# 《Android移动应用基础教程》习题答案

# 第1章 Android基础入门

1. 填空题
2. Linux、操作系统
3. 应用程序层、应用程序框架层、核心类库、Linux内核
4. adb start-server
5. java代码文件
6. 打包成.apk文件
7. 判断题

1、× 2、√ 3、√ 4、× 5、√

1. 选择题

1、C 2、ABD 3、B 4、C 5、B

四、简答题

1、Android体系结构总共包含四层，分别是：

* 应用程序层： 设备上安装的软件应用都属于这一层
* 应用程序框架层 ：包含应用API
* 核心类库：包含系统库和运行环境，系统库包含了底层C代码；运行环境包含了Java的核心库和Dalvik虚拟机
* Linux内核：提供Android的底层驱动。

2、SDK包含了Android的API源代码、各种工具、示例工程、用到的各种资源模板等。

# 第2章 Android UI开发

1. 填空题
2. RelativeLayout、LinearLayout、TableLayout、GirdLayout、FrameLayout、AbsoluteLayout
3. android:layout\_alignParentLeft
4. 水平、竖直
5. RelativeLayout
6. 显示全部信息、显示调试信息、显示一般信息、显示警告信息、显示错误信息
7. 判断题

1、√ 2、√ 3、× 4、√ 5、×

1. 选择题

1、D 2、D 3、C 4、D 5、C

1. 简答题

1、使用Toast可以直接调用Tosast的静态方法：

Toast.makeText(this, text, duration).show();

也可以创建出Toast对象，如：

Toast toast = new Toast(this);

toast.setText("提示信息");

toast.show();

2、一共有六种布局，它们的作用分别是：

* RelativeLayout：相对于其他控件或者容器决定控件的位置；
* LinearLayout：使控件以竖直或者水平方向排列；
* TableLayout：使控件以表格形式排列；
* GirdLayout：能使控件交错显示，能够避免因布局嵌套对设备性能的影响，更利于自由布局的开发；
* FrameLayout：使控件按照创建顺序在屏幕的左上角重叠显示；
* AbsoluteLayout：通过绝对的坐标控制控件摆放的位置。

# 第3章 Activity

1. 填空题
2. 运行状态、暂停状态、停止状态
3. standard、singleTop、singleTask、singleInstance
4. 隐式Intent、显示Intent
5. onRestart()
6. startActivityForResult()
7. 判断题

1、√ 2、× 3、√ 4、× 5、√

1. 选择题

1、A 2、D 3、D 4、C 5、C

四、简答题

1、四种启动模式分别是：

* Standard：每当启动一个新的Activity，它就会进入任务栈，并处于栈顶的位置，对于使用standard模式的Activity，系统不会判断该Activity在栈中是否存在，每次启动都会创建一个新的实例。
* SingleTop：与standard类似，不同的是，当启动的Activity已经位于栈顶时，则直接使用它不创建新的实例。如果启动的Activity没有位于栈顶时，则创建一个新的实例位于栈顶。
* SingleTask：创建的Activity只存在一个实例。已经存在则直接使用该实例，并将当前Activity之上的所有Activity出栈，如果没有发现则创建一个新的实例。
* singleInstance：启动一个新的任务栈管理Activity

2、Activity三种状态分别是：

* 运行状态：该状态下可以对界面进行操作
* 暂停状态：界面可见但不具有焦点无法点击
* 停止状态：界面不可见，一般在停止状态时保存数据以防止程序被系统强制杀死。

# 第4章 数据存储

1. 填空题
2. 可保持或传输
3. 内存、SD卡
4. DOM、SAX、PULL
5. 配置参数
6. 文件、Sharedpreferences、Sqlite、网络、ContentProvider
7. 判断题

1、√ 2、√ 3、× 4、× 5、√

1. 选择题

1、C 2、D 3、D 4、BD 5、C

四、简答题

1、五种存储方式各自的特点分别是：

* 文件存储：可以存储较大文件，以IO流形式传输数据。
* Sqlite：占用内存小，所有文件都存放在单一文件中。
* 网络：通过网络将数据传输到服务器存储，避免了客户端的安全隐患。
* Sharedpreference：使用方便，可以存储配置信息，以键值对的形式存于内存中
* ContentProvider：提供自身暴漏的数据给第三方应用使用，统一了数据访问格式，以数据库形式保存数据。

2、使用Sharedpreference储存数据时因为它本身没有写文件的功能，因此需要使用Editor对象将数据存入内存，比如：

SharedPreferences sp = getSharedPreferences("config", 0);

Editor editor = sp.edit();

editor.putString("a", "a");

editor.commit();

# 第5章 SQLite数据库

1. 填空题
2. BaseAdapter、SimpleAdapter、ArrayAdapter
3. SQLiteOpenHelper
4. onCreate()、onUpgrade()
5. Cursor、close
6. Id
7. 判断题

1、× 2、× 3、√ 4、× 5、×

1. 选择题

1、AC 2、B 3、B 4、C 5、B

四、简答题

1、首先创建一个类继承SQLiteOpenHelper，重写onCreate()方法并在该方法中创建表，使用创建出的SQLiteOpenHelper的子类对象的getWritableDatabase()方法获得一个可读写的数据库对象。

2、四个抽象方法分别是：

* getCount：得到Item的总数
* getItem：根据position得到某个Item的对象
* getItemId：根据position得到某个Item的id
* getView：得到相应position对应的Item视图，position当前Item的位置，convertView复用的View对象。

# 第6章 内容提供者

1. 填空题
2. UriMatcher
3. onChange()
4. insert、delete、update、query
5. 保存、检索
6. ContentResolver
7. 判断题

1、× 2、√ 3、√ 4、× 5、√

1. 选择题

1、ABC 2、C 3、ACD 4、ABD 5、C

四、简答题

1、通过ContentProvider共享数据统一了数据访问方式，使用起来更规范，通过数据库存储并指定了URI，只有通过特定URI才能访问数据，使数据更安全。

2、ContentProvider可以共享自己的数据给外部应用访问，要访问ContentProvider暴漏的数据就要用到ContentResolver。而ContentObserver就相当于中间人的角色，它可以事实监听ContentProvider的数据是否发生变化，如果发生变化就会触发onChange()方法，Contentresolver可以在onChange()方法中查询的哪些数据发生了变化再对数据进行操作。