**《Android移动应用开发》**

**实验指导书**

**2018年3月**

**目录**

[实验一 Android开发环境 3](#_Toc300061075)

[实验二 Android应用程序 6](#_Toc300061076)

[实验三 Android生命周期 10](#_Toc300061077)

[实验四 Android用户界面 13](#_Toc300061078)

[实验五 组件通信与广播 24](#_Toc300061079)

[实验六 后台服务 36](#_Toc300061080)

[实验七 数据存储与访问 48](#_Toc300061081)

[实验八 位置服务于地图应用 71](#_Toc300061082)

[实验九 Android NDK开发 77](#_Toc300061083)

# 实验一 Android开发环境

一、实验目的

1．掌握Android开发环境的安装配置方法。

2．掌握Android SDK安装和配置。

3．掌握Android模拟器的使用。

二、实验内容

搭建Android应用程序的开发环境。

三、实验仪器、设备

硬件：PC 微型计算机、1G以上内存，40G以上硬盘

软件：Windows XP，Eclipse , JDK , Android SDK

四、实验步骤

1．安装JDK和Eclipse

安装JDK，假设安装目录：C:\Program Files\Java\jdk1.6.0\_14\



安装后，配置环境变量。

安装Eclipse

选择下载：Eclipse IDE for Java Developers (92MB)，解压到相应的目录中。

启动Eclipse

双击目录中的eclipse.exe，假设选择工作目录：E:\Android\workplace，建议选择复选框，将工作目录设成默认工作目录。

2．安装Android SDK

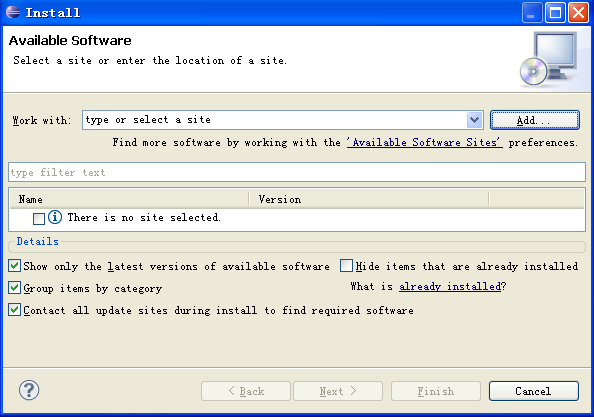
Android开发站点：<http://code.google.com/android>，点击“下载SDK”

选择Windows版本的Android SDK，文件名：android-sdk-windows-1.5\_r3.zip，Android SDK只需解压，无需安装。解压目录：E:\Android\android-sdk-windows-1.5\_r3（假定）。

3、安装ADT插件

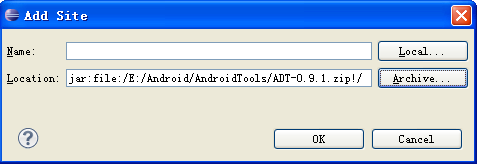
安装ADT有两种方法。手动下载ADT插件，下载网址：<http://dl-ssl.google.com/android/eclipse/>或Android SDK帮助文档中的下载页面，下载页面位置：<Android SDK>/docs/sdk/adt\_download.html，选择下载：ADT\_0.9.1.zip。

安装插件：启动Eclipse.选择Help → Install New Software,打开Eclipse插件的安装界面。点击“Add”。

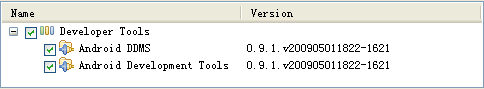


点击Archive

选择ADT保存位置



选择ADT插件安装选项：复选Android DDMS和Android Development Tools



认可ADT插件的许可证

提示重启Eclipse，即完成。

自动下载ADT插件

自动下载ADT插件的方法和手动方法安装相似，不同之处在于：直接输入插件压缩包的下载地址<http://dl-ssl.google.com/android/eclipse/>。

设置Android SDK的保存路径

选择：Windows→Preferences 打开Eclipse的配置界面，输入Android SDK 的保存路径，点击“Apply”。

五、实验报告要求

1．严格按照实验报告格式完成实验报告并上交服务器。

2．按照步骤对实验过程进行截图和分析。

六、实验注意事项

注意将各实验步骤进行截图保存，Appname应包含学号，并在截图中显示出来。

七、思考题

如何测试你的Android 应用程序开发环境已经搭建成功？

# 实验二 Android应用程序

一、实验目的

1．掌握使用Eclipse开发Android应用程序的步骤。

2．掌握AVD的创建和使用方法。

3．了解Android应用程序结构。

二、实验内容

建立一个HelloAndroid第一个Android程序。

三、实验仪器、设备

硬件：PC 微型计算机、1G以上内存，40G以上硬盘

软件：Windows XP，Eclipse , JDK , Android SDK

四、实验步骤

1、建立HelloAndroid的步骤

1）创建Android工程

打开Android工程向导：File→New→Project…︱Android→Android Project 或 File →New→Other …︱Android→Android Project。

填写工程名称：工程名称必须唯一，不能与已有的工程重名，在Project name中填入HelloAndroid。

选择创建方式

选择编译目标

填写相关信息

应用程序名称，即Android程序在手机中显示的名称，显示在手机的顶部，等信息。

进入Eclipse。

2）Android虚拟设备（AVD）

1.5版本的Android SDK中没有附带任何配置好的AVD，需建立一个AVD。

3）启动选项

Run→Run Configuration或Run→Debug Configuration配置模拟器的启动选项，可以选择不同的AVD、配置网络速度、网络延迟等属性。

4）启动Android程序

Run→Run ∣Android Application或Run→Debug ∣Android Application，在程序调试完毕后，可直接再次运行Android程序，以便节约启动模拟器的时间。

5)代码实现如下：

HelloAndroid.java文件代码：

1. package edu.scse.HelloAndroid;
2. import android.app.Activity;
3. import android.os.Bundle;
4. public class HelloAndroid extends Activity {
5. /\*\* Called when the activity is first created. \*/
6. @Override
7. public void onCreate(Bundle savedInstanceState) {
8. super.onCreate(savedInstanceState);
9. setContentView(R.layout.main);
10. }
11. }

R.java文件如下：

1. package edu.scse.HelloAndroid;
2. public final class R {
3. public static final class attr {
4. }
5. public static final class drawable {
6. public static final int icon=0x020000;
7. }
8. public static final class layout {
9. public static final int main=0x030000;
10. }
11. public static final class string {
12. public static final int app\_name=0x040001;
13. public static final int hello=0x040000;
14. }
15. }

main.xml文件的代码：

1. <?xml version="1.0" encoding="utf-8"?>
2. <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3. android:orientation="vertical"
4. android:layout\_width="fill\_parent"
5. android:layout\_height="fill\_parent"
6. >
7. <TextView
8. android:layout\_width="fill\_parent"
9. android:layout\_height="wrap\_content"
10. android:text="@string/hello"
11. />
12. </LinearLayout>

Strings.xml文件的代码：

1. <?xml version="1.0" encoding="utf-8"?>
2. <resources>
3. <string name="hello">Hello World, HelloAndroid!</string>
4. <string name="app\_name">HelloAndroid</string>
5. </resources>

AndroidManifest.xml文件的代码：

1. <?xml version="1.0" encoding="utf-8"?>
2. <manifest xmlns:android="http://schemas.android.com/apk/res/android"
3. package="edu.scse.HelloAndroid"
4. android:versionCode="1"
5. android:versionName="1.0">
6. <application android:icon="@drawable/icon"
7. android:label="@string/app\_name">
8. <activity android:name=".HelloAndroid"
9. android:label="@string/app\_name">
10. <intent-filter>
11. <action android:name="android.intent.action.MAIN" />
12. <category android:name="android.intent.category.LAUNCHER" />
13. </intent-filter>
14. </activity>
15. </application>
16. <uses-sdk android:minSdkVersion="3" />
17. </manifest>

五、实验报告要求

1、上机输入和调试自己所编的程序。一人一组，独立上机调试，上机时出现的问题，最好独立解决。

2、严格按照实验报告格式完成实验报告并上交服务器。

六、实验注意事项

注意将各实验步骤进行截图保存，Appname应包含学号，并在截图中显示出来。

七、思考题

1. 如何编写一个Android的应用程序，R.java和AndroidMainifest.xml文件有什么用途？

# 实验三 Android生命周期

一、实验目的

1. 巩固使用Eclipse开发Android应用程序方法和步骤。

2．加深了解和体验Activity的生命周期中各状态的变化关系。

3．掌握Android应用程序的调试方法和工具。

二、实验内容

1.建立在Android生命周期各阶段添加日志点的Android工程，工程名称：ActivityLifeCycle，包名称：edu.scse.ActivityLifeCycle，Activity名称：ActivityLifeCycle。

三、实验仪器、设备

硬件：PC 微型计算机、1G以上内存，40G以上硬盘

软件：Windows XP，Eclipse , JDK , Android SDK

四、实验步骤

1. 建立工程请参考实验二。

具体代码如下：

ActivityLifeCycle.java文件的代码：

1. package edu.scse.ActivityLifeCycle;
3. import android.app.Activity;
4. import android.os.Bundle;
5. import android.util.Log;
7. public class ActivityLifeCycle extends Activity {
8. private static String TAG = "LIFTCYCLE";
9. @Override //完全生命周期开始时被调用，初始化Activity
10. public void onCreate(Bundle savedInstanceState) {
11. super.onCreate(savedInstanceState);
12. setContentView(R.layout.main);
13. Log.i(TAG, "(1) onCreate()");
14. }
16. @Override //可视生命周期开始时被调用，对用户界面进行必要的更改
17. public void onStart() {
18. super.onStart();
19. Log.i(TAG, "(2) onStart()");
20. }
22. @Override //在onStart()后被调用，用于恢复onSaveInstanceState()保存的用户界面信息
23. public void onRestoreInstanceState(Bundle savedInstanceState) {
24. super.onRestoreInstanceState(savedInstanceState);
25. Log.i(TAG, "(3) onRestoreInstanceState()");
26. }
28. @Override //在活动生命周期开始时被调用，恢复被onPause()停止的用于界面更新的资源
29. public void onResume() {
30. super.onResume();
31. Log.i(TAG, "(4) onResume()");
32. }

33.

1. @Override // 在onResume()后被调用，保存界面信息
2. public void onSaveInstanceState(Bundle savedInstanceState) {
3. super.onSaveInstanceState(savedInstanceState);
4. Log.i(TAG, "(5) onSaveInstanceState()");
5. }
7. @Override //在重新进入可视生命周期前被调用，载入界面所需要的更改信息
8. public void onRestart() {
9. super.onRestart();
10. Log.i(TAG, "(6) onRestart()");
11. }
13. @Override //在活动生命周期结束时被调用，用来保存持久的数据或释放占用的资源
14. public void onPause() {
15. super.onPause();
16. Log.i(TAG, “(7) onPause()”);、
17. }

51.

1. @Override //在可视生命周期结束时被调用，一般用来保存持久的数据或释放占用的资源
2. public void onStop() {
3. super.onStop();
4. Log.i(TAG, "(8) onStop()");
5. }
7. @Override //在完全生命周期结束时被调用，释放资源，包括线程、数据连接等
8. public void onDestroy() {
9. super.onDestroy();
10. Log.i(TAG, "(9) onDestroy()");
11. }
12. }

五、实验报告要求

1、上机输入和调试自己所编的程序。一人一组，独立上机调试，上机时出现的问题，最好独立解决。

2、严格按照实验报告格式完成实验报告并上交服务器

六、实验注意事项

注意将各实验步骤进行截图保存，Appname应包含学号，并在截图中显示出来。

七、思考题

1. LogCat工具如何使用？

2．DevTools工具如何使用？

# 实验四 Android用户界面

一、实验目的

1. 加深对用户界面的控件的理解和使用。

2．掌握各种界面布局的用法。

3．掌握选项菜单、子菜单和快捷菜单的使用方法。

二、实验内容

1. 建立一个“ButtonDemo”的程序，包含Button和ImageButton两个按钮，上方是“Button按钮”，下方是一个ImageButton控件？

2. 建立一个“CheckboxRadiobuttonDemo”程序，包含五个控件，从上至下分别是TextView01、CheckBox01、 CheckBox02、RadioButton01、RadioButton02，当选择RadioButton01， RadioButton02则无法选择？

3．建立一个“TabDemo”程序，包含三个XML文件，分别为tab1.xml、tab2.xml和tab3.xml，这3个文件分别使用线性布局、相对布局和绝对布局示例中的main.xml的代码，并将布局的ID分别定义为layout01、layout02和layout03？

4．建立一个“MyContextMenu”程序，来完成Android菜单的功能。

三、实验仪器、设备

硬件：PC 微型计算机、1G以上内存，40G以上硬盘

软件：Windows XP，Eclipse , JDK , Android SDK

四、实验步骤

1. “ButtonDemo”源代码如下：

ButtonDemo.java文件源如下：

package edu.scse.ButtonDemo;

public class ButtonDemo extends Activity {

/\*\* Called when the activity is first created. \*/

@Override

public void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.main);

Button button = (Button)findViewById(R.id.Button01);

ImageButton imageButton = (ImageButton)findViewById(R.id.ImageButton01);

button.setText("Button按钮");

imageButton.setImageResource(R.drawable.download);

final TextView textView = (TextView)findViewById(R.id.TextView01);

//（1）按钮注册到各自的监听器

button.setOnClickListener(new View.OnClickListener() {

public void onClick(View view) {

textView.setText("Button按钮");

}

});

imageButton.setOnClickListener(new View.OnClickListener() {

public void onClick(View view) {

textView.setText("ImageButton按钮");

}

});

（2）按钮注册到同一个监听器

Button.OnClickListener buttonListener = new Button.OnClickListener(){

@Override

public void onClick(View v) {

switch(v.getId()){

case R.id.Button01:

textView.setText("Button按钮");

return;

case R.id.ImageButton01:

textView.setText("ImageButton按钮");

return;

}

}};

button.setOnClickListener(buttonListener);

imageButton.setOnClickListener(buttonListener);

}

}

Main.xml源代码如下：

<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:id="@+id/TextView01"

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content"

android:text="@string/hello"/>

<Button android:id="@+id/Button01"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Button01" >

</Button>

<ImageButton android:id="@+id/ImageButton01"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content">

</ImageButton>

</LinearLayout>

String.xml源代码如下：

<resources>

<string name="hello">Hello World, ButtonDemo!</string>

<string name="app\_name">ButtonDemo</string>

</resources>

AndroidManifest.xml文件源代码如下：

<manifest xmlns:android="http://schemas.android.com/apk/res/android"

package="edu.scse.ButtonDemo"

android:versionCode="1"

android:versionName="1.0">

<application android:icon="@drawable/icon" android:label="@string/app\_name">

<activity android:name=".ButtonDemo"

android:label="@string/app\_name">

<intent-filter>

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

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

</intent-filter>

</activity>

</application>

<uses-sdk android:minSdkVersion="3" />

</manifest>

2．“CheckboxRadiobuttonDemo”源代码如下：

CheckboxRadiobuttonDemo.java文件的源代码如下：

package edu.hrbu.CheckboxRadiobuttonDemo;

public class CheckboxRadiobuttonDemo extends Activity {

/\*\* Called when the activity is first created. \*/

@Override

public void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.main);

final TextView textView = (TextView)findViewById(R.id.TextView01);

//(1)CheckBox

final CheckBox checkBox1= (CheckBox)findViewById(R.id.CheckBox01);

final CheckBox checkBox2= (CheckBox)findViewById(R.id.CheckBox02);

CheckBox.OnClickListener checkboxListener = new CheckBox.OnClickListener(){

@Override

public void onClick(View v) {

switch(v.getId()){

case R.id.CheckBox01:

textView.setText("CheckBox01, isChecked:"+String.valueOf(checkBox1.isChecked()));

return;

case R.id.CheckBox02:

textView.setText("CheckBox02, isChecked:"+String.valueOf(checkBox2.isChecked()));

return;

}

}};

checkBox1.setOnClickListener(checkboxListener);

checkBox2.setOnClickListener(checkboxListener);

//(2)RadioButton

final RadioButton radioButton1 = (RadioButton)findViewById(R.id.RadioButton01);

final RadioButton radioButton2 = (RadioButton)findViewById(R.id.RadioButton02);

RadioButton.OnClickListener radioButtonListener = new RadioButton.OnClickListener(){

@Override

public void onClick(View v) {

switch(v.getId()){

case R.id.RadioButton01:

textView.setText("RadioButton1 is selected");

return;

case R.id.RadioButton02:

textView.setText("RadioButton2 is selected");;

return;

}

}};

radioButton1.setOnClickListener(radioButtonListener);

radioButton2.setOnClickListener(radioButtonListener);

}

}

Main.xml文件的源代码如下：

<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:id="@+id/TextView01"

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content"

android:text="@string/hello"/>

<CheckBox android:id="@+id/CheckBox01"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="CheckBox01" >

</CheckBox>

<CheckBox android:id="@+id/CheckBox02"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="CheckBox02" >

</CheckBox>

<RadioGroup android:id="@+id/RadioGroup01"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content">

<RadioButton android:id="@+id/RadioButton01"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="RadioButton01" >

</RadioButton>

<RadioButton android:id="@+id/RadioButton02"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="RadioButton02" >

</RadioButton>

</RadioGroup>

</LinearLayout>

String.xml文件的源代码如下：

<resources>

<string name="hello">Hello World, CheckboxRadiobuttonDemo!</string>

<string name="app\_name">CheckboxRadiobuttonDemo</string>

</resources>

Androidmanifest.xml文件的源代码如下：

<manifest xmlns:android="http://schemas.android.com/apk/res/android"

package="edu.hrbu.CheckboxRadiobuttonDemo"

android:versionCode="1"

android:versionName="1.0">

<application android:icon="@drawable/icon" android:label="@string/app\_name">

<activity android:name=".CheckboxRadiobuttonDemo"

android:label="@string/app\_name">

<intent-filter>

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

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

</intent-filter>

</activity>

</application>

<uses-sdk android:minSdkVersion="3" />

</manifest>

3．“TabDemo”源代码如下：

TabDemo.java文件源代码如下:

package edu.scse.TabDemo;

public class TabDemo extends TabActivity {

/\*\* Called when the activity is first created. \*/

@Override

public void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

TabHost tabHost = getTabHost();

LayoutInflater.from(this).inflate(R.layout.tab1, tabHost.getTabContentView(),true);

LayoutInflater.from(this).inflate(R.layout.tab2, tabHost.getTabContentView(),true);

LayoutInflater.from(this).inflate(R.layout.tab3, tabHost.getTabContentView(),true);

tabHost.addTab(tabHost.newTabSpec("TAB1").

setIndicator("线性布局").setContent(R.id.layout01));

tabHost.addTab(tabHost.newTabSpec("TAB2").

setIndicator("绝对布局").setContent(R.id.layout02));

tabHost.addTab(tabHost.newTabSpec("TAB3").

setIndicator("相对布局").setContent(R.id.layout03));

}

}

Tab1.xml文件的源代码如下：

<LinearLayout android:id = "@+id/layout01"

xmlns:android="http://schemas.android.com/apk/res/android"

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content"

android:orientation="vertical">

<TextView android:id="@+id/label"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="用户名：" >

</TextView>

<EditText android:id="@+id/entry"

android:layout\_height="wrap\_content"

android:layout\_width="fill\_parent">

</EditText>

<Button android:id="@+id/ok"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="确认：">

</Button>

<Button android:id="@+id/cancel"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="取消：" >

</Button>

</LinearLayout>

Tab2源代码如下：

<AbsoluteLayout android:id="@+id/layout02"

android:layout\_width="fill\_parent"

android:layout\_height="fill\_parent" xmlns:android="http://schemas.android.com/apk/res/android">

  <TextView android:id="@+id/label"

android:layout\_x="40dip"

android:layout\_y="40dip"

android:layout\_height="wrap\_content"

android:layout\_width="wrap\_content"

android:text="用户名：" />

  <EditText android:id="@+id/entry"

android:layout\_x="40dip"

android:layout\_y="60dip"

android:layout\_height="wrap\_content"

android:layout\_width="150dip" />

  <Button android:id="@+id/ok"

android:layout\_width="70dip"

android:layout\_height="wrap\_content"

android:layout\_x="40dip"

android:layout\_y="120dip"

android:text="确认" />

  <Button android:id="@+id/cancel"

android:layout\_width="70dip"

android:layout\_height="wrap\_content"

android:layout\_x="120dip"

android:layout\_y="120dip"

android:text="取消" />

  </AbsoluteLayout>

Tab3.xml源代码如下：

<RelativeLayout android:id="@+id/layout03"

android:layout\_width="fill\_parent"

android:layout\_height="fill\_parent"

xmlns:android="http://schemas.android.com/apk/res/android">

<TextView android:id="@+id/label"

android:layout\_height="wrap\_content"

android:layout\_width="fill\_parent"

android:text="**用户名：**">

</TextView>

<EditText android:id="@+id/entry"

android:layout\_height="wrap\_content"

android:layout\_width="fill\_parent"

android:layout\_below="@id/label">

</EditText>

<Button android:id="@+id/cancel"

android:layout\_height="wrap\_content"

android:layout\_width="wrap\_content"

android:layout\_alignParentRight="true"

android:layout\_marginLeft="10dip"

android:layout\_below="@id/entry"

android:text="取消：" >

</Button>

<Button android:id="@+id/ok"

android:layout\_height="wrap\_content"

android:layout\_width="wrap\_content"

android:layout\_toLeftOf="@id/cancel"

android:layout\_alignTop="@id/cancel"

android:text="确认：">

</Button>

</RelativeLayout>

String.xml文件的源代码如下：

<resources>

<string name="hello">Hello World, TabDemo!</string>

<string name="app\_name">TabDemo</string>

</resources>

Androidmanifest.xml文件的源代码如下：

<manifest xmlns:android="http://schemas.android.com/apk/res/android"

package="edu.scse.TabDemo"

android:versionCode="1"

android:versionName="1.0">

<application android:icon="@drawable/icon" android:label="@string/app\_name">

<activity android:name=".TabDemo"

android:label="@string/app\_name">

<intent-filter>

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

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

</intent-filter>

</activity>

</application>

<uses-sdk android:minSdkVersion="3" />

</manifest>

4．“MyContextMenu”源代码如下

MyContextMenu.java文件的源代码如下：

package edu.scse.MyContextMenu;

public class MyContextMenu extends Activity {

/\*\* Called when the activity is first created. \*/

final static int CONTEXT\_MENU\_1 = Menu.FIRST;

final static int CONTEXT\_MENU\_2 = Menu.FIRST+1;

final static int CONTEXT\_MENU\_3 = Menu.FIRST+2;

TextView LabelView = null;

@Override

public void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.main);

LabelView = (TextView)findViewById(R.id.label);

registerForContextMenu(LabelView);

}

@Override

public void onCreateContextMenu(ContextMenu menu,

View v, ContextMenuInfo menuInfo){

menu.setHeaderTitle("快捷菜单标题");

menu.add(0, CONTEXT\_MENU\_1, 0,"菜单子项1");

menu.add(0, CONTEXT\_MENU\_2, 1,"菜单子项2");

menu.add(0, CONTEXT\_MENU\_3, 2,"菜单子项3");

}

@Override

public boolean onContextItemSelected(MenuItem item){

switch(item.getItemId()){

case CONTEXT\_MENU\_1:

LabelView.setText("菜单子项1");

return true;

case CONTEXT\_MENU\_2:

LabelView.setText("菜单子项2");

return true;

case CONTEXT\_MENU\_3:

LabelView.setText("菜单子项3");

return true;

}

return false;

}

}

Main.xml文件的源代码如下：

<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:id="@+id/label"

android:layout\_width="fill\_parent"

android:layout\_height="fill\_parent"

android:text="@string/hello"

/>

</LinearLayout>

String.xml文件的源代码如下：

<resources>

<string name="hello">Hello World, MyContextMenu!</string>

<string name="app\_name">MyContextMenu</string>

</resources>

Androidmanifest.xml文件的源代码如下：

<manifest xmlns:android="http://schemas.android.com/apk/res/android"

package="edu.scse.MyContextMenu"

android:versionCode="1"

android:versionName="1.0">

<application android:icon="@drawable/icon" android:label="@string/app\_name">

<activity android:name=".MyContextMenu"

android:label="@string/app\_name">

<intent-filter>

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

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

</intent-filter>

</activity>

</application>

<uses-sdk android:minSdkVersion="3" />

</manifest>

五、实验报告要求

1、上机输入和调试自己所编的程序。一人一组，独立上机调试，上机时出现的问题，最好独立解决。

2、严格按照实验报告格式完成实验报告并上交服务器

六、实验注意事项

注意将各实验步骤进行截图保存，Appname应包含学号，并在截图中显示出来。

七、思考题

1.建立一个程序“SpinnerDemo”包含3个子项Spinner控件？

2．建立一个“ListViewDemo”程序，包含四个控件，从上至下分别为TextView01、ListView01、 ListView02和 ListView03？

3．Android界面框架是如何实现按键事件和触摸事件的？

# 实验五 组件通信与广播

一、实验目的

1.使学生学会使用Intent启动Activity的方法。

2．使学生掌握获取Activity返回值的方法。

3. 使学生掌握发送和接收广播消息的方法。

二、实验内容

1、使用“IntentDemo”程序来完成如何使用Intent启动新的Activity？

2、使用“ActivityCommunication”说明了如何以Sub-Activity方式启动子Activity，以及使用Intent进行组件间通信？

3、使用“IntentResolutionDemo”说明了如何在AndroidManifest.xml文件中注册Intent过滤器，以及如何设置<intent-filter>节点属性来捕获指定的Intent？

4、使用“BroadcastReceiverDemo”说明了如何在应用程序中注册BroadcastReceiver，并接收指定类型的广播消息？

三、实验仪器、设备

硬件：PC 微型计算机、1G以上内存，40G以上硬盘

软件：Windows XP，Eclipse , JDK , Android SDK

四、实验步骤

1.“IntentDemo”程序源代码：

IntentDemo.java文件的源代码:

package edu.scse.IntentDemo;

public class IntentDemo extends Activity {

/\*\* Called when the activity is first created. \*/

@Override

public void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.main);

Button button = (Button)findViewById(R.id.btn);

button.setOnClickListener(new OnClickListener(){

public void onClick(View view){

Intent intent = new Intent(IntentDemo.this, ActivityToStart.class);

startActivity(intent);

}

});

}

}

ActivityToStart.java文件的源代码：

package edu.scse.IntentDemo;

public class ActivityToStart extends Activity {

/\*\* Called when the activity is first created. \*/

@Override

public void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activitytostart);

}

}

Main.xml文件的源代码：

<LinearLayout xmlns:android="**http://schemas.android.com/apk/res/android**" android:orientation="**vertical**"

android:layout\_width="**fill\_parent**"

android:layout\_height="**fill\_parent**">

  <Button android:id="**@+id/btn**"

android:layout\_width="**wrap\_content**" android:layout\_height="**wrap\_content**"

android:text="**启动Activity**" />

  </LinearLayout>

activitytostart.xml 文件的源代码如下：

<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="**ActivityToStart**" />

</LinearLayout>

String.xml文件的源代码如下：

<resources>

<string name="hello">Hello World, IntentDemo!</string>

<string name="app\_name">IntentDemo</string>

</resources>

2．“ActivityCommunication”源代码：

ActivityCommunication.java文件源代码如下：

package edu.scse.ActivityCommunication;

public class ActivityCommunication extends Activity {

private static final int SUBACTIVITY1 = 1;

private static final int SUBACTIVITY2 = 2;

TextView textView;

/\*\* Called when the activity is first created. \*/

@Override

public void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.main);

textView = (TextView)findViewById(R.id.textShow);

final Button btn1 = (Button)findViewById(R.id.btn1);

final Button btn2 = (Button)findViewById(R.id.btn2);

btn1.setOnClickListener(new OnClickListener(){

public void onClick(View view){

Intent intent = new Intent(ActivityCommunication.this, SubActivity1.class);

startActivityForResult(intent, SUBACTIVITY1);

}

});

btn2.setOnClickListener(new OnClickListener(){

public void onClick(View view){

Intent intent = new Intent(ActivityCommunication.this, SubActivity2.class);

startActivityForResult(intent, SUBACTIVITY2);

}

});

}

@Override

protected void onActivityResult(int requestCode, int resultCode, Intent data) {

super.onActivityResult(requestCode, resultCode, data);

switch(requestCode){

case SUBACTIVITY1:

if (resultCode == RESULT\_OK){

Uri uriData = data.getData();

textView.setText(uriData.toString());

}

break;

case SUBACTIVITY2:

break;

}

}

}

SubActivity1.java文件源代码如下：

package edu.scse.ActivityCommunication;

public class SubActivity1 extends Activity {

@Override

public void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.subactivity1);

final EditText editText = (EditText)findViewById(R.id.edit);

Button btnOK = (Button)findViewById(R.id.btn\_ok);

Button btnCancel = (Button)findViewById(R.id.btn\_cancel);

btnOK.setOnClickListener(new OnClickListener(){

public void onClick(View view){

String uriString = editText.getText().toString();

Uri data = Uri.parse(uriString);

Intent result = new Intent(null, data);

setResult(RESULT\_OK, result);

finish();

}

});

btnCancel.setOnClickListener(new OnClickListener(){

public void onClick(View view){

setResult(RESULT\_CANCELED, null);

finish();

}

});

}

}

SubActivity2.java

package edu.scse.ActivityCommunication;

public class SubActivity2 extends Activity {

/\*\* Called when the activity is first created. \*/

@Override

public void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.subactivity2);

Button btnReturn = (Button)findViewById(R.id.btn\_return);

btnReturn.setOnClickListener(new OnClickListener(){

public void onClick(View view){

setResult(RESULT\_CANCELED, null);

finish();

}

});

}

}

Main.xml

<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:id="**@+id/textShow**"

android:layout\_width="**fill\_parent**"

android:layout\_height="**wrap\_content**"

android:text="" />

  <Button android:id="**@+id/btn1**"

android:layout\_width="**wrap\_content**" android:layout\_height="**wrap\_content**"

android:text="**启动Activity1**" />

  <Button android:id="**@+id/btn2**"

android:layout\_width="**wrap\_content**"

android:layout\_height="**wrap\_content**"

android:text="**启动Activity2**" />

  </LinearLayout>

SubActivity1.xml源代码：

<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:id="@+id/textShow"

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content"

android:text="SubActivity 1"

/>

<EditText

android:id="@+id/edit"

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content"

android:text=""

/>

<Button

android:id="@+id/btn\_ok"

android:layout\_width="100dip"

android:layout\_height="wrap\_content"

android:text="**接受**"

/>

<Button

android:id="@+id/btn\_cancel"

android:layout\_width="100dip"

android:layout\_height="wrap\_content"

android:text="**撤销**"

/>

/>

</LinearLayout>

SubActivity2.xml源代码：

<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:id="@+id/textShow"

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content"

android:text="SubActivity 2"

/>

<Button

android:id="@+id/btn\_return"

android:layout\_width="100dip"

android:layout\_height="wrap\_content"

android:text="关闭"

/>

</LinearLayout>

Androidmanifest.xml文件代码如下：

package edu.scse.ActivityCommunication;

public class SubActivity2 extends Activity {

/\*\* Called when the activity is first created. \*/

@Override

public void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.subactivity2);

Button btnReturn = (Button)findViewById(R.id.btn\_return);

btnReturn.setOnClickListener(new OnClickListener(){

public void onClick(View view){

setResult(RESULT\_CANCELED, null);

finish();

}

});

}

}

3．“IntentResolutionDemo”源代码：

IntentResolutionDemo.java源代码如下：

package edu.scse.IntentResolutionDemo;

public class IntentResolutionDemo extends Activity {

/\*\* Called when the activity is first created. \*/

@Override

public void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.main);

Button button = (Button)findViewById(R.id.btn);

button.setOnClickListener(new OnClickListener(){

public void onClick(View view){

Intent intent = new Intent(Intent.ACTION\_VIEW, Uri.parse("schemodemo://edu.scse/path"));

startActivity(intent);

}

});

}

}

ActivityToStart.java源代码如下：

package edu.scse.IntentResolutionDemo;

public class ActivityToStart extends Activity {

/\*\* Called when the activity is first created. \*/

@Override

public void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activitytostart);

}

}

main.xml源代码：

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"

android:orientation="vertical"

android:layout\_width="fill\_parent"

android:layout\_height="fill\_parent"

>

<TextView

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content"

android:text="@string/hello"

/>

<Button

android:id="@+id/btn"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="启动Activity"

/>

</LinearLayout>

activitytostart.xml源代码：

<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="Hello, Activity"

/>

</LinearLayout>

AndroidManifest.xml源代码如下：

<manifest xmlns:android="http://schemas.android.com/apk/res/android"

package="edu.scse.IntentResolutionDemo"

android:versionCode="1"

android:versionName="1.0">

<application android:icon="@drawable/icon" android:label="@string/app\_name">

<activity android:name=".IntentResolutionDemo"

android:label="@string/app\_name">

<intent-filter>

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

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

</intent-filter>

</activity>

<activity android:name=".ActivityToStart"

android:label="@string/app\_name">

<intent-filter>

<action android:name="android.intent.action.VIEW" />

<category android:name="android.intent.category.DEFAULT" />

<data android:scheme="schemodemo" android:host="edu.scse" />

</intent-filter>

</activity>

</application>

<uses-sdk android:minSdkVersion="3" />

</manifest>

4．“BroadcastReceiverDemo”源代码：

BroadcastReceiverDemo.java代码如下：

package edu.scse.BroadcastReceiverDemo;

public class BroadcastReceiverDemo extends Activity {

private EditText entryText ;

private Button button;

@Override

public void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.main);

entryText = (EditText)findViewById(R.id.entry);

button = (Button)findViewById(R.id.btn);

button.setOnClickListener(new OnClickListener(){

public void onClick(View view){

Intent intent = new Intent("edu.scse.BroadcastReceiverDemo");

intent.putExtra("message", entryText.getText().toString());

sendBroadcast(intent);

}

});

}

}

MyBroadcastReceiver.java代码如下：

package edu.scse.BroadcastReceiverDemo;

public class MyBroadcastReceiver extends BroadcastReceiver {

@Override

public void onReceive(Context context, Intent intent) {

String msg = intent.getStringExtra("message");

Toast.makeText(context, msg, Toast.LENGTH\_SHORT).show();

}

}

Main.xml源代码：

<manifest xmlns:android="http://schemas.android.com/apk/res/android"

package="edu.scse.BroadcastReceiverDemo"

android:versionCode="1"

android:versionName="1.0">

<application android:icon="@drawable/icon" android:label="@string/app\_name">

<activity android:name=".BroadcastReceiverDemo"

android:label="@string/app\_name">

<intent-filter>

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

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

</intent-filter>

</activity>

<receiver android:name=".MyBroadcastReceiver">

<intent-filter>

<action android:name="edu.scse.BroadcastReceiverDemo" />

</intent-filter>

</receiver>

</application>

<uses-sdk android:minSdkVersion="3" />

</manifest>

AndroidManifest.xml源代码如下：

<manifest xmlns:android="http://schemas.android.com/apk/res/android"

package="edu.scse.BroadcastReceiverDemo"

android:versionCode="1"

android:versionName="1.0">

<application android:icon="@drawable/icon" android:label="@string/app\_name">

<activity android:name=".BroadcastReceiverDemo"

android:label="@string/app\_name">

<intent-filter>

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

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

</intent-filter>

</activity>

<receiver android:name=".MyBroadcastReceiver">

<intent-filter>

<action android:name="edu.scse.BroadcastReceiverDemo" />

</intent-filter>

</receiver>

</application>

<uses-sdk android:minSdkVersion="3" />

</manifest>

五、实验报告要求

1、上机输入和调试自己所编的程序。一人一组，独立上机调试，上机时出现的问题，最好独立解决。

2、严格按照实验报告格式完成实验报告并上交服务器

六、实验注意事项

注意将各实验步骤进行截图保存，Appname应包含学号，并在截图中显示出来。

七、思考题

1. 使用“WebViewIntentDemo”说明如何隐式启动Activity？

2．简述Intent过滤器的原理与匹配机制？

3．简述如何获取Activity返回值？

# 实验六 后台服务

一、实验目的

1．使同学掌握服务的隐式启动和显式启动方法。

1. 使同学学会使用线程的启动、挂起和停止方法。
2. 使同学学会跨线程的界面更新。

二、实验内容

1.使用“SimpleRandomServiceDemo”程序显式启动服务在应用程序中建立Service？

2．使用“ThreadRandomServiceDemo”程序使用线程持续产生随机数？

3．使用“SimpleMathServiceDemo”程序使用绑定方式使用Service？

4．使用“RemoteMathServiceDemo”程序，说明如何创建跨进程？

三、实验仪器、设备

硬件：PC 微型计算机、1G以上内存，40G以上硬盘

软件：Windows XP，Eclipse , JDK , Android SDK

四、实验步骤

1. SimpleRandomServiceDemo程序

SimpleRandomServiceDemo.java源代码如下：

package edu.scse.SimpleRandomServiceDemo;

public class SimpleRandomServiceDemo extends Activity {

/\*\* Called when the activity is first created. \*/

@Override

public void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.main);

Button startButton = (Button)findViewById(R.id.start);

Button stopButton = (Button)findViewById(R.id.stop);

final Intent serviceIntent = new Intent(this, RandomService.class);

startButton.setOnClickListener(new Button.OnClickListener(){

public void onClick(View view){

startService(serviceIntent);

}

});

stopButton.setOnClickListener(new Button.OnClickListener(){

public void onClick(View view){

stopService(serviceIntent);

}

});

}

}

RandomService.java源码如下：

package edu.scse.SimpleRandomServiceDemo;

public class RandomService extends Service{

@Override

public void onCreate() {

super.onCreate();

Toast.makeText(this, "(1) 调用onCreate()",

Toast.LENGTH\_LONG).show();

}

@Override

public void onStart(Intent intent, int startId) {

super.onStart(intent, startId);

Toast.makeText(this, "(2) 调用onStart()",

Toast.LENGTH\_SHORT).show();

double randomDouble = Math.random();

String msg = "随机数："+ String.valueOf(randomDouble);

Toast.makeText(this,msg, Toast.LENGTH\_SHORT).show();

}

@Override

public void onDestroy() {

super.onDestroy();

Toast.makeText(this, "(3) 调用onDestroy()",

Toast.LENGTH\_SHORT).show();

}

@Override

public IBinder onBind(Intent intent) {

return null;

}

}

Main.xml代码：

<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:id="@+id/label"

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content"

android:text="@string/hello">

</TextView>

<Button android:id="@+id/start"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="启动Service" >

</Button>

<Button android:id="@+id/stop"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="停止Service" >

</Button>

</LinearLayout>

AndroidManifest.xml代码如下：

<manifest xmlns:android="http://schemas.android.com/apk/res/android"

package="edu.scse.SimpleRandomServiceDemo"

android:versionCode="1"

android:versionName="1.0">

<application android:icon="@drawable/icon" android:label="@string/app\_name">

<activity android:name=".SimpleRandomServiceDemo"

android:label="@string/app\_name">

<intent-filter>

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

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

</intent-filter>

</activity>

<service android:name=".RandomService"/>

</application>

<uses-sdk android:minSdkVersion="3" />

</manifest>

2．ThreadRandomServiceDemo程序

ThreadRandomServiceDemo.java源代码如下：

package edu.scse.ThreadRandomServiceDemo;

public class ThreadRandomServiceDemo extends Activity {

private static Handler handler = new Handler();

private static TextView labelView = null;

private static double randomDouble ;

public static void UpdateGUI(double refreshDouble){

randomDouble = refreshDouble;

handler.post(RefreshLable);

}

private static Runnable RefreshLable = new Runnable(){

@Override

public void run() {

labelView.setText(String.valueOf(randomDouble));

}

};

@Override

public void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.main);

labelView = (TextView)findViewById(R.id.label);

Button startButton = (Button)findViewById(R.id.start);

Button stopButton = (Button)findViewById(R.id.stop);

final Intent serviceIntent = new Intent(this, RandomService.class);

startButton.setOnClickListener(new Button.OnClickListener(){

public void onClick(View view){

startService(serviceIntent);

}

});

stopButton.setOnClickListener(new Button.OnClickListener(){

public void onClick(View view){

stopService(serviceIntent);

}

});

}

}

RandomService源代码如下：

package edu.scse.ThreadRandomServiceDemo;

public class RandomService extends Service{

private Thread workThread;

@Override

public void onCreate() {

super.onCreate();

Toast.makeText(this, "(1) 调用onCreate()",

Toast.LENGTH\_LONG).show();

workThread = new Thread(null,backgroudWork,"WorkThread");

}

@Override

public void onStart(Intent intent, int startId) {

super.onStart(intent, startId);

Toast.makeText(this, "(2) 调用onStart()",

Toast.LENGTH\_SHORT).show();

if (!workThread.isAlive()){

workThread.start();

}

}

@Override

public void onDestroy() {

super.onDestroy();

Toast.makeText(this, "(3) 调用onDestroy()",

Toast.LENGTH\_SHORT).show();

workThread.interrupt();

}

@Override

public IBinder onBind(Intent intent) {

return null;

}

private Runnable backgroudWork = new Runnable(){

@Override

public void run() {

try {

while(!Thread.interrupted()){

double randomDouble = Math.random();

ThreadRandomServiceDemo.UpdateGUI(randomDouble);

Thread.sleep(1000);

}

} catch (InterruptedException e) {

e.printStackTrace();

}

}

};

}

Main.xml源代码如下：

<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:id="@+id/label"

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content"

android:text="@string/hello">

</TextView>

<Button android:id="@+id/start"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="启动Service" >

</Button>

<Button android:id="@+id/stop"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="停止Service" >

</Button>

</LinearLayout>

AndroidManifest.xml源代码如下：

<manifest xmlns:android="http://schemas.android.com/apk/res/android"

package="edu.scse.ThreadRandomServiceDemo"

android:versionCode="1"

android:versionName="1.0">

<application android:icon="@drawable/icon" android:label="@string/app\_name">

<activity android:name=".ThreadRandomServiceDemo"

android:label="@string/app\_name">

<intent-filter>

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

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

</intent-filter>

</activity>

<service android:name=".RandomService"/>

</application>

<uses-sdk android:minSdkVersion="3" />

</manifest>

* 1. SimpleMathServiceDemo程序

SimpleMathServiceDemo.java源代码如下：

package edu.scse.SimpleMathServiceDemo;

public class SimpleMathServiceDemo extends Activity {

private MathService mathService;

private boolean isBound = false;

TextView labelView;

@Override

public void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.main);

labelView = (TextView)findViewById(R.id.label);

Button bindButton = (Button)findViewById(R.id.bind);

Button unbindButton = (Button)findViewById(R.id.unbind);

Button computButton = (Button)findViewById(R.id.compute);

bindButton.setOnClickListener(new View.OnClickListener(){

@Override

public void onClick(View v) {

if(!isBound){

final Intent serviceIntent = new Intent(SimpleMathServiceDemo.this,MathService.class);

bindService(serviceIntent,mConnection,Context.BIND\_AUTO\_CREATE);

isBound = true;

}

}

});

unbindButton.setOnClickListener(new View.OnClickListener(){

@Override

public void onClick(View v) {

if(isBound){

isBound = false;

unbindService(mConnection);

mathService = null;

}

}

});

computButton.setOnClickListener(new View.OnClickListener(){

@Override

public void onClick(View v) {

if (mathService == null){

labelView.setText("未绑定服务");

return;

}

long a = Math.round(Math.random()\*100);

long b = Math.round(Math.random()\*100);

long result = mathService.Add(a, b);

String msg = String.valueOf(a)+" + "+String.valueOf(b)+

" = "+String.valueOf(result);

labelView.setText(msg);

}

});

}

private ServiceConnection mConnection = new ServiceConnection() {

@Override

public void onServiceConnected(ComponentName name, IBinder service) {

mathService = ((MathService.LocalBinder)service).getService();

}

@Override

public void onServiceDisconnected(ComponentName name) {

mathService = null;

}

};

}

MathService.java源代码如下：

package edu.scse.SimpleMathServiceDemo;

public class MathService extends Service{

private final IBinder mBinder = new LocalBinder();

public class LocalBinder extends Binder{

MathService getService() {

return MathService.this;

}

}

@Override

public IBinder onBind(Intent intent) {

Toast.makeText(this, "本地绑定：MathService",

Toast.LENGTH\_SHORT).show();

return mBinder;

}

@Override

public boolean onUnbind(Intent intent){

Toast.makeText(this, "取消本地绑定：MathService",

Toast.LENGTH\_SHORT).show();

return false;

}

public long Add(long a, long b){

return a+b;

}

}

Main.xml源代码如下：

<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:id="@+id/label"

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content"

android:text="@string/hello">

</TextView>

<Button android:id="@+id/bind"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="**服务绑定**" >

</Button>

<Button android:id="@+id/unbind"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="**取消绑定**" >

</Button>

<Button android:id="@+id/compute"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="加法运算" >

</Button>

</LinearLayout>

AndroidManifest.xml源代码如下：

<manifest xmlns:android="http://schemas.android.com/apk/res/android"

package="edu.scse.SimpleMathServiceDemo"

android:versionCode="1"

android:versionName="1.0">

<application android:icon="@drawable/icon" android:label="@string/app\_name">

<activity android:name=".SimpleMathServiceDemo"

android:label="@string/app\_name">

<intent-filter>

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

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

</intent-filter>

</activity>

<service android:name=".MathService"/>

</application>

<uses-sdk android:minSdkVersion="3" />

</manifest>

4．RemoteMathServiceDemo程序

MathService.java

package edu.scse.RemoteMathServiceDemo;

public class MathService extends Service{

private final IMathService.Stub mBinder = new IMathService.Stub() {

public long Add(long a, long b) {

return a + b;

}

};

@Override

public IBinder onBind(Intent intent) {

Toast.makeText(this, "远程绑定：MathService",

Toast.LENGTH\_SHORT).show();

return mBinder;

}

@Override

public boolean onUnbind (Intent intent){

Toast.makeText(this, "取消远程绑定：MathService",

Toast.LENGTH\_SHORT).show();

return false;

}

}

IMathService.aidl

package edu.scse.RemoteMathServiceDemo;

interface IMathService {

long Add(long a, long b);

}

Main.xml源代码如下：

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"

android:orientation="vertical"

android:layout\_width="fill\_parent"

android:layout\_height="fill\_parent"

>

<TextView

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content"

android:text="@string/hello"

/>

</LinearLayout>

AndroidManifest.xml源代码如下：

<manifest xmlns:android="http://schemas.android.com/apk/res/android"

package="edu.scse.RemoteMathServiceDemo"

android:versionCode="1"

android:versionName="1.0">

<application android:icon="@drawable/icon" android:label="@string/app\_name">

<service android:name=".MathService"

android:process=":remote">

<intent-filter>

<action android:name="edu.scse.RemoteMathServiceDemo.MathService" />

</intent-filter>

</service>

</application>

<uses-sdk android:minSdkVersion="3" />

</manifest>

五、实验报告要求

1、上机输入和调试自己所编的程序。一人一组，独立上机调试，上机时出现的问题，最好独立解决。

2、严格按照实验报告格式完成实验报告并上交服务器

六、实验注意事项

注意将各实验步骤进行截图保存，Appname应包含学号，并在截图中显示出来。

七、思考题

1. 简述Service的原理和用途？

2．如何通过Service实现MP3播放器音乐播放器？

3．使用“ParcelMathServiceDemo”程序，说明如何在跨进程服务中使用自定义数据类型?

# 实验七 数据存储与访问

一、实验目的

1. 使学生掌握SharedPreferences的使用方法。

2．使学生掌握SQLite数据库的建立和操作方法。

3．使学生掌握ContentProvider的创建与使用方法。

二、实验内容

1. 通过SimplePreferenceDemo说明SharedPreferences的文件保存位置和保存格式。

2．通过InternalFileDemo程序实现在内部存储器上进行文件写入和读取。

3．通过ResourceFileDemo程序实现如何在程序运行时访问资源文件。

4．通过SQLiteDemo来实现对数据库操作的一个示例。

三、实验仪器、设备

硬件：PC 微型计算机、1G以上内存，40G以上硬盘

软件：Windows XP，Eclipse , JDK , Android SDK

四、实验步骤

1. SimplePreferenceDemo程序

SimplePreferenceDemo.java源代码如下：

package edu.scse.SimplePreferenceDemo;

public class SimplePreferenceDemo extends Activity {

private EditText nameText;

private EditText ageText;

private EditText heightText;

public static final String PREFERENCE\_NAME = "SaveSetting";

public static int MODE = Context.MODE\_WORLD\_READABLE + Context.MODE\_WORLD\_WRITEABLE;

@Override

public void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.main);

nameText = (EditText)findViewById(R.id.name);

ageText = (EditText)findViewById(R.id.age);

heightText = (EditText)findViewById(R.id.height);

}

@Override

public void onStart(){

super.onStart();

loadSharedPreferences();

}

@Override

public void onStop(){

super.onStop();

saveSharedPreferences();

}

private void loadSharedPreferences(){

SharedPreferences sharedPreferences = getSharedPreferences(PREFERENCE\_NAME, MODE);

String name = sharedPreferences.getString("Name","Tom");

int age = sharedPreferences.getInt("Age", 20);

float height = sharedPreferences.getFloat("Height",1.81f);

nameText.setText(name);

ageText.setText(String.valueOf(age));

heightText.setText(String.valueOf(height));

}

private void saveSharedPreferences(){

SharedPreferences sharedPreferences = getSharedPreferences(PREFERENCE\_NAME, MODE);

SharedPreferences.Editor editor = sharedPreferences.edit();

editor.putString("Name", nameText.getText().toString());

editor.putInt("Age", Integer.parseInt(ageText.getText().toString()));

editor.putFloat("Height", Float.parseFloat(heightText.getText().toString()));

editor.commit();

}

}

Main.xml代码如下：

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"

android:id="@+id/RelativeLayout01"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content" >

<EditText android:id="@+id/name"

android:text=""

android:layout\_width="280dip"

android:layout\_height="wrap\_content"

android:layout\_alignParentRight="true"

android:layout\_marginLeft="10dip" >

</EditText>

<TextView android:id="@+id/name\_label"

android:text="姓名："

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_alignParentLeft="true"

android:layout\_toRightOf="@id/name"

android:layout\_alignBaseline="@+id/name">

</TextView>

<EditText android:id="@+id/age"

android:text=""

android:layout\_width="280dip"

android:layout\_height="wrap\_content"

android:layout\_alignParentRight="true"

android:layout\_marginLeft="10dip"

android:layout\_below="@id/name"

android:numeric="integer">

</EditText>

<TextView android:id="@+id/age\_label"

android:text="年龄："

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_alignParentLeft="true"

android:layout\_toRightOf="@id/age"

android:layout\_alignBaseline="@+id/age" >

</TextView>

<EditText android:id="@+id/height"

android:layout\_width="280dip"

android:layout\_height="wrap\_content"

android:layout\_alignParentRight="true"

android:layout\_marginLeft="10dip"

android:layout\_below="@id/age"

android:numeric="decimal">

</EditText>

<TextView android:id="@+id/height\_label"

android:text="身高："

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_alignParentLeft="true"

android:layout\_toRightOf="@id/height"

android:layout\_alignBaseline="@+id/height">

</TextView>

</RelativeLayout>

AndroidManifest.xml源代码如下：

<manifest xmlns:android="http://schemas.android.com/apk/res/android"

package="edu.scse.SimplePreferenceDemo"

android:versionCode="1"

android:versionName="1.0">

<application android:icon="@drawable/icon" android:label="@string/app\_name">

<activity android:name=".SimplePreferenceDemo"

android:label="@string/app\_name">

<intent-filter>

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

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

</intent-filter>

</activity>

</application>

<uses-sdk android:minSdkVersion="3" />

</manifest>

2. InternalFileDemo程序

InternalFileDemo.java源代码如下：

package edu.scse.InternalFileDemo;

public class InternalFileDemo extends Activity {

private final String FILE\_NAME = "fileDemo.txt";

private TextView labelView;

private TextView displayView;

private CheckBox appendBox ;

private EditText entryText;

@Override

public void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.main);

labelView = (TextView)findViewById(R.id.label);

displayView = (TextView)findViewById(R.id.display);

appendBox = (CheckBox)findViewById(R.id.append);

entryText = (EditText)findViewById(R.id.entry);

Button writeButton = (Button)findViewById(R.id.write);

Button readButton = (Button)findViewById(R.id.read);

writeButton.setOnClickListener(writeButtonListener);

readButton.setOnClickListener(readButtonListener);

entryText.selectAll();

entryText.findFocus();

}

OnClickListener writeButtonListener = new OnClickListener() {

@Override

public void onClick(View v) {

FileOutputStream fos = null;

try {

if (appendBox.isChecked()){

fos = openFileOutput(FILE\_NAME,Context.MODE\_APPEND);

}

else {

fos = openFileOutput(FILE\_NAME,Context.MODE\_PRIVATE);

}

String text = entryText.getText().toString();

fos.write(text.getBytes());

labelView.setText("文件写入成功，写入长度："+text.length());

entryText.setText("");

} catch (FileNotFoundException e) {

e.printStackTrace();

}

catch (IOException e) {

e.printStackTrace();

}

finally{

if (fos != null){

try {

fos.flush();

fos.close();

} catch (IOException e) {

e.printStackTrace();

}

}

}

}

};

OnClickListener readButtonListener = new OnClickListener() {

@Override

public void onClick(View v) {

displayView.setText("");

FileInputStream fis = null;

try {

fis = openFileInput(FILE\_NAME);

if (fis.available() == 0){

return;

}

byte[] readBytes = new byte[fis.available()];

while(fis.read(readBytes) != -1){

}

String text = new String(readBytes);

displayView.setText(text);

labelView.setText("文件读取成功，文件长度："+text.length());

} catch (FileNotFoundException e) {

e.printStackTrace();

}

catch (IOException e) {

e.printStackTrace();

}

}

};

}

Main.xml代码如下：

<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:id="**@+id/label**"

android:layout\_width="**fill\_parent**"

android:layout\_height="**wrap\_content**"

android:text="**@string/hello**" />

  <EditText android:id="**@+id/entry**"

android:text="**输入文件内容**"

android:layout\_width="**fill\_parent**"

android:layout\_height="**wrap\_content**" />

<LinearLayout

android:id="**@+id/LinearLayout01**" android:layout\_width="**wrap\_content**" android:layout\_height="**wrap\_content**">

  <Button android:id="**@+id/write**"

android:text="**写入文件**"

android:layout\_width="**wrap\_content**" android:layout\_height="**wrap\_content**" />

  <Button android:id="**@+id/read**"

android:text="**读取文件**"

android:layout\_width="**wrap\_content**" android:layout\_height="**wrap\_content**" />

  </LinearLayout>

  <CheckBox android:id="**@+id/append**"

android:text="**追加模式**"

android:layout\_width="**wrap\_content**" android:layout\_height="**wrap\_content**" />

  <TextView android:id="**@+id/display**"

android:text="**文件内容显示区域**"

android:layout\_width="**fill\_parent**"

android:layout\_height="**fill\_parent**"

android:background="**#FFFFFF**"

android:textColor="**#000000**" />

  </LinearLayout>

AndroidManifest.xml源代码如下：

<manifest xmlns:android="http://schemas.android.com/apk/res/android"

package="edu.scse.InternalFileDemo"

android:versionCode="1"

android:versionName="1.0">

<application android:icon="@drawable/icon" android:label="@string/app\_name">

<activity android:name=".InternalFileDemo"

android:label="@string/app\_name">

<intent-filter>

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

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

</intent-filter>

</activity>

</application>

<uses-sdk android:minSdkVersion="3" />

</manifest>

3. ResourceFileDemo程序

ResourceFileDemo源代码如下：

package edu.scse.ResourceFileDemo;

public class ResourceFileDemo extends Activity {

private Resources resources;

private TextView displayView;

@Override

public void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.main);

Button readRawButton = (Button)findViewById(R.id.read\_raw);

Button readXmlButton = (Button)findViewById(R.id.read\_xml);

Button clearButton = (Button)findViewById(R.id.clear);

readRawButton.setOnClickListener(readRawButtonListener);

readXmlButton.setOnClickListener(readXmlButtonListener);

clearButton.setOnClickListener(clearButtonListener);

this.displayView = (TextView)findViewById(R.id.display);

this.resources = this.getResources();

}

OnClickListener readRawButtonListener = new OnClickListener() {

@Override

public void onClick(View v) {

InputStream inputStream = null;

try {

inputStream = resources.openRawResource(R.raw.raw\_file);

byte[] reader = new byte[inputStream.available()];

while (inputStream.read(reader) != -1) {

}

displayView.setText(new String(reader,"utf-8"));

} catch (IOException e) {

Log.e("ResourceFileDemo", e.getMessage(), e);

} finally {

if (inputStream != null) {

try {

inputStream.close();

}

catch (IOException e) { }

}

}

}

};

OnClickListener readXmlButtonListener = new OnClickListener() {

@Override

public void onClick(View v) {

XmlPullParser parser = resources.getXml(R.xml.people);

String msg = "";

try {

while (parser.next() != XmlPullParser.END\_DOCUMENT) {

String people = parser.getName();

String name = null;

String age = null;

String height = null;

if ((people != null) && people.equals("person")) {

int count = parser.getAttributeCount();

for (int i = 0; i < count; i++) {

String attrName = parser.getAttributeName(i);

String attrValue = parser.getAttributeValue(i);

if ((attrName != null) && attrName.equals("name")) {

name = attrValue;

} else if ((attrName != null) && attrName.equals("age")) {

age = attrValue;

} else if ((attrName != null) && attrName.equals("height")) {

height = attrValue;

}

}

if ((name != null) && (age != null) && (height != null)) {

msg += "姓名："+name+"，年龄："+age+"，身高："+height+"\n";

}

}

}

} catch (Exception e) {

Log.e("ResourceFileDemo", e.getMessage(), e);

}

displayView.setText(msg);

}

};

OnClickListener clearButtonListener = new OnClickListener() {

@Override

public void onClick(View v) {

displayView.setText("");

}

};

}

Main.xml代码如下：

<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:id="@+id/label"

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content"

android:text="@string/hello">

</TextView>

<LinearLayout android:id="@+id/LinearLayout01"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content">

<Button android:id="@+id/read\_raw"

android:text="**读取原始文件**"

android:layout\_width="120dip"

android:layout\_height="wrap\_content">

</Button>

<Button android:id="@+id/read\_xml"

android:text="**读取XML文件**"

android:layout\_width="120dip"

android:layout\_height="wrap\_content">

</Button>

<Button android:id="@+id/clear"

android:text="清除显示"

android:layout\_width="80dip"

android:layout\_height="wrap\_content">

</Button>

</LinearLayout>

<TextView android:id="@+id/display"

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content"

android:text="">

</TextView>

</LinearLayout>

AndroidManifest.xml源代码如下：

<manifest xmlns:android="http://schemas.android.com/apk/res/android"

package="edu.scse.ResourceFileDemo"

android:versionCode="1"

android:versionName="1.0">

<application android:icon="@drawable/icon" android:label="@string/app\_name">

<activity android:name=".ResourceFileDemo"

android:label="@string/app\_name">

<intent-filter>

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

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

</intent-filter>

</activity>

</application>

<uses-sdk android:minSdkVersion="3" />

</manifest>

4. SQLiteDemo程序

SQLiteDemo.java源代码：

package edu.scse.SQLiteDemo;

public class SQLiteDemo extends Activity {

/\*\* Called when the activity is first created. \*/

private DBAdapter dbAdepter ;

private EditText nameText;

private EditText ageText;

private EditText heightText;

private EditText idEntry;

private TextView labelView;

private TextView displayView;

@Override

public void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.main);

nameText = (EditText)findViewById(R.id.name);

ageText = (EditText)findViewById(R.id.age);

heightText = (EditText)findViewById(R.id.height);

idEntry = (EditText)findViewById(R.id.id\_entry);

labelView = (TextView)findViewById(R.id.label);

displayView = (TextView)findViewById(R.id.display);

Button addButton = (Button)findViewById(R.id.add);

Button queryAllButton = (Button)findViewById(R.id.query\_all);

Button clearButton = (Button)findViewById(R.id.clear);

Button deleteAllButton = (Button)findViewById(R.id.delete\_all);

Button queryButton = (Button)findViewById(R.id.query);

Button deleteButton = (Button)findViewById(R.id.delete);

Button updateButton = (Button)findViewById(R.id.update);

addButton.setOnClickListener(addButtonListener);

queryAllButton.setOnClickListener(queryAllButtonListener);

clearButton.setOnClickListener(clearButtonListener);

deleteAllButton.setOnClickListener(deleteAllButtonListener);

queryButton.setOnClickListener(queryButtonListener);

deleteButton.setOnClickListener(deleteButtonListener);

updateButton.setOnClickListener(updateButtonListener);

dbAdepter = new DBAdapter(this);

dbAdepter.open();

}

OnClickListener addButtonListener = new OnClickListener() {

@Override

public void onClick(View v) {

People people = new People();

people.Name = nameText.getText().toString();

people.Age = Integer.parseInt(ageText.getText().toString());

people.Height = Float.parseFloat(heightText.getText().toString());

long colunm = dbAdepter.insert(people);

if (colunm == -1 ){

labelView.setText("添加过程错误！");

} else {

labelView.setText("成功添加数据，ID："+String.valueOf(colunm));

}

}

};

OnClickListener queryAllButtonListener = new OnClickListener() {

@Override

public void onClick(View v) {

People[] peoples = dbAdepter.queryAllData();

if (peoples == null){

labelView.setText("数据库中没有数据");

return;

}

labelView.setText("数据库：");

String msg = "";

for (int i = 0 ; i<peoples.length; i++){

msg += peoples[i].toString()+"\n";

}

displayView.setText(msg);

}

};

OnClickListener clearButtonListener = new OnClickListener() {

@Override

public void onClick(View v) {

displayView.setText("");

}

};

OnClickListener deleteAllButtonListener = new OnClickListener() {

@Override

public void onClick(View v) {

dbAdepter.deleteAllData();

String msg = "数据全部删除";

labelView.setText(msg);

}

};

OnClickListener queryButtonListener = new OnClickListener() {

@Override

public void onClick(View v) {

int id = Integer.parseInt(idEntry.getText().toString());

People[] peoples = dbAdepter.queryOneData(id);

if (peoples == null){

labelView.setText("数据库中没有ID为"+String.valueOf(id)+"的数据");

return;

}

labelView.setText("数据库：");

displayView.setText(peoples[0].toString());

}

};

OnClickListener deleteButtonListener = new OnClickListener() {

@Override

public void onClick(View v) {

long id = Integer.parseInt(idEntry.getText().toString());

long result = dbAdepter.deleteOneData(id);

String msg = "删除ID为"+idEntry.getText().toString()+"的数据" + (result>0?"成功":"失败");

labelView.setText(msg);

}

};

OnClickListener updateButtonListener = new OnClickListener() {

@Override

public void onClick(View v) {

People people = new People();

people.Name = nameText.getText().toString();

people.Age = Integer.parseInt(ageText.getText().toString());

people.Height = Float.parseFloat(heightText.getText().toString());

long id = Integer.parseInt(idEntry.getText().toString());

long count = dbAdepter.updateOneData(id, people);

if (count == -1 ){

labelView.setText("更新错误！");

} else {

labelView.setText("更新成功，更新数据"+String.valueOf(count)+"条");

}

}

};

}

People.java

package edu.scse.SQLiteDemo;

public class People {

public int ID = -1;

public String Name;

public int Age;

public float Height;

@Override

public String toString(){

String result = "";

result += "ID：" + this.ID + "，";

result += "姓名：" + this.Name + "，";

result += "年龄：" + this.Age + "， ";

result += "身高：" + this.Height + "，";

return result;

}

}

DBAdapter.java源代码如下：

package edu.scse.SQLiteDemo;

public class DBAdapter {

private static final String DB\_NAME = "people.db";

private static final String DB\_TABLE = "peopleinfo";

private static final int DB\_VERSION = 1;

public static final String KEY\_ID = "\_id";

public static final String KEY\_NAME = "name";

public static final String KEY\_AGE = "age";

public static final String KEY\_HEIGHT = "height";

private SQLiteDatabase db;

private final Context context;

private DBOpenHelper dbOpenHelper;

public DBAdapter(Context \_context) {

context = \_context;

}

/\*\* Close the database \*/

public void close() {

if (db != null){

db.close();

db = null;

}

}

/\*\* Open the database \*/

public void open() throws SQLiteException {

dbOpenHelper = new DBOpenHelper(context, DB\_NAME, null, DB\_VERSION);

try {

db = dbOpenHelper.getWritableDatabase();

}

catch (SQLiteException ex) {

db = dbOpenHelper.getReadableDatabase();

}

}

public long insert(People people) {

ContentValues newValues = new ContentValues();

newValues.put(KEY\_NAME, people.Name);

newValues.put(KEY\_AGE, people.Age);

newValues.put(KEY\_HEIGHT, people.Height);

return db.insert(DB\_TABLE, null, newValues);

}

public People[] queryAllData() {

Cursor results = db.query(DB\_TABLE, new String[] { KEY\_ID, KEY\_NAME, KEY\_AGE, KEY\_HEIGHT},

null, null, null, null, null);

return ConvertToPeople(results);

}

public People[] queryOneData(long id) {

Cursor results = db.query(DB\_TABLE, new String[] { KEY\_ID, KEY\_NAME, KEY\_AGE, KEY\_HEIGHT},

KEY\_ID + "=" + id, null, null, null, null);

return ConvertToPeople(results);

}

private People[] ConvertToPeople(Cursor cursor){

int resultCounts = cursor.getCount();

if (resultCounts == 0 || !cursor.moveToFirst()){

return null;

}

People[] peoples = new People[resultCounts];

for (int i = 0 ; i<resultCounts; i++){

peoples[i] = new People();

peoples[i].ID = cursor.getInt(0);

peoples[i].Name = cursor.getString(cursor.getColumnIndex(KEY\_NAME));

peoples[i].Age = cursor.getInt(cursor.getColumnIndex(KEY\_AGE));

peoples[i].Height = cursor.getFloat(cursor.getColumnIndex(KEY\_HEIGHT));

cursor.moveToNext();

}

return peoples;

}

public long deleteAllData() {

return db.delete(DB\_TABLE, null, null);

}

public long deleteOneData(long id) {

return db.delete(DB\_TABLE, KEY\_ID + "=" + id, null);

}

public long updateOneData(long id , People people){

ContentValues updateValues = new ContentValues();

updateValues.put(KEY\_NAME, people.Name);

updateValues.put(KEY\_AGE, people.Age);

updateValues.put(KEY\_HEIGHT, people.Height);

return db.update(DB\_TABLE, updateValues, KEY\_ID + "=" + id, null);

}

/\*\* 静态Helper类，用于建立、更新和打开数据库\*/

private static class DBOpenHelper extends SQLiteOpenHelper {

public DBOpenHelper(Context context, String name, CursorFactory factory, int version) {

super(context, name, factory, version);

}

private static final String DB\_CREATE = "create table " +

DB\_TABLE + " (" + KEY\_ID + " integer primary key autoincrement, " +

KEY\_NAME+ " text not null, " + KEY\_AGE+ " integer," + KEY\_HEIGHT + " float);";

@Override

public void onCreate(SQLiteDatabase \_db) {

\_db.execSQL(DB\_CREATE);

}

@Override

public void onUpgrade(SQLiteDatabase \_db, int \_oldVersion, int \_newVersion) {

\_db.execSQL("DROP TABLE IF EXISTS " + DB\_TABLE);

onCreate(\_db);

}

}

}

Main.xml代码如下：

<LinearLayout xmlns:android="**http://schemas.android.com/apk/res/android**" android:orientation="**vertical**"

android:layout\_width="**fill\_parent**"

android:layout\_height="**fill\_parent**">

<RelativeLayout android:id="**@+id/RelativeLayout01**"

android:layout\_width="**wrap\_content**" android:layout\_height="**wrap\_content**">

  <EditText android:id="**@+id/name**"

android:text=""

android:layout\_width="**280dip**"

android:layout\_height="**wrap\_content**" android:layout\_alignParentRight="**true**"

android:layout\_marginLeft="**10dip**" />

  <TextView android:id="**@+id/name\_label**"

android:text="**姓名：**"

android:layout\_width="**wrap\_content**" android:layout\_height="**wrap\_content**" android:layout\_alignParentLeft="**true**" android:layout\_toRightOf="**@id/name**" android:layout\_alignBaseline="**@+id/name**" />

  <EditText android:id="**@+id/age**"

android:text=""

android:layout\_width="**280dip**"

android:layout\_height="**wrap\_content**" android:layout\_alignParentRight="**true**"

android:layout\_marginLeft="**10dip**"

android:layout\_below="**@id/name**"

android:numeric="**integer**" />

  <TextView android:id="**@+id/age\_label**"

android:text="**年龄：**"

android:layout\_width="**wrap\_content**" android:layout\_height="**wrap\_content**" android:layout\_alignParentLeft="**true**"

android:layout\_toRightOf="**@id/age**" android:layout\_alignBaseline="**@+id/age**" />

  <EditText android:id="**@+id/height**"

android:layout\_width="**280dip**"

android:layout\_height="**wrap\_content**" android:layout\_alignParentRight="**true**"

android:layout\_marginLeft="**10dip**"

android:layout\_below="**@id/age**"

android:numeric="**decimal**" />

  <TextView android:id="**@+id/height\_label**"

android:text="**身高：**"

android:layout\_width="**wrap\_content**" android:layout\_height="**wrap\_content**"

android:layout\_alignParentLeft="**true**" android:layout\_toRightOf="**@id/height**" android:layout\_alignBaseline="**@+id/height**" />

  </RelativeLayout>

<LinearLayout android:id="**@+id/LinearLayout01**"

android:layout\_width="**fill\_parent**"

android:layout\_height="**wrap\_content**">

  <Button android:id="**@+id/add**"

android:text="**添加数据**"

android:layout\_width="**wrap\_content**" android:layout\_height="**wrap\_content**"

android:padding="**5dip**"

android:layout\_weight="**1**" />

  <Button android:id="**@+id/query\_all**"

android:text="**全部显示**"

android:layout\_width="**wrap\_content**" android:layout\_height="**wrap\_content**"

android:padding="**5dip**"

android:layout\_weight="**1**" />

  <Button android:id="**@+id/clear**"

android:text="**清除显示**"

android:layout\_width="**wrap\_content**" android:layout\_height="**wrap\_content**"

android:padding="**5dip**"

android:layout\_weight="**1**" />

  <Button android:id="**@+id/delete\_all**"

android:text="**全部删除**"

android:layout\_width="**wrap\_content**" android:layout\_height="**wrap\_content**"

android:padding="**5dip**"

android:layout\_weight="**1**" />

  </LinearLayout>

<LinearLayout android:id="**@+id/LinearLayout03**"

android:layout\_width="**fill\_parent**" android:layout\_height="**wrap\_content**">

  <TextView android:text="**ID：**"

android:layout\_width="**wrap\_content**" android:layout\_height="**wrap\_content**"

android:padding="**3dip**" />

  <EditText android:id="**@+id/id\_entry**"

android:layout\_width="**50dip**"

android:layout\_height="**wrap\_content**"

android:padding="**3dip**"

android:layout\_weight="**1**" />

  <Button android:id="**@+id/delete**"

android:text="**ID删除**"

android:layout\_width="**50dip**"

android:layout\_height="**wrap\_content**"

android:padding="**3dip**" android:layout\_weight="**1**" />

  <Button android:id="**@+id/query**"

android:text="**ID查询**"

android:layout\_width="**50dip**"

android:layout\_height="**wrap\_content**"

android:padding="**3dip**"

android:layout\_weight="**1**" />

  <Button android:id="**@+id/update**"

android:text="**ID更新**"

android:layout\_width="**50dip**"

android:layout\_height="**wrap\_content**"

android:padding="**3dip**"

android:layout\_weight="**1**" />

  </LinearLayout>

  <TextView android:id="**@+id/label**"

android:text="**查询结果：**"

android:layout\_width="**wrap\_content**" android:layout\_height="**wrap\_content**" />

<ScrollView android:layout\_width="**fill\_parent**"

android:layout\_height="**fill\_parent**">

<LinearLayout android:layout\_width="**fill\_parent**"

android:layout\_height="**wrap\_content**" android:orientation="**vertical**">

  <TextView android:id="**@+id/display**"

android:text=""

android:layout\_width="**wrap\_content**" android:layout\_height="**wrap\_content**" />

  </LinearLayout>

  </ScrollView>

  </LinearLayout>

AndroidManifest.xml源代码如下：

<manifest xmlns:android="http://schemas.android.com/apk/res/android"

package="edu.scse.SQLiteDemo"

android:versionCode="1"

android:versionName="1.0">

<application android:icon="@drawable/icon" android:label="@string/app\_name">

<activity android:name=".SQLiteDemo"

android:label="@string/app\_name">

<intent-filter>

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

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

</intent-filter>

</activity>

</application>

<uses-sdk android:minSdkVersion="3" />

</manifest>

五、实验报告要求

1、上机输入和调试自己所编的程序。一人一组，独立上机调试，上机时出现的问题，最好独立解决。

2、严格按照实验报告格式完成实验报告并上交服务器

六、实验注意事项

注意将各实验步骤进行截图保存，Appname应包含学号，并在截图中显示出来。

七、思考题

1、简述各种文件存储的区别与适用情况。

2、简述SQLite数据库的特点和体系结构。

3、简述ContentProvider的用途和原理。

4．如何通过一个无界面的ContentProviderDemo程序，实现提供一个ContentProvider组件，供其他应用程序进行数据交换，底层使用SQLite数据库，支持数据的添加、删除、更新和查询等基本操作。

5．如何使用SDcardFileDemo程序来实现将数据保存在SD卡。

# 实验八 位置服务于地图应用

一、实验目的

1. 使学生更好的理解位置服务的概念。

1. 使学生掌握地图密钥的申请和获取位置信息的方法。
2. 使学生掌握MapView和MapController的用法。

二、实验内容

1. 使用“CurrentLocationDemo”程序来完成显示当前位置新的功能，并能够监视设备的位置变化的功能。

2．使用“MapOverlayDemo”来实现如何在Google地图上添加覆盖层，并在预订的物理坐标上显示提示信息。

三、实验仪器、设备

硬件：PC 微型计算机、1G以上内存，40G以上硬盘

软件：Windows XP，Eclipse , JDK , Android SDK

四、实验步骤

1. CurrentLocationDemo程序

CurrentLocationDemo.java源代码

package edu.scse.CurrentLocationDemo;

public class CurrentLocationDemo extends Activity {

/\*\* Called when the activity is first created. \*/

@Override

public void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.main);

String serviceString = Context.LOCATION\_SERVICE;

LocationManager locationManager = (LocationManager)getSystemService(serviceString);

String provider = LocationManager.GPS\_PROVIDER;

Location location = locationManager.getLastKnownLocation(provider);

getLocationInfo(location);

locationManager.requestLocationUpdates(provider, 2000, 0, locationListener);

}

private void getLocationInfo(Location location){

String latLongInfo;

TextView locationText = (TextView)findViewById(R.id.label);

if (location != null){

double lat = location.getLatitude();

double lng = location.getLongitude();

latLongInfo = "Lat: " + lat + "\nLong: " + lng;

}

else{

latLongInfo = "No location found";

}

locationText.setText("Your Current Position is:\n" + latLongInfo);

}

private final LocationListener locationListener = new LocationListener(){

@Override

public void onLocationChanged(Location location) {

getLocationInfo(location);

}

@Override

public void onProviderDisabled(String provider) {

getLocationInfo(null);

}

@Override

public void onProviderEnabled(String provider) {

getLocationInfo(null);

}

@Override

public void onStatusChanged(String provider, int status, Bundle extras) {

}

};

}

Main.xml 源代码如下:

<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:id="@+id/label"

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content"

android:text="@string/hello"/>

</LinearLayout>

AndroidManifest.xml源代码如下：

<manifest xmlns:android="http://schemas.android.com/apk/res/android"

package="edu.scse.CurrentLocationDemo"

android:versionCode="1"

android:versionName="1.0">

<application android:icon="@drawable/icon" android:label="@string/app\_name">

<activity android:name=".CurrentLocationDemo"

android:label="@string/app\_name">

<intent-filter>

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

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

</intent-filter>

</activity>

</application>

<uses-permission android:name="android.permission.ACCESS\_FINE\_LOCATION"/>

<uses-sdk android:minSdkVersion="3" />

</manifest>

2．MapOverlayDemo程序

MapOverlayDemo.java源代码如下：

package edu.scse.MapOverlayDemo;

public class MapOverlayDemo extends MapActivity {

private MapView mapView;

private MapController mapController;

private TextOverlay textOverlay;

@Override

public void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.main);

mapView = (MapView)findViewById(R.id.mapview);

mapController = mapView.getController();

Double lng = 126.676530486 \* 1E6;

Double lat = 45.7698895661 \* 1E6;

GeoPoint point = new GeoPoint(lat.intValue(), lng.intValue());

mapController.setCenter(point);

mapController.setZoom(11);

mapController.animateTo(point);

textOverlay = new TextOverlay();

List<Overlay> overlays = mapView.getOverlays();

overlays.add(textOverlay);

}

@Override

protected boolean isRouteDisplayed() {

// TODO Auto-generated method stub

return false;

}

}

TextOverlay源代码如下：

package edu.scse.MapOverlayDemo;

public class TextOverlay extends Overlay {

private final int mRadius = 5;

@Override

public void draw(Canvas canvas, MapView mapView, boolean shadow) {

Projection projection = mapView.getProjection();

if (shadow == false){

Double lng = 126.676530486 \* 1E6;

Double lat = 45.7698895661 \* 1E6;

GeoPoint geoPoint = new GeoPoint(lat.intValue(), lng.intValue());

Point point = new Point();

projection.toPixels(geoPoint, point);

RectF oval = new RectF(point.x - mRadius, point.y - mRadius,

point.x + mRadius, point.y + mRadius);

Paint paint = new Paint();

paint.setARGB(250, 250, 0, 0);

paint.setAntiAlias(true);

paint.setFakeBoldText(true);

canvas.drawOval(oval, paint);

canvas.drawText("标记点", point.x+2\*mRadius, point.y, paint);

}

super.draw(canvas, mapView, shadow);

}

@Override

public boolean onTap(GeoPoint p, MapView mapView) {

return false;

}

}

Main.xml 源代码如下:

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"

android:orientation="vertical"

android:layout\_width="fill\_parent"

android:layout\_height="fill\_parent">

<TextView android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content"

android:text="@string/hello"/>

<com.google.android.maps.MapView

android:id="@+id/mapview"

android:layout\_width="fill\_parent"

android:layout\_height="fill\_parent"

android:enabled="true"

android:clickable="true"

android:apiKey="0mVK8GeO6WUz4S2F94z52CIGSSlvlTwnrE4DsiA"/> //自己申请

</LinearLayout>

AndroidManifest.xml源代码如下:

<manifest xmlns:android="http://schemas.android.com/apk/res/android"

package="edu.scse.MapOverlayDemo"

android:versionCode="1"

android:versionName="1.0">

<application android:icon="@drawable/icon" android:label="@string/app\_name">

<activity android:name=".MapOverlayDemo"

android:label="@string/app\_name">

<intent-filter>

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

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

</intent-filter>

</activity>

<uses-library android:name="com.google.android.maps"></uses-library>

</application>

<uses-sdk android:minSdkVersion="3" />

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

</manifest>

五、实验报告要求

1、上机输入和调试自己所编的程序。一人一组，独立上机调试，上机时出现的问题，最好独立解决。

2、严格按照实验报告格式完成实验报告并上交服务器

六、实验注意事项

注意将各实验步骤进行截图保存，Appname应包含学号，并在截图中显示出来。

七、思考题

1、简述Android应用程序如何使用Google地图？

2、简述Google地图覆盖层的使用方法？

# 实验九 Android NDK开发

一、实验目的

1．使学生更好的理解Android NDK概念和用途。

2．使学生掌握Android NDK的开发技术。

二、实验内容

使用AndroidNdkDemo程序进行加法运算，程序会随机产生两个整数，然后调用C语言开发的共享库对这两个整数进行加法运算，最后将运算结果显示在用户界面上。

三、实验仪器、设备

硬件：PC 微型计算机、1G以上内存，40G以上硬盘

软件：Windows XP，Eclipse , JDK , Android SDK

四、实验步骤

1、AndroidNdkDemo程序

AndroidNdkDemo.java文件的源代码如下：

package edu.scse.AndroidNdkDemo;

public class AndroidNdkDemo extends Activity {

/\*\* Called when the activity is first created. \*/

@Override

public void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.main);

final TextView displayLable = (TextView)findViewById(R.id.display);

Button btn = (Button)findViewById(R.id.add\_btn);

btn.setOnClickListener(new View.OnClickListener(){

@Override

public void onClick(View v) {

double randomDouble = Math.random();

long x = Math.round(randomDouble\*100);

randomDouble = Math.random();

long y = Math.round(randomDouble\*100);

System.loadLibrary("add-module");

long z = add(x, y);

String msg = x+" + " +y+" = "+z;

displayLable.setText(msg);

}

});

}

public native long add(long x, long y);

// public long add(long x, long y){

// return x+y;

// }

}

Main.xml源代码如下：

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"

android:orientation="vertical"

android:layout\_width="fill\_parent"

android:layout\_height="fill\_parent"

>

<TextView

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content"

android:text="@string/hello">

</TextView>

<TextView android:id="@+id/display"

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content"

android:text="">

</TextView>

<Button android:id="@+id/add\_btn"

android:text="**加法运算**"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content">

</Button>

</LinearLayout>

AndroidManifest.xml源代码如下：

<manifest xmlns:android="http://schemas.android.com/apk/res/android"

package="edu.scse.AndroidNdkDemo"

android:versionCode="1"

android:versionName="1.0">

<application android:icon="@drawable/icon" android:label="@string/app\_name">

<activity android:name=".AndroidNdkDemo"

android:label="@string/app\_name">

<intent-filter>

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

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

</intent-filter>

</activity>

</application>

<uses-sdk android:minSdkVersion="3" />

</manifest>

add-module.c源代码如下：

#include <jni.h>

jlong Java\_edu\_scse\_AndroidNdkDemo\_AndroidNdkDemo\_add( JNIEnv\* env,

jobject this,

jlong x,

jlong y )

{

return x+y;

}

Android.mk源代码如下：

LOCAL\_PATH := $(call my-dir)

include $(CLEAR\_VARS)

LOCAL\_MODULE := add-module

LOCAL\_SRC\_FILES := add-module.c

include $(BUILD\_SHARED\_LIBRARY)

五、实验报告要求

1、上机输入和调试自己所编的程序。一人一组，独立上机调试，上机时出现的问题，最好独立解决。

2、严格按照实验报告格式完成实验报告并上交服务器

六、实验注意事项

注意将各实验步骤进行截图保存，Appname应包含学号，并在截图中显示出来。

七、思考题

1．简要说明Android NDK的用途。