

“无之以为用，有之以为利”

---老子.道德经

# 什么是3G<—>

**3G**，全称为3rd Generation，中文含义就是指第三代数字通信。

所谓**3G**，是指将无线通信与**国际互联网**等多媒体通信结合的新一代移动通信系统。**3G**只是一种通信技术标准，符合这个标准的技术有WCDMA、CDMA2000、TD-SCDMA三种无线接口标准。中国联通使用的是WCDMA（世界上大部分**3G**网络都采用的是该标准）；中国电信使用的是CDMA2000（日、韩和北美使用）；中国移动使用的是具有自主知识产权的TD-SCDMA（只有中国才使用）。相对第一代模拟制式手机(1G)和第二代GSM、CDMA等数字手机(2G)，**3G**手机能处理图像、音乐、视频等多种媒体形式，提供包括网页浏览、电话会议、电子商务等多种信息服务。**3G**网络与前两代的主要区别是整合了互联网并且数据传输速度有很大的提升。

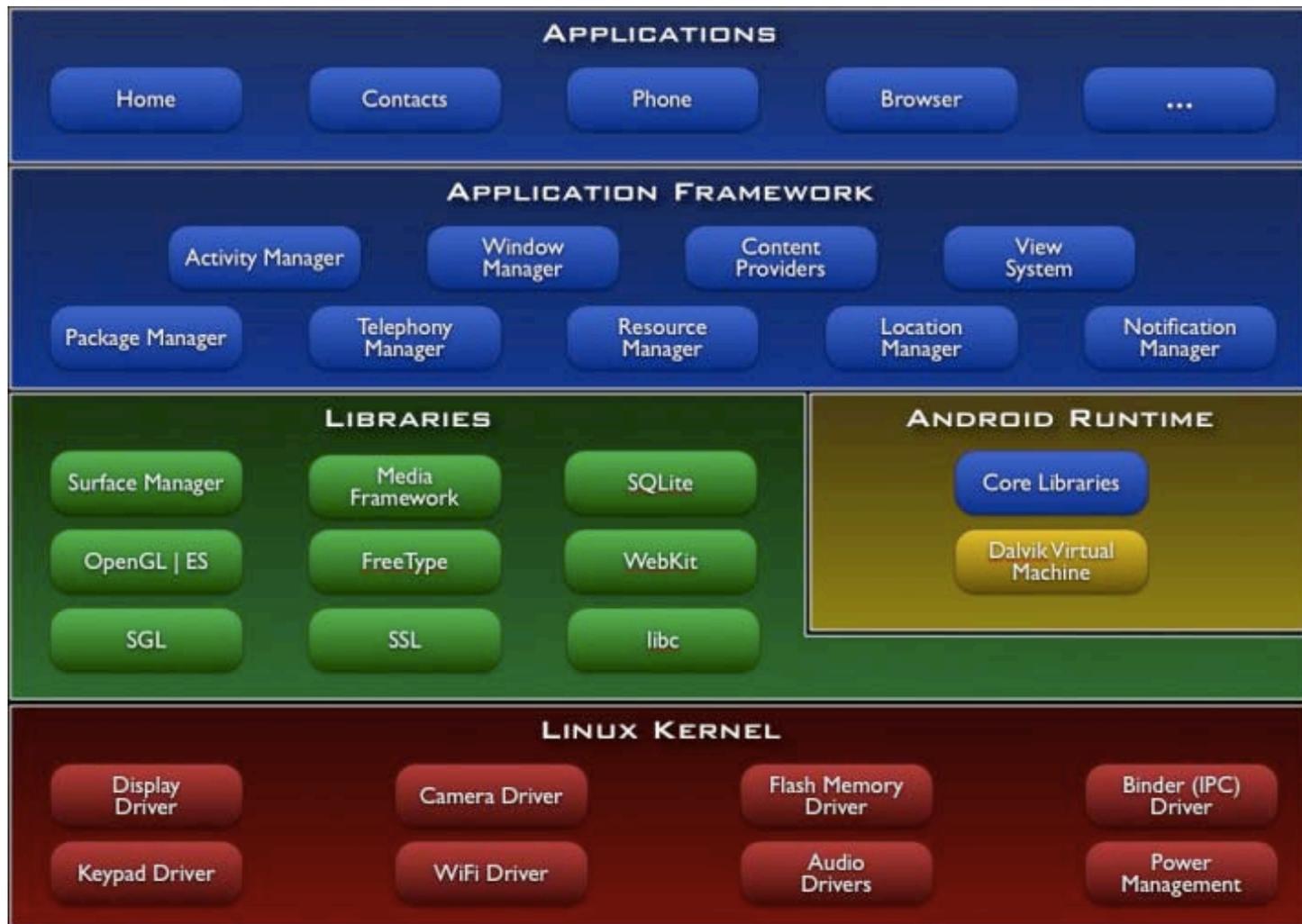
# 什么是3G<二>

目前中国正在建设3G网络，大城市的3G网络基本上已经铺设好，像北京铺设好了超过2000个基站，深圳铺设了1000多个基站。但是要全民普及到3G手机尚且需要几年时间。现在，大家使用的手机大多还是2.5G手机，符合2.5G网络的接口标准有CDMA2000 1X和GPRS，中国联通使用的是CDMA2000 1X标准，中国移动使用的是GPRS标准。目前，我们可以把2.5G移动通信技术看作是2G迈向3G的衔接性技术，在2.5G网络下出现了如WAP、蓝牙(Bluetooth)等技术。对于2.5G网络，我们应该也要有所了解，因为以后大家到企业中有可能面对2.5G网络下的应用，如：wap项目。

# 什么是Android

- Android是Google在2007年11月5日推出的开源手机操作系统。目前Android在中国的发展是非常火的，为了节省研发费用，国内很多的手机厂商和移动运营商已经开始使用上android操作系统了，其中包括中国移动，中国联通,中国电信，华为，联想和一些山寨手机厂商。
- 另外android应用的范围不仅仅在手机，还被应用在汽车、平板电脑、和智能上网设备上。
- 因为目前大家使用的手机大多属于傻瓜手机，也就是打打电话、发发短信。从今天开始，大家对智能手机要有个重新的认识，智能手机除了可以打电话发短信之外还能完成电脑所能完成一切，所以大家需要把智能手机看作是一台电脑。

# Android软件堆栈



# 智能手机软件平台

智能手机软件平台有：

Symbian, **Windows Phone**, RIM BlackBerry, **Android**, **iPhone**,  
Java/J2ME。

2011年市场份额(美国2010年11月到2011年1月之间):

Android 29%

iPhone 27%

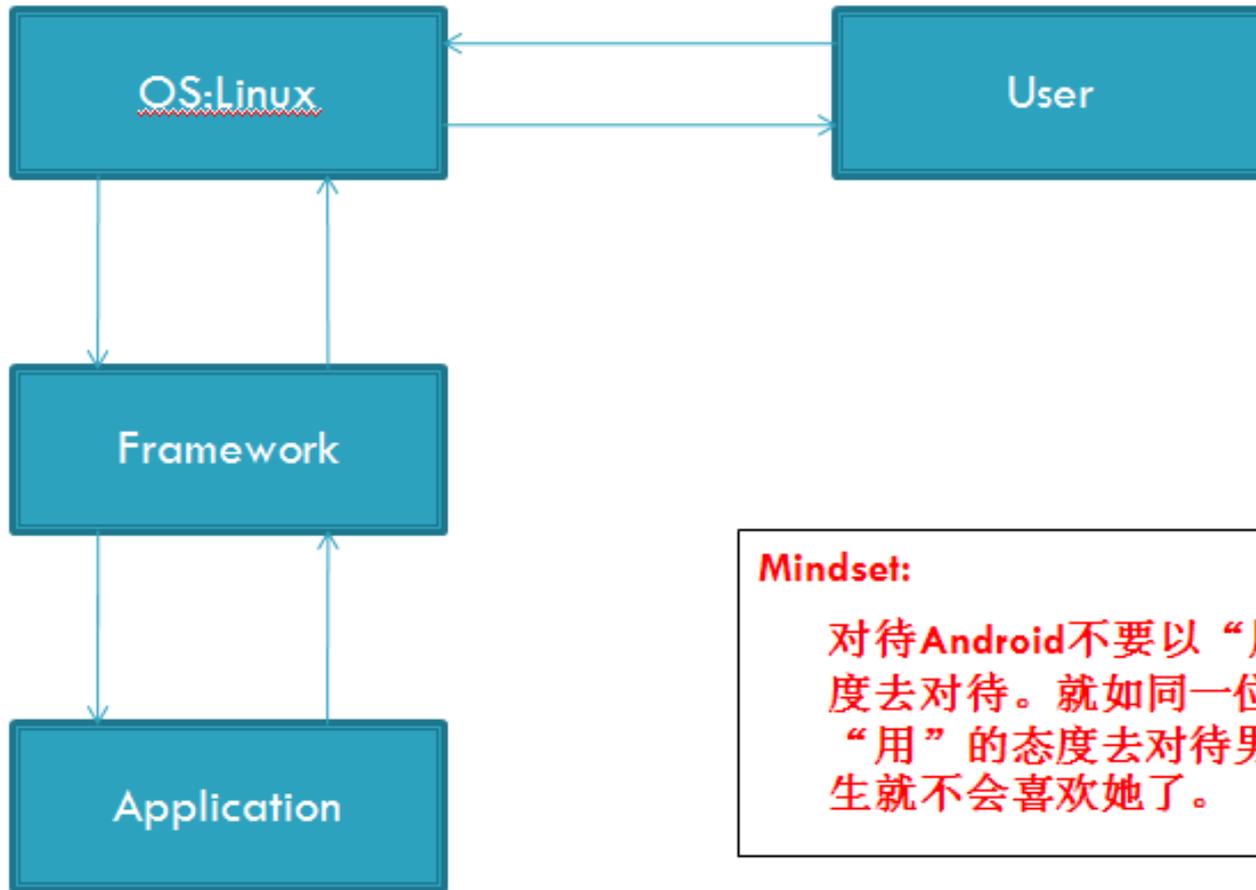
windows Phone 10%

Symbian 2%

# Android应用框架的特点

- 目的：让应用软件工程师派生具体子类
- 应用框架中的函数常需要调用应用程序中的函数
- 既定的类别之间的关系，预设对象之间的交互关系和行为
- 拥有**预设的函数实现**（**Default Behavior**），Android应用开发工程师可以根据需要在子类中修改默认行为。

# Android应用程序与使用者的交互关系图



## Mindset:

对待Android不要以“用”的态度去对待。就如同一位少女以“用”的态度去对待男生，男生就不会喜欢她了。

# 如何安装 Android SDK 和Eclipse 插件<一>

- 所需开发环境:JDK 5 或 JDK 6 (仅有JRE不够)、 Eclipse 3.4 或以上版本
- 下载安装**Android SDK**: Android SDK包含了开发Android应用所依赖的jar文件、运行环境及相关工具。

下载地址: [http://dl.google.com/android/android-sdk\\_r13-windows.zip](http://dl.google.com/android/android-sdk_r13-windows.zip)

下载完SDK后,把.zip文件解压到你电脑上合适位置。使用解压包中的SDK Manager安装需要的组件,我们在这里安装的Android2.2.启动Eclipse,选择window->preferences,在打开的视图左边点击android,在右边的SDK Location中选择**Android SDK**所在位置。

# 如何安装 Android SDK 和Eclipse 插件<二>

## 安装 Eclipse 插件 (ADT)

启动 Eclipse，选择 Help > Install New Software，在出现的对话框里，点击Add按钮，在对话框的name一栏输入“ADT”，然后点击Archive...，浏览和选择已经下载的**ADT** 插件压缩文件。

点击 OK.。返回可用软件的视图，你会看到这个插件，然后选择Developer Tools (会选中下面的“Android Developer Tools”和“Android Editors“)，点击 Next，最后重启 Eclipse。

# 如何安装 Android SDK 和Eclipse 插件<三>

## □ 安装 Eclipse 插件 (ADT)

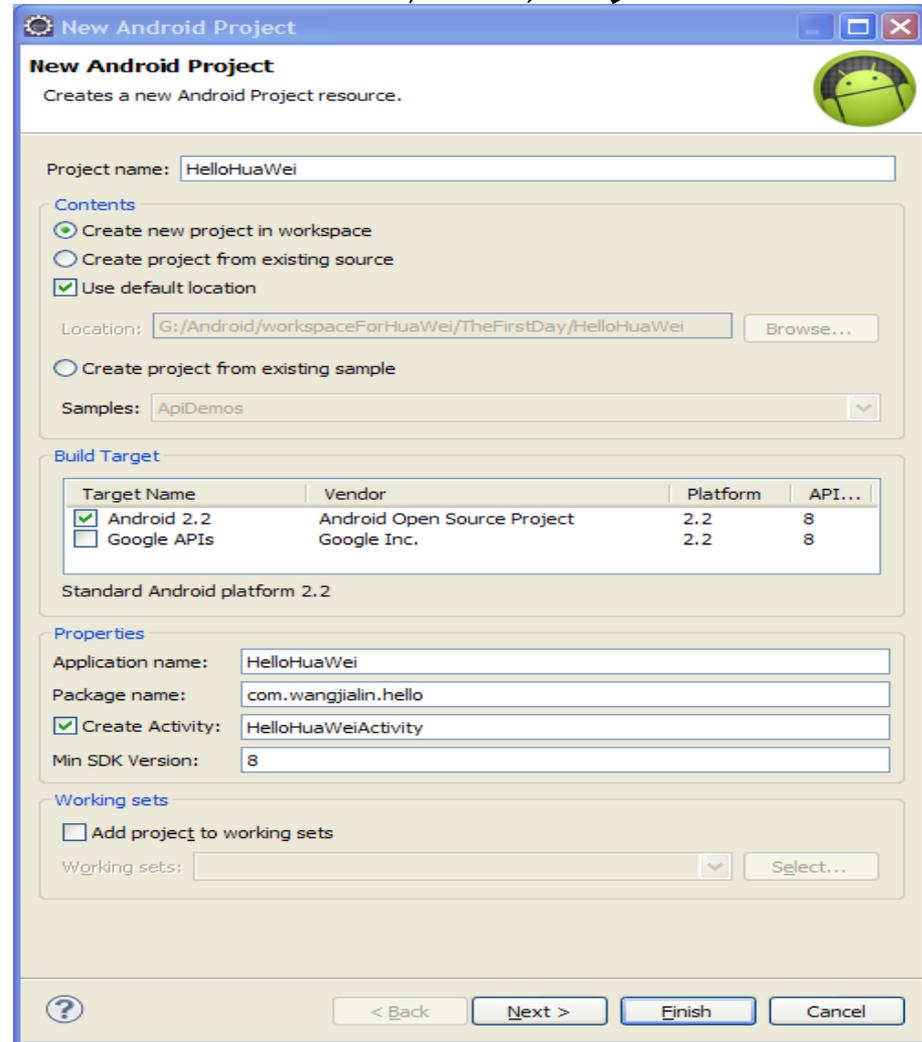
方法一：启动 Eclipse，选择 Help > Install New Software，在出现的对话框里，点击Add按钮，在对话框的name一栏输入“ADT”，然后点击Archive...，浏览和选择已经下载的**ADT**插件压缩文件。

点击 OK。返回可用软件的视图，你会看到这个插件，然后选择Developer Tools (会选中下面的“Android Developer Tools”和“Android Editors“)，点击 Next，最后重启 Eclipse。

方法二： **Help > Install New Software> click Add>**  
Location输入 <https://dl-ssl.google.com/android/eclipse/>

# 开发第一个Android应用

打开Eclipse，新建项目(点击File→New→Project)，在项目列表中展开Android目录，选择Android Project:



# 开发第一个Android应用

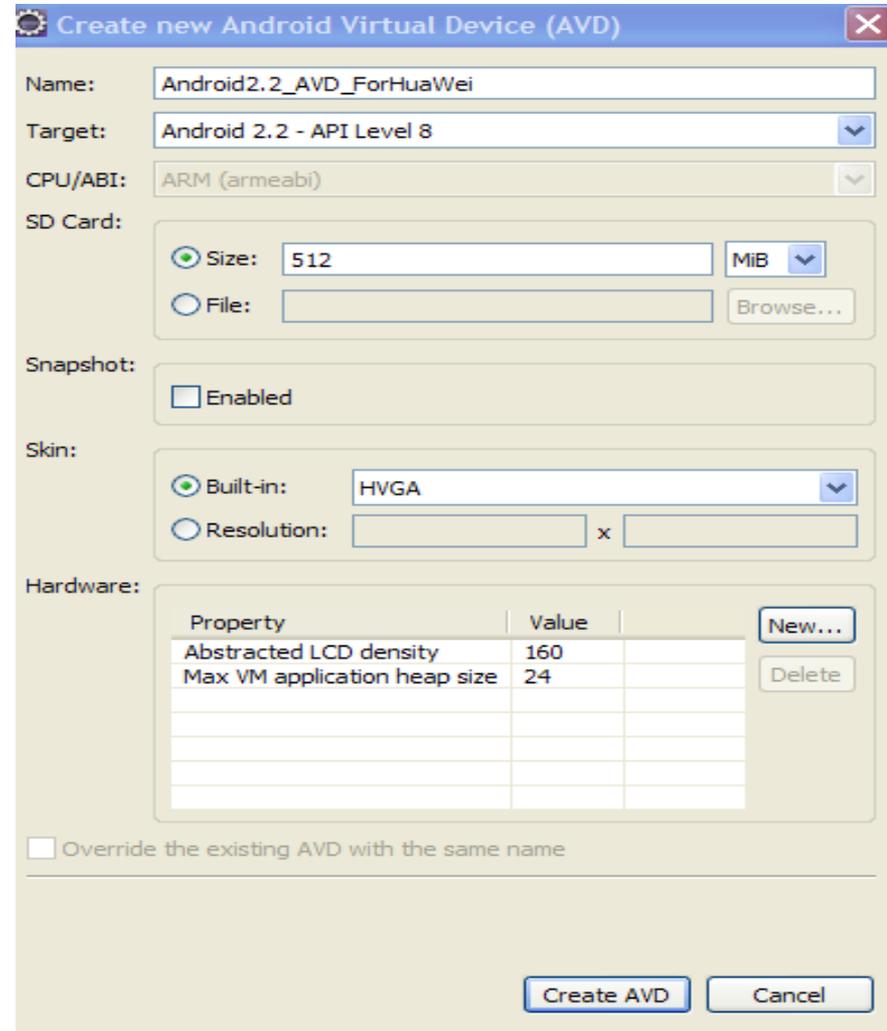
点击“finish”即可完成项目的创建，创建后的项目已经是一个可运行的Android应用，我们可以通过下面方式运行此应用：

点击工具栏上机器人形状的虚拟设备管理器（简称“AVD“），如下：



# 开发第一个Android应用

在打开的虚拟设备管理器  
中创建一个虚拟手机：



# 屏幕分辨率

- 屏幕分辨率
- HVGA:  $320 \times 480$
- QVGA:  $240 \times 320$
- WQVGA400:  $240 \times 400$
- WQVGA432:  $240 \times 432$
- WVGA800:  $480 \times 800$
- WVGA854:  $480 \times 854$
  
- Android软件开发普遍支持的机型: HTC G1,G2,G3,G4 和 Moto Droid
  
- 模拟器屏幕切换Ctrl+F12

# 开发第一个Android应用

- 在项目上右键点击run as → Android application，如下图：



# Android应用程序架构<一>

**src/** java原代码存放目录

**gen/** 自动生成目录

**gen** 目录中存放所有由Android开发工具自动生成的文件。目录中最重要的就是R.java文件。这个文件由Android开发工具自动产生的。Android开发工具会自动根据你放入res目录的资源，同步更新修改R.java文件。正因为R.java文件是由开发工具自动生成的，所以我们应该避免手工修改R.java。R.java在应用中起到了字典的作用，它包含了各种资源的id，通过R.java，应用可以很方便地找到对应资源。

**res/** 资源(Resource)目录

在这个目录中我们可以存放应用使用到的各种资源，如xml界面文件，图片或数据。具体请看ppt下方备注栏。

# Android应用程序架构<二>

## **assets**资源目录

Android除了提供/res目录存放资源文件外，在/**assets**目录也可以存放资源文件，而且/**assets**目录下的资源文件不会在R.java自动生成ID，所以读取/**assets**目录下的文件必须指定文件的路径，如：

```
file:///android_asset/xxx.3gp
```

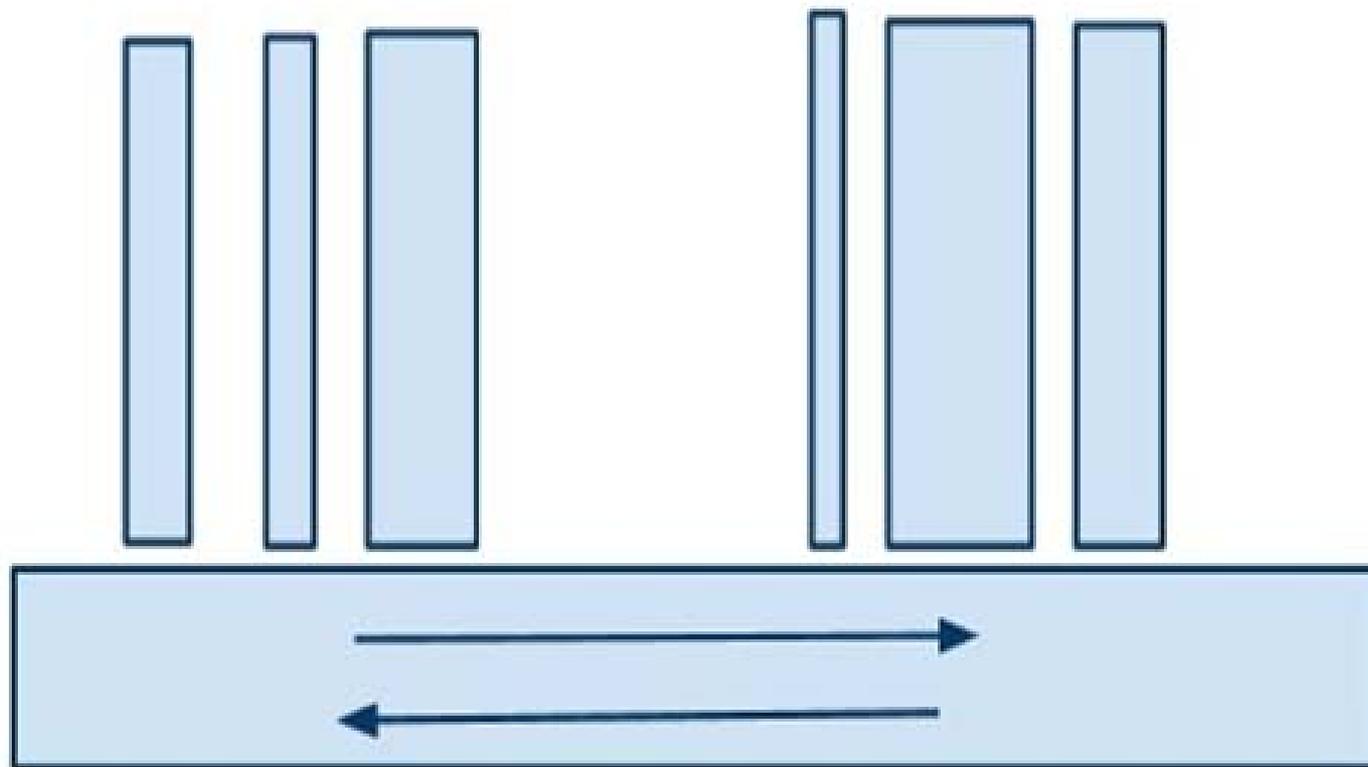
## **AndroidManifest.xml** 项目清单文件

这个文件列出了应用程序所提供的功能，以后你开发好的各种组件（**Activity**、**ContentProvider**、**BroadcastReceiver**、**Service**）需要在该文件中进行配置，如果应用使用到了系统内置的应用(如电话服务、互联网服务、短信服务、**GPS**服务等等)，你还需在该文件中声明使用权限。

**default.properties** 项目环境信息，一般是不需要修改此文件

# Android应用程序架构<三>

In Android



# Android应用程序架构<四>

`res/drawable` 专门存放png、jpg等图像资源。在代码中使用 `getResources().getDrawable(resourceId)` 获取该目录下的资源。

- `drawable-hdpi`、`drawable-mdpi`、`drawable-ldpi`的区别：
- (1)`drawable-hdpi`里面存放高分辨率的图片,如WVGA (480x800),FWVGA (480x854)
- (2)`drawable-mdpi`里面存放中等分辨率的图片,如HVGA (320x480)
- (3)`drawable-ldpi`里面存放低分辨率的图片,如QVGA (240x320)
- 系统会根据机器的分辨率来分别到这几个文件夹里面去找对应的图片。
- 在开发程序时为了兼容不同平台不同屏幕,建议各自文件夹根据需求均存放不同版本图片。

`res/layout` 专门存放xml界面文件,xml界面文件和HTML文件一样,主要用于显示用户操作界面。

# Android应用程序架构<五>

`res/values` 专门存放应用使用到的各种类型数据。不同类型的数据存放在不同的文件中，如下：

- `strings.xml` 定义字符串和数值，在Activity中使用`getResources().getString(resourceId)`或`getResources().getText(resourceId)`取得资源。它的作用和struts中的国际化资源文件一样。

```
<?xml version="1.0" encoding="UTF-8"?>
<resources>
  <string name="wangjialin">王家林</string>
</resources>
```

- `arrays.xml` 定义数组。

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
  <string-array name="colors">
    <item>red</item>
    <item>yellow</item>
    <item>green</item>
    <item>blue</item>
  </string-array>
</resources>
```

# Android应用程序架构<六>

- colors.xml 定义颜色和颜色字串数值，你可以在Activity中使用getResources().getDrawable(resourceId) 以及getResources().getColor(resourceId)取得这些资源。例子如下：

```
<?xml version="1.0" encoding="UTF-8"?>
<resources>
  <color name="contents_text">#ff0000</color>
</resources>
```

- dims.xml 定义尺寸数据，在Activity中使用getResources().getDimension(resourceId) 取得这些资源

```
<?xml version="1.0" encoding="UTF-8"?>
<resources>
  <dimen name="key_height">50dip</dimen>
</resources>
```

# Android应用程序架构<七>

- styles.xml 定义样式。

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <style name="wangjialinText" parent="@style/Text">
        <item name="android:textSize">18sp</item>
        <item name="android:textColor">#0066FF</item>
    </style>
</resources>
```

res/anim/ 存放定义动画的XML文件。

res/xml/ 在Activity中使用getResources().getXML() 读取该目录下的XML资源文件。

- res/raw/ 该目录用于存放应用使用到的原始文件，如音效文件等。编译软件时，这些数据不会被编译，它们被直接加入到程序安装包里。为了在程序中使用这些资源，你可以调用getResources().openRawResource(ID)，参数ID形式：*R.raw.somefilename*。

# 电话拨号器

效果图:



# 电话拨号器

因为应用要使用手机的电话服务，所以要在清单文件  
AndroidManifest.xml中添加电话服务权限：

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.wangjialin.action"
    android:versionCode="1"
    android:versionName="1.0">
    略....
    <uses-sdk android:minSdkVersion="8" />
    <uses-permission
        android:name="android.permission.CALL_PHONE"/>
</manifest>
```

# 电话拨号器

界面布局:

```
<?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/inputmobile"/>

    <EditText android:layout_width="fill_parent" android:layout_height="wrap_content"
        android:id="@+id/mobile"/>

    <Button android:layout_width="wrap_content" android:layout_height="wrap_content"
        android:text="@string/button"
        android:id="@+id/button"/>
</LinearLayout>
```

LinearLayout (线性布局)、[AbsoluteLayout\(绝对布局\)](#)、RelativeLayout(相对布局)、TableLayout(表格布局)、FrameLayout(帧布局)

# Android中的显示单位

- px (pixels)像素  
一般HVGA代表320x480像素，这个用的比较多。
  - dip或dp (device independent pixels)设备独立像素  
这个和设备硬件有关，一般为了支持WVGA、HVGA和QVGA 推荐使用这个，不依赖像素。
  - sp (scaled pixels — best for text size)比例像素  
主要处理字体的大小，可以根据系统的字体自适应。
- 除了上面三个显示单位，下面还有几个不太常用：
- in (inches)英寸
  - mm (millimeters)毫米
  - pt (points)点， 1/72英寸
- 为了适应不同分辨率，不同的像素密度，推荐使用dip，文字使用sp。

# 电话拨号器

Activity:

```
public class DialerAction extends Activity {
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
        Button button = (Button)findViewById(R.id.button);
        button.setOnClickListener(new View.OnClickListener(){
            public void onClick(View v) {
                EditText editText = (EditText)findViewById(R.id.mobile);
                Intent intent = new Intent(Intent.ACTION_CALL, Uri.parse("tel:"+ editText.getText()));
                DialerAction.this.startActivity(intent);
            }
        });
    }
}
```

# 电话拨号器

测试步骤:

1>在Eclipse中运行此应用

2>开启一个Android模拟器:



3>在电话播号器中输入上图现显的电话号码

# “尚未注册网络”错误信息的解决办法<一>



打开Android模拟器时，出现无信号，拨打电话或发短信时，提示“尚未注册网络”错误信息的解决方案如下。

- **场景一：你的电脑没有连接上互联网，同时也没有在局域网。**

解决办法：右键点击网上邻居,选择"属性",在网络连接窗口中右键点击"本地连接",选择"属性",设置TCP/IP属性如下:

IP地址:192.168.1.100

子网掩码:255.255.255.0

默认网关:192.168.1.100

首选DNS服务器:192.168.1.100

## “尚未注册网络” 错误信息的解决办法<二>

- 场景二：你的电脑没有连接上互联网，但在局域网。

解决办法：右键点击网上邻居,选择"属性",在网络连接窗口中右键点击"本地连接",选择"属性",设置TCP/IP属性如下:

IP地址:设置成你所在局域网的IP, 如: 192.168.1.100

子网掩码:设置成你所在局域网的掩码, 如: 255.255.255.0

默认网关:设置成你所在局域网的网关, 一般网关的IP格式为: \*.\*.\*.1, 如: 192.168.1.1

首选DNS服务器:设置成你所在局域网的路由器IP, 一般路由器的IP格式为: \*.\*.\*.1, 如: 192.168.1.1

- 最后一种解决方案是：让你的电脑连接上互联网。

# 短信发送器

效果图:



# 短信发送器

因为应用要使用手机的短信服务，所以要在清单文件  
AndroidManifest.xml中添加短信服务权限：

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.wangjialin.sms"
    android:versionCode="1"
    android:versionName="1.0">
    略....
    <uses-sdk android:minSdkVersion="8" />
    <uses-permission
        android:name="android.permission.SEND_SMS"/>
</manifest>
```

# 短信发送器

界面布局:

```
<?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/inputmobile"/>

    <EditText android:layout_width="fill_parent" android:layout_height="wrap_content"
        android:id="@+id/mobile"/>

    <TextView android:layout_width="fill_parent" android:layout_height="wrap_content"
        android:text="@string/content"/>

    <EditText android:layout_width="fill_parent" android:layout_height="wrap_content"
        android:minLines="3"
        android:id="@+id/content"/>

    <Button android:layout_width="wrap_content" android:layout_height="wrap_content"
        android:text="@string/button"
        android:id="@+id/button"/>
</LinearLayout>
```

# 短信发送器

Activity主要代码:

```
String mobile = mobileView.getText().toString();
String content = contentView.getText().toString();
SmsManager smsManager = SmsManager.getDefault();
PendingIntent sentIntent = PendingIntent.getBroadcast(SMSSender.this, 0, new
Intent(), 0);
//如果字数超过70,需拆分成多条短信发送
List<String> msgs = smsManager.divideMessage(content);
for(String msg : msgs){
    smsManager.sendTextMessage(mobile, null, msg, sentIntent, null);
    //第四个参数用于告之短信发送状态,最后一个参数用于告之短信接收状态
}
Toast.makeText(SMSSender.this, "短信发送完成", Toast.LENGTH_LONG).show();
```

# 发送彩信

可以通过调用系统自带的短信程序发送彩信：

```
Intent intent = new Intent(Intent.ACTION_SEND);  
intent.putExtra(Intent.EXTRA_STREAM,  
    Uri.parse("file:///sdcard/cong.png"));  
intent.putExtra("address", "13677789999");  
intent.putExtra("exit_on_sent", true);  
intent.putExtra("subject", "it's subject");  
intent.putExtra("sms_body", "it's content");  
intent.setType("image/jpeg"); // 视频: video/mpeg* , 文本: text/plain  
startActivity(intent);
```

# 对应用进行单元测试<一>

在实际开发中，开发android软件的过程需要不断地进行测试。使用JUnit测试框架，是正规Android开发的必用技术，在JUnit中可以得到组件，可以模拟发送事件和检测程序处理的正确性。

第一步：首先在AndroidManifest.xml中加入下面红色代码：

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.wangjialin.action" android:versionCode="1"
    android:versionName="1.0">
    <application android:icon="@drawable/icon" android:label="@string/app_name">
        <uses-library android:name="android.test.runner" />
        ....
    </application>
    <uses-sdk android:minSdkVersion="8" />
    <instrumentation android:name="android.test.InstrumentationTestRunner"
        android:targetPackage="com.wangjialin.action" android:label="Tests for My App"
        />
</manifest>
```

上面targetPackage指定的包要和应用的package相同。

# 对应用进行单元测试<二>

第二步：编写单元测试代码（选择要测试的方法，右键点击“Run As”--“Android Junit Test”）：

```
import android.test.AndroidTestCase;
import android.util.Log;
public class XMLTest extends AndroidTestCase {
    public void testSomething() throws Throwable {
        Assert.assertTrue(1 + 1 == 3);
    }
}
```

# 数据存储与访问

很多时候我们开发的软件需要对处理后的数据进行存储,以供再次访问。Android为数据存储提供了如下几种方式:

- 文件
- **SharedPreferences**(偏好参数保存)
- **SQLite**数据库
- 内容提供者 (Content provider)
- 网络

# 使用文件进行数据存储<一>

首先给大家介绍使用文件如何对数据进行存储，Activity提供了openFileOutput()方法可以用于把数据输出到文件中，具体的实现过程与在J2SE环境中保存数据到文件中是一样的。

```
public class FileActivity extends Activity {  
    @Override public void onCreate(Bundle savedInstanceState) {  
        ...  
        FileOutputStream outputStream = this.openFileOutput("wangjialin.txt",  
Context.MODE_PRIVATE);  
        outputStream.write("王家林".getBytes());  
        outputStream.close();  
    }  
}
```

openFileOutput()方法的第一参数用于指定文件名称，不能包含路径分隔符“/”，如果文件不存在，Android会自动创建它。

# 使用文件进行数据存储<二>

创建的文件保存在/data/data/<package name>/files目录，如：  
[/data/data/com.wangjialin.action/files/wangjialin.txt](#)，通过点击Eclipse菜单“Window”-“Show View”-“Other”，在对话框中展开android文件夹，选择下面的File Explorer视图，然后在File Explorer视图中展开/data/data/<package name>/files目录就可以看到该文件。

openFileOutput()方法的第二参数用于指定操作模式，有四种模式，分别为：  
Context.MODE\_PRIVATE = 0

Context.MODE\_APPEND = 32768

Context.MODE\_WORLD\_READABLE = 1

Context.MODE\_WORLD\_WRITEABLE = 2

# 读取文件内容<一>

如果要打开存放在/data/data/<package name>/files目录应用私有的文件，可以使用Activity提供openFileInput()方法。

```
FileInputStream inStream =  
    this.getContext().openFileInput("wangjialin.txt");  
Log.i("FileTest", readInStream(inStream));  
readInStream()的方法请看本页下面备注。
```

或者直接使用文件的绝对路径：`File file = new File("/data/data/com.wangjialin.action/files/wangjialin.txt");`  
`FileInputStream inStream = new FileInputStream(file);`  
`Log.i("FileTest", readInStream(inStream));`

**注意：**上面文件路径中的“`com.wangjialin.action`”为应用所在包，当你在编写代码时应替换为你自己应用使用的包。

# 读取文件内容<二>

对于私有文件只能被创建该文件的应用访问，如果希望文件能被其他应用读和写，可以在创建文件时，指定 `Context.MODE_WORLD_READABLE` 和 `Context.MODE_WORLD_WRITEABLE` 权限。

Activity还提供了 `getCacheDir()` 和 `getFilesDir()` 方法：

`getCacheDir()` 方法用于获取 `/data/data/<package name>/cache` 目录

`getFilesDir()` 方法用于获取 `/data/data/<package name>/files` 目录

# 读取文件内容<三>

- `public static String readInStream(FileInputStream inStream){`
- `try {`
- `ByteArrayOutputStream outputStream = new ByteArrayOutputStream();`
- `byte[] buffer = new byte[1024];`
- `int length = -1;`
- `while((length = inStream.read(buffer)) != -1 ){`
- `outputStream.write(buffer, 0, length);`
- `}`
- `outputStream.close();`
- `inStream.close();`
- `return outputStream.toString();`
- `} catch (IOException e) {`
- `Log.i("FileTest", e.getMessage());`
- `}`
- `return null;`
- `}`

# 把文件存放在SDCard<一>

使用Activity的openFileOutput()方法保存文件，文件是存放在手机空间上，一般手机的存储空间不是很大，存放些小文件还行，如果要存放像视频这样的大文件，是不可行的。对于像视频这样的大文件，我们可以把它存放在SDCard。SDCard是干什么的？你可以把它看作是移动硬盘或U盘。



在模拟器中使用SDCard，你需要先创建一张SDCard卡（当然不是真的SDCard，只是镜像文件）。创建SDCard可以在Eclipse创建模拟器时随同创建，也可以使用DOS命令进行创建，如下：

在Dos窗口中进入android SDK安装路径的tools目录，输入以下命令创建一张容量为2G的SDCard，文件后缀可以随便取，建议使用.img：

```
mksdcard 2048M F:\Android\sdcard.img
```

# 把文件存放在SDCard<二>

在程序中访问**SDCard**，你需要申请访问**SDCard**的权限。

在AndroidManifest.xml中加入访问SDCard的权限如下：

```
<!-- 在SDCard中创建与删除文件权限 -->
```

```
<uses-permission
```

```
    android:name="android.permission.MOUNT_UNMOUNT_FILESYS  
    TEMS"/>
```

```
<!-- 往SDCard写入数据权限 -->
```

```
<uses-permission
```

```
    android:name="android.permission.WRITE_EXTERNAL_STORAGE  
    "/>
```

# 把文件存放在SDCard<一>

要往SDCard存放文件，程序必须先判断手机是否装有SDCard，并且可以进行读写。

**注意：**访问SDCard必须在AndroidManifest.xml中加入访问SDCard的权限

```
if(Environment.getExternalStorageState().equals(Environment.MEDIA_
MOUNTED)){
    File sdCardDir = Environment.getExternalStorageDirectory();//获
    取SDCard目录
    File saveFile = new File(sdCardDir, "wangjialin.txt");
    FileOutputStream outputStream = new FileOutputStream(saveFile);
    outputStream.write("王家林".getBytes());
    outputStream.close();
}
```

# 把文件存放在SDCard<二>

`Environment.getExternalStorageState()`方法用于获取SDCard的状态，如果手机装有SDCard，并且可以进行读写，那么方法返回的状态等于`Environment.MEDIA_MOUNTED`。

`Environment.getExternalStorageDirectory()`方法用于获取SDCard的目录，当然要获取SDCard的目录，你也可以这样写：

```
File sdCardDir = new File("/mnt/sdcard"); //获取SDCard目录
```

```
File saveFile = new File(sdCardDir, "wangjialin.txt");
```

```
//上面两句代码可以合成一句： File saveFile = new  
File("/mnt/sdcard/wangjialin.txt");
```

```
FileOutputStream outputStream = new FileOutputStream(saveFile);
```

```
outputStream.write("Hey,WangJialin".getBytes());
```

```
outputStream.close();
```

# 使用SAX或者DOM或者pull解析XML文件

在Android平台上可以使用Simple API for XML(SAX)、Document Object Model(DOM)和Android附带的pull解析器解析XML文件。下面是本例子要解析的XML文件：

文件名称： wangjialin.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<persons>
  <person id="23">
    <name>wangjialin</name>
    <age>58</age>
  </person>
  <person id="20">
    <name>Android</name>
    <age>4</age>
  </person>
</persons>
```

# XML文件的javabean

例子定义了一个javabean用于存放上面解析出来的xml内容，这个javabean为Person，

```
public class Person {  
    private Integer id;  
    private String name;  
    private Short age;  
  
    public Integer getId() {  
        return id;  
    }  
    public void setId(Integer id) {  
        this.id = id;  
    }  
    public String getName() {  
        return name;  
    }  
    public void setName(String name) {  
        this.name = name;  
    }  
    public Short getAge() {  
        return age;  
    }  
    public void setAge(Short age) {  
        this.age = age;  
    }  
}
```

# 使用SAX读取XML文件<一>

SAX是一个解析速度快并且占用内存少的xml解析器，非常适合用于Android等移动设备。SAX解析XML文件采用的是事件驱动，也就是说，它并不需要解析完整整个文档，在按内容顺序解析文档的过程中，SAX会判断当前读到的字符是否合法XML语法中的某部分，如果符合就会触发事件。所谓事件，其实就是一些回调（callback）方法，这些方法(事件)定义在ContentHandler接口。下面是一些ContentHandler接口常用的方法：

## **startDocument()**

当遇到文档的开头的时候，调用这个方法，可以在其中做一些预处理的工作。

## **endDocument()**

和上面的方法相对应，当文档结束的时候，调用这个方法，可以在其中做一些善后的工作。

# 使用SAX读取XML文件<二>

**startElement(String namespaceURI, String localName, String qName, Attributes atts)**

当读到一个开始标签的时候，会触发这个方法。namespaceURI就是命名空间，localName是不带命名空间前缀的标签名，qName是带命名空间前缀的标签名。通过atts可以得到所有的属性名和相应的值。要注意的是SAX中一个重要的特点就是它的流式处理，当遇到一个标签的时候，它并不会纪录下以前所碰到的标签，也就是说，在startElement()方法中，所有你所知道的信息，就是标签的名字和属性，至于标签的嵌套结构，上层标签的名字，是否有子元素等等其它与结构相关的信息，都是不得而知的，都需要你的程序来完成。这使得SAX在编程处理上没有DOM来得那么方便。

**endElement(String uri, String localName, String name)**

这个方法和上面的方法相对应，在遇到结束标签的时候，调用这个方法。

**characters(char[] ch, int start, int length)**

这个方法用来处理在XML文件中读到的内容，第一个参数为文件的字符串内容，后面两个参数是读到的字符串在这个数组中的起始位置和长度，使用new String(ch,start,length)就可以获取内容。

# 使用SAX读取XML文件<三>

```
<?xml version="1.0" encoding="UTF-8"?>
<persons>
  <person id="23">
    <name>王家林</name>
    <age>58</age>
  </person>
  <person id="20">
    <name>Android</name>
    <age>4</age>
  </person>
</persons>
```

解析wangjialin.xml触发的事件为:

读到的标签及内容	触发事件
{文档开始}	startDocument()
<persons>	startElement(, "persons", null, "{Attributes}")
"\n\t"	characters("<persons>...</persons>", "12", "2")
<person>	startElement(, "person", null, "{Attributes}")
"\n\t\t"	characters("<persons>...</persons>", "31", "3")
<name>	startElement(, "name", null, "{Attributes}")
"王家林"	characters("<persons>...</persons>", "40", "2")
</name>	endElement("", "name", null)

# 使用SAX读取XML文件<四>

```
"\n\t\t"      characters("<persons>...</persons>", "50", "3")
<age>        startElement(, "age", null, "{Attributes}")
"30"         characters("<persons>...</persons>", "58", "2")
</age>       endElement("", "age", null)
"\n\t"       characters("<persons>...</persons>", "67", "2")
</person>    endElement("", "person", null)
"\n\t"       characters("<persons>...</persons>", "79", "2")
<person>     startElement(, "person", null, "{Attributes}")
"\n\t\t"     characters("<persons>...</persons>", "98", "3")
<name>       startElement(, "name", null, "{Attributes}")
"Android"    characters("<persons>...</persons>", "107", "3")
</name>      endElement("", "name", null)
"\n\t\t"     characters("<persons>...</persons>", "118", "3")
<age>        startElement(, "age", null, "{Attributes}")
"25"         characters("<persons>...</persons>", "126", "2")
</age>       endElement("", "age", null)
"\n\t"       characters("<persons>...</persons>", "135", "2")
</person>    endElement("", "person", null)
"\n"         characters("<persons>...</persons>", "147", "1")
</persons>   endElement("", "persons", null)
{文档结束}   endDocument()
```

# Android应用框架魅力的源泉：**IoC**

- 应用框架最独特之处：框架里的函数能够调用应用程序里的函数。
- **IoC**源于OO语言的类别继承体系。
- 预知和**控制**未来

# IoC的定义和两种方式

- **IoC**: 框架拥有控制权而主动调用应用程序的情形同称为**IoC**
- **IoC**的两种方式: 继承接口、依赖注入 (**Dependency Inject**) 。
- **DI**: 此时的框架是一个**Container**, 掌握对象的**Lifecycle**和**Structure**。

# 创建框架的步骤

- 抽象：抽出共同之现象
  - 观察
  - 分别异同点
  - 抽取相同点
- 派生或者**Contain**
- **Interface**

# The Beginning