二、简答

1、简述Android应用开发程序中的四个最主要组件及作用。

四种基本组件:Activity、Service、BroadcaseReceiver和ContentProvider

Activity是Android程序的呈现层，显示可视化的用户界面，并接收与用户交互所产生的界面事件，用于提示用户程序已经正常启动。

Service一般用于没有用户界面，但需要长时间在后台运行的应用。BroadcaseReceiver是用来接受并响应广播消息的组件。

ContentProvider是Android系统提供的一种标准的共享数据的机制，应用程序可以通过ContentProvider访问其他应用程序的私有数据。

2、描述Android平台体系结构的层次划分，并说明各个层次的作用。

Android采用了软件堆层的架构，共分为四层： Linux内核、中间件层、应用程序框架层和应用程序层。

（1）Linux内核提供由操作系统内核管理的底层基础功能。

（2）中间件层包含函数库和Android运行时

（3）应用程序框架提供了Android平台基本的管理功能和组件重用机制。

（4）应用程序提供了一系列核心应用程序。

3、简述Activity事件回调函数的作用和调用顺序。

Activity事件回调函数具体分为Activity生命周期的事件回调函数和Activity状态保存和恢复函数的事件回调函数，Activity生命周期

的事件回调函数的作用主要是为了让Activity程序了解自身状态的变化；Activity状态保存和恢复函数的事件回调函数的作用主要是保存或恢复Activity的状态信息。

Activity事件的调用顺序是：

(1)onCreate,完全生命周期开始，初始化 Activity；

(2)onStart,可视生命周期开始，对用户界面进行必要的更改；

(3)onRestoreInstanceState,恢复onSaveLnstanceState保存的用户界面信息；

(4)onResume,活动生命周期开始，保存界面信息；

（5）onSaveInstanceState,在 onResume后，保存界面信息；

（6）onRestart,重新进入可视生命周期前，载入界面所需要的更改信息；

（7）onPause,活动生命周期结束，保存持久地数据或释放占用的资源；

（8）onStop,可视生命周期结束，保存持久地数据或释放占用的资源；

（9）onDestory,完全生命周期结束，释放资源。

4、简述5种界面布局及其特点。

6种界面布局分别为：线性布局、框架布局、表格布局、相对布局、绝对布局和网格布局。

（1）线性布局的子元素可垂直或者水平排列，但是每一列/行中，只能有一个界面元素。

（2）框架布局是只存放一个元素的空白空间，且位置只能是空白空间的左上角，如果有多个子元素，后放置的子元素会遮挡先放置的子元素。

（3）表格布局将屏幕划分网格，把界面元素添加到网格中，支持嵌套，也可添加其他的界面布局。

（4）相对布局中的界面元素的位置是通过与其他的元素的相对位置确定的，具有灵活性。

（5）绝对布局的界面元素的位置是通过坐标确定的，他是不推荐使用的一种布局。

（6）网格布局将用户界面划分为网格，界面元素可随意摆放在这些网格中，界面元素可以占用多个网格的，使界面设计更为灵活。

5、简述Intent的定义和用途。

Intent是一个动作的完整描述，包含了动作的产生组件、接收组件和传递的数据信息。Intent为Activity、Service和BroadcastReceiver等组件提供交互能力，将一个组件的数据和动作传递给另一个组件。Intent的一个最常见的用途就是启动Activity和Service；另一个用途是在Android系统上发布广播消息，广播消息可以是接收到特定数据或消息，也可以是手机的信号变化或电池的电量过低等信息。

6、简述Intent过滤器的定义和功能。

Intent过滤器是一种根据Intent中的动作（Action）、类别（Categorie）和数据（Data）等内容，对适合接收该Intent的组件进行匹配和筛选的机制。

Intent过滤器的作用是匹配数据类型、路径和协议，还包括可以用来确定多个匹配项顺序的优先级。每个Intent过滤器可以定义多个<category>标签，程序开发人员可以使用自定义的类别，或使用Android系统提供的类别。

7、什么是Service以及描述下它的生命周期。

Android Service是运行在后台的代码,不能与用户交互,可以运行在自己的进程,也可以运行在其他应用程序进程的上下文里.需要通过某一个Activity或者其他Context对象来调用。

Service生命周期包括完全生命周期和活动生命周期，还有三个回调函数，分别是onCreate()、onStart()和onDestroy()如图：

完整生命周期从onCreate()开始到onDestroy()结束，在onCreate()中完成Service的初始化工作，在onDestroy()中释放所有占用的资源

活动生命周期从onStart()开始，但没有与之对应的“停止”函数，因此可以粗略的认为活动生命周期是以onDestroy()标志结束。

8、简述Service的基本原理和用途。

Service能够长期在后台运行，适用于无需用户干预，且规则或长期运行的后台功能。首先，因为Service没有用户界面，更加有利于降低系统资源的消耗，而且Service比Activity具有更高的优先级，因此在系统资源紧张时，Service不会被Android系统优先终止。即使Service被系统终止，在系统资源恢复后Service也将自动恢复运行状态，因此可以认为Service是在系统中永久运行的组件。Service除了可以实现后台服务功能，还可以用于进程间通信（Inter Process Communication，IPC），解决不同Android应用程序进程之间的调用和通讯问题。

9、Content Provider是如何实现数据共享的？

在创建ContentProvider前，首先要实现底层的数据源，数据源包括数据库、文件系统或网络等，然后继承ContentProvider类中实现基本数据操作的接口函数，包括添加、删除、查找和更新等功能

调用者不能直接调用ContentProvider的接口函数，而需要使用ContentResolver对象。