目录

[一. 简答题： 2](#_Toc121180831)

[1. Android平台体系架构可以分为哪几层？ 2](#_Toc121180832)

[2. Intent有两种类型？这两种类型之间有何区别？ 2](#_Toc121180833)

[3. 活动在其生命周期内可能有哪些状态？ 2](#_Toc121180834)

[4. 活动在其生命周期内可能调用哪些生命周期回调方法？ 2](#_Toc121180835)

[5. 简述如何使用SharedPreferences存储数据。 2](#_Toc121180836)

[6. Android中的广播可有哪两种类型？分别有什么特点？ 2](#_Toc121180837)

[7. AsyncTask类的哪些方法在主线程中执行？哪些方法在子线程中执行？P129 3](#_Toc121180838)

[8. 服务有哪两种类型？分别有什么特点 3](#_Toc121180839)

[二. 常用的View组件属性： 3](#_Toc121180840)

[9. **id属性** 4](#_Toc121180841)

[10. **text属性** 5](#_Toc121180842)

[11. **hint属性** 7](#_Toc121180843)

[12. **textsize属性** 8](#_Toc121180844)

[13. **imageView的src属性** 9](#_Toc121180845)

[14. **checkbox的checked属性** 10](#_Toc121180846)

[三. 与考试相关的项目及主要考点 11](#_Toc121180847)

[1. GuessNumber： 11](#_Toc121180848)

[1) 考察在Activity中的事件处理的方法。 11](#_Toc121180849)

[2) 考察字符串资源文件的使用（资源文件的定义，在代码中使用字符串资源文件的方法） 14](#_Toc121180850)

[3) 使用隐式Intent启动拨号与短信应用Activity的方法 16](#_Toc121180851)

[4) 考察ListView组件使用的基本步骤 16](#_Toc121180852)

[5) 考察SharedPreference和SQLite数据库这两种存取数据方式的使用，以及SQLiteOpenHelper类的使用。 17](#_Toc121180853)

[2. ExampleService:（3-4个空） 20](#_Toc121180854)

[1) 考察启动两种类型服务组件的方法 20](#_Toc121180855)

[2) 考察绑定型服务使用的基本步骤。 20](#_Toc121180856)

[3. ExampleBroadcast： 21](#_Toc121180857)

# 简答题：

## Android平台体系架构可以分为哪几层？

**应用层，应用框架层，系统运行库层，硬件抽象层，Linux内核层。**

## Intent有两种类型？这两种类型之间有何区别？

**Intent 可分为显式 Intent 和隐式 Intent 。显式 Intent 需要指定动作的执行对象，即执行某个动作的应用组件的组件名称；当使用隐式intent的时候，android系统将会找到合适的组件并启动这个组件。**

## 活动在其生命周期内可能有哪些状态？

**运行状态、暂停状态、停止状态、销毁状态。**

## 活动在其生命周期内可能调用哪些生命周期回调方法？

1. **onCreat（）：活动第一次被创建时调用，在此方法中完成活动的初始化操作。**
2. **onStart()：启动Activity时被回调。**
3. **onResume()：在活动准备好和用户进行交互时调用。返回栈顶，且处于运行状态。**
4. **onPause()：启动或恢复另一个活动时调用。**
5. **onStop()：当活动完全不可见时调用。**
6. **onDestroy()：被销毁前调用。**
7. **onRestart()：活动由停止状态变为运行状态前调用。重启。**

## 简述如何使用SharedPreferences存储数据。

**创建sp编辑器，以键值对的形式加入数据，以String Key为索引来取出数据，清除。**

## Android中的广播可有哪两种类型？分别有什么特点？

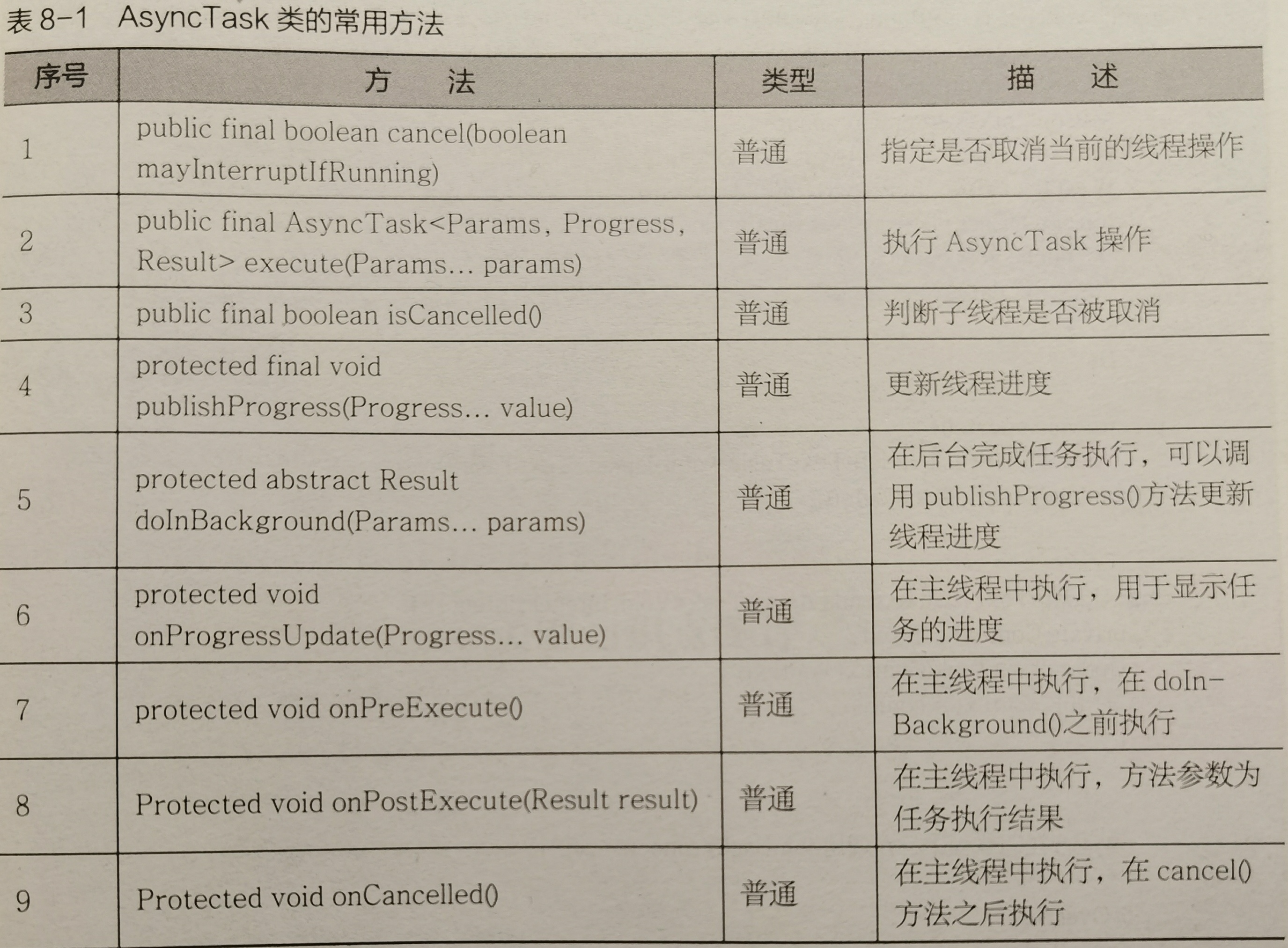
**标准广播和有序广播。**

**标准广播：完全异步执行的广播，广播接收器没有顺序，广播效率高但无法被截断。**

**有序广播：同步执行的广播，接收器有先后顺序，可以截断正在传递的广播。**

## AsyncTask类的哪些方法在主线程中执行？哪些方法在子线程中执行？P129

**在主线程执行的方法有：OnProgressUpdate（Progress... value）、onPreExecute（）、onPostExecute(Result result)、onCancelled（）；在子线程执行的方法有：doInBackground（params... params）。**



## 服务有哪两种类型？分别有什么特点

**服务分为启动型服务和绑定型服务，**

**启动型服务：应用组件通过调用startService（）方法启动服务，服务与调用者没有绝对关联，当调用者关闭后服务还会一直在后台运行。**

**绑定型服务：应用组件通过调用bindService（）方法与服务绑定，服务与调用者的生命周期所关联，当调用者关闭时，服务也会关闭。**

# 常用的View组件属性：

**希望大家可以理解并运用常用的View组件的属性，会通过选择题和综合题进行考察。**

**1、id属性：给当前的元素定义一个唯一的标识符**

**2、text属性：显示的内容信息**

**3、hint属性：文本框中初始显示的用来提示用户输入的属性**

**4、textsize属性：定义文字大小范围和尺寸**

**5、imageView的src属性：图片的来源**

**6、checkbox的checked属性：用于表示是否选中**

9. **id属性**

p60的例子

|  |  |  |
| --- | --- | --- |
| 类型 | 属性 | 值 |
| TextView | id | @+id/label |
| id | @+id/entry |
| id | @+id/ok |
| id | @+id/cancel |

* 作用：设置一个组件id，通过findViewById()的方法获取到该对象，然后进行相关设置。**特定文档中唯一的名称，**其值必然是具有唯一标识功能的名称。
* ***id="@\*"*属性主要分为三种：**

（1）android:id="@+id/new\_name"

开发者为当前的控件或者布局新定义一个id名称。该ID名称在R.java文件中会分配一个唯一的int型常量，用于对资源引用的索引，形如：

|  |
| --- |
| XML <span style="font-size:18px;">public static final int new\_name=0x7f070000; </span> |

添加资源的命名规则：资源文件只能以小写字母和下划线做首字母，随后的名字中只能出现 [a-z0-9\_.] 这些字符，否则R.java文件不会自动更新，并且eclipse会提示错误。（PS:但是实际上，使用ADT21.0.0以上的版本之后，如果开发Android 4.0.3以上版本，可以使用A-Z这些字母）

（2）android:id="@id/defined\_name"

引用一个名称为defined\_name的id资源。该引用针对的是开发者自定义的id资源。

（3）android:id="@android:id/sys\_name"

引用名称为sys\_name的系统内部资源。例如：

|  |
| --- |
| XML <span style="font-size:18px;">android:id="@android:id/tabcontent"  </span> |

* 简单来讲：  
  @+id 新增一个资源id  
  ＠id和android：id，引用现有的资源id

10. **text属性**

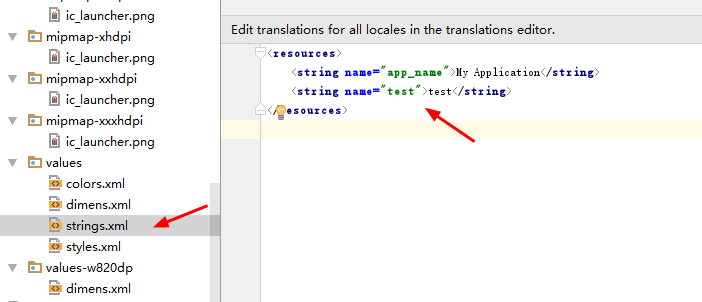
p60的例子

|  |  |  |
| --- | --- | --- |
| 类型 | 属性 | 值 |
| EditView | text | 用户名： |
| text | 确认 |
| text | 取消 |

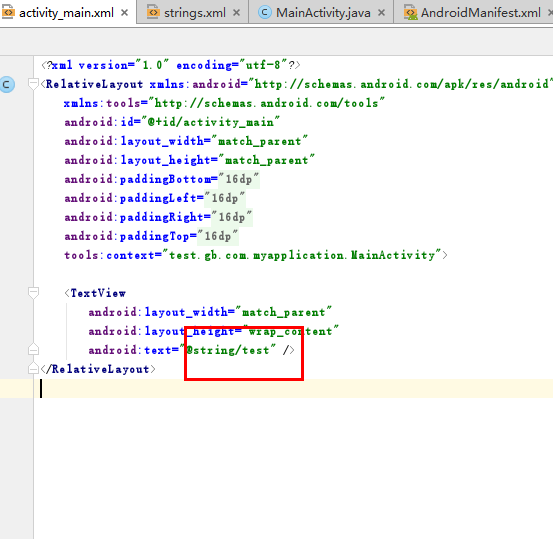
* 作用：设置显示文本

|  |  |  |
| --- | --- | --- |
| TextView对象的方法 | xml属性 | 作用 |
| SetText | text | 设置文本 |

将字符串放置在value文件夹的strings.xml文件下：



在activity\_main.xml中使用@string/\*来调用这个字符串资源，“\*”号代表string定义的name:



效果如下：



11. **hint属性**

* 作用：hint是**文本框中初始显示的用来提示用户输入的属性**，在用户点击输入框开始输入时就会消失。
* android:hint=" "

android:hint=" " 输入的是提示信息，在输入框里面做指定提示语

|  |
| --- |
| XML <EditText  android:id="@+id/Text"  android:layout\_width="320px"  android:layout\_height="50px"   <span style="color:#ff0000;">android:hint="请输入要播放文件地址："</span>   /> |

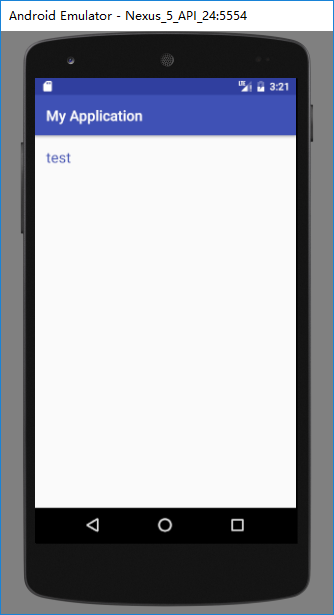
12. **textsize属性**

* 作用：设置文字的大小

|  |  |  |
| --- | --- | --- |
| TextView对象的方法 | xml属性 | 作用 |
| setTextSize | textSize | 设置文字的大小 |



效果如下：



* **扩展：**设置大小的单位（字体，也包括控件大小）

**dp:** 设备独立像素，不同设备有不同的显示效果,这个和设备硬件有关，不依赖像素。

**px:** 像素，在 不同设备显示的效果相同。

**pt:** 标准的长度单位，简单易用，单位换算1pt＝1/72英寸。

**sp:** 用于字体显示。

13. **imageView的src属性**

* background指的是背景,foreground指的是前景,**src指的是内容**;三者可以同时使用;
* **src填入图片时,是按照图片大小直接填充,并不会进行拉伸;**而使用background和foreground填入图片,则会根据ImageView的大小进行拉伸
* background和foreground是所有view的属性,总是缩放到view大小,不受scaleType影响;而**src是ImageView的特有属性,它会受到scaleType的影响**
* src设置图片的方式：

|  |
| --- |
| XML android:src="@mipmap/ic\_launcher" |

|  |
| --- |
| Java //1. setImageDrawable(Drawable drawable) image.setImageDrawable(getResources().getDrawable(R.drawable.ic\_launcher));//不会变形 //2. setImageBitmap(Bitmap bitmap) stringPath = Environment.getExternalStorageDirectory()+"/test.jpg"; Bitmap bitmap = BitmapFactory.decodeFile(path); image.setImageBitmap(bitmap);//不会变形 //3. setImageResource(int resId) image.setImageResource(R.drawable/ic\_launcher);//不会变形 |

14. **checkbox的checked属性**

checkbox：多选按钮

* 作用：android:checked="true" 设置默认选中

**checked 当前是否勾选。 Boolean**

# 与考试相关的项目及主要考点

**（综合题用，一定在下列考点中出题，示例项目在学习通的资源中）：**

## GuessNumber：

### 考察在Activity中的事件处理的方法。



响应用户交互2



响应用户交互3



事件处理函数



|  |
| --- |
| Java protected void onCreate(Bundle savedInstanceState) {  super.onCreate(savedInstanceState);  setContentView(R.layout.*activity\_main*);  edtNumber = findViewById(R.id.*edtNumber*);  btnGuess = findViewById(R.id.*btnGuess*);  btnGenerate = findViewById(R.id.*btnGenerate*);  txvInfo = findViewById(R.id.*txvInfo*); |

### 考察字符串资源文件的使用（资源文件的定义，在代码中使用字符串资源文件的方法）





|  |
| --- |
| Java private void guess() {  int num;  num = Integer.parseInt(edtNumber.getText().toString());  int flag = number.compareNumber(num);  String info;  int color;  if (flag < 0) {  info = getString(R.string.*text\_smaller*, num, number.getCount());  color = getResources().getColor(R.color.*color\_smaller*);  }  else if (flag > 0) {  info = getString(R.string.*text\_bigger*, num, number.getCount());  color = getResources().getColor(R.color.*color\_bigger*);  }  else {  info = getString(R.string.*text\_right*, num, number.getCount());  btnGenerate.setEnabled(true);  btnGuess.setEnabled(false);  btnRecord.setEnabled(true);  btnEmail.setEnabled(true);  btnMessage.setEnabled(true);  edtId.setEnabled(true);  color = getResources().getColor(R.color.*color\_right*);  }  txvInfo.setText(info);  txvInfo.setTextColor(color); } |

### 使用隐式Intent启动拨号与短信应用Activity的方法

|  |
| --- |
| Java  private void shareToEmail(){  Intent intent = new Intent();  intent.setAction(Intent.ACTION\_SENDTO);  intent.setData(Uri.parse("mai1to:"));  intent.putExtra(Intent.EXTRA\_SUBJECT, getString(R.string.*mail\_subject*));  intent.putExtra(Intent.EXTRA\_TEXT, getString(R.string.*recommend\_messgae*, number.getCount(), number.getTargetNumber()));  Intent chooseIntent = In·tent.createChooser(intent, getString(R.string.*txvShare\_text*));  startActivity(chooseIntent);  //startActivity(intent);  } |

|  |
| --- |
| Java private void shareToMessage(){  Intent intent = new Intent();  intent.setAction(Intent.ACTION\_SENDTO);  intent.setData(Uri.parse("sms"));  intent.putExtra("sms\_body", getString(R.string.*recommend\_messgae*, number.getCount(), number.getTargetNumber()));  startActivity(intent); } |

### 考察ListView组件使用的基本步骤

|  |
| --- |
| Java private void showDataBySQLite() {  ArrayList<String> records = new ArrayList<String>();  MyDBHelper dbHelper = new MyDBHelper(this, "GNDB", null, 1);  SQLiteDatabase db = dbHelper.getReadableDatabase();  Cursor cursor = db.query("RECORD",  new String[]{"userid", "number", "count"},  null, null, null, null,null);  while(cursor.moveToNext()){  String uid = cursor.getString(0);  int number = cursor.getInt(1);  int counts = cursor.getInt(2);  records.add(getString(R.string.*txvMessage\_text*,  uid, number, counts));  }  cursor.close();  db.close();  ListView lsvRecord = findViewById(R.id.*lsvRecord*);  ArrayAdapter adapter = new ArrayAdapter(this,  android.R.layout.simple\_list\_item\_1,  records);  //填写ListView组件填充数据的方法名  lsvRecord.setAdapter(adapter); } |

### 考察SharedPreference和SQLite数据库这两种存取数据方式的使用，以及SQLiteOpenHelper类的使用。

|  |
| --- |
| Java private void saveDataBySP(Record record){  SharedPreferences sp = this.getSharedPreferences("record",MODE\_PRIVATE);  SharedPreferences.Editor editor = sp.edit();  editor.putString("userid", record.getUserid());  editor.putInt("target", record.getTargetNumber());  editor.putInt("count", record.getCount());  editor.apply();  } |

|  |
| --- |
| Java private void saveDataBySQlite(Record record) {  MyDBHelper dbHelper = new MyDBHelper(this, "GNDB", null, 1);  SQLiteDatabase db = dbHelper.getWritableDatabase();  ContentValues recordValues = new ContentValues();  recordValues.put("userid", record.getUserid());  recordValues.put("number", record.getTargetNumber());  recordValues.put("count", record.getCount());  db.insert("RECORD",null, recordValues); } |

|  |
| --- |
| Java private void showDataBySP() {  SharedPreferences sp = this.getSharedPreferences("record", MODE\_PRIVATE);  String userid = sp.getString("userid","");  int target = sp.getInt("target", 0);  int count = sp.getInt("count", 0);  TextView txvMessage = findViewById(R.id.*txvMessage*);  txvMessage.setText(getString(R.string.*txvMessage\_text*,  userid, target, count)); } |

|  |
| --- |
| Java private void showDataBySQLite() {  ArrayList<String> records = new ArrayList<String>();  MyDBHelper dbHelper = new MyDBHelper(this, "GNDB", null, 1);  SQLiteDatabase db = dbHelper.getReadableDatabase();  Cursor cursor = db.query("RECORD",  new String[]{"userid", "number", "count"},  null, null, null, null,null);  while(cursor.moveToNext()){  String uid = cursor.getString(0);  int number = cursor.getInt(1);  int counts = cursor.getInt(2);  records.add(getString(R.string.*txvMessage\_text*,  uid, number, counts));  }  cursor.close();  db.close();  ListView lsvRecord = findViewById(R.id.*lsvRecord*);  ArrayAdapter adapter = new ArrayAdapter(this,  android.R.layout.simple\_list\_item\_1,  records);  lsvRecord.setAdapter(adapter); } |

SQLiteOpenHelper类

|  |
| --- |
| Java **public class** MyDBHelper **extends** SQLiteOpenHelper {   **public** MyDBHelper(@Nullable Context context, @Nullable String name, @Nullable SQLiteDatabase.CursorFactory factory, **int** version) {  **super**(context, name, factory, version);  }   @Override  **public void** onCreate(SQLiteDatabase sqLiteDatabase) {  String sql=**"create table RECORD (\_id INTEGER PRIMARY KEY AUTOINCREMENT, "** + **"userid TEXT, "** + **"number INTEGER, "** +**"count INTEGER);"**;  sqLiteDatabase.execSQL(sql);  } } |

## ExampleService:（3-4个空）

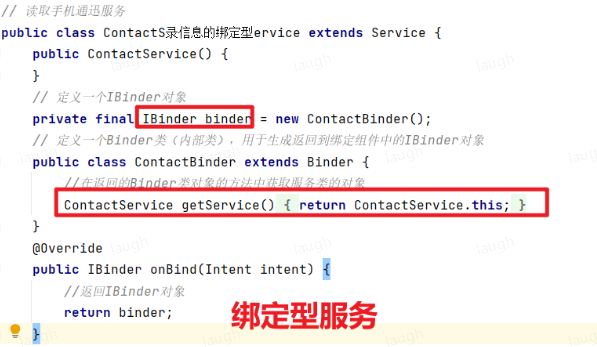
### 考察启动两种类型服务组件的方法

|  |
| --- |
| Java // 在启动型服务中运行需要长时间执行的代码 Intent intent = new Intent(this, MyIntentService.class); intent.putExtra("second", second); startService(intent); |

|  |
| --- |
| Java //在启动Activity时绑定服务 Intent intent = new Intent(this, ContactService.class); bindService(intent, connection, Context.*BIND\_AUTO\_CREATE*); |

### 考察绑定型服务使用的基本步骤。

|  |
| --- |
| Java // 定义一个服务连接类的对象访问绑定服务 private ServiceConnection connection = new ServiceConnection() {  @Override  public void onServiceConnected(ComponentName componentName, IBinder iBinder) {  ContactService.ContactBinder binder =  (ContactService.ContactBinder)iBinder;  service = binder.getService();  bound = true;  } |



## ExampleBroadcast：

**主要考察广播接收器组件的基本结构，广播的动态注册与取消注册的方法。**

|  |
| --- |
| Java  public class SmsReceiver extends BroadcastReceiver {  @RequiresApi(api = Build.VERSION\_CODES.M)  @Override  public void onReceive(Context context，Intel  Bundle bundle = intent.getExtras();  SmsMessage msg = null; |

|  |
| --- |
| Java private void send(){  MyReceiver1 myReceiver1 = new MyReceiver1();  IntentFilter intentFilter = new IntentFilter();  intentFilter.addAction("com.example.broad");  intentFilter.setPriority(10);  //填写动态注册广播的方法名  registerReceiver(myReceiver1, intentFilter);  MyReceiver2 myReceiver2 = new MyReceiver2();  intentFilter.setPriority(50);  registerReceiver(myReceiver2, intentFilter);   Intent intent = new Intent();  intent.setAction("com.example.broad");  intent.putExtra("msg","this is a broadcast!");  //sendBroadcast(intent);  sendOrderedBroadcast(intent, null); } |

|  |
| --- |
| Java  protected void onStop() {  super.onStop();  //填写注销广播的方法名  unregisterReceiver(receiver);  } |