## 第1章 Android基础入门

### 【思考题】

1、请思考Java虚拟机和Dalvik虚拟机的区别。

2、请思考如何使用DDMS工具打开SD卡目录。

### 【答案】

1、Java虚拟机和Dalvik虚拟机主要有两大区别，一是他们编译后的文件不同，二是它们基于的架构不同，具体如下：

1. 编译后的文件不同

Java虚拟机运行的是.class字节码文件，而Dalvik虚拟机运行的则是其专有的.dex文件，在Java程序中Java类会被翻译成一个或者多个字节码文件（.class）然后打包到JAR文件，之后Java虚拟机会从相应的class文件和JAR文件中获取相应的字节码。Android程序虽然也是使用Java语言进行编程，但是在翻译成.class文件后，还会通过工具将所有的.class文件转换成一个.dex文件，然后Dalvik虚拟机从其中读取指令和数据，最后的.odex是为了在运行过程中进一步提高性能，对.dex文件的进一步优化，能加快软件的加载速度和开启速度。

1. 基于的架构不同

Java虚拟机是基于栈的架构，大家知道，栈是一个连续的内存空间，取出和存入的速度比较慢，而Dalvik是基于寄存器的架构，寄存器是CPU上的一块缓存，寄存器的存取速度要比从内存中存取的速度快很多，这样就可以根据硬件来最大的优化设备，更适合移动设备的使用。

2、DDMS视图的右侧窗口中，可以看到Threads、Allocation Tracker、Network Statistics、File Explorer、Emulator Control、System Information选项卡，这些选项卡分别显示线程统计信息、内存分配跟踪器（每个程序占用的内存）、网络统计信息、文件资源管理器、Android系统信息。选中File Explorer选项卡，展开mnt目录该目录中的sdcard目录就是SD卡的目录。

## 第2章 Android UI开发

### 【思考题】

1、请思考Android中有几种布局，并说明每种布局的特点。

2、请思考在使用单元测试时的步骤。

### 【答案】

1、Android中的布局分为五种，分别是相对布局、线性布局、表格布局、帧布局、绝对布局。

1）相对布局

在Eclipse中开发Android程序时，默认采用的就是相对布局。相对布局通常有两种形式，一种是相对于容器而言的，一种是相对于控件而言的，为了能准确定位布局中的控件。

2）线性布局

线性布局是Android中较为常用的布局方式，它使用<LinearLayout>标签表示。线性布局主要有两种形式，一种是水平线性布局，一种是垂直线性布局。

3）表格布局

表格布局就是让控件以表格的形式来排列控件的，只要将控件或信息放在单元格中，控件就可以整齐的排列。

4）帧布局

帧布局是Android布局中最简单的一种，帧布局为每个加入其中的控件创建一个空白区域（称为一帧，每个控件占据一帧）。采用帧布局方式设计界面时，只能在屏幕左上角显示一个控件，如果添加多个控件，这些控件会按照顺序在屏幕的左上角重叠显示，且会透明显示之前控件的文本。

5）绝对布局

绝对布局需要通过指定x、y坐标来控制每一个控件的位置，放入该布局的控件需要通过android:layout\_x和android:layout\_y两个属性指定其准确的坐标值，并显示在屏幕上。

2、步骤如下：

1）添加test()测试方法

项目创建时Android Studio就已经默认创建了一个androidTest包和ApplicationTest类，双击打开ApplicationTest类，在该类中添加一个test()测试方法，对比两个参数值，具体代码如下：

1 package cn.itcast.androidui;

2 import android.app.Application;

3 import android.test.ApplicationTestCase;

4 /\*\*

5 \* <a href="http://d.android.com/tools/testing/

6 \* testing\_android.html">Testing Fundamentals</a>

7 \*/

8 public class ApplicationTest extends ApplicationTestCase<Application> {

9 public ApplicationTest() {

10 super(Application.class);

11 }

12 public void test()throws Exception{

13 final int expected =1;

14 final int reality =1;

15 //断言,expected期望的参数值与reality相同

16 assertEquals(expected, reality);

17 }

18 }

2）设置测试选项

编写好代码之后，在类class ApplicationTest…处点击右键，选择【Create 'ApplicationTest'…】选项，此时会弹出一个设置对话框，如下图1所示。

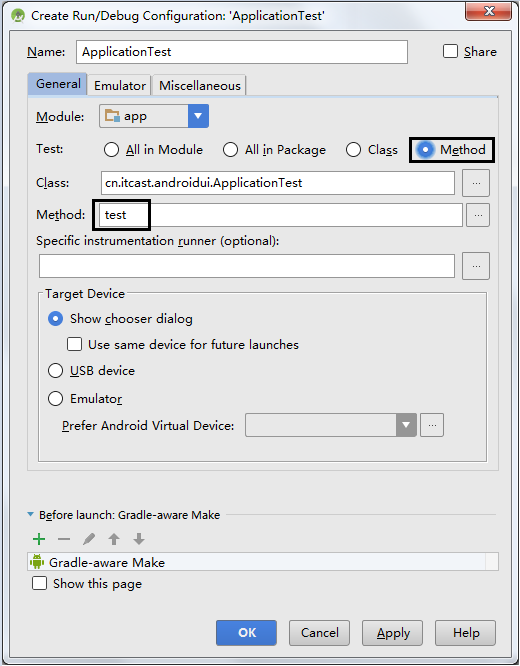


图1 设置对话框

在此窗口中，需要将测试类型修改为Method，此时会出现一栏输入框，填写ApplicationTest类中需要测试的方法名test。也可点击输入框右侧 图标选择需要测试的方法。然后点击“OK”后就可以运行程序了。

3）运行测试

启动程序后，在下方点击图标查看结果，运行结果如图2所示。

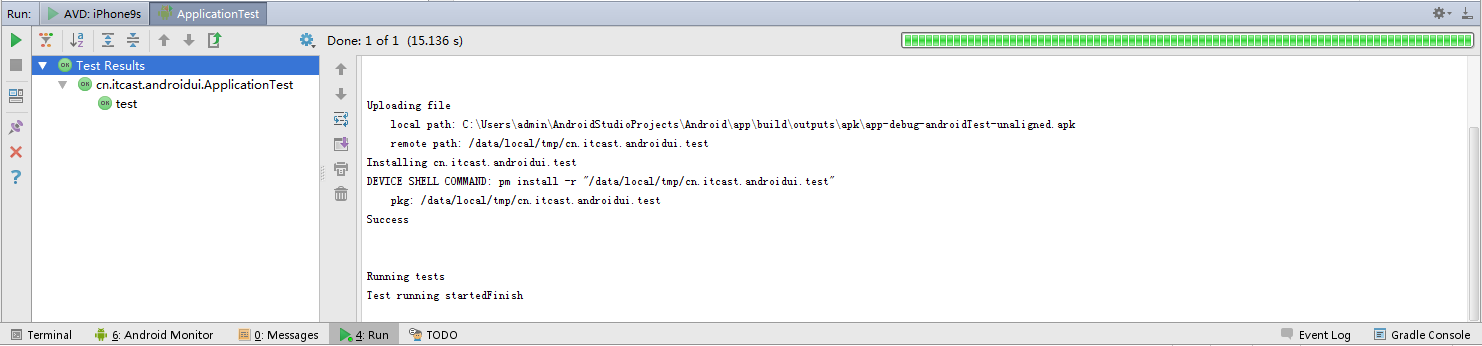


图2 测试结果

从图2可以看出，测试窗口中出现一个绿条，说明方法中两个参数相同，测试结果正常。接下来修改上述程序代码，让其显示错误信息，修改如下：

final int reality =2;

运行程序，此时会发现测试窗口显示红条，测试未通过，说明方法中两个参数不同，测试结果错误，如图3所示。

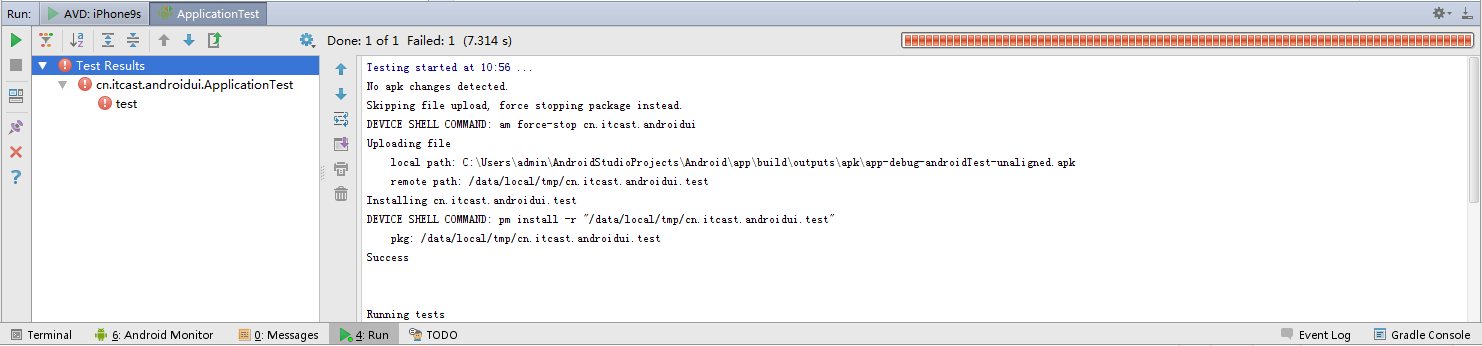


图3 测试结果

## 第3章 Activity

### 【思考题】

1、请思考什么是Activity，以及Activity的作用。

2、请思考Activity生命周期中包含有哪几种状态。

### 【答案】

1、Activity是Android应用程序的四大组件之一，它负责管理Android应用程序的用户界面，为用户提供可视化界面及操作。一个应用程序通常包含多个Activity，每个Activity负责管理一个用户界面。这些界面可以添加多个控件，每个控件负责实现不同功能。

在应用程序中，Activity就像一个界面管理员，用户在界面上的操作是通过Activity来管理的。

2、生命周期就是一个对象从创建到销毁的过程，每一个对象都有自己的生命周期。同样，Activity也具有相应的生命周期，Activity的生命周期分为五种状态，分别是启动状态、运行状态、暂停状态、停止状态和销毁状态，其中启动状态和销毁状态是过渡状态，Activity不会在这两个状态停留。接下来将针对Activity生命周期的五种状态进行详细地讲解。

1）启动状态

Activity的启动状态很短暂，一般情况下，当Activity启动之后便会进入运行状态。

2）运行状态

Activity在此状态时处于屏幕最前端，它是可见、有焦点的，可以与用户进行交互，如点击、双击、长按事件等。

值得一提的是，当Activity处于运行状态时，Android会尽可能地保持它的运行，即使出现内存不足的情况，Android也会先销毁栈底的Activity，来确保当前Activity正常运行。

3）暂停状态

在某些情况下，Activity对用户来说仍然可见，但它无法获取焦点，用户对它操作没有响应，此时它就处于暂停状态。例如，当前Activity上覆盖了一个透明或者非全屏的Acitvity时，被覆盖的Activity就处于暂停状态。

4）停止状态

当Activity完全不可见时，它就处于停止状态，但仍然保留着当前状态和成员信息。如果系统内存不足，那么这种状态下的Activity很容易被销毁。

5）销毁状态

当Activity处于销毁状态时，将被清理出内存。

## 第4章 数据存储

### 【思考题】

1、请思考Android中有几种数据存储方式以及各自特点。

2、请思考在Android中如何使用SharedPreferences类。

### 【答案】

1、Android系统中的五种数据存储方式，每种方式都有其不同的特点，下面将针对这五种方式进行简单介绍。

* 文件存储：以IO流形式把数据存入手机内存或者SD卡，可以存储大数据，如音乐、图片或者视频等。
* SharedPreferences：它本质上是一个XML文件，以Map<Object,Object>形式存入手机内存中。常用于存储较简单的参数设置，如QQ登陆账号密码的存储，窗口功能状态的存储等，使用起来简单、方便。
* SQLite数据库：SQLite是一个轻量级、跨平台的数据库。数据库中所有信息都存在单一文件内，占用内存小，并且支持基本SQL语法，是项目中经常被采用的一种数据存储方式，通常用于存储用户信息等。
* ContentProvider：又称为内容提供者，是Android四大组件之一，以数据库形式存入手机内存，可以共享自己的数据给其他应用使用，相对于其他对外共享数据的方式而言，ContentProvider统一了数据访问方式，使用起来更规范。
* 网络存储：把数据存储到服务器，不存在本地，使用的时候直接从网络获取，避免了手机端信息丢失以及其他的安全隐患。

需要注意的是，在Android中应用程序存储的数据都属于应用私有，如果要将程序中的私有数据分享给其他应用程序，可以使用文件存储、SharedPreferences以及ContentProvider，但是推荐使用ContentProvider共享数据。

2、SharedPreferences是Android平台上一个轻量级的存储类，主要用于存储一些应用程序的配置参数，例如用户名、密码、自定义参数的设置等。SharedPreferences中存储的数据是以key/value键值对的形式保存在XML文件中，该文件位于“data/data/<packagename>/shared\_prefs”文件夹中。需要注意的是，SharedPreferences中的value值只能是float、int、long、boolean、String、StringSet类型数据。

使用SharedPreferences存储数据时，需要先获取SharedPreferences对象，通过该对象获取到Editor对象，然后通过Editor对象的相关方法存储数据，具体代码如下：

//获取sp对象，参数data表示文件名，MODE\_PRIVATE表示文件操作模式

SharedPreferences sp = getSharedPreferences("data",MODE\_PRIVATE);

SharedPreferences.Editor editor = sp.edit(); // 获取编辑器

editor.putString("name", "传智播客"); // 存入String类型数据

editor.putInt("age", 8); // 存入int类型数据

editor.commit(); // 提交修改

SharedPreferences获取数据时比较简单，只需要创建SharedPreferences对象，然后使用该对象获取相应key的值即可，具体代码如下：

SharedPreferences sp = context.getSharedPreferences();

String data = sp.getString("name",""); // 获取用户名

SharedPreferences删除数据时与存储数据类似，同样需要先获取到Editor对象，然后通过该对象删除数据，最后提交，具体代码如下：

SharedPreferences sp = context.getSharedPreferences ();

Editor editor = sp.edit();

editor.remove("name"); // 删除一条数据

editor.clear(); // 删除所有数据

editor.commit(); // 提交修改

## 第5章 SQLite数据库

### 【思考题】

1、请思考Android中的SQLite数据库具有哪些特点。

2、请思考ListView控件在实际生活中有哪些应用场景。

### 【答案】

1、SQLite是一个轻量级数据库，第一个版本诞生于2000年5月。它最初是为嵌入式设计的，占用资源非常低，在内存中只需要占用几百KB的存储空间。这也是Android移动设备采用SQLite数据库的重要原因之一。

SQLite是遵守ACID关联式的数据库管理系统。这里的ACID是指数据库事务正确执行的四个基本要素，即原子性（Atomicity）、一致性（Consistency）、隔离性（Isolation）、持久性（Durability）。同时SQLite还支持SQL语言、事务处理等功能。

SQLite没有服务器进程，它通过文件保存数据，该文件是跨平台的，可以放在其他平台中使用。并且在保存数据时，支持 NULL、INTEGER、REAL（浮点数字）、TEXT(字符串文本)和BLOB(二进制对象)五种数据类型。但实际上SQLite也接受varchar(n)、char(n)、decimal(p,s) 等数据类型，只不过在运算或保存时会转换成对应的五种数据类型。因此，可以将各种类型的数据保存到任何字段中，而不用关心字段声明的数据类型。这也是SQLite数据库的最大特点。

2、在日常生活中，我们经常会使用新闻客户端、淘宝客户端等应用程序。这些应用程序，通常会有一个页面能展示多个条目信息，并且每个条目信息的布局都是一样的。为此，我们会使用Android系统中提供了一个ListView控件，该控件可以解决上述问题，而且更易维护。

## 第6章 BroadcastReceiver（广播接收者）

### 【思考题】

1、请思考Android中广播接收者的作用。

2、请思考广播有几种类型，以及不同类型的区别。

### 【答案】

1、在Android系统中，广播是一种运用在应用程序之间传递消息的机制，例如电池电量低时会发送一条提示广播。要过滤并接收广播中的消息就需要使用BroadcastReceiver（广播接收者），广播接收者是Android四大组件之一，通过广播接收者可以监听系统中的广播消息，并实现在不同组件之间的通信。

2、在Android系统中，根据广播的执行顺序不同，可将其分为有序广播和无序广播，接下来将针对这两种广播分别进行讲解。

1）无序广播

无序广播是一种完全异步执行的广播，在广播发出去之后，所有监听了这个广播事件的广播接收器几乎都会在同一时刻接收到这条广播，他们之间没有任何先后顺序可言，这种广播的效率会比较高，但同时意味着它是无法被截断的。无序广播的工作流程如图1所示。

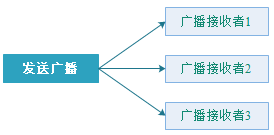


图1 无序广播

2）有序广播

有序广播则是一种同步执行的广播，在广播发出之后，同一时刻只会有一个广播接收器能够接收到这条消息，当这个广播接收器中的逻辑执行完毕后，广播才会继续传递。相比无序广播的效率较低，但此类型是有先后顺序的，并可被拦截。有序广播的工作流程如图2所示。

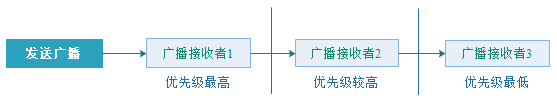


图2 有序广播

## 第7章 Service（服务）

### 【思考题】

1、请思考服务有几种开启方式，每种开启方式的特点。

2、请思考如何在Android系统中完成不同进程之间的通信。

### 【答案】

1、启动服务有两种方式，分别是通过startService()方式启动服务、bindService()方式启动服务。通过startService()方式开启的服务会长期的运行在后台，并且服务的状态与开启者的状态没有关系。通过bindService()方式开启的服务与开启者的状态有关，当调用者销毁了，服务也会被销毁。

2、在Android系统中，各个应用程序都运行在自己的进程中，进程之间一般无法直接进行通信，如果想要完成不同进程之间的通信，就需要使用远程服务通信。远程服务通信是通过AIDL(Android Interface Definition Language)实现的，它是一种接口定义语言(Interface Definition Language)，其语法格式非常简单，与Java中定义接口很相似，但是存在几点差异，具体如下：

* AIDL定义接口的源代码必须以.aidl结尾；
* AIDL接口中用到的数据类型，除了基本数据类型、String、List、Map、CharSequence之外，其他类型全部都需要导入包，即使它们在同一个包中。

## 第8章 ContentProvider（内容提供者）

### 【思考题】

1、请思考在程序中如何使用内容提供者操作数据。

2、请思考什么是内容观察者，内容观察者如何应用。

### 【答案】

1、ContentProvider（内容提供者）是Android系统四大组件之一，其功能是在不同程序之间实现数据共享。在Android系统中，应用程序之间是相互独立的，分别运行在自己的进程中。若应用程序之间需要共享数据，就需要用到ContentProvider。它不仅允许一个程序访问另一个程序中的数据，同时还可以选择只对哪一部分数据进行共享，从而保证了程序中的隐私数据不被泄露。

ContentProvider是不同应用程序之间进行数据交换的标准API，它以Uri的形式对外提供数据，允许其他应用操作本应用数据。其他应用则使用ContentResolver，并根据ContentProvider提供的Uri操作指定数据。接下来通过图例的方式来讲解ContentProvider的工作原理，如下图1所示。

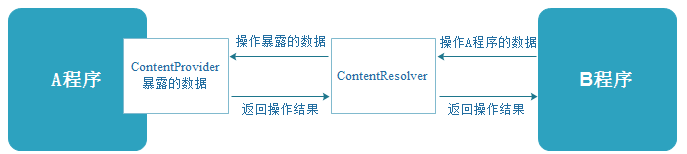


图1 ContentProvider工作原理图

从图1可以看出，A应用需要使用ContentProvider暴露数据，才能被其他应用操作。B应用必须通过ContentResolver操作A应用暴露出来的数据，而A应用会将操作结果返回给ContentResolver，然后ContentResolver再将操作结果返回给B应用。

2、内容观察者（ContentObserver）是用来观察指定Uri所代表的数据。当ContentObserver观察到指定Uri代表的数据发生变化时，就会触发ContentObserver的onChange()方法。此时在onChange()方法里使用ContentResovler可以查询到变化的数据。为了让初学者更好地理解，接下来通过一个图例的方式来讲解ContentObserver的工作原理，如图2所示。

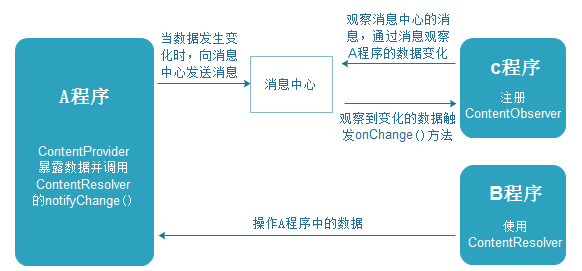


图2 ContentObserver工作原理图

从图2中可以看出，使用ContentObserver观察A应用的数据时，首先要在A应用的ContentProvider中调用ContentResolver的notifyChange()方法。调用了这个方法之后当数据发生变化时，它就会向“消息中心”发送数据变化的消息。然后C应用观察到“消息中心”有数据变化时，就会触发ContentObserver的onChange()方法。

## 第9章 网络编程

### 【思考题】

1、请思考GET与POST请求方式的使用及两种方式的区别。

2、请思考Handler机制中的四个关键对象的作用。

### 【答案】

1. 提交方式如下

1）GET方式提交数据

GET方式是以实体的方式得到由请求URL所指向的资源信息，它向服务器提交的参数跟在请求URL后面。使用GET方式访问网络URL的长度一般要小于1K。

2）POST方式提交数据

使用POST方式向服务器发出请求时需要在请求后附加实体。它向服务器提交的参数在请求后的实体中，POST方式对URL的长度是没有限制的。使用POST方式请求网络，请求参数跟在请求实体中。用户不能在浏览器中看到向服务器提交的请求参数，因此POST方式要比GET方式相对安全。

2、Handler机制主要包括四个关键对象，分别是：Message、Handler、MessageQueue、Looper。下面对这四个关键对象进行简要的介绍。

1）Message

Message是在线程之间传递的消息，它可以在内部携带少量的信息，用于在不同线程之间交换数据。Message的what字段可以用来携带一些整型数据，obj字段可以用来携带一个Object对象。

2）Handler

Handler是处理者的意思，它主要用于发送消息和处理消息。一般使用Handler对象的sendMessage()方法发送消息，发出的消息经过一系列的处理后，最终会传递到Handler对象的handlerMessage()方法中。

3）MessageQueue

MessageQueue是消息队列的意思，它主要用来存放通过Handler发送的消息。通过Handler发送的消息会存在MessageQueue中等待处理，每个线程中只会有一个MessageQueue对象。

4）Looper

Looper是每个线程中的MessageQueue的管家。调用Looper的loop()方法后，就会进入到一个无限循环中。每当发现MessageQueue中存在一条消息，就会将它取出，并传递到Handler的handlerMessage()方法中。此外，每个线程也只会有一个Looper对象。在主线程中创建Handler对象时，系统已经默认存在一个Looper对象，所以不用手动创建Looper对象，而在子线程中的Handler对象，需要调用Looper.loop()方法开启消息循环。

## 第10章 高级编程

### 【思考题】

1、请思考MediaPlayer播放音频的步骤。

2、请思考什么是Fragment，以及Fragment的作用。

### 【答案】

1、为了让初学者更好的掌握MediaPlayer的使用，接下来通过示例代码来演示MediaPlayer播放音频的完整过程，具体步骤如下：

1）创建MediaPlayer

MediaPlayer mediaPlayer = new MediaPlayer(); //创建MediaPlayer

mediaPlayer.setAudioStreamType(AudioManager.STREAM\_MUSIC);//设置声音流类型

2）设置数据源

设置数据源有三种方式，分别是设置播放应用自带的音频文件、设置播放SD卡中的音频文件、设置播放网络音频文件，具体代码如下 ：

//播放应用res/raw目录下自带的音频文件：

mediaPlayer.create(this, R.raw.xxx);

// 播放SD卡中的音频文件

mediaPlayer.setDataSource("mnt/sdcard/xxx.mp3");

//播放网络音频文件：

mediaPlayer.setDataSource("http://www.xxx.mp3");

3）播放音乐

播放本地音乐文件与播放网络文件有所不同，当准备播放本地文件时使用的是prepare();方法通知底层框架准备播放音乐，而准备播放网络音频文件使用prepareAsync()方法，具体代码如下：

mediaPlayer.prepare(); //播放本地音乐文件

mediaPlayer.start(); //执行start()开始播放音乐

播放网络文件

mediaPlayer.prepareAsync(); //播放网络音乐文件

mediaPlayer.setOnPreparedListener(new OnPreparedListener){

public void onPrepared(MediaPlayer player){

mediaPlayer.start();

}

}

需要注意的是，当播放网络中的音频文件时，需要添加访问网络的权限，具体如下：

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

4）暂停播放

暂停播放使用的是pause()方法，但是在暂停播放之前先要判断MediaPlayer对象是否存，在并且是否正在播放音乐，具体代码如下：

if(mediaPlayer!=null && mediaPlayer.isPlaying()){

mediaPlayer.pause();

}

5）重新播放

重新播放使用的是seekTo()方法，该方法是MediaPlayer中快退快进的方法，它接收时间的参数表示毫秒值，代表要把播放时间定位到哪一毫秒，这里定位到0毫秒就是重头开始播放，具体代码如下：

//播放状态下进行重播

if(mediaPlayer!=null && mediaPlayer.isPlaying()){

mediaPlayer.seekTo(0);

return;

}

//暂停状态下进行重播，要手动调用start();

if(mediaPlayer!=null){

mediaPlayer.seekTo(0);

mediaPlayer.start();

}

6）停止播放

停止播放音频使用的是stop()方法，停止播放之后还要调用MediaPlayer的release()方法将占用的资源释放并将MediaPlayer置为空，具体代码如下：

if(mediaPlayer!=null && mediaPlayer.isPlaying()){

mediaPlayer.stop();

mediaPlayer.release();

mediaPlayer = null;

}

2、Fragment（碎片）是一种可以嵌入在Activity中的UI片段，它能让程序更加合理地利用大屏幕空间，因而Fragment在平板上应用的非常广泛。Fragment与Activity十分相似，它能包含布局，同时也具有自己的生命周期。

Fragment需要包含在Activity中，一个Activity里面可以包含一个或者多个Fragment，而且一个Activity可以同时展示多个Fragment。同时，Fragment也能具有自己的布局。接下来通过一个图例的方式来讲解Fragment的用途，具体如图1所示。



图1 Fragment的用途

从图1中可以看出， 在普通手机上展示新闻列表和新闻内容各需要一个Activity，由于普通手机尺寸较小，因此展示一屏展示新闻列表，一屏展示新闻内容是合理的。如果在平板电脑上这样做就太浪费屏幕空间了，因此通常在平板上都是用一个Activity展示两个Fragment，其中一个Fragment用来展示新闻列表，另一个用来展示新闻内容。