
@Override  
public void onActivityCreated(  
    @Nullable Bundle savedInstanceState) {  
    super.onActivityCreated(savedInstanceState);  
    TextView txtGioiThieu = (TextView)  
        viewGioiThieu.findViewById(R.id.txt_gioithieu);  
    txtGioiThieu.setText("Nội dung giới thiệu Đà Lạt");  
// Nhận và xử lý sự kiện nút lệnh trong Fragment_Cách 1  
    Button btnDong = (Button)  
        viewGioiThieu.findViewById(R.id.btn_dong);  
    btnDong.setOnClickListener(  
        new View.OnClickListener() {  
            @Override  
            public void onClick(View view) {  
                // Mã lệnh xử lý công việc  
            }  
        } );  
}
```

Activity dạng Tab (1)

- Activity chính: dạng Action Bar Tabs (with ViewPager)
- Mỗi thành phần: đối tượng Fragment

Du lịch Đà Lạt

GIỚI THIỆU

HÌNH ẢNH

Đà Lạt từ lâu đã trở thành một trong những thành phố du lịch nổi tiếng nhất của Việt Nam, với nhiều cảnh quan thiên nhiên đẹp và thơ mộng. Có ai đó đã từng ví Đà Lạt cũng giống một thành phố trong rừng, nơi khí trời se lạnh, thiên nhiên và con người gần gũi như luôn hoà quyện, đan xen vào nhau.

Ai đến Đà Lạt đều phải thốt lên rằng: Ôi! Sao Đà Lạt đẹp đến thế! Đẹp mê hồn người lữ khách, đẹp cả trong tim người chưa đến Đà Lạt bao giờ. Phải chăng người ta yêu Đà Lạt bởi những dòng thác cuộn trào, ảo ảnh sương khói của hương hoa, của nước Hồ Xuân Hương dịu mát.

Núi Lang Biang nằm trên địa bàn huyện Lạc Dương, cách trung tâm thành phố Đà Lạt 12km về phía bắc. Núi Lang Biang còn được gọi là Núi Mẹ, gồm 2 ngọn, có độ cao 2.167m.

DÓNG ACTIVITY

Du lịch Đà Lạt

GIỚI THIỆU

HÌNH ẢNH

Activity dạng Tab (2)

- Các đối tượng liên quan khi sử dụng
 - + TabLayout: thuộc lớp android.support.design.widget.TabLayout, là ViewGroup hỗ trợ thiết kế giao diện với nhiều Tab
 - + TabItem: thuộc lớp android.support.design.widget.TabItem, kế thừa từ View hỗ trợ thiết kế từng Tab thành phần.
 - + ViewPager: hỗ trợ quản lý các thành phần Action Bar Tabs, thuộc lớp android.support.v4.view.ViewPager.
 - + Fragment: hỗ trợ tạo Fragment thành phần, thuộc lớp android.support.v4.app.Fragment.
 - + FragmentManager: hỗ trợ quản lý Fragment, thuộc lớp android.support.v4.app.FragmentManager.
 - + FragmentTransaction: hỗ trợ quản lý các giao tác (chọn hoặc bỏ chọn Fragment) thuộc lớp android.support.v4.app.FragmentTransaction.
 - + FragmentPagerAdapter: hỗ trợ quản lý đối tượng thành phần, thuộc lớp android.support.v4.app.FragmentPagerAdapter.

Activity dạng Tab (3)

- Tạo các Fragment thành phần
- Tạo Activity dạng Tabbed Activity, chọn kiểu điều hướng (Navigation Style) là Action Bar Tabs (with ViewPager)
 - Activity thực thi giao diện ActionBar.TabListener với các phương thức: onTabSelected, onTabUnselected, onTabReselected
 - Khởi tạo và gán thuộc tính cho các đối tượng quản lý Fragment
 - Khai báo lớp SectionsPagerAdapter kế thừa FragmentPagerAdapter để quản lý và gán đối tượng hiển thị cho từng thành phần
 - Gán Fragment hiển thị trong phương thức onTabSelected

Activity dạng Tab (4)

- Giao diện: khai báo TabLayout với các Tab thành phần

```
<android.support.design.widget.TabLayout  
    android:id="@+id/tablayout"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:background="#CCDEEF"  
    app:tabIndicatorColor="#33FFBB"  
    app:tabIndicatorHeight="5dp">  
  
<android.support.design.widget.TabItem  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="@string/gioi_thieu" />
```

Activity dạng Tab (5)

- Giao diện: khai báo ViewPager chứa các thành phần

```
<android.support.v4.view.ViewPager  
    android:id="@+id/container"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
    app:layout_behavior="@string/behavior" />
```

Activity dạng Tab (6)

- Mã lệnh: Activity thực thi giao diện ActionBar.TabListener, khai báo các biến quản lý Fragment

```
public class DaLat extends AppCompatActivity
    implements ActionBar.TabListener {

    private SectionsPagerAdapter mSectionsPagerAdapter;
    private ViewPager mViewPager;
```

Activity dạng Tab (7)

- Mã lệnh: khởi tạo cho các đối tượng quản lý Fragment

@Override

```
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_dalat);  
    mSectionsPagerAdapter = new SectionsPagerAdapter(  
        getSupportFragmentManager());  
    mViewPager = (ViewPager) findViewById(R.id.container);  
    mViewPager.setAdapter(mSectionsPagerAdapter);  
    TabLayout tabLayout =  
        (TabLayout) findViewById(R.id.tablayout);  
    mViewPager.addOnPageChangeListener(  
        new TabLayout.TabLayoutOnPageChangeListener(  
            tabLayout));  
    tabLayout.addOnTabSelectedListener(  
        new TabLayout.ViewPagerOnTabSelectedListener(  
            mViewPager));  
}
```

Activity dạng Tab (8)

- Mã lệnh: ghi đè các phương thức: onTabSelected, onTabUnselected, onTabReselected

```
@Override  
public void onTabSelected(ActionBar.Tab tab,  
                           FragmentTransaction ft) {  
    mViewPager.setCurrentItem(tab.getPosition());  
}  
  
@Override  
public void onTabUnselected(ActionBar.Tab tab,  
                           FragmentTransaction ft) {  
}  
  
@Override  
public void onTabReselected(ActionBar.Tab tab,  
                           FragmentTransaction ft) {  
}  
}
```

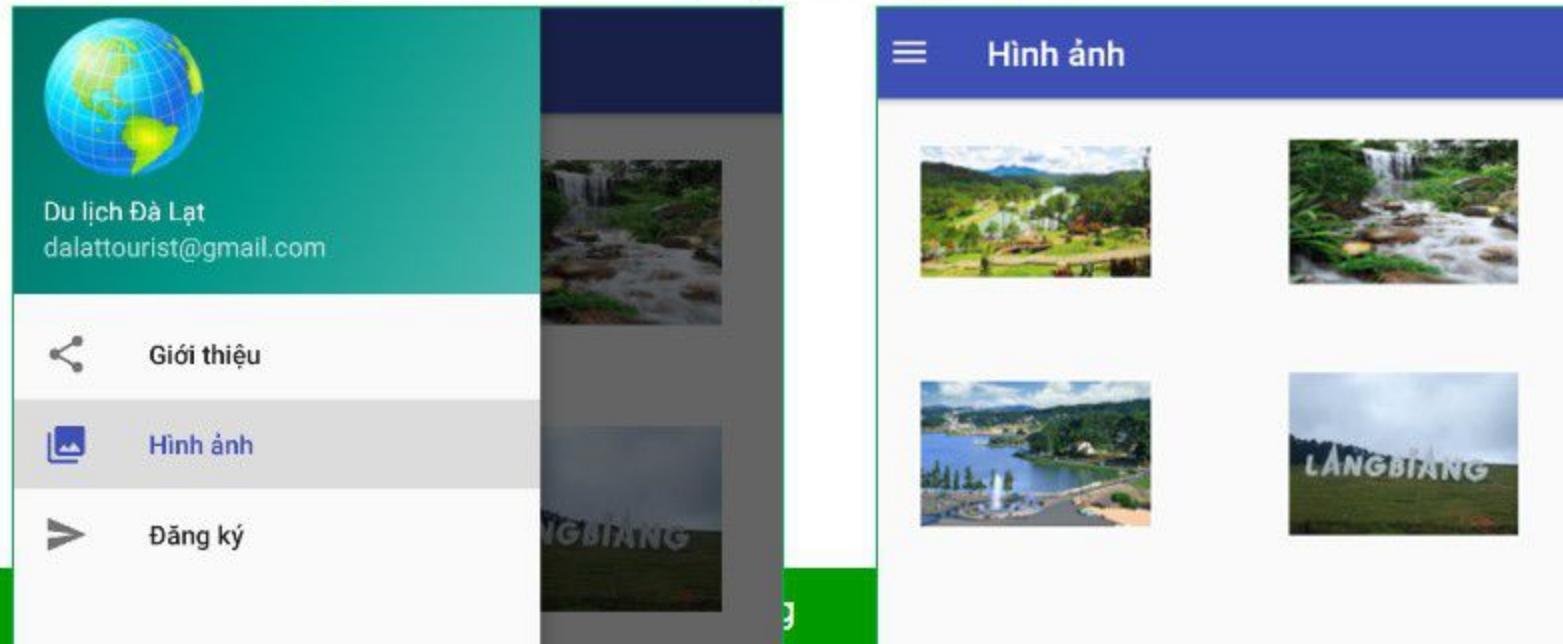
Activity dạng Tab (9)

- Mã lệnh: lớp SectionsPagerAdapter quản lý và gán đối tượng hiển thị

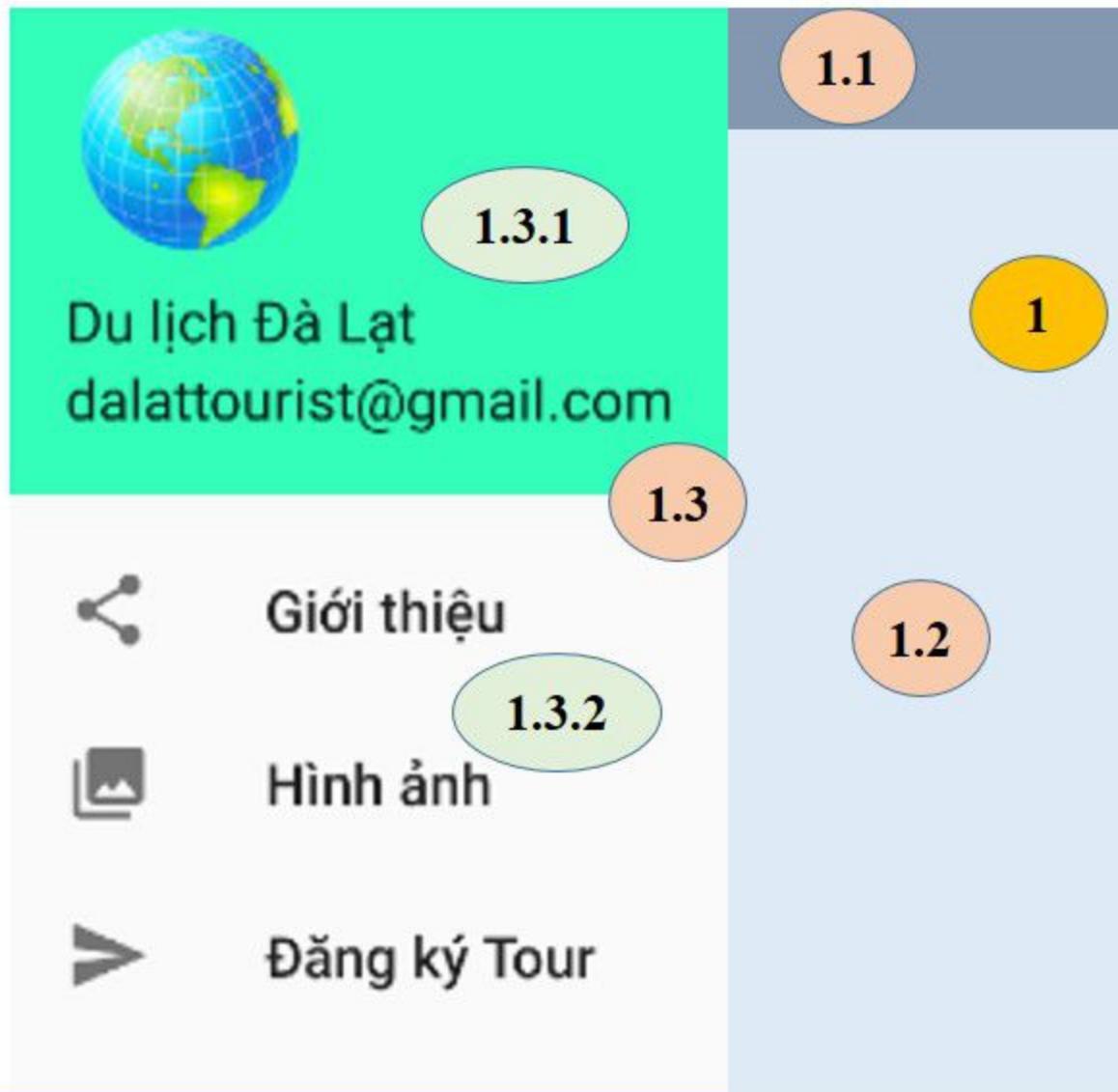
```
public class SectionsPagerAdapter  
        extends FragmentPagerAdapter {  
  
    public SectionsPagerAdapter(FragmentManager fm) {  
        super(fm);  
    }  
  
    @Override  
    public Fragment getItem(int position) {  
        switch (position) {  
            case 1:  
                return new HinhAnh();  
            default:  
                return new GioiThieu();  
        }  
    }  
  
    @Override  
    public int getCount() {  
        return 2;  
    }  
}
```

Activity điều hướng (1)

- Navigation Drawer (Left Menu hoặc Slide Menu): thanh thực đơn tự động ân ở bên trái màn hình
- Được hiển thị khi người dùng nhấn vào biểu tượng ≡ bên trái Action Bar hoặc vuốt màn hình từ trái sang phải
- Mỗi thành phần: đối tượng Fragment



Activity điều hướng (2)



Activity điều hướng (3)

- **Vùng 1:** toàn bộ Activity điều hướng, chia thành 3 cung con
- **Vùng 1.1:** vùng cho thanh hành động ActionBar
- **Vùng 1.2:** vùng thể hiện nội dung chính của giao diện khi thực thi các chức năng thành phần (Fragment)
- **Vùng 1.3:** vùng thể hiện thanh điều hướng tự động ẩn phía bên trái, chia thành 2 cung con
 - **Vùng 1.3.1:** vùng thể hiện tiêu đề của thanh điều hướng
 - **Vùng 1.3.2:** vùng thể hiện thực đơn chức năng

Activity điều hướng (4)

- Các đối tượng liên quan khi sử dụng
 - DrawerLayout: thuộc lớp android.support.v4.widget.DrawerLayout, là ViewGroup hỗ trợ thiết kế giao diện cho Activity điều hướng, chứa các cửa sổ thành phần
 - FrameLayout: thuộc lớp android.widget.FrameLayout, là một ViewGroup thiết kế một khu vực trên màn hình để hiển thị một thành phần.
 - Fragment: hỗ trợ tạo Fragment thành phần, thuộc lớp android.support.v4.app.Fragment.
 - FragmentManager: hỗ trợ quản lý Fragment, thuộc lớp android.support.v4.app.FragmentManager.

Activity điều hướng (5)

- Các đối tượng liên quan khi sử dụng
 - FragmentTransaction: hỗ trợ quản lý các giao tác (chọn hoặc bỏ chọn Fragment) thuộc lớp android.support.v4.app.FragmentTransaction.
Để thay thế Fragment này bằng Fragment khác, ta thực hiện:

```
FragmentTransaction transaction =  
    getSupportFragmentManager().beginTransaction();  
transaction.replace(old Fragment, new Fragment);  
transaction.addToBackStack("Chuỗi nhận dạng Fragment");  
transaction.commit();
```

- NavigationView: là một ViewGroup hỗ trợ tạo thực đơn điều hướng cho ứng dụng, thuộc lớp android.support.design.widget.NavigationView.
- ActionBarDrawerToggle: hỗ trợ quản lý xử lý sự kiện đóng/mở thực đơn điều hướng, thuộc lớp android.support.v7.app.ActionBarDrawerToggle.

Activity điều hướng (6)

- Tạo các Fragment thành phần
- Tạo Activity dạng Navigation Drawer Activity
 - Kiểm tra và điều chỉnh các tập tin giao diện
 - Activity thực thi giao diện
NavigationView.OnNavigationItemSelectedListener
 - Ghi đè phương thức onNavigationItemSelected
 - Gán Fragment cần mở tương ứng id của thực đơn được chọn,
 - Khai báo đối tượng quản lý giao tác (FragmentTransaction), gán các thuộc tính cần thiết, thực hiện thay thế Fragment,
 - Gán trạng thái được chọn cho thực đơn và tiêu đề cho Fragment

Activity điều hướng (7)

```
public class DuLichDaLat extends AppCompatActivity
implements NavigationView.OnNavigationItemSelected {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.dulich_dalat_main);

        Toolbar toolbar =
                (Toolbar) findViewById(R.id.toolbar);
        setSupportActionBar(toolbar);

        ActionBar myActionBar = getSupportActionBar();
        myActionBar.setDisplayHomeAsUpEnabled(true);
        myActionBar.setHomeButtonEnabled(true);
```

Activity điều hướng (8)

```
DrawerLayout drawer = (DrawerLayout)
    findViewById(R.id.drawer_layout);
ActionBarDrawerToggle toggle =
    new ActionBarDrawerToggle(this, drawer,
        toolbar, R.string.navigation_drawer_open,
        R.string.navigation_drawer_close);
drawer.addDrawerListener(toggle);
toggle.syncState();

NavigationView navigationView =
    (NavigationView) findViewById(R.id.nav_view);
navigationView.setNavigationItemSelected(this);

// Gán Fragment mặc định khi khởi tạo Activity
onNavigationItemSelected(
    navigationView.getMenu().findItem(
        R.id.nav_gioithieu));
```

Activity điều hướng (9)

```
@SuppressWarnings("StatementWithEmptyBody")
@Override
public boolean onNavigationItemSelected(MenuItem item) {
    int id = item.getItemId();
    Fragment fragment = null;
    Class fragmentClass = null;
    if (id == R.id.nav_gioithieu) {
        fragmentClass = GioiThieu.class;
    } else if (id == R.id.nav_hinhanh) {
        fragmentClass = HinhAnh.class;
    } else {
        fragmentClass = GioiThieu.class;
    }
    try {
        fragment = (Fragment) fragmentClass.newInstance();
    } catch (Exception e) {
    }
}
```

Activity điều hướng (10)

```
// Thay thế Fragment theo thực đơn được chọn
    FragmentTransaction transaction =
        getSupportFragmentManager().beginTransaction();
    transaction.replace(R.id.layout_content, fragment);
    transaction.addToBackStack(
        item.getTitle().toString());
    transaction.commit();

    item.setChecked(true);
    setTitle(item.getTitle());

    DrawerLayout drawer = (DrawerLayout)
        findViewById(R.id.drawer_layout);
    drawer.closeDrawer(GravityCompat.START);
    return true;
}
```

Tạo giao diện cho nhiều loại màn hình (1)

- Vấn đề
 - Màn hình với kích thước (Screen size) khác nhau: *small, normal, large, and extra-large*
 - Màn hình với mật độ điểm ảnh (Screen density) khác nhau: *low, medium, high, extra-high, extra-extra-high, and extra-extra-extra-high*
 - Hướng màn hình (Orientation): *portrait, landscape*

Tạo giao diện cho nhiều loại màn hình (2)

- Hướng giải quyết
 - Tạo nhiều tập tin layout tương ứng với kích thước và hướng màn hình
 - Tạo nhiều hình ảnh với kích thước khác nhau
 - Sử dụng tỷ lệ hiển thị đối tượng hình so với màn hình chuẩn sử dụng
 - Kiểm tra ứng dụng với nhiều màn hình khác nhau
 - Khai báo trong AndroidManifest cụ thể loại màn hình ứng dụng hỗ trợ, khai báo lắng nghe sự thay đổi cấu hình để tránh Activity khởi động lại

Tạo giao diện cho nhiều loại màn hình (3)

- Hướng giải quyết
 - Kiểm tra sự thay đổi cấu hình và xử lý phù hợp
 - Ghi đè phương thức onConfigurationChanged và onWindowFocusChanged nhận biết sự thay đổi cấu hình

VD: kiểm tra và xử lý sự thay đổi hướng màn hình

```
if (getResources().getConfiguration().orientation ==  
    Configuration.ORIENTATION_LANDSCAPE) {  
    // Mã lệnh với màn hình hướng ngang  
} else {  
    // Mã lệnh với màn hình hướng đứng  
}
```