第一章：

1、Android四大组件：Activity、Service、ContentProvider、BroadcastReceiver

1. Intent（意图）：

Intent用来描述应用程序的功能。

Intent是利用消息实现应用程序间的交互机制，能够在程序运行的过程中连接两个不同的组件。

Intent描述了应用中一次操作的动作、数据以及附加数据，向Android表达某种请求或者意愿，Android系统会根据Intent描述的内容来选择适当的组件来响应，并将Intent传递给该组件，完成组件的调用。

1. Service（服务）：

服务是Android应用程序中具有较长的生命周期但是没有用户界面的程序。

Service运行在后台，并且可以与其他程序进行交互。

Service跟Activity的级别差不多，但是不能独立运行，需要通过某一个Activity来调用。

Android应用程序的生命周期是由Android系统来决定的，不由具体的应用程序线程来控制。

如果应用程序要求在没有界面显示的情况还能正常运行（要求有后台线程，而且直到线程结束，后台线程才会被系统回收），此时就需要用到Service。

1. 小结：

Android是Google公司推出的开源手机操作系统

Android上的应用程序开发使用Java语言

Android平台采用了软件栈的架构，主要分为活动、意图、服务和内容提供器四部分

活动（Activity）是最基本的Android应用程序组件

意图（Intent）利用消息实现应用程序间的交互机制

服务（Service）是Android应用程序中具有较长的生命周期但是没有用户界面的程序

内容提供器（Content Provider）提供了一种多应用间数据共享的方式

1. ：
2. Activity：

Activity（活动）是Android应用程序中最基本的组成单位。

Activity主要负责创建显示窗口，一个Activity对象通常就代表了一个单独的屏幕。

Activity是用户唯一可以看得到的组件，用来与用户进行交互的。

Activity是通过一个Activity栈来进行管理，当前显示的Activity被放到栈顶 。

1. Activity生命周期：

Activity具有生命周期，在生命周期的过程中共有四种状态：

激活或者运行状态：此时Activity运行在屏幕的前台；

暂停状态：此时Activity失去了焦点但是仍然对用户可见，例如在该Activity上遮挡了一个 透明的或者非全屏的Activity；

停止状态：此时Activity被其他Activity完全覆盖；

终止状态：此时Activity将会被系统销毁（清理出内存）。

1. Activity的用来维护生命周期的七个方法：

onCreate()、onStart()、onRestart()、onResume()、onPause()、onStop()、onDestroy()

1. Activity生命周期：
2. 完整生命期：从Activity最初调用onCreate()方法到最终调用onDestroy()方法的这个过程称为完整生命周期。
3. 可见生命周期：从Activity调用onStart()方法开始，到调用对应的onStop()方法为止的这个过程称为可见生命周期。在这段时间内，用户可以在屏幕上看到这个Activity，尽管并不一定是在前台显示，也不一定可以与其交互。

（3）前台生命周期：从Activity调用onResume()方法开始，到调用对应的onPause()方法为止的这个过程称为前台生命周期，这段时间当前的Activity处于其他所有Activity的前面，且可以与用户交互。

5、日志Log：

静态方法 级别分类 功能说明

v() verbose，对应DDMS的Log Cat视图中的V 最低级别，所有信息

d() debug，对应DDMS的Log Cat视图中的D 调试信息

i() info，对应DDMS的Log Cat视图中的I 一般信息

w() warn，对应DDMS的Log Cat视图中的W 警告信息

e() error，对应DDMS的Log Cat视图中的E 错误信息

6、在Android工程中，资源文件分别存放在res和assets两个文件夹中：

res目录存放Android程序能通过R资源类直接访问的资源；

assets目录存放Android程序不能直接访问的资源（原生文件），如MP3文件，必须通过AssetManager类以二进制流的形式读取。

1. Android资源类型：

目录结构 存放的资源类型

res/anim 动画文件

res/drawable 图片文件

res/layout 布局文件

res/xml 任意的XML文件

res/raw 直接复制到设备中的原生文件

res/menu 菜单文件

res/values 各种XML资源文件：

strings.xml：字符串文件

colors.xml：颜色文件

arrays.xml：数组文件

styles.xml：样式文件

dimens.xml：尺寸文件

8、小结：

（1）Activity是Android应用程序中最基本的组成单位

（2）大部分的Android应用中包含多个Activity类

（3）Activity共有四种状态：激活或者运行状态、暂停状态、停止状态、终止状态

（4）控制Activity屏幕横竖屏显示方式，设置参数“android:screenOrientation”为landscape或portrait

（5）Activity被系统销毁时，保存必要信息，保证在Activity恢复时不被丢失

（6）每个Activity类在定义时都必须继承android.app.Activity

（7）android.app.Activity类中的方法定义了Activity完整的生命周期

（8）Android中的资源是指非代码部分，是代码中使用的外部文件

（9）对于字符串资源，在其他资源文件中使用“@string/资源名称”的形式访问，在代码中可通过“R.string.资源名称”的形式访问

（10）对于图片资源，在其他资源文件中使用“@drawable/资源名称”的形式访问，在代码中可通过“R.drawable.资源名称”的形式访问

第三章：用户界面：

1、Android中的界面元素主要由以下几个部分构成：

视图组件（View）

视图容器（ViewGroup）

布局管理（Layout）

1. 视图组件（View）：

View视图组件是用户界面的基础元素

View对象存储了Andiord屏幕上一个特定的矩形区域的布局和内容属性的数据体

通过View对象，可实现对布局、绘图、焦点变换、滚动条、屏幕区域的按键、用户交互等功能

Android的窗体功能是通过Widget（窗体部件）类实现的，而View类是Widget的基类

1. 视图容器组件（ViewGroup）：

ViewGroup是View的容器，可将View添加到ViewGroup中，一个ViewGroup也可以加入到另外一个ViewGroup里。

1. 布局管理Layout：

LinearLayout：线性布局。

RelativeLayout：相对布局。

TableLayout：表格布局。

AbsoluteLayout：绝对布局。

FrameLayout：框架布局。

GridLayout：网格布局。

5、事件处理机制

Android系统中引用Java的事件处理机制，包括事件、事件源和事件监听器三个事件模型

1. 提示信息 （Toast）

提示信息（Toast）是Android中用来显示提示信息的一种机制，与对话框不同，Toast是没有焦点的，而且Toast显示时间有限，过一定的时间会自动消失。

Toast类定义在android.widget包中。

调用Toast的静态方法makeText()创建一个指定文本和时长的提示信息。

调用Toast的show()方法显示提示信息

1. 菜单

Android中菜单分为两种类型：

选项菜单（Option Menu）

上下文菜单（Context Menu）

8、创建选项菜单的3个步骤：

覆盖Activity的onCreateOptionsMenu()方法，当第一次打开菜单时该方法被自动调用。

调用Menu的add()方法添加菜单项（MenuItem）,可以调用MenuItem的setIcon()方法为菜单设置图标。

当菜单项被选择时，覆盖Activity的onOptionsItemSelected()方法来响应事件。

1. 小结：

Android中的界面元素主要由View、ViewGroup和Layout几个部分构成

Android系统中引用Java的事件处理机制，包括事件、事件源和事件监听器三个事件模型

Android中提供了两种创建布局的方式：XML布局文件和代码直接实现

Android的布局包含LinearLayout、RelativeLayout、TableLayout、AbsoluteLayout、FrameLayout和GridView等多种布局

提示信息（Toast）是Android中用来显示提示信息的一种机制，与对话框不同，Toast是没有焦点的，而且Toast显示时间有限，过一定的时间会自动消失

Android系统中提供了四种对话框：AlertDialog、ProgressDialog、DatePickerDialog和TimePickerDialog

常用的Widget组件有：按钮（Button）、文本框（TextView）、编辑框（EditText）、复选框（CheckBox）、单选按钮组（RadioGroup）、下拉列表（Spinner）

Android的菜单有两种：选项菜单（Options Menus）和上下文菜单（Context Menus）

ActionBar的主要元素包括：图标和标题部分、ActionButton、OverFlow、ActionView、Tabs

1. ：Intent概述 ：
2. Intent由动作、数据、分类、类型、组件和扩展信息等内容组成，每个组成都由相应的属性进行表示，并提供设置和获取相应属性的方法。

2、Intent的属性：Action属性、Data属性、Category属性、Component属性、Extra属性

3、<intent-filter>元素

<intent-filter>元素中常用<action>、<data>和<category>这些子元素

Category属性的默认值“android.intent.category.DEFAULT”是启动Activity的默认值

1. 广播和接收Intent机制：

实现广播和接收Intent机制包含四个步骤：

注册相应的广播接收器（Broadcast Receiver），广播接收器是接收广播消息并对消息作出反应的组件；

发送广播，该过程将消息内容和用于过滤的信息封装起来，并广播给广播接收器；

满足条件的广播接收器执行接收方法onReceiver()；

销毁广播接收器。

1. Intent启动：

Android应用程序的三个核心组件：活动(Activity)、广播接收器(BroadcastReceiver)、服务（Service）都可以通过Intent来启动或激活。

6、小结：

Intent由动作、数据、分类、类型、组件和扩展信息等内容组成

Action属性用于描述Intent要完成的动作，对要执行的动作进行一个简要描述

Intent的Data属性是执行动作的URI和MIME类型

Category属性指明一个执行Action的分类

Component属性用于指明Intent的目标组件的类名称

Extra属性用于添加一些附加信息

多Activity的Android应用程序可通过startActivity()方法指定相应的Intent对象来启动另外一个Activity

Intent的Extra属性用于添加一些附加信息，利用该属性可以进行消息的传递

将传递的信息存放到Extra属性中有如下两种方式：一种是直接法将信息添加到Extra属性中，另一种是将数据封装到Bundle包中

Intent可以通过显示方式或隐式方式找到目标组件，显示方式是直接通过设置组件名来实现，而隐式方式则是通过Intent Filter过滤实现（default与隐示Intent呼应）

在AndroidManifest.xml配置文件中，Intent Filter以<intent-filter>元素来指定，一个组件中可以有多个<intent-filter>元素，每个<intent-filter>元素描述不同的能力

广播接收器（Broadcast Receiver）是接收广播消息并对消息作出反应的组件

Activity的sendBroadcast()、sendOrderBroadcast()和sendStrikyBroadcast()方法都可以广播Intent到广播接收器

大数据消息传递用Bundle。

1. Service：

1、Service特点：

没有用户界面，不与用户交互；

长时间运行，不占程序控制权；

比Activity的优先级高，不会轻易被Android系统终止，即使Service被系统终止，在系统资源恢复后Service仍将自动运行。

用于进程间通信（Inter Process Communication，IPC），解决两个不同进程之间的调用和通信问题。

1. 启动和绑定Service

Service类创建好之后，可以通过两种方式启动Service：

启动方式：

使用Context.startService()方法启动Service，调用者与Service之间没有关联，即使调用者退出，Service服务依然运行。

绑定方式：

通过Context.bindService()启动Service，调用者与Service之间绑定在一起，调用者一旦退出，Service服务也就终止。

1. 启动方式的Service生命周期：

startService（）→onCreate（）→onStart（）→运行Service→stopService（）→onDestroy（）

绑定方式的Service生命周期：

bindService（）→onCreate（）→onBind（）→运行Service→unbindService（）→onDestroy（）

1. 停止Service：

使用Context.startService()方法启动的Service，通过调用Context.stopService()或Service.stopSelf()方法结束。

使用Context.bindService()绑定的Service，通过调用Context.unbindservice()解除绑定。

1. 小结：

Service提供程序的后台服务，分为本地服务和远程服务两种类型

定义一个Service类需要只要继承Service类，并实现其生命周期中的方法

Service必须在AndroidManifest.xml配置文件中通过<service>元素进行声明

Service有启动方式和绑定方式两种启动方式。

Service的启动方式使用Context.startService()方法来启动一个Service，调用者与Service之间没有关联，即使调用者退出，Service服务依然运行

Service的绑定方式通过Context.bindService()来启动一个Service，调用者与Service之间绑定在一起，调用者一旦退出，Service服务也就终止

使用Context.startService()方法启动的Service，通过调用Context.stopService()或Service.stopSelf()方法结束服务

使用Context.bindService()绑定的Service，通过调用Context.unbindservice()解除绑定的服务

Android提供大量的系统服务，这些系统服务用于完成不同的功能，通过Context.getSystemService()方法可以获取不同服务管理对象

NotificationManager类是系统的通知服务管理类，它能够将通知Notification信息显示在状态栏上

1. 数据存储：

1、Android提供了四种数据存储方式：

SharedPreferences存储数据：Preference采用“键-值”对方式组织和管理数据，其数据存储在XML文件中。是一个轻量级的存储机制。该方式实现比较简单，适合简单数据的存储。

File存储数据：文件存储是一个“重量级”存储机制，比Preference方式更适合存储较大的数据；不同于SQLite，不适合结构化的数据存储。

SQLite存储数据：SQLite使用数据库作为存储方式，适合大数据量的数据存储，能够很容易的对数据进行增加、插入、删除、更新等操作。

网络存储数据。

2、使用SharedPreferences的步骤：

首先使用getSharedPreferences()生成SharedPreferences对象。

使用SharedPreferences.Editor的putXXX()方法保存数据。

使用SharedPreferences.Editor的commit()方法将上一步保存的数据写到XML文件。

使用SharedPreferences的getXXX()方法获取相应数据。

1. File存储数据：
2. 通过Context.openFileInput()方法可以获取标准的文件输入流（FileInputStream），读取设备上的文件；通过Context.openFileOuput()方法可以获取标准的文件输出流（FileOutputStream）。
3. Mode主要有四种模式：

MODE\_APPEND追加

MODE\_PRIVATE 私有

MODE\_WORLD\_READABLE 可读

MODE\_WORLD\_WRITEABLE 可写

1. 数据共享Content Provider ：

（1）数据模型 ：

ContentProvider将其存储的数据以数据表的形式提供给访问者。

在数据表中每一行为一条记录，而每一列为具有特定类型和意义的字段。

每一条数据记录都包括一个“\_ID”数据列，该字段唯一标识一个记录。

URI ：每一个ContentProvider都对外提供一个自身数据集的唯一标识

1. ContentResolver：

ContentResolver提供了对ContentProvider的数据进行查询、插入、修改和删除等操作的方法，在开发过程中是间接的通过操作ContentResolver来操作ContentProvider的。

使用getContentResolver()方法获取ContentResolver实例对象；

使用query()方法查询；

使用insert()方法插入数据，注意先将数据封装到ContentValues对象中；

使用delete()方法删除数据：

使用update()方法实现记录的更新操作。

1. 小结：

Preference提供了一种轻量级的数据存储方式，以“key-value”方式将数据保存在一个XML配置文件中

通过Context.openFileInput()方法可以获取标准的文件输入流（FileInputStream），读取设备上的文件；通过Context.openFileOuput()方法可以获取标准的文件输出流（FileOutputStream）

Android中通过SQLite数据库实现结构化数据存储

SQLiteOpenHelper是SQLiteDatabase的一个帮助类，用来管理数据库的创建和版本更新

Content Provider（内容提供器）是所有应用程序之间数据存储和检索的一个桥梁，其作用就是使得各个应用程序之间实现数据共享

Android系统为一些常见的应用（如音乐、视频、图像、联系人列表等）定义了相应的ContentProvider，它们被定义在android.provider包下。

总结：

1. Android的四大组件是什么，它们的作用。

答：Activity，Service，ContentProvider，BroadCastReceiver

Activity是Android程序与用户交互的窗口，是Android构造块中最基本的一种，它需要为保持个界面的状态，做很多持久化的事情，妥善管理生命周期以及一些跳转逻辑。

Service是后台服务于Activity，封装有一个完整的功能逻辑实现，接受上层指令，完成相关的事物，定义好需要接受的Intent提供同步和异步的接口。

ContentProvider是Android提供的第三方应用数据的访问方案，可以派生ContentProvider类，对外提供数据，可以像数据库一样进行选择排序，屏蔽内部数据的存储细节，向外提供统一的接口模型，大大简化上层应用，对数据的整合提供了更方便的途径。

BroadCastReceiver接受一种或者多种Intent做触发事件，接受相关消息，做一些简单处理，转换成一条Notification，统一了Android的事件广播模型

1. Android中的onCreate（）方法是其生命周期的开始。