



אנדרואיד

**FossComm Λάρισσα 9-10/5/2009**



**Greek Android Community**

**Πολυχρόνης Κωνσταντίνος**

**AndroidDev.gr**

# Android is an Open platform

Το Android έχει τα εξής χαρακτηριστικά:

- Συγκεκριμένο H/W design
- Linux
- OpenSource libraries
- Runtime
- Application framework
- User Interface framework
- Preinstalled applications

# Android και Java

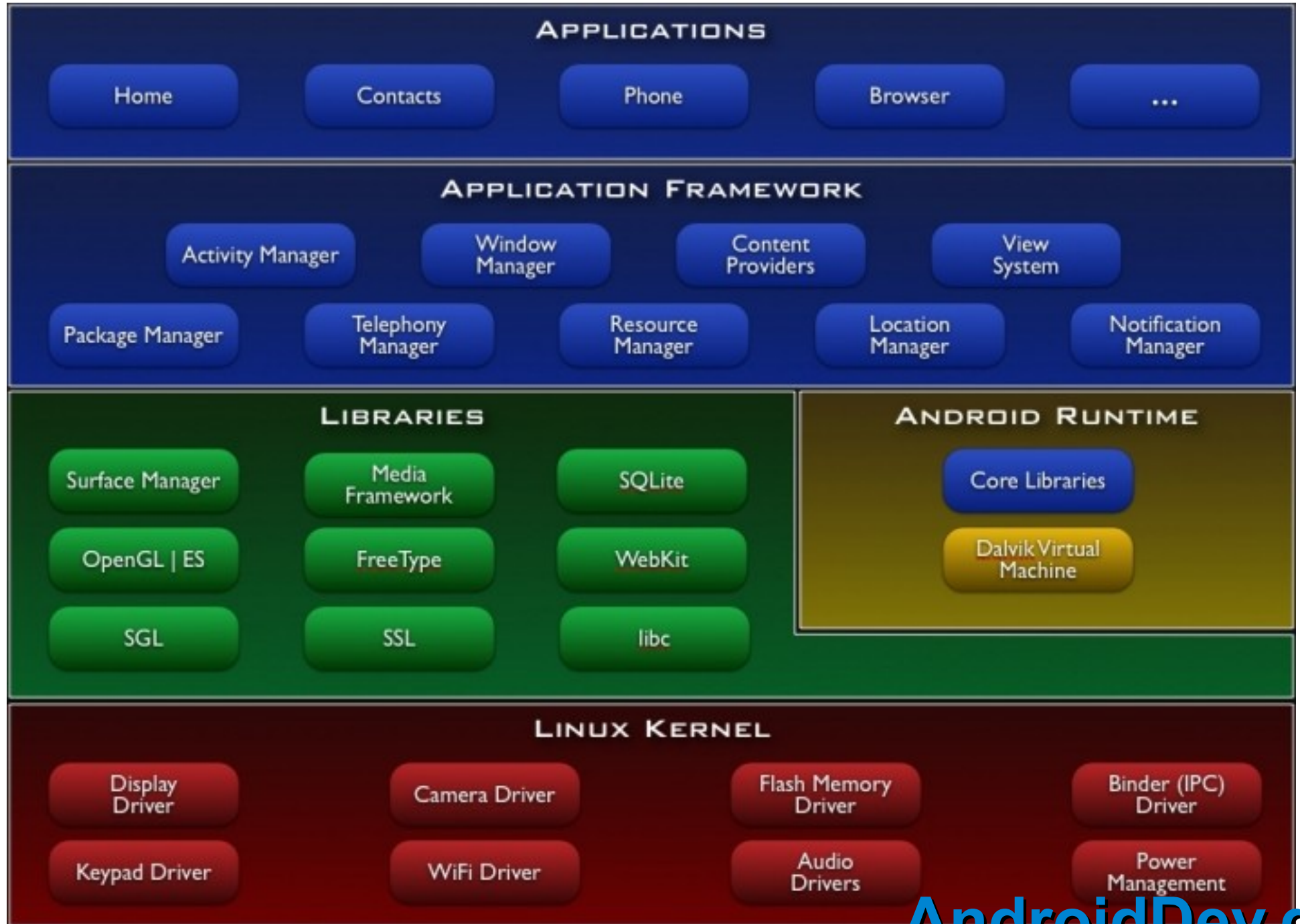
Γιατί Java;



Προφανής απάντηση:

- Εύκολη
- Γνωστή
- Ανοιχτή και δωρεάν

# Android Architecture



# Android Runtime

Το Android περιλαμβάνει ένα σετ από βιβλιοθήκες που παρέχουν το απαραίτητο **functionality** που υπάρχει και στις βιβλιοθήκες της Java.

Κάθε εφαρμογή του Android τρέχει σε δική του **process**, με δικό του στιγμιότυπο (**instance**) από την **Dalvik virtual machine**. Η Dalvik έχει γραφτεί με τέτοιο τρόπο ώστε μία συσκευή να μπορεί να τρέχει πολλά VMs με μεγάλη απόδοση. Η Dalvik VM εκτελεί τα αρχεία στο **Dalvik Executable (.dex) format**, το οποίο είναι βελτιωμένο για ελάχιστη χρήση μνήμης.

Η Dalvik VM βασίζεται στον **Linux kernel** για τις λειτουργίες του κατώτερου επιπέδου όπως το **threading** και το **low-level memory management**.

# Android Features

- **Application framework** το οποίο επιτρέπει την επαναχρησιμοποίηση και αντικατάσταση των components
- **Dalvik virtual machine** optimized για κινητές συσκευές
- **Integrated browser** βασισμένος στην open source μηχανή WebKit (-> Chrome)
- **Optimized graphics** που υποστηρίζονται απο μια custom 2D graphics library  
3D graphics βασισμένα στο OpenGL ES 1.0 specification (hardware acceleration optional)
- **SQLite** for structured data storage
- **Multimedia** υποστήριξη για σχεδόν όλα τα διάσημα formats video ήχου και εικόνας
- **GSM Telephony** (hardware dependent)
- **Bluetooth, EDGE, 3G, και WiFi** (hardware dependent)
- **Camera, GPS, compass, και accelerometer** (hardware dependent)
- **Πλούσιο περιβάλλον ανάπτυξης** που περιλαμβάνει έναν εξομοιωτή, εργαλεία για debugging, memory και performance profiling, και ένα plugin για το Eclipse IDE

# Βιβλιοθήκες του Android

Το Android περιλαμβάνει ένα σετ απο C/C++ βιβλιοθήκες που χρησιμοποιούνται απο διάφορα components του συστήματος του Android. Αυτές οι δυνατότητες διαθέτοντε στους developers μέσα απο το Android application framework.

Μερικές απο τις βασικές βιβλιοθήκες είναι οι παρακάτω:

**System C library** - a BSD-derived implementation of the standard C system library (libc), tuned for embedded Linux-based devices

**Media Libraries** - based on PacketVideo's OpenCORE; the libraries support playback and recording of many popular audio and video formats, as well as static image files, including MPEG4, H.264, MP3, AAC, AMR, JPG, and PNG

**Surface Manager** - manages access to the display subsystem and seamlessly composites 2D and 3D graphic layers from multiple applications



# Βιβλιοθήκες του Android

**LibWebCore** - a modern web browser engine which powers both the Android browser and an embeddable web view

**SGL** - the underlying 2D graphics engine

**3D libraries** - an implementation based on OpenGL ES 1.0 APIs; the libraries use either hardware 3D acceleration (where available) or the included, highly optimized 3D software rasterizer

**FreeType** - bitmap and vector font rendering

**SQLite** - a powerful and lightweight relational database engine available to all applications

# Android Emulator



# Design Philosophy

Οι εφαρμογές στο Android πρέπει να είναι:

**Γρήγορες**

**Άμεσα Ανταποκρίσιμες**

**Seamless**

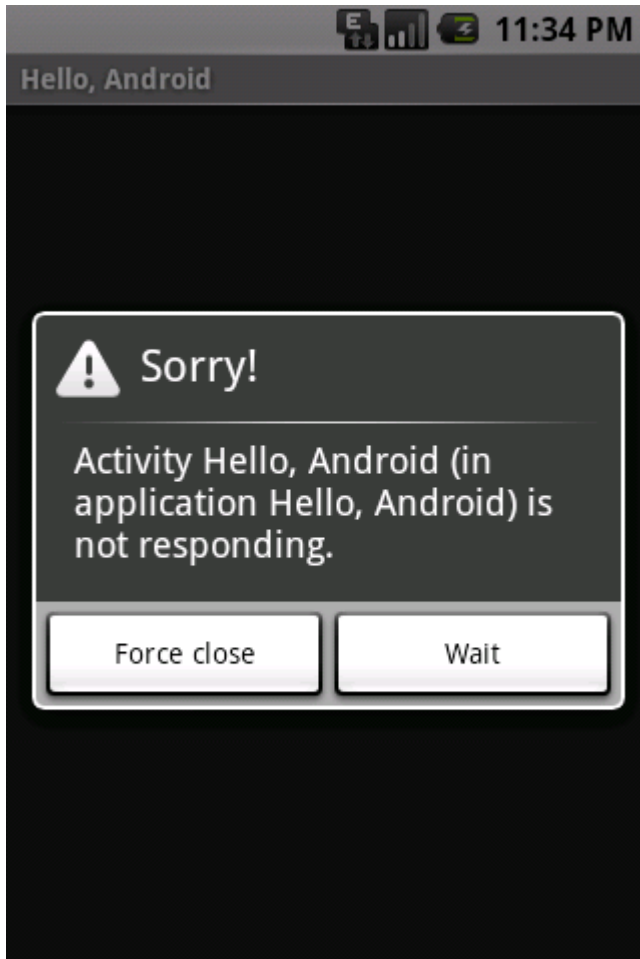
Action	Time
Add a local variable	1
Add a member variable	4
Call String.length()	5
Call empty static native method	5
Call empty static method	12
Call empty virtual method	12.5
Call empty interface method	15
Call Iterator.next() on a HashMap	165
Call put() on a HashMap	600
Inflate 1 View from XML	22,000
Inflate 1 LinearLayout containing 1 TextView	25,000
Inflate 1 LinearLayout containing 6 View objects	100,000
Inflate 1 LinearLayout containing 6 TextView objects	135,000
Launch an empty activity	3,000,000

# Σχεδιασμός για απόδοση

- \* Avoid Creating Objects
- \* Use Native Methods
- \* Prefer Virtual Over Interface
- \* Prefer Static Over Virtual
- \* Avoid Internal Getters/Setters
- \* Cache Field Lookups
- \* Declare Constants Final
- \* Use Enhanced For Loop Syntax With Caution
- \* Avoid Enums
- \* Use Package Scope with Inner Classes
- \* Avoid Float

# Σχεδιασμός για ανταποκρισιμότητα

## Application Not Responding (ANR) dialog



\* No response to an input event (e.g. key press, screen touch) within 5 seconds

\* A `BroadcastReceiver` hasn't finished executing within 10 seconds

# Σχεδιασμός για Seamlessness

- \* Don't Drop Data
- \* Don't Expose Raw Data
- \* Don't Interrupt the User
- \* Got a Lot to Do? Do it in a Thread
- \* Don't Overload a Single Activity Screen
- \* Extend System Themes
- \* Design Your UI to Work with Multiple Screen

## Resolutions

- \* Assume the Network is Slow
- \* Don't Assume Touchscreen or Keyboard
- \* Do Conserve the Device Battery

# Application Components

Οι εφαρμογές στο Android αποτελούνται από τα παρακάτω:

- **Activities**

Κυρίως παράθυρα, GUI -συγκροτούνται από Views

- **Services**

Υπηρεσίες που εκτελούνται στο background

- **Broadcast receivers**

Components που περιμένουν να ενεργοποιηθούν από ένα συμβάν

- **Content providers**

Αποθηκεύουν τα δεδομένα και τα κάνουν διαθέσιμα σε άλλες εφαρμογές



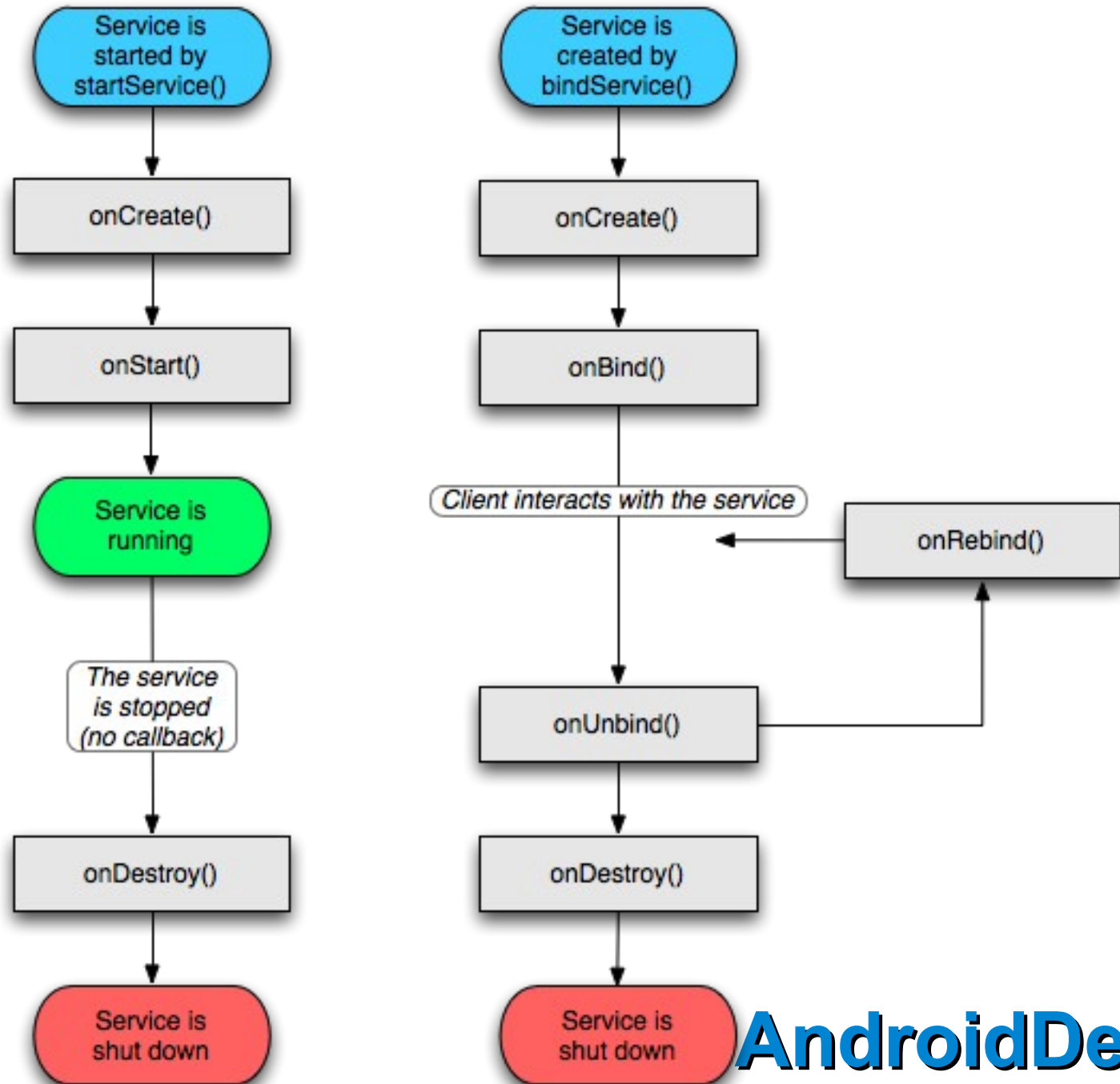


# Application Priority

Καθορίζουμε τι προτεραιότητα έχουν τα applications και με ποια σειρά θα “σκοτωθούν” σε περιπτώσεις λίγης μνήμης.

- **Foreground Process**
  - **Visible Process**
  - **Service Process**
- **Background Process**
  - **Empty Process**

# Service Lifecycle



# Activating components: intents

Τα Activities, Services και Broadcast receivers  
επικοινωνούν μεταξύ τους με Intents.

Τι είναι τα intents;  
Μηχανισμός μεταφοράς δεδομένων

# Activating components: intents

```
Intent intent1 = new Intent(CurrentClass.this,  
NextClass.class);
```

```
startActivity(intent1);
```

# Activating components: intents

```
Intent intent1 = new Intent(CurrentClass.this,  
NextClass.class);
```

```
intent1.putExtra("name", "Ventrix");
```

```
startActivity(intent1);
```

# Activating components: intents

```
Intent intent1 = new  
Intent(Intent.ACTION_VIEW,  
Uri.parse("http://www.site.com"));  
  
startActivity(intent1);
```

# The AndroidManifest.xml File

Το κεντρικό αρχείο της εφαρμογής μας

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"  
    package="com.my_domain.app.helloactivity">
```

```
    <application android:label="@string/app_name">
```

```
        <activity android:name=".HelloActivity">
```

```
            <intent-filter>
```

```
                <action android:name="android.intent.action.MAIN"/>
```

```
                <category android:name="android.intent.category.LAUNCHER"/>
```

```
            </intent-filter>
```

```
        </activity>
```

```
    </application>
```

```
</manifest>
```

# The Permission System

```
<uses-permission android:name="android.permission.***" />
```

READ\_CONTACTS

INTERNET

SEND\_SMS

RECEIVE\_SMS

VIBRATE

CHANGE\_WIFI\_STATE

...



# Resource Types

1. Simple Values
  1. Color Values
  2. Strings and Styled Text
  3. Dimension Values
2. Drawables
  1. Bitmap Files
  2. Color Drawables
  3. Nine-Patch (Stretchable) Images
3. Animation
4. Menus
5. Layout
  1. Custom Layout Resources
6. Styles and Themes

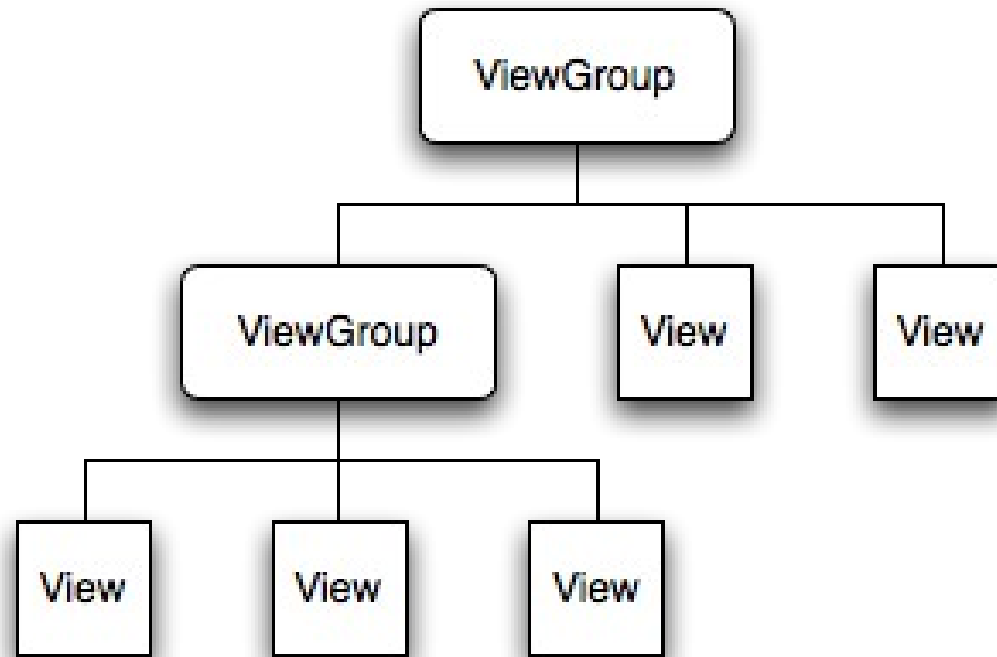
# Resource Types Example

string.xml

```
<resources>  
  <string name="hello">Welcome!</string>  
  <string name="hello2">I'm glad to see you!</string>  
</resources>
```

```
String str = getString(R.string.hello2);
```

# User Interface



# View Groups

**FrameLayout**

**Gallery**

**GridView**

**LinearLayout**

**ListView**

**RelativeLayout**

**ScrollView**

**Spinner**

**SurfaceView**

**TabHost**

**TableLayout**

**ViewFlipper**

# User Interface

## Δυο τρόποι δημιουργίας GUI

**Declare UI elements in XML.**

**Android provides a straightforward XML vocabulary that corresponds to the View classes and subclasses, such as those for widgets and layouts.**

# User Interface

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical" >
    <TextView android:id="@+id/text"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello, I am a TextView" />
    <Button android:id="@+id/button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello, I am a Button" />
</LinearLayout>
```

# User Interface

Χρησιμοποιώντας το XML File

```
public void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.main_layout);  
}
```

# User Interface

## Δυο τρόποι δημιουργίας GUI

**Instantiate layout elements at runtime.**

**Your application can create View and ViewGroup objects (and manipulate their properties) programmatically.**



# Android Developers Site



<http://developer.android.com/>

# FossComm Λάρισα 9-10/5/2009



## Real World Example