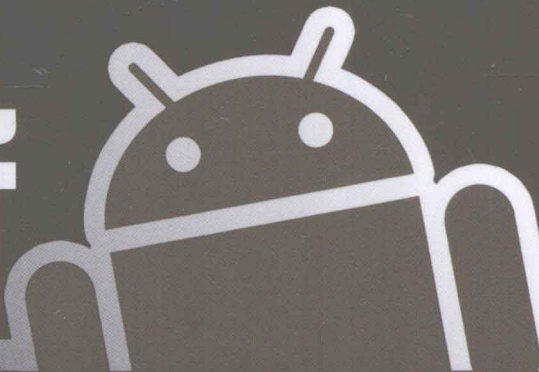


Analyzing

Google Android SDK——The Smart Phone Development Paradigm Bible

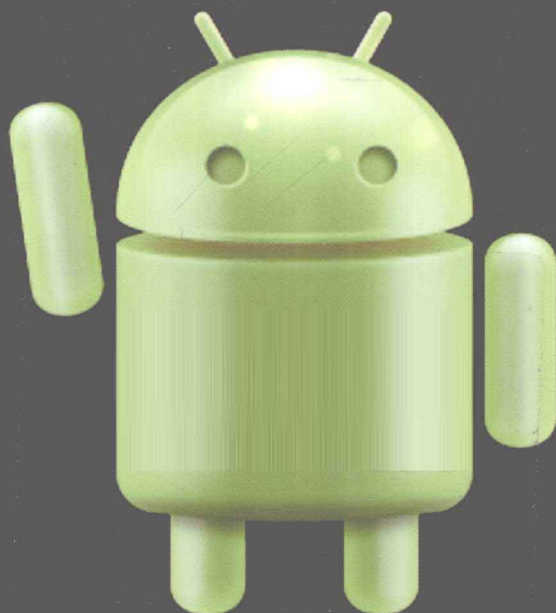
解析



薛显亮 编著

Google Android SDK

——智能手机开发范例手册



ANDROID

中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

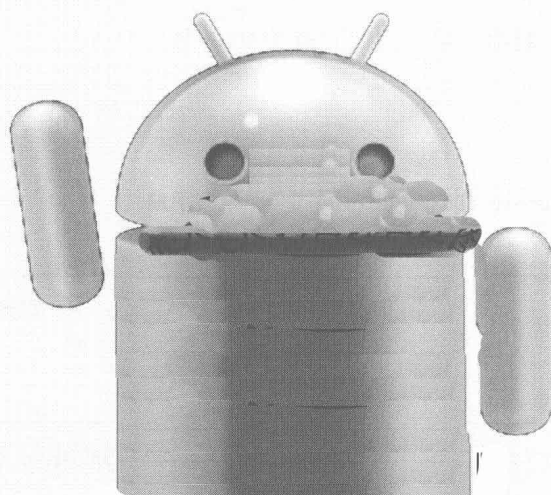
解析



薛显亮 编著

Google Android SDK

——智能手机开发范例手册



ANDROID

中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

内 容 简 介

全书基于 Android SDK 2.3 版本,分为 15 章,深入浅出地讲解了 Android 基于 SDK 开发的基本原理、关键技术和典型应用等各方面的知识,其中绝大部分理论知识来自于 Android 2.3 开发文档,力求保持原生态。

本书对于已经掌握 Android 基础知识,且具备一定的软件开发经验,而又想进一步提高开发技能的程序员具有明显的能力提升作用;同时可作为 Android 程序员随时查阅各种知识和技术的工具手册。

图书在版编目(CIP)数据

解析 Google Android SDK : 智能手机开发范例手册
/ 薛显亮编著. — 北京 : 中国铁道出版社, 2012.8
ISBN 978-7-113-14359-6

I. ①解… II. ①薛… III. ①移动电话机—应用程序
—程序设计 IV. ①TN929.53

中国版本图书馆 CIP 数据核字(2012)第 034285 号

书 名: 解析 Google Android SDK——智能手机开发范例手册
作 者: 薛显亮 编著

策 划: 荆 波
责任编辑: 荆 波
责任印制: 赵星辰

读者服务热线: 010-63560056
编辑助理: 王 婷

出版发行: 中国铁道出版社(北京市西城区右安门西街 8 号 邮政编码: 100054)
印 刷: 北京新魏印刷厂
版 次: 2012 年 8 月第 1 版 2012 年 8 月第 1 次印刷
开 本: 787mm×1092mm 1/16 印张: 31.75 字数: 750 千
书 号: ISBN 978-7-113-14359-6
定 价: 65.00 元

版权所有 侵权必究

凡购买铁道版图书,如有印制质量问题,请与本社发行部联系调换。

在今天, Android 一词已经越来越被人们熟知, 截止到 2011 年 8 月, Android 应用程序数量已经突破 10 万, 巨大的数字可以证明 Android 智能手机操作系统已经是当今世界上最流行的系统之一。在优势方面, Android 平台的最大优势是其代码开放性, 开放的平台允许任何移动终端厂商加入到 Android 联盟中来, 可以使其拥有更多的开发者。随着用户和应用的日益丰富, 一个崭新的平台也将很快走向成熟。

在过去很长的一段时间里, 特别是在欧美地区, 手机应用往往受到运营商制约, 如使用什么功能、接入什么网络, 几乎都受到运营商的控制。自从 iPhone 上市, 用户可以更加方便地连接网络, 运营商的制约减少。随着 EDGE、HSDPA 这些 2G/3G 移动网络的逐步过渡和提升, 手机随意接入网络已经成为现实。

此外, 各个厂商推出的硬件也各具特色, 极大地促进了 Android 的发展。

Android 平台提供给第三方开发商一个十分宽泛、自由的环境。因此, 不会受到各种条条框框的限制, 可想而知, 会有多少新颖别致的软件由此诞生。

从搜索巨人到全面的互联网渗透, Google 服务如地图、邮件、搜索等已经成为连接用户和互联网的重要纽带, 而 Android 平台手机将无缝结合这些优秀的 Google 服务。

本书主要对 Android SDK 2.3 进行了详细介绍, 绝大部分理论知识来自于 Android 2.3 开发文档, 力求保持原生态。目前, 其他版本的 Android 主要用于平板电脑开发, 而且成熟度上略差, 同时也向下兼容 2.3。掌握成熟的 2.3 版本, 是一个关键点, 若后面发布成熟版本, 其中也完全包含这些内容, 读者只需要学习新增的 API 即可。

本书的章节安排

本书分为 15 章, 由浅入深地讲解 Android 基于 SDK 开发的方方面面。

第 1 章: 介绍了 Android 开发的必备基础知识, 包括四大组件及其生命周期、意图与意图过滤器、manifest 文件结构、进程与线程。

第 2 章: 开发第一个应用 Hello World, 部署到真机, 并对 Android 常用的调试手段和测试框架进行了详细介绍。

第 3 章: 讲解了 Android 中的 UI 部分。包含 UI 概述、布局、菜单、UI 事件、Dialog、通知、主题与样式、自定义组件、AdapterView 与数据绑定。

第 4 章: 全面介绍了 Android 的应用资源。首先对应用资源类型进行介绍, 然后对怎样提供应用资源进行说明, 接着讲解了如何使用应用资源, 最后介绍了运行时配置改变的应用场景。

第 5 章: 介绍了 Android 中的数据存储部分。涉及的知识点有共享偏好、内部存储、外部存储、数据库存储和网络存储。

第 6 章：介绍了 Android 中蓝牙的使用。涉及的知识点有蓝牙基础 API、设置蓝牙、寻找设备、建立连接、管理连接等。

第 7 章：介绍了 Android 与 Internet 的有关知识。主要讲解了 HTTP 协议相关的内容和 WebView 的使用。

第 8 章：介绍了 Android 的搜索框架。包含搜索框的使用，增加最近搜索建议，增加自定义的搜索建议，以及搜索配置。

第 9 章：介绍了 Android 的安全与权限。

第 10 章：介绍了应用小部件。本章教你一步一步创建自己的 App Widget。

第 11 章：介绍了 Android 一些常用的应用开发，包含了打电话应用、发短信应用等。

第 12 章：介绍了 Google 位置服务及 Google Maps API 的使用。

第 13 章：介绍了一些实践的技巧和方法，如避免内存泄露，布局的技巧讲解，怎么样轻松使用线程等。

第 14 章：讲解了怎样共享 APK，应用程序的原理及如何为应用增加广告等。

第 15 章：对于拷贝和粘贴的介绍和讲解。

附录 A：Android 中解析 XML，包含 SAX、DOM、Pull 等解析。

附录 B：高级组件的进一步介绍。

附录 C：开发调试的介绍和演示。

本书的读者对象

- 具备一定软件开发经验，想快速进入 Android 开发领域的程序员。
- 已经掌握 Android 的基础知识，想进一步提高 Android 开发技能的程序员。
- 为了个人兴趣和职业技能储备而想要学习 Android 的爱好者。
- 想将本书作为枕边书，随时查阅 Android 的各种知识和技术的读者。
- 可作为各大培训机构的培训教材。

致谢

感谢常立兵的参与，他为本书编写了两个附录，使得本书更加完善。

感谢所有参与编写、出版这本书的人，尤其是本书的责任编辑。

编 者

2012 年 5 月

目 录

Contents

第 1 章 Android 应用程序基础

1.1 四大组件	2
1.1.1 Activity	2
1.1.2 Service	4
1.1.3 Broadcast receiver	4
1.1.4 Content providers	5
1.2 Intent 与 IntentFilter	13
1.2.1 Intent	13
1.2.2 Intent 解析	16
1.2.3 Note Pad 例子	20
1.3 Manifest 文件	24
1.4 Activity 与任务	26
1.4.1 关系 (affinity) 和新任务	27
1.4.2 启动模式	28
1.4.3 栈的清除	29
1.4.4 任务的启动	29
1.5 进程与线程	30
1.5.1 进程	30
1.5.2 线程	32
1.5.3 远程调用	33
1.6 组件生命周期	36
1.6.1 活动 (Activity) 生命周期	36
1.6.2 服务生命周期	40
1.6.3 广播接收者的生命周期	41

第 2 章 Android 之 HelloWorld

2.1 安装 Android SDK 与 ADT 插件	42
2.2 建立第一个 Android 项目	44
2.2.1 在 Eclipse 中新建工程	44
2.2.2 为应用添砖加瓦	46
2.2.3 使用模拟器	48

2.2.4 使用真机进行开发	52
2.3 开发 Android 程序常用的调试手段	52
2.4 使用测试用例	57
2.4.1 测试 API 简介	58
2.4.2 Activity 测试	62
2.4.3 Content Provider 测试	65
2.4.4 Service 测试	66
2.5 测试 HelloWorld 工程	67
2.5.1 创建测试工程	67
2.5.2 创建测试用例类	69
2.5.3 运行测试用例并查看结果	71
2.6 Android SDK 提供的一些工具	72
2.7 精彩案例：应用更新模块	73

第 3 章 认识可爱的 UI

3.1 UI 概览	80
3.2 丰富的布局	83
3.2.1 FrameLayout	83
3.2.2 LinearLayout	84
3.2.3 AbsoluteLayout	86
3.2.4 RelativeLayout	86
3.2.5 TableLayout	87
3.3 创建菜单	88
3.3.1 通过 xml 文件声明菜单项	89
3.3.2 为 Menu 填充资源	89
3.3.3 增加 item 选择事件监听	90
3.3.4 为菜单增加 Checkbox（复选框）和 radio（单选框）	90
3.3.5 为 Menu 增加快捷键	91
3.4 处理 UI 事件	91
3.4.1 事件监听器	91
3.4.2 事件处理器	93
3.5 Dialog 介绍	93
3.5.1 AlertDialog	93
3.5.2 ProgressDialog	98
3.5.3 DatePickerDialog	100

3.5.4	TimePickerDialog	101
3.5.5	自定义 Dialog	101
3.6	提醒用户	103
3.6.1	Toast 提醒	103
3.6.2	状态栏提醒	105
3.7	主题与样式	110
3.8	自定义组件	112
3.8.1	完全自定义组件	113
3.8.2	复合控件	117
3.8.3	修改已存在的 View 类型	118
3.9	AdapterView 与数据绑定	119
3.9.1	为 AdapterView 填充数据	119
3.9.2	处理用户选择事件	120
3.10	精彩案例: SurfaceView 的使用	120

第 4 章 应用资源

4.1	应用资源类型	126
4.1.1	动画资源	126
4.1.2	颜色状态列表	132
4.1.3	Drawable 资源	134
4.1.4	布局资源	151
4.1.5	菜单资源	153
4.1.6	字符串资源	156
4.1.7	style 资源	159
4.2	为应用提供资源	160
4.2.1	组织资源类型	160
4.2.2	提供可替换的资源	162
4.2.3	最佳兼容性	165
4.2.4	Android 如何找到最匹配的资源	166
4.3	访问应用资源	167
4.3.1	在代码中访问资源	168
4.3.2	在 XML 文件中访问资源	169
4.3.3	访问平台资源	170
4.4	处理运行时配置改变	170
4.4.1	在配置改变期间维护一个对象	170

4.4.2 自行处理配置的改变	171
4.5 精彩案例：本地化.....	172

第5章 数据存储

5.1 使用 Shared Preferences	175
5.2 内部存储器	177
5.3 外部存储器	181
5.4 使用数据库	183
5.4.1 创建数据库	183
5.4.2 对数据库进行操作	184
5.4.3 SQLite 事务操作	187
5.4.4 数据库在 NotePad 例子中的应用	188
5.5 网络存储	192
5.6 精彩案例：文件浏览器.....	197

第6章 蓝牙

6.1 基础 API	205
6.2 设置蓝牙	206
6.3 寻找设备	208
6.4 建立连接	210
6.5 管理连接	215
6.6 精彩案例：BluetoothChat 讲解	216

第7章 Android 与 Internet

7.1 以 GET 方式发送 HTTP 请求	229
7.2 以 Post 方式发送 HTTP 请求	233
7.3 WebView	237
7.4 把网络图片设置为桌面壁纸	242
7.5 小巧练歌器	246
7.5.1 MediaPlayer	246
7.5.2 范例原理	249
7.6 精彩案例：网络 apk 安装器	257

第8章 Android 搜索框架

8.1 使用搜索框	267
-----------------	-----

8.1.1	基本知识.....	268
8.1.2	创建搜索配置.....	268
8.1.3	创建搜索 Activity.....	269
8.1.4	使用搜索框.....	271
8.1.5	增加语音搜索.....	273
8.2	增加最近搜索建议.....	274
8.2.1	基础知识.....	274
8.2.2	创建提供搜索建议的内容提供者.....	275
8.2.3	修改搜索配置.....	276
8.2.4	保存查询文本.....	276
8.2.5	实例.....	277
8.3	增加自定义的搜索建议.....	282
8.3.1	基础知识.....	282
8.3.2	创建内容提供者.....	283
8.3.3	为建议声明 Intent.....	285
8.4	搜索配置.....	287
8.5	精彩案例：词典应用.....	290

第 9 章 安全和权限

9.1	应用程序签名.....	299
9.2	用户 ID 和文件访问.....	303
9.3	使用权限.....	304
9.4	声明和实施权限.....	304
9.4.1	在清单文件里声明权限.....	304
9.4.2	在清单文件里实施权限.....	306
9.5	URI 权限.....	309
9.6	精彩案例：使用 URI 权限.....	309

第 10 章 应用小部件

10.1	基础知识.....	316
10.1.1	增加 AppWidgetProviderInfo 元数据.....	317
10.1.2	创建应用小部件布局.....	318
10.1.3	使用 AppWidgetProvider 类.....	320
10.1.4	创建应用小部件配置 Activity.....	322
10.2	精彩案例：我的应用小部件.....	323

10.2.1	AppWidgetProvider 子类	324
10.2.2	广播接收者 MyBroadcastReceiver	327
10.2.3	笑话内容显示 Activity	328
10.2.4	AppWidgetProviderInfo 配置文件	328

第 11 章 Android 常用程序开发

11.1	应用开场动画	330
11.2	制作自己的开机程序	333
11.3	自制拨号器	336
11.4	短信的发送与接收	342
11.5	查看自己的 SIM 卡属于哪个运营商	348
11.6	神奇的传感器	351
11.7	制作九宫图	354
11.8	动态更改屏幕朝向	356
11.9	铃声音量调节器	358
11.10	PreferenceActivity 的使用	362
11.11	JNI 编程	365

第 12 章 Google 位置服务

12.1	获取用户位置	373
12.1.1	确定用户位置面临的挑战	373
12.1.2	请求位置更新	374
12.1.3	定义一个最佳性能模式	375
12.1.4	常见的应用案例	378
12.1.5	提供模拟位置数据	379
12.2	谷歌地图 (Google Maps) 外部库	381
12.2.1	获得 Maps API Key	382
12.2.2	创建 Google Map 工程	384
12.2.3	显示地图	385
12.2.4	显示缩放视图	387
12.2.5	改变地图视图	388
12.2.6	显示一个特定位置	388
12.2.7	增加标记	390
12.2.8	获取触摸位置	392
12.2.9	地理编码与反向地理编码	394
12.3	精彩案例: 超市促销	395

第 13 章 最佳实践

13.1 避免内存泄露.....	409
13.2 屏幕朝向改变提速.....	411
13.3 布局技巧之一——创建高效的布局.....	413
13.4 布局技巧之二——使用 ViewStubs	417
13.5 布局技巧之三——合并布局.....	419
13.6 Android 多任务	424
13.7 轻松使用线程.....	426
13.8 跟踪内存的分配.....	428

第 14 章 共享 APK

14.1 应用程序原理.....	432
14.2 为应用增加广告.....	441

第 15 章 拷贝和粘贴 Copy and Paste

15.1 剪切板框架概述.....	443
15.1.1 剪切板类.....	444
15.1.2 强制转换剪切板中的数据为文本.....	445
15.1.3 复制数据到剪贴板	446
15.1.4 粘贴来自剪切板的数据	447
15.2 使用内容提供者拷贝复杂类型数据.....	450
15.2.1 在 URI 上编码标识符	450
15.2.2 拷贝数据结构	451
15.2.3 拷贝数据流.....	453
15.2.4 设计有效的复制粘贴功能	454

附录 A Android 中解析 XML

A.1 准备工作.....	455
A.2 SAX 解析.....	456
A.3 DOM 解析	460
A.4 Pull 解析	461

附录 B 组件

B.1 文本框 TextView 介绍与应用	467
------------------------------	-----

B.2	编辑框 EditText 介绍与应用	469
B.3	按钮 Button 介绍与应用	470
B.4	图片视图 ImageView 介绍与应用	471
B.5	图片按钮 ImageButton 介绍与应用	472
B.6	网格视图 GridView 的介绍与应用	474
B.7	循环图片集 Gallery 介绍与应用	477
B.8	选项卡 TabWidget 介绍与应用	479
B.9	自动完成文本框 AutoCompleteTextView 介绍与应用	483
B.10	拖动条 SeekBar 介绍与应用	484
B.11	图片切换 ImageSwitcher 介绍与应用	486
B.12	小结	487

附录 C 开发调试

C.1	DDMS 介绍	488
C.1.1	Logcat 介绍	488
C.1.2	Devices 介绍	489
C.1.3	Emulator Control 介绍	490
C.1.4	File Explorer 介绍	491
C.1.5	Allocation Track 介绍	491
C.2	Android JUnit 单元测试	491
C.3	如何调试项目	493
C.4	小结	494

第 1 章 Android 应用程序基础

Android 应用程序是用 java 语言编写的，通过 Android SDK 工具进行 Android 代码的编译，把代码、数据和资源文件编译到一个以 apk 为后缀的文件中。apk 文件是 Android 系统中应用程序的通用安装文件，是发布或安装应用到手机上的一个载体。我们可以把一个 apk 文件看作是一个 Android 应用。

Android 应用被安装到设备上以后，只在自己的安全沙箱内运行，要了解安全沙箱，需要掌握以下几点知识：

1. Android 操作系统是一个多用户 Linux 系统，每个应用代表不同的用户。
2. 默认情况下，系统会为每个用户指定一个唯一的 Linux 用户 ID（这个 ID 只会被系统使用，应用本身并不知道这个 ID 是多少）。系统会为每个应用下的所有文件设置了权限，只有指定了相应用户 ID 的应用才可以对这些文件进行访问。
3. 每个进程都有他自己的虚拟机对象（VM），所以应用程序之间的代码运行是彼此隔离的。
4. 默认情况下，每个应用运行在自己的 Linux 进程里。应用里的任何组件（本章接下来会对 Android 的四种组件进行详细介绍）需要运行时，Android 就会启动一个进程。当应用所有的组件停止运行，或是系统为了让其他应用运行而必须回收内存时，就会关闭当前应用所在的进程。

通过这种方式，Android 系统实现了最小特权原则，也就是说，每个应用只能访问自己进程空间的内容。这创建了一个非常安全的环境，例如，一个应用不能够访问系统没有授予访问权限的服务。

然而，有许多种方式可以实现应用之间数据的共享和访问系统的服务，下面介绍两种方式：

1. 安排两个应用程序共享一个 Linux 用户 ID 是可以的，这种情况下他们是可以访问彼此的文件的。为了节约系统资源，拥有相同 ID 的应用也能运行在一个相同的 Linux 进程中，并且共享相同的 VM。
2. 一个应用可以请求特定的权限来访问设备上的数据，例如联系人信息、短信、SD 卡、照相机、蓝牙等。所有应用权限在用户安装应用程序时被授予。

以上内容介绍了应用是如何存在于 Android 系统之中，相信大家对 Android 应用有了一定意义的了解，本章接下来的内容包括：

1. Android 系统的四大组件，这是整个 Android 程序开发中最核心的东西。这些组件有 Activity、Service、Content Provider 和 Broadcast Receiver。
2. Intent 和 IntentFilter，两者在 Android 程序开发中也是非常基础的。通过 Intent 和 IntentFilter，我们可以实现 Android 系统中组件之间的解耦，让我们开发出更加强壮的程序。
3. manifest.xml 文件的介绍。manifest.xml 文件是整个 Android 应用程序的核心配置文件。
4. Activity 与 Task 一节主要介绍了 Activity 与 Task 之间的关系。另外也要多多体会 Task 这一设计概念在移动操作系统 Android 中的巧妙之处。
5. 通过对 Android 系统中进程与线程的深入解读，有利于我们开发出更健壮更优雅的应用。
6. 最后，我们讨论了组件的生命周期，是对本文开始部分的补充。

1.1 四大组件

Android 中一个核心的概念就是一个应用能使用另一个应用的组件（如果另一个应用允许的话）即组件的重用，比如，你的应用需要显示一个滚动的图片列表，其他的应用已经开发好一个合适的图片列表滚动器，并且设置其对其他应用可用；在这种情况下，你不必开发自己的滚动器，而可以重复使用现有的组件。

更方便的是，你的应用中不需要包含你用到的另一个应用的代码，也不需要链接那些代码。相反地，当你想实现某个功能，其他应用的组件恰好提供了这个功能，就启动另一个应用相应的组件（不是让另一个应用全部启动）来完成你的工作。为了对这一特性进行支持，在 Android 中，当一个应用的任何部分需要被使用时，系统必须能启动这个应用，并且将相应部分实例化成 java 对象，所以，和其他大多数系统不同的是，Android 应用程序没有一个单独的程序入口（例如：没有 main() 函数）；相反地，Android 只是提供一些必要的组件以便需要时系统能实例化并运行它。

Android 中有四种组件，无论何时，一个请求都应该由一个特定的组件来处理，Android 系统会确保包含这个组件的应用进程运行；如果需要就实例化并启动它。

首先，我们讨论 Android 中四种组件之首——Activity（活动），它是 Android 应用的视图层。

1.1.1 Activity

一个 Activity 是一个可见的、用户可以与之进行交互的界面。例如，一个 Activity 可以显示菜单，用户能够选择菜单中的一项；一个短信应用中，其中一个 Activity 负责显示联系人列表，第二个 Activity 负责向第一个 Activity 选择的联系人写消息，第三个 Activity 负责显示短信历史，虽然彼此结合形成一个应用，但每个 Activity 都是彼此独立的，每个都是 Activity 的子类。

一个应用程序可能由一个或多个 Activity 组成，Activity 显示什么，需要多少个 Activity，依赖于这个应用的设计者。一般地，应用中的一个 Activity 应该被标记为应用启动时第一个呈现给用户的，就是所谓的“main”Activity。从某个 Activity 到另一个 Activity 是通过当前的 Activity 启动另一个 Activity 来实现的。

每个 Activity 默认绘制在一个窗口里。通常，这个窗口会占满整个屏幕，但是它可以比屏幕小，浮在另一个窗口的上面。

一个窗口中的可见内容是由一些具有层次关系的 view 组成的，都是继承自 View 类，每个 view 都控制一个窗口中的特定的矩形框区域，父 view 包含子 view 和组织子 view 的布局，叶子 view（那些在继承层次中底层的 view）绘制在它们所控制的矩形框中，并且对用户的动作做出直接的回应，因此 view 就是 Activity 和用户交互的地方，Android 开发中有很多的 view 可以直接使用，包括 buttons、text fields、scroll bars、menu items、check boxes 等。

一个 view 树是通过 Activity.setContentView()方法被放到一个 Activity 的 window 中的，content view 是 view 树中最顶端的那个 view。

Activity 可以通过调用它自己的 finish()方法来关闭，某一个 Activity 也可以通过调用 finishActivity()来关闭另一个 Activity(这个 Activity 是通过调用 startActivityForResult()来启动的)。

例如：

```
/**
 * MainActivity
 * 主要作用: 1.启动 OtherActivity 2.处理 OtherActivity 返回的数据
 */
private static final int REQUEST_CODE = 1;
@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.main);
    //启动 OtherActivity, REQUEST_CODE 代表请求码
    startActivityForResult(new Intent(this, OtherActivity.class), REQUEST_CODE);
}
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    switch(resultCode) {
        case OtherActivity.RESULT_OK: //这个常量代表 OtherActivity 返回成功
            //Toast 显示 OtherActivity 返回的信息
            Toast toast=Toast.makeText(getApplicationContext(),
            data.getStringExtra("msgFromOther"), Toast.LENGTH_SHORT);
            toast.setGravity(Gravity.CENTER|Gravity.TOP, 0, 0);
            //设置 toast 的显示位置
            toast.show(); //显示 toast 提醒
            break;
        default:
            break;
    }
}

/**
 * OtherActivity
 * 主要作用: 接受 MainActivity 调用, 返回处理结果
 */
```

```
public static final int RESULT_OK = 1;
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    Intent data = new Intent(OtherActivity.this, MainActivity.class);
    //定义 Intent
    data.putExtra("msgFromOther", "来自 OtherActivity 的消息");
    this.setResult(RESULT_OK, data); //这个方法会返回到调用它的那个组件里
    finish(); //调用此方法 Activity 被关闭, 即调用了 onDestroy() 方法
}
```

效果如图 1-1 所示。

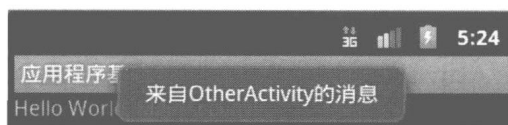


图 1-1 运行效果图

1.1.2 Service

service 是一个用户不可见的组件：在一段不确定的时间里运行在系统后台，每个 **service** 都继承自 **Service** 类。

例如，一个服务可以在后台播放音乐，在网络中获取数据或执行复杂的计算。

可以连接或者绑定到一个正在运行的 **service**（如果这个 **service** 还没运行的话就启动它），当连接到 **service** 后，你可以通过一个 **service** 暴露出来的接口和这个 **service** 交流，对音乐 **service** 来说，其接口允许用户暂停、后退、停止和重新播放音乐。

和 **Activity** 或者其他组件一样，**service** 运行在这个应用进程的主线程中，所以它不应该阻塞其他的组件或者用户界面，它们经常为那些耗时长的任务单独开一个线程。

一个 **service** 可以通过调用自己的 **stopSelf()**，或者在其他应用组件中调用 **stopService(Intent service)** 来关闭。

1.1.3 Broadcast receiver

一个 **Broadcast receiver** 是这样—个组件，它接收广播并作出反应；在系统中有很多已有的广播，比如反映时区变化的、电池变化的、用户修改了系统语言时的广播等，应用程序也可以自定义广播，比如定义这样一个广播：让其他的应用知道某些数据已经下载完毕可以使用了。

一个应用可以有任意多个 **Broadcast receiver** 来对它所关心的广播进行监听并作出反应。所有的 **receiver** 都继承自 **Broadcast Receiver** 类。

Broadcast receiver 不显示在用户界面上，然而，它可以启动一个 **Activity** 来作为它接收到的信息一种反应，或者它可以使用 **NotificationManager** 来提示用户，**Notifications** 可以通过不同的方式获得用户的注意，比如点亮闪光灯、震动、播放一个声音等，它们一般放一个图标在状态栏上，让用户可以打开获得这些信息，关于 **Notifications** 请参考后面章节的内容。