# Android课后题答案

## 第1章 Android基础入门

一、填空题

1、dex

2、@color

3、AndroidManifest.xml

4、LogCat

二、判断题

1、对 2、对 3、对 4、错 5、错 6、错

三、选择题

1、C 2、C 3、D 4、C 5、A

四、简答题

**1. 简述如何搭建Android开发环境**。

答：Android开发环境的搭建分为3步，具体如下：

1. Android Studio工具的下载、安装、以及相关配置。在安装过程中选择该工具的安装路径，以及配置Android Studio相关设置，如Android Studio工具的主题。
2. 创建模拟器。首先单击ADV Manager的图标，在弹出框中选择模拟设备，接着下载Android SDK System Image，下载完成后创建并打开模拟设备。
3. 下载SDK。分别在Default Settings窗口中下载SDK版本和SDK Tools工具。

**2. 简述Android源代码的编译过程**

答：Java源文件经过JDK编译器编译为class文件后，Dalvik虚拟机中的Dx工具会将部分class文件转换成dex文件，dex文件还会在Dalvik虚拟机中进一步优化成odex文件。

**3.简述Android系统架构包含的层次以及各层的特点**

答：Android系统架构从高到低分为四层，分别为依次是应用程序层（Applications）、应用程序框架层（Application Framework）、核心类库（Libraries）和Linux内核（Linux Kernel），各层的特点具体如下：

1. 应用程序层：一个核心应用程序的集合，安装在手机中的应用程序都属于这一层。
2. 应用程序架构层：主要提供了构建应用程序时用到的各种API。例如活动管理器（Activity Manager）。
3. 核心类库：主要包含了系统库和Android运行环境。
4. Linux内核：他为Android设备的各种硬件提供了底层的驱动，如：显示驱动。

## 第2章 Android常见界面布局

一、填空题

1、ViewGroup

2、LinearLayout

3、TableRow

4、RelativeLayout

5、int

二、判断题

1、对 2、错 3、对 4、对 5、对

三、选择题

1、A 2、C 3、A 4、D 5、C

四、简答题

**1. 列举Android中的常用布局，并简述他们各自的特点。**

Android中有五种常用布局，分别为RelativeLayout（相对布局）、LinearLayout（线性布局）、FrameLayout（帧布局）、TableLayout（表格布局）、ConstraintLayout（约束布局），他们的特点具体如下：

1. RelativeLayout（相对布局）：该布局是通过相对位置的方式指定该布局内子控件的位置。
2. LinearLayout(线性布局)：该布局可通过android:orientation属性指定该布局内的子控件水平和竖直排列。
3. TableLayout（表格布局）：采用行、列的形式来管理控件。他通常搭配TableRow布局使用，一个TableRow代表一行，在TableRow中添加的控件代表一列。
4. FrameLayout（帧布局）：该布局会在屏幕上创建一块空白区域，添加到该区域中的每个子控件占一帧，这些帧会一个一个叠加在一起，后加入的控件会叠加在上一个控件上层。默认情况下，帧布局中的所有控件会与左上角对齐。
5. ConstraintLayout（约束布局）：该布局是Android2.2新添加的布局，他适用于可视化的方式编写界面布局，他有相对定位、居中定位和倾向、Chain等性质。

五、编程题

## 第3章 Android常见界面控件

一、判断题

1、对 2、错 3、错 4、对 5、对

二、选择题

1、B 2、C 3、D 4、A 5、C 6、B 7、A 8、A

三、简答题

**1. 简述ListView与RecyclerView的区别**

答：ListView与RecyclerView的区别具体如下：

1、展示效果：RecyclerView控件可以通过LayoutManager类实现横向或竖向的列表效果、瀑布流效果和GridView效果，而ListView控件只能实现竖直的列表效果。

2、适配器：RecyclerView控件使用的是RecyclerView.Adapter适配器，该适配器将BaseAdapter中的getView()方法拆分为onCreateViewHolder()方法和onBindViewHolder()方法，强制使用ViewHolder类，使代码编写规范化，避免了初学者写的代码性能不佳。

3、复用效果：RecyclerView控件复用Item对象的工作由该控件自己实现，而ListView控件复用Item对象的工作需要开发者通过convertView的setTag()方法和getTag()方法进行操作。

4、动画效果：RecyclerView控件可以通过setItemAnimator()方法为Item添加动画效果，而ListView控件不可以通过该方法为Item添加动画效果。

**2. 简述实现Button按钮的点击事件的方式有哪几种？**

答：实现Button按钮的点击事件一共有3种方式，具体如下：

1. 在布局文件中指定onClick属性的方式设置点击事件。

2、使用匿名内部类的方式设置点击事件。

3、通过为Activity实现OnClickListener接口的方式设置点击事件。

**3.简述AlertDialog对话框的创建过程**

1、 调用AlertDialog的静态内部类Builder创建AlertDialog.Builder的对象。

2、 调用AlertDialog.Builder的setTitle()和setIcon()方法分别设置AlertDialog对话框的标题名称和图标。

3、 调用AlertDialog.Builder的setMessage()、setSingleChoiceItems()或者setMultiChoiceItems()方法设置AlertDialog对话框的内容为简单文本、单选列表或者为多选列表。

4、 调用AlertDialog.Builder的setPositiveButton()和setNegativeButton()方法设置AlertDialog对话框的确定和取消按钮。

5、 调用AlertDialog.Builder的create()方法创建AlertDialog对象。

6、 调用AlertDialog对象的show()方法显示该对话框。

7、 调用AlertDialog对象的dismiss()方法取消该对话框。

四、编程题

## 第4章 程序活动单元Activity

一、填空题

1、singleInstance

2、startActivityForResult()

3、IntentFilter

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

5、finish()

二、判断题

1、对 2、错 3、对 4、错 5、对

三、选择题

1、C 2、B 3、B 4、A 5、B

四、简答题

**1. 简述Activity的生命周期的方法有哪些**

答：Activity一共有7个方法，这些方法和调用的时机具体如下：

1、onCreate()：Activity创建时调用，通常做一些初始化设置。

2、onStart()：Activity即将可见时调用。

3、onResume()：Activity获取焦点时调用**。**

4、onPause()：当前Activity被其他Activity覆盖或屏幕锁屏时调用。

5、onStop()：Activity对用户不可见时调用。

6、onDestroy()：Activity销毁时调用。

7、onRestart()：Activity从停止状态到再次启动时调用。

**2. 简述Activity的四种启动模式及其特点**

答：Activity的四种启动模式分别为standard、singleTop、singleTask和singleInstance，这些模式的特点具体如下：

1. standard：每启动一个Activity就会在栈顶创建一个新的实例。
2. singleTop：当被启动的Activity位于栈顶时，复用该Activity实例。
3. singleTask：当被启动的Activity在栈中存在实例时，会直接复用此Activity实例，并把当前Activity上面的所有实例弹出栈。
4. singleInstance：会启动一个新的任务栈来管理该Activity。

**3. 简述Activity、Intent、IntentFilter的作用**

答：Activity表示活动，他是Android的四大组件之一，Intent表示意图，IntentFilter表示过滤器，他们的作用具体如下：

(1) Activity是一个负责与用户交互的组件，每个Android应用中都会用Activity来显示界面以及处理界面上一些控件的事件。

(2) Intent被称为意图，是程序中各组件间进行交互的一种重要方式，它不仅可以指定当前组件要执行的动作，还可以在不同组件之间进行数据传递。

(3) IntentFilter为过滤器，当发送一个隐式Intent后，Android系统会将它与程序中的每一个组件的过滤器进行匹配，匹配属性有 action、data、category，需要这三个属性都匹配成功才能唤起相应的组件。

## 第5章 数据存储

一、判断题

1、对 2、对 3、错 4、对 5、错 6、对

二、选择题

1、C 2、B 3、B 4、C 5、C 6、D

三、简答题

**1.简述数据库事物的4个基本要素要素**

答：事务的操作比较严格，他必须满足ACID，ACID是指数据库事务正确执行的四个基本要素的缩写，这些要素包括原子性（Atomicity）、一致性（Consistency）、隔离性（Isolation）、持久性（Durability），接下来针对这四个基本要素进行详细解释。

* 原子性：表示事务是一个不可再分割的工作单位，事务中的操作要么全部成功，要么全部失败回滚。
* 一致性：表示事务开始之前和结束之后，数据库的完整性没有被破坏。也就是说数据库事务不能破坏关系数据的完整性以及业务逻辑上的一致性。
* 隔离性：表示并发的事务是相互隔离的，也就是一个事务内部的操作都必须封锁起来，不会被其他事务影响到。
* 持久性：表示事务一旦提交后，该事务对数据做的更改便持久保存在数据库中，并不会被回滚，即使出现了断电等事故，也不会影响数据库中的数据。

**2. 简述Android数据存储的方式**

答：Android平台提供的五种数据存储方式，分别为文件存储、SharedPreferences、SQLite数据库、ContentProvider和网络存储，这些存储方式的特点如下。

* 文件存储：Android提供了openFileInput()和openFileOutput()方法来读取设备上的文件，其读取方式与Java中I/O程序是完全一样的。
* SharedPreferences：这是Android提供的用来存储一些简单的配置信息的一种机制，他采用了XML格式将数据存储到设备中。通常情况下，我们使用SharedPreferences存储一些应用程序的各种配置信息，如用户名、密码等。
* SQLite数据库：SQLite是Android自带的一个轻量级的数据库，他运算速度快，占用资源少，还支持基本SQL语法，一般使用他作为复杂数据的存储引擎，可以存储用户信息等。
* ContentProvider：Android四大组件之一，主要用于应用程序之间的数据交换，他可以将自己的数据共享给其他应用程序使用。
* 网络存储：需要与Android网络数据包打交道，将数据存储到服务器上，通过网络提供的存储空间来存储/获取数据信息。

四、编程题

**1.使用SQLite数据库的事务操作，编写一段模拟银行转账的逻辑代码。**

PersonSQLiteOpenHelper helper = new PersonSQLiteOpenHelper (getApplication());

//获取一个可读写的SQLiteDataBase对象

SQLiteDatabase db = helper.getWritableDatabase();

// 开始数据库的事务

db.beginTransaction();

try {

//执行转出操作

db.execSQL("update person set account = account-1000 where name =?",

new Object[] { "张三" });

//执行转入操作

db.execSQL("update information set account = account +1000 where name =?",

new Object[] { "王五" });

//标记数据库事务执行成功

db.setTransactionSuccessful();

}catch (Exception e) {

Log.i("事务处理失败", e.toString());

} finally {

db.endTransaction(); //关闭事务

db.close(); //关闭数据库

}

## 第7章 使用内容提供者共享数据

一、判断题

1、对 2、对 3、对 4、错 5、对

二、选择题

1、C 2、B 3、B 4、C 5、D

三、简答题

**1. 简述内容提供者的工作原理**

答：假设B程序需要操作A程序数据库中的数据，一般需要A程序使用ContentProvider

暴露数据，才能被其他程序操作。B程序通过ContentResolver操作A程序暴露出来的数据，而A程序会将操作结果返回给ContentResolver，然后ContentResolver再将操作结果返回给B程序。

**2. 简述内容观察者的工作原理**

使用ContentObserver观察A程序的数据时，首先要在A程序的ContentProvider中调用ContentResolver的notifyChange()方法。调用此方法后，当B程序操作A程序中的数据时，A程序会向“消息中心”发送数据变化的消息，此时C程序会观察到“消息中心”的数据有变化，会触发ContentObserver的onChange()方法。

四、编程题

## 第8章 广播机制

一、填空题

1、BroadcastReceiver

2、 动态注册和静态注册

二、判断题

1、对 2、对 3、错 4、对 5、错

三、选择题

A、 BC 2、ABCD

四、简答题

**1. 广播机制的实现过程**

答：Android中的广播使用了观察者模式，即基于消息的发布/订阅事件的模式。广播发送者和接收者分别处于观察者模式中的消息发布和订阅两端。广播机制的实现过程具体如下：

1. 广播接收者通过Binder机制在AMS(Activity Manager Service)中进行注册。

2. 广播发送者通过Binder机制向AMS发送广播。

3. AMS查找符合相应条件（IntentFilter/Permission）的广播接收者，将广播发送到相应的消息循环队列中。

4. 执行消息循环时获取到发送的广播，然后回调广播接收者中的onReceive()方法并在该方法中进行相关处理。

**2. 简述有序广播和无序广播的区别**

答：广播分为有序广播和无序广播，他们有以下几点的区别。

(1) 发送广播时，使用的方法不同。有序广播使用sendOrderedBroadcast()发送广播，而无序广播使用sendBroadcast()方法发送广播。

(2) 广播接收者执行的顺序

1. 有序广播的接收者是顺序执行的。

有序广播按照广播接收者声明的优先级别被依次接收。当在高级别的广播接收者逻辑执行完毕之后，广播才会继续传递。当优先级相同时，先注册的广播接受者优先执行。

1. 无序广播是完全异步执行的。

当发送无序广播时，所有监听这个广播的广播接收者都会接收到此广播消息，但接收和执行的顺序不确定。

（2）拦截广播

有序广播的接收者可拦截广播。如果优先级较高的广播接收者将广播终止，那么广播将不再向后传递。而无序广播则不能被拦截。

（3）效率

有序广播的效率比无序广播低。

五、编程题

## 第9章 服务

一、填空题

1、unBindService() 2、本地服务 远程服务 3、AIDL

二、判断题

1、错 2、错 3、错 4、错 5、错

三、选择题

1、C 2、D 3、D

四、简答题

**1.简述Service的两种启动方式**

答：Service的启动方式分别可以调用startService()、bindService()方法，这两个启动方式的区别如下所示：

1. 生命周期

* startService():使用该方法开启Service时，执行的生命周期方法依次为onCreate()、onStartCommand()、onDestroy()。
* bindService()：使用该方法开启Service时，执行的生命周期方法依次为：onCreate()、onBind()、onUnbind()、onDestroy()。

1. 停止服务的方法

* startService():调用stopSelf()、stopService()方法停止服务。
* bindService():调用unbindService()方法停止服务。

1. 组件的关联

* startService():当一个组件通过此方法开启服务时，服务与开启该组件没有关联，即使开启服务的组件被销毁，服务依旧运行。
* bindService():当一个组件通过此方法开启服务时，服务会与该组件绑定，组件一旦被销毁，该服务也会被销毁。

**2. 简述Service的生命周期？**

答：使用不同的方式启动服务，其生命周期会不同。开启服务的方法分别为startService()、bindService()，当通过startService()方法启动服务时，执行的生命周期方法依次为onCreate()、onStartCommand()、onDestroy()。当通过bindService()方法启动服务时，执行的生命周期方法依次为onCreate()、onBind()、onUnbind()、onDestroy()。其生命周期方法的具体含义如下：

* onCreate ():第一次创建服务时执行的方法。
* onStartCommand():调用startService()方法启动服务时执行的方法。
* onBind():调用bindService()方法启动服务时执行的方法。
* onUnbind():调用unBindService()方法断开服务绑定时执行的方法。
* onDestory():服务被销毁时执行的方法。

五、编程题

## 第10章 Android事件处理

1. 填空题
2. onKeyDown()
3. GestureDetector
4. OnClickListener
5. 判断题

1、错 2、对 3、错 4、对 5、对

1. 选择题

1、ABCD 2、ABCD 3、ABCD

1. 简答题

**1.简述Handler消息机制的原理**

答： Handler消息处理机制主要包括四个关键对象，分别为Handler、Message、MessageQueue、Looper。具体功能如下：

1. Handler主要用于发送消息和处理消息。
2. Message是在线程之间传递的消息，它可以在内部携带少量的信息，用于在不同线程之间交换数据。
3. MessageQueue为消息队列，它主要用于存放通过Handler发送的消息。
4. Looper主要通过调用loop()方法，不断的从MessageQueue中获取消息，并分发到Handler中。

Handler消息机制是通过上述四个关键对象的配合使用而完成的。在UI线程中创建Handler对象，并通过该对象的sendMessage()方法发送消息到MessageQueue中，接着通过Looper调用loop()方法不断的从MessageQueue中获取消息，并分发到Handler中，最终通过Handler的handleMessage()方法处理获取的消息。

**2.简述事件监听处理的实现原理**

答：事件监听处理主要涉及三个对象，分别是Event Source（事件源）、Event（事件）、Event Listener（事件监听器），这三个对象的具体介绍如下：

1. Event Source（事件源）：事件发生的场所，通常是指各个组件。
2. Event（事件）：封装了界面组件发生的特定事情。
3. Event Listener（事件监听器）：负责监听事件源所发生的事件，并对各种事件做出相应的响应。

事件监听的处理主要围绕上述3个对象实现，首先为事件源设置监听器，用于监听用户操作，当用户执行该操作时，会触发事件源的监听器并生成对应的事件对象，接着将生成的事件对象作为参数传递给事件监听器，事件监听器会对该事件对象进行判断，执行对应的事件处理器。

## 第11章 网络与数据处理

1. 填空题
2. URLConnection
3. WebKit
4. JSONObject
5. 判断题

1、对 2、对 3、对 4、对 5、对 6、错

1. 选择题

1、ABCD 2、ABD

1. 简答题

**1. 简述使用HttpURLConnection 访问网络的步骤。**

答：使用HttpURLConnection访问网络的步骤如下：

1. 创建URL对象。
2. 调用URL对象的openConnection()方法获取HttpURLConnection对象。
3. 调用setRequestMethod()方法设置http请求的方式。
4. 通过setConnectTimeout()方法设置连接的超时时间。
5. 调用getInputStream()方法获取服务器返回的输入流。
6. 最后调用disconnect()方法关闭http连接。
7. 编程题

**1. 请写出使用JSONArray类解析JSON数据的主要逻辑代码。JSON数据如下所示。**

[{"name":"LiLi","score":"95"},{"name":"LiLei","score":"99"},

{"name":"王小明","score":"100"},{"name":"LiLei","score":"89"}]

答：使用JSONArray类解析JSON数据的逻辑代码如下：

public void getJson(){

String json = "["

+ "{" + "\"" + "name" + "\"" + ":" + "\"" + "LiLi" + "\"" + ","+"\"" + "score" + "\"" + ":" + "\"" + "95" + "\""+ "},"

+ "{" + "\"" + "name" + "\"" + ":" + "\"" + "LiLei" + "\"" + ","+"\"" + "score" + "\"" + ":" + "\"" + "99" + "\""+ "},"

+ "{" + "\"" + "name" + "\"" + ":" + "\"" + "李小明" + "\"" + ","+"\"" + "score" + "\"" + ":" + "\"" + "100" + "\""+ "},"

+ "{" + "\"" + "name" + "\"" + ":" + "\"" + "王小敏" + "\"" + ","+"\"" + "score" + "\"" + ":" + "\"" + "89" + "\""+ "}"+ "]";

JSONArray jsonArray = null;

try {

jsonArray = new JSONArray(json);

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

JSONObject jsonObj = jsonArray.getJSONObject(i);

String name = jsonObj.optString("name");

int score = jsonObj.optInt("score");

}

} catch (JSONException e) {

e.printStackTrace();

}

}

## 第13章 图形图像处理

1. 判断题

1、对 2、对 3、对

1. 选择题

1、ABC 2、ABCD 3、ABCD 4、B 5、C

1. 简答题

**1.简述逐帧动画的工作原理。**

答：逐帧(Frame)动画的工作原理非常简单，类似与动画片的工作原理，他按照事先准备好的静态图片顺序进行播放，并利用人眼的“视觉暂留”原理，使用户产生动画的错觉。

## 第14章 多媒体应用开发

1. 判断题

1、对 2、对 3、错 4、对 5、对

1. 选择题

1、ABC 2、ABC 3、D

1. 简答题
2. **简述使用MediaPlayer播放音频的步骤。**

答：使用MediaPlayer播放音频包含以下几个步骤：

1. 创建MediaPlayer对象
2. 调用setAudioStreamType()设置音频的类型
3. 设置数据源
4. 调用start()方法播放音频文件
5. 调用stop()方法停止播放音频文件
6. 调用release()方法释放MediaPlayer对象占用的资源

2. 简要介绍一下SoundPool使用的场景。

答：SoundPool又称为音频池，他适合在应用程序中播放按键音或者消息提示音等短小的多个音频。

1. 编程题