



中国电子教育学会高教分会推荐
普通高等教育电子信息类“十三五”课改规划教材

11010.....
0010100101.....
11001.....
11010.....
0010101001.....
11001.....
10101001.....



Android 平台应用软件开发

主编 朱小军 柴方林



西安电子科技大学出版社
<http://www.xduph.com>

中国电子教育学会高教分会推荐

普通高等教育电子信息类“十三五”课改规划教材

Android 平台应用软件开发

主编 朱小军 柴方林

参编 祁建宏 文 静

西安电子科技大学出版社

内 容 简 介

本书全面且详细地讲述了 Android 应用程序开发用到的相关知识,包括 Android 开发环境的搭建与布局、Android 控件、Android 数据存储、Android 网络开发和 Android 中的特色编程等。本书注重实际动手能力的培养,每个知识点都以小案例的方式进行讲解,以项目驱动方式使读者能够更好地理解知识点,最后两章的综合项目能够提高读者对综合项目的编程能力。

本书配套光盘中包含了搭建开发环境所用的全部软件 and 所有章节案例的源代码程序,不仅可以作为参考,也可以直接使用,以方便读者快速掌握 Android 应用开发技术。

本书可供具备 Java 编程基本功底并有志于 Android 应用开发的初中级读者使用,也适合作为高等学校计算机专业 Android 方向学生的教材。

图书在版编目(CIP)数据

Android 平台应用软件开发/朱小军,柴方林主编. —西安:西安电子科技大学出版社,2016.9

中国电子教育学会高教分会推荐

普通高等教育电子信息类“十三五”课改规划教材

ISBN 978-7-5606-4170-6

I. ① A… II. ① 朱… ② 柴… III. ① 移动终端—应用程序—程序设计—高等学校—教材
IV. ① TN929.53

中国版本图书馆 CIP 数据核字(2016)第 211940 号

策 划 刘玉芳

责任编辑 马 静 阎 彬

出版发行 西安电子科技大学出版社(西安市太白南路 2 号)

电 话 (029)88242885 88201467 邮 编 710071

网 址 www.xduph.com 电子邮箱 xdupfb001@163.com

经 销 新华书店

印刷单位 陕西华沐印刷科技有限责任公司

版 次 2016 年 9 月第 1 版 2016 年 9 月第 1 次印刷

开 本 787 毫米×1092 毫米 1/16 印 张 19.5

字 数 462 千字

印 数 1~3000 册

定 价 40.00 元(含光盘)

ISBN 978 - 7 - 5606 - 4170 - 6/TN

XDUP 4462001-1

如有印装问题可调换

前 言

随着移动互联网和物联网技术的快速发展及应用,移动互联技术在企业项目开发中的应用变得越来越广泛。而 Android 操作系统是以 Linux 为基础的开放源代码操作系统,主要适用于便携设备,是目前市面上大部分智能手机与平板电脑的主流操作系统。来自互联网的统计数据显示,Android 已经成为目前使用最为广泛的移动操作系统,远超 Apple 公司的 iOS 和 Microsoft 公司的 Windows Phone。基于 Android 系统的一些移动物联网相关核心技术也已经成为计算机、移动互联网相关专业学生未来就业的必备技术,许多开发人员将其作为一项专项技能进行学习和掌握。

本书是作者基于多年从事本课程教学的经验和体会,与甘肃省河图信息有限公司有多年实际开发项目经验的工程师合作完成的。本书由浅入深、循序渐进地介绍了移动软件开发技术的知识点,既有详尽的理论阐述,又有丰富的案例程序,使读者能轻松、快速、全面地掌握移动环境软件编程技术。

全书共分 12 章,主要内容如下:

第 1 章介绍 Android 系统和软件开发的框架;第 2 章介绍 Android 软件开发环境的搭建;第 3 章介绍 Android 中的五种布局方式;第 4 章介绍 Android 中的 UI 控件;第 5 章介绍 Android 中的交互与菜单;第 6 章介绍 Android 中的数据存储;第 7 章介绍 Android 中的四大组件;第 8 章介绍 Android 中访问网络的方法;第 9 章介绍 Android 中的一些特色编程;第 10 章介绍 Android 中调用其他语言的方法;第 11 章和第 12 章是综合项目。前 10 章选择功能单一、结构简单的小项目说明所讲知识点,最后两章给出两个综合项目,对所学知识进行汇总练习。

本书配套光盘中包含了书中搭建开发环境所用的所有软件及各章案例的源代码。读者可以将案例导入 Eclipse 中观看案例效果,再根据书中所讲步骤进行项目练习。

本书由兰州城市学院 Android 软件开发课程主讲教师朱小军和甘肃省河图信息有限公司 Android 工程师柴方林主编,参加编写工作的还有兰州城市学院的祁建宏老师和文静老师。其中柴方林工程师编写了第 7 章、第 11 章、第 12 章;祁建宏老师编写了第 1 章、第 9 章、第 10 章;文静老师编写了第 2 章、第 5 章;朱小军老师编写了第 3 章、第 4 章、第 6 章、第 8 章,并负责全书的审核、统稿。

兰州城市学院信息工程学院魏邦龙教授审定了本书稿，在此表示衷心感谢！

由于时间的原因以及个人学术与经验的欠缺，书中难免存在不妥之处，恳请同行专家学者和广大读者批评指正。

本书所带工具的下载地址：

<https://yunpan.cn/cPyzRe3kQqzWB>

访问密码：1885

编 者

2016年6月

目 录

第 1 章 绪论.....	1	第 4 章 Android 中的 UI 控件.....	37
1.1 Android 操作系统简介.....	1	4.1 UI 控件概念介绍.....	37
1.1.1 Android 平台的特性.....	2	4.1.1 基础控件.....	37
1.1.2 Android 系统架构.....	2	4.1.2 高级控件.....	39
1.2 软件系统架构.....	4	4.2 UI 控件使用案例.....	40
1.2.1 C/S 结构.....	4	4.2.1 基础控件案例.....	40
1.2.2 B/S 结构.....	5	4.2.2 高级控件案例.....	50
1.2.3 C/S 结构与 B/S 结构的分析比较.....	6	第 5 章 Android 中的交互与菜单.....	68
1.2.4 软件系统架构总结.....	6	5.1 Android 中的 Toast 提示框.....	68
第 2 章 Android 开发环境搭建.....	9	5.1.1 Toast 提示框的创建.....	68
2.1 Android 开发环境搭建.....	9	5.1.2 Toast 提示框的优化.....	68
2.1.1 搭建环境工具介绍.....	9	5.2 Android 的对话框.....	70
2.1.2 工具软件的安装.....	10	5.2.1 常用的设置对话框方法.....	70
2.1.3 Android 模拟器的创建.....	13	5.2.2 创建对话框的关键代码.....	71
2.2 Android 程序目录结构.....	13	5.3 Android 事件处理.....	73
2.2.1 HelloWorld 程序.....	14	5.3.1 Android 中按钮点击事件处理的 四种方法.....	73
2.2.2 Android 程序目录结构.....	15	5.3.2 Android 中的键盘响应事件.....	77
2.2.3 第一个自己设计界面的 Android 程序.....	17	5.3.3 Android 中的 onTouchEvent 事件.....	78
2.3 Android 开发的几个辅助资源.....	19	5.4 Intent 的进阶.....	79
2.3.1 Android SDK 的 API 文档.....	19	5.4.1 什么是 Intent.....	79
2.3.2 Dalvik Debug Monitor Service.....	19	5.4.2 使用 Intent 进行界面间数据的传递... ..	79
2.3.3 ADB 命令行工具.....	22	5.5 Android 中的 Notification.....	80
第 3 章 Android 中的五种布局.....	24	5.5.1 PendingIntent 简介.....	80
3.1 五种布局概念介绍.....	24	5.5.2 创建 Notification 的关键代码.....	81
3.2 五种布局案例.....	25	5.6 Android 中的 Menu.....	82
3.2.1 线性布局案例.....	25	5.6.1 菜单的创建与响应.....	82
3.2.2 相对布局案例.....	28	5.6.2 实现 SubMenu.....	85
3.2.3 帧布局案例.....	30	5.6.3 实现上下文菜单 Context Menu.....	86
3.2.4 表格布局案例.....	32	5.6.4 使用 XML 生成菜单.....	89
3.2.5 绝对布局案例.....	35	5.6.5 Android 中的 PopupWindow.....	90
		5.7 Android 中的多媒体播放.....	93

5.7.1 MediaPlayer 简介	93	8.1.2 工具软件的安装	147
5.7.2 MediaPlayer 案例	94	8.1.3 搭建服务器	148
第 6 章 数据存储	98	8.2 Android 利用 HttpURLConnection	
6.1 Android 中数据存储的方式	98	访问网络	149
6.1.1 SharedPreferences 概述	98	8.2.1 HttpURLConnection 介绍	149
6.1.2 SharedPreferences 存储数据的实现	99	8.2.2 HttpURLConnection 的使用方法	150
6.1.3 SharedPreferences 存储数据的优势 ..	100	8.2.3 使用 HttpURLConnection 访问	
6.2 文件存储数据	100	服务器	151
6.2.1 文件存储数据的关键代码	101	8.3 Android 利用 HttpClient 访问网络	154
6.2.2 文件方式读写 SD 卡上的数据	101	8.3.1 HttpClient 的介绍	154
6.3 SQLite 数据库存储数据	102	8.3.2 使用 HttpClient 来执行 GET 调用	154
6.3.1 SQLite 数据库的操作	104	8.3.3 使用 HttpClient 来执行 POST 调用 ..	157
6.3.2 SQLite 数据库操作的关键代码	104	8.4 Android 从服务器下载文件	161
6.4 ContentProvider 存储数据和网络		8.4.1 从服务器下载图片	161
存储数据	108	8.4.2 从服务器下载文件	162
6.5 项目演练	109	8.5 xml 的生成与解析	164
6.5.1 项目分析	109	8.5.1 什么是 xml	164
6.5.2 项目创建步骤	109	8.5.2 在服务器端生成 xml	164
第 7 章 Android 中的四大组件	117	8.5.3 在客户端解析 xml 数据	166
7.1 Activity 组件	117	8.6 JSON 的生成与解析	169
7.1.1 Activity 的生命周期	117	8.6.1 什么是 JSON	169
7.1.2 Activity 的任务栈	123	8.6.2 JSON 在服务端的生成	169
7.2 BroadcastReceiver 组件	124	8.6.3 JSON 在客户端的解析	171
7.2.1 创建 BroadcastReceiver	124	8.7 服务器与数据库的通信	173
7.2.2 广播接收器的生命周期	125	8.7.1 数据库的安装	173
7.2.3 广播的类型和发送	126	8.7.2 服务器与 mysql 的连接	175
7.2.4 广播接收器案例	126	8.7.3 数据库的增、删、改、查语句	176
7.3 Service 组件	130	8.8 本章项目实战	178
7.3.1 Service 的分类	130	8.8.1 项目流程分析	178
7.3.2 创建和启动 Service 组件的方法	130	8.8.2 做项目之前的准备	179
7.3.3 Service 组件案例	132	8.8.3 开始做项目	179
7.4 ContentProvider 组件	135	第 9 章 Android 中的特色编程	207
7.4.1 Uri 类介绍	136	9.1 Android 的 Fragment	207
7.4.2 UriMatcher 类使用介绍	136	9.1.1 Fragment 的相关使用	207
7.4.3 ContentProvider 数据共享	137	9.1.2 通过布局文件将 Fragment 加入	
7.4.4 ContentProvider 案例	138	Activity	208
第 8 章 Android 中的网络访问	147	9.1.3 通过 Java 代码将 Fragment 加入	
8.1 在 Eclipse 上搭建服务器	147	Activity	209
8.1.1 搭建服务器环境工具介绍	147	9.2 Android 中 ViewPager 的使用	212

9.2.1 ViewPager 简介	212	10.2 Android NDK 编程	241
9.2.2 ViewPager 案例	213	10.2.1 Android NDK 简介	241
9.3 Android 中的动画	216	10.2.2 Android NDK 使用的原因	242
9.3.1 四种补间动画的介绍	217	10.2.3 Android NDK 开发环境的搭建	242
9.3.2 用 xml 布局实现动画	218	10.3 Android NDK 编程案例	244
9.3.3 用 Java 代码实现动画	222	第 11 章 项目实战——短信提醒	247
9.4 Android 常用传感器的操作	223	11.1 本章导读	247
9.4.1 Android 传感器的类型	223	11.2 制定计划	247
9.4.2 Android 传感器的简介	223	11.3 需求分析	248
9.4.3 Android 传感器的使用	225	11.4 软件设计	250
9.4.4 Android 传感器案例	227	11.5 程序编码	255
9.5 Android 中的定位	232	11.6 软件测试	286
9.5.1 Android 中定位方法介绍	232	11.7 运行与维护	287
9.5.2 基站定位案例	233	第 12 章 项目实战——天气预报	288
9.6 Android 的硬件操作	234	12.1 本章导读	288
9.6.1 Android 中的 wifi 开发	234	12.2 制定计划	288
9.6.2 wifi 开发实例项目	235	12.3 需求分析	289
9.6.3 Android 中的蓝牙开发	237	12.4 软件设计	289
第 10 章 Android NDK 编程	240	12.5 程序编码	292
10.1 JNI 概述	240	12.6 软件的后测试与维护	303
10.1.1 JNI 机制的使用过程	240	参考文献	304
10.1.2 使用 JNI 机制产生的副作用	240		
10.1.3 JNI 机制的使用场合	241		

第1章 绪 论

学习目标

- 了解 Android 操作系统基本概念;
- 掌握计算机软件开发的基本架构。

1.1 Android 操作系统简介

从第一台手机——摩托罗拉 DynaTAC 8000X 问世至今,手机的发展仅有 30 多年的时间。但随着手机硬件技术和移动网络技术的发展,手机的功能已不仅仅拘泥于单纯的打电话、发短信,还具有了多媒体播放、高速网络浏览、GPS 定位、无线连接甚至更多的功能。随着移动通信技术的飞速发展和移动互联时代的到来,手机作为人们必备的移动通信工具,已从简单的通话工具演变成一个移动的个人信息收集和处理的平台。借助操作系统和丰富的应用软件,手机成了一台智能移动终端。这类终端的出现改变了很多人的生活方式及对高新技术下通信设备的渴求,人们不再满足于手机的基本功能,而开始追求手机强大的操作系统给人们带来更多、更强、更具个性的服务。

现在,市场上流行的智能手机操作系统有 Symbian OS、Android OS、Windows Phone、iOS、Blackberry 等。其中 Android OS 和 iOS 系统不仅仅在智能手机市场份额中遥遥领先,而且这种优势仍在不断增加。根据全球领先的移动互联网第三方数据挖掘与整合营销机构艾媒咨询数据显示,2012 年中国智能手机市场 Android OS 份额达到 68.6%,占据绝对主流地位。

Android 英文原意为“机器人”。2003 年 Andy Rubin 在美国创办了一家名为 Android 的公司,其主要经营业务为手机软件和手机操作系统。2005 年 Google 公司收购了 Android 公司。此后 Google 公司与摩托罗拉、中国移动、宏达、高通和 T-Mobile 等 30 多家无线通信领军企业组成了开放手机联盟,合作开发基于 Linux 的开源手机操作系统,并于 2007 年 11 月 5 日正式推出了基于 Linux 2.6 标准内核的开源手机操作系统,命名为 Android。

Android 操作系统是首个为移动终端开发的真正开放的、完整的移动操作系统,支持该系统的手机厂商有摩托罗拉、HTC、三星、LG、索尼爱立信、联想、中兴等。Android 系统的最大优势是开源性,可以提供给第三方开发商广泛、自由的开发环境。因此第三方开发商推出了众多功能各异的应用产品。

1.1.1 Android 平台的特性

1. 采用 Java 语言

Java 是一种面向对象的语言，可以在很多设备和操作系统上使用，学习起来比较容易，上手比较快。而且学习 Java 语言可以帮助开发者更好地了解其他操作系统的应用开发过程，起到一通皆通的效果。

2. 可移植性好

Android 应用程序使用 Java 编程语言开发，因为 Java 语言的跨平台性，Android 应用程序能够很容易地移植到其他移动操作系统上。例如，可以非常容易地将 Android 应用移植到 Blackberry、Symbian、Ubuntu 等移动操作系统上，而 Android 应用程序也可以轻松移植到 Chrome OS。

3. 良好的开发环境

谷歌公司为 Android 开发者营造了良好的开发环境。Android Studio 开发工具有利于开发者为 Android 系统开发新的应用和游戏，该工具的开发环境和模式非常丰富和便捷，可以为开发者提供相关的测试工具和各种数据分析。

4. 开发门槛低

想要为 iOS 操作系统开发一款应用，首先开发者得有一台 Mac 电脑。而 Android 应用开发者在 Windows、Mac 还有 Linux 操作系统上都能完成 Android 应用程序的开发工作。此外，如果要登记为苹果应用程序商店的开发人员，一年需要的费用为 99 美元，而如果登记为 Google Play 的开发人员，一年只需要支付 25 美元。

5. 市场份额

2014 年年底的数据统计显示全球有 16 亿的 Android 用户，与此同时，iOS 用户只有 3 亿，而 Blackberry 以及其他移动操作系统的用户数量加起来只有 9100 万。对于开发者而言，用户数量是影响应用程序使用量最直接、最重要的因素。

6. 盈利能力

Google 是互联网公司，靠服务挣钱；而 Apple 是设备商，靠设备去赚钱。Apple 要做的是 Apple 认为最人性化的设备，Apple 的软件是为 Apple 的设备服务的。因此 iPhone 平台中很多游戏和应用都是收费的。但是 Android 不同，Android 平台的大多数应用和游戏都是免费的，因此在 Android 平台中，会有更多的用户愿意去尝试新的应用和游戏，其盈利多来自应用和游戏广告。这种模式更容易被用户接受。在 Android 平台下，2014 年全年用户为开发者贡献的广告收入比 2013 年提升了将近 50%。

1.1.2 Android 系统架构

Android 是一个开放源代码的操作系统。Android 和其操作系统一样，采用了分层架构。从架构图看，Android 分为四层，从高层到低层分别是应用程序层(APPLICATIONS)、应用程序框架层(APPLICATION FRAMEWORK)、系统库层(LIBRARIES)和 Linux 核心层(LINUX KERNEL)，如图 1-1 所示。

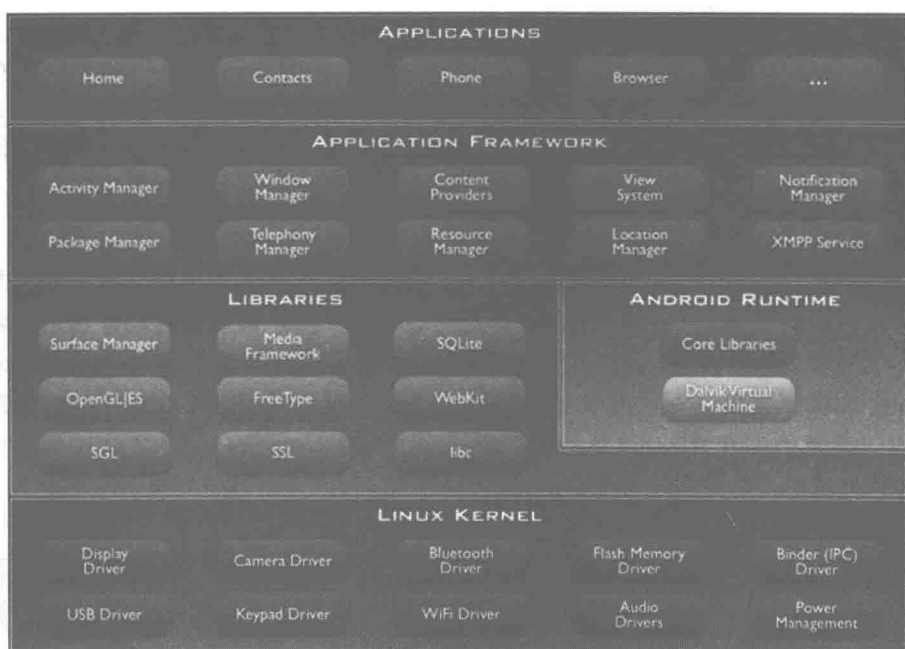


图 1-1 Android 系统构架图

1. 应用程序层

Android 操作系统发布时，系统中就包含了一系列核心应用程序包，包括 E-mail 客户端、SMS 短消息程序、日历、地图、浏览器、联系人管理程序等。所有的应用程序都是使用 Java 语言编写的，同时开发者也可以利用 Java 语言设计和编写属于自己的应用程序，而这些程序与那些 Android 操作系统包含的核心应用程序是彼此平等的，并且不会产生冲突。

2. 应用程序框架层

Android 的应用程序框架为应用程序层的开发者提供了简单方便的 API 框架，应用程序层的开发者可以使用该层提供的 API 开发出自己的应用程序。应用程序框架层包括活动管理器(Activity Manager)、窗口管理器(Window Manager)、内容提供者(Content Providers)、视图系统(View System)、通知管理器(Notification Manager)、包管理器(Package Manager)、电话管理器(Telephony Manager)、资源管理器(Resource Manager)、位置管理器(Location Manager)和 XMPP 服务(XMPP Service)十个部分。在 Android 平台上，任何一个应用程序都可以发布它的功能块(不过得遵循框架的安全性限制)并且任何其他的应用程序都可以使用其所发布的功能块。Android 的应用程序框架帮助程序员快速地开发程序，并且该应用程序重用机制也使用户可以方便地替换程序组件。

3. 系统库层和 Android 运行环境

Android 的系统库层分成两个部分，一个是系统包含的各种库，另一个是 Android 运行环境。

系统库包括九个使用库，分别是图层管理(Surface Manager)、媒体库(Media Framework)、SQLite、OpenGL ES、FreeType、WebKit、SGL、SSL 和 libc。这些库能被 Android 系统中不同的组件使用。它们大多是使用 C 和 C++实现的，可以通过 Android 应用程序框架为开

发者提供服务。

Android 运行环境主要指的是虚拟机技术——Dalvik。Dalvik 虚拟机和一般 Java 虚拟机不同，Dalvik 虚拟机执行扩展名为 dex 的 Dalvik 可执行文件，该格式文件针对小内存使用做了优化。每一个 Android 应用程序都在它自己的进程中运行，都拥有一个独立的 Dalvik 虚拟机实例。Dalvik 被设计成一个设备可以同时高效地运行多个虚拟系统。

4. Linux 核心层

Linux 操作系统具有一些重要的特点，首先是开源，其次强大的内存管理和进程管理方案、基于权限的安全模式、经过认证的驱动模型等特征都非常符合移动平台的要求，因此 Android 系统选择采用 Linux 内核。虽然 Android 采用了 Linux 内核，但也根据移动平台特征对 Linux 内核系统做了一些改动。例如 Android 并没有采用 glibc 作为 C 库，而是采用 Google 自己开发的一套 Bionic Libc 来代替 glibc。此外，Android 还对 Linux 设备驱动进行了增强等。

1.2 软件系统架构

计算机发展初期，软件系统架构是 Mainframe 结构，该结构下的客户、数据和程序被集中在主机上，通常只有少量的 GUI 界面，这种结构体系对远程数据库的访问比较困难。随着 PC 的广泛应用，在 20 世纪 80 年代中期出现了 Client/Server 分布式计算结构，也就是我们常说的 C/S 结构，这种体系结构的应用程序的处理在客户机和服务器两者中实现；应用程序的请求通常是通过客户机提交给关系型数据库来处理的，客户机在接收到服务器返回的数据后实现显示和业务逻辑。Client/Server 结构因为其灵活性得到了极其广泛的应用。

随着网络的发展，越来越多的商业系统被搬上 Internet，一种新的、更具生命力的体系结构被广泛采用，这就是我们熟知的“三层/多层体系结构”，也就是我们常说的 B/S 机构。三层体系结构中，客户层是用户接口和用户请求的发出地，典型的应用是网络浏览器；服务器层是负责处理客户层提交请求的计算机，典型的应用是 Web 服务器和运行业务代码的应用程序服务器；数据层是为整个软件系统提供数据的数据源基地，典型的应用是关系型数据库和其他后端数据资源，如 Oracle、SAP、R/3 等。三层体系结构中，请求信息、处理请求和数据被物理隔离。

三层结构是个更灵活的体系结构，它把显示逻辑从业务逻辑中分离出来，这就意味着业务代码是独立的，可以不关心如何显示、在哪里显示。业务逻辑层处于中间层，不需要关心由哪种类型的客户来显示数据，也可以与后端系统保持相对独立性，有利于系统扩展。三层结构具有更好的移植性，可以跨不同类型的平台工作，允许用户请求在多个服务器间进行负载均衡。三层结构中的安全性也更易于实现，因为应用程序已经同客户隔离。

1.2.1 C/S 结构

C/S 结构，即 Client/Server(客户机/服务器)结构。此结构把数据库内容放在远程的服务器上，而在客户机上安装了相应软件。C/S 软件一般采用两层结构，由两部分构成：前端

是客户机,即用户界面(Client)结合了表示与业务逻辑,接收用户的请求,并向数据库服务提出请求,通常是一个 PC;后端是服务器,即数据管理(Server)将数据提交给客户端,客户端将数据进行计算并将结果呈现给用户,还要提供完善的安全保护及对数据的完整性处理等操作,并允许多个客户同时访问同一个数据库。在这种结构中,服务器的硬件必须具有足够的处理能力,这样才能满足不同客户的要求。

C/S 结构在技术上很成熟,它的主要特点是交互性强、具有安全的存取模式、网络通信量低、响应速度快、有利于处理大量数据。但是具有该结构的程序的针对性较强,变更不够灵活,维护和管理的难度较大,通常只局限于小型局域网,不利于扩展。并且,由于具有该结构的每台客户机都需要安装相应的客户端程序,分布功能弱且兼容性差,不能实现快速部署安装和配置,因此缺少通用性,具有较大的局限性,且要求具有一定专业水准的技术人员才能完成。

1.2.2 B/S 结构

B/S 结构,即 Browser/Server(浏览器/服务器)结构。此结构只安装维护一个服务器(Server),而客户端采用浏览器(Browse)。它是随着 Internet 技术的兴起,对 C/S 结构的一种变化和改进,主要利用了不断成熟的 WWW(万维网)浏览器技术,结合了多种 Script 语言(VBScript、javascript 等)和 ActiveX 技术,是一种全新的软件系统构造技术。

B/S 三层体系结构采用三层客户/服务器结构,在数据管理层(Server)和用户界面层(Client)增加了一层结构,称为中间件(Middleware),使整个体系结构成为三层。三层结构是伴随着中间件技术的成熟而兴起的,核心概念是利用中间件将应用分为表示层、业务逻辑层和数据存储层三个不同的处理层次。这三个层次是从逻辑上划分的,具体的物理分法可以有多种组合。中间件作为构造三层结构应用系统的基础平台,提供了以下主要功能:负责客户机与服务器、服务器与服务器间的连接和通信;实现应用与数据库的高效连接;提供一个三层结构应用的开发、运行、部署和管理的平台。这种三层结构在层与层之间相互独立,任何一层的改变不会影响其他层的功能。

在 B/S 体系结构系统中,用户通过浏览器向分布在网络上的许多服务器发出请求,服务器对浏览器的请求进行处理,将用户所需信息返回到浏览器,而其余工作如数据请求、加工、结果返回以及动态网页生成、对数据库的访问和应用程序的执行等全部由 Web Server 完成。随着 Windows 将浏览器技术植入操作系统内部,这种结构已成为当今应用软件的首选体系结构。显然 B/S 结构应用程序相对于传统的 C/S 结构应用程序是一个非常大的进步。

B/S 结构的主要特点是分布性强、维护方便、开发简单、共享性强、总体拥有成本低。但同时,数据安全问题显现、数据传输速度慢、软件的个性化特点明显降低,且对服务器要求过高,这些缺点是显而易见的,难以实现传统模式下的特殊功能要求。例如通过浏览器进行大量数据输入、报表的应答、专用性打印输出比较困难,实现复杂的应用构造也有较大的困难。虽然可以用 ActiveX、Java 等技术开发较为复杂的应用,但是相对于发展已非常成熟的 C/S 一系列应用工具来说,这些技术的开发复杂,并没有完全成熟的技术工具供使用。

1.2.3 C/S 结构与 B/S 结构的分析比较

1. 硬件环境不同

C/S 建立在局域网的基础上,通过专门服务器提供连接和数据交换服务,所处理的用户不仅固定,并且处于相同区域,要求拥有相同的操作系统。B/S 建立在广域网的基础上,自己管理信息,有比 C/S 更强的适应范围,一般只要有操作系统和浏览器就行,与操作系统平台关系较小,面向不可知的用户群。

2. 结构不同

C/S 软件一般采用两层结构,而 B/S 采用三层结构。这两种结构的不同点是两层结构中客户端参与运算,而三层结构中客户端不参与运算,只是简单地接收用户的请求,显示最后的结果。由于三层结构中的客户端并不需要参与计算,所以对客户端的计算机配置要求较低。虽然 B/S 采用了逻辑上的三层结构,但在物理上的网络结构仍然是原来的以太网或环形网。这样,第一层与第二层结构之间的通信、第二层与第三层结构之间的通信都需占用同一条网络线路,网络通信量大。而 C/S 只有两层结构,网络通信量只包括 Client 与 Server 之间的通信量,网络通信量低。所以,C/S 处理大量信息的能力是 B/S 无法比拟的。

3. 处理模式不同

B/S 的处理模式与 C/S 相比,大大简化了客户端,只要装上操作系统、网络协议软件以及浏览器即可,这时的客户机称为瘦客户机,而服务器则集中了所有的应用逻辑。

4. 系统维护不同

系统维护是在软件生存周期中开销最大的一部分。C/S 程序由于其本身的整体性,必须整体考察并处理出现的问题,而 B/S 结构的客户端不必安装及维护。B/S 结构在构件组成方面只变更个别构件,开发、维护等工作都集中在服务器端;当需要升级时,只需更新服务器端的软件,而不必更换客户端软件,可实现系统的无缝升级。这样就减轻了系统维护与升级的成本和工作量,使用户的总体拥有成本(TCO)大大降低。

5. 对安全要求不同

由于 C/S 采用点对点的配对结构模式,并采用适用于局域网、安全性比较好的网络协议(例如 NT 的 NetBEUI 协议),安全性可得到较好的保证。C/S 一般面向相对固定的用户群,程序更加注重流程,它可以对权限进行多层次校验,提供了更安全的存取模式,对信息安全的控制能力很强。一般高度机密的信息系统适宜采用 C/S 结构。而 B/S 采用单点对多点、多点对多点这种开放的结构模式,并采用 TCP/IP 这一类运用于 Internet 的开放性协议,其安全性只能靠数据服务器上管理密码的数据库来保证,所以 B/S 对安全以及访问速度比 C/S 有更高的要求。而 Internet 技术中这些关键的安全问题尚未解决。

1.2.4 软件系统架构总结

通过对软件系统架构的了解,我们可以得出这样的结论,无论采用怎样的软件系统架构,一个软件系统的构成大体可分为三层:图形界面层、业务逻辑处理层和数据层。

第一层是图形界面层,该层的主要作用是提供软件系统的界面,用来和用户进行交互。

在 C/S 结构中,这一层主要表现为客户机上的客户端程序,因此这层主要是以客户端程序的编程语言来实现的。在 B/S 结构中,该层主要表现在浏览器访问页面上,因此该层主要是以 html 语言、css、JavaScript 语言来实现的。在 Android 软件开发中,这一层主要是由 Android 中的四大组件和 UI 控件来实现的。

软件系统的最底层是整个系统的数据层,这一层的主要作用是为整个软件系统提供需要的数据或者对用户从图形界面层传入的数据进行保存,无论在 C/S 结构还是 B/S 结构中,这一层都是由数据库来实现的,例如我们常见的 MySQL、SQLServer、Oracle 等数据库。在 Android 软件开发中,该层既可以通过访问远程的数据库服务器来实现,也可以通过 Android 本身自带的数据存储形式来实现。例如 Android 可以通过 SharedPreferences 类将数据类存放到本地文件,也可以将数据存放到 Android 本身自带的嵌入式数据库 SQLite 中。

软件系统的中间层是业务逻辑处理层,该层的主要作用是将用户界面层中用户提交的数据进行处理后存放到数据层,或者从数据层取得数据进行处理后提供给图形界面层使用。这一层主要根据软件系统开发时所使用的语言来实现。Android 软件开发使用的主要工具是 Java 语言,因此在 Android 软件开发中该层也主要是由 Java 语言来实现的。

有了软件系统构成的思路,我们就可以明确 Android 软件开发中要学习的内容,首先要学习 Android 应用开发中程序界面的构成,其次要学习 Android 应用开发中数据的处理和存储,最后要学习 Android 应用程序中处理业务逻辑的场景和方法。

1. Android 界面开发技术

界面开发技术是 Android 软件开发中的基本技术,所有的应用程序都有与用户进行交互的界面。在 Android 软件开发中,使用 xml 文件首先对界面进行布局,然后在 Activity 里进行引用是最常见的界面开发技术,这种技术包含的学习内容主要有以下几种:

(1) Android 最常见的五种 Layout 布局。

深刻理解五种布局组织控件的使用是 Android 开发的基础,只有这样,才可以在实际的界面中灵活地使用每种布局结构,构造出需要的界面。

(2) 常用控件。

Android 开发中能够使用的界面控件有很多,至少要知道每种控件是做什么的,至于里面的特定参数和方法可以在需要的时候再去查。

(3) 界面美化。

系统的控件很多时候无法满足要求,需要通过添加背景、更换图片等进行美化,因此要熟悉每种控件如何去美化。

掌握了上面的内容后,通过布局的嵌套就可以制作出各种界面,如果再熟悉 include 的使用就可以减少 xml 的重复,方便后期的修改和维护。

2. Android 数据存储技术

Android 中的数据存储方式主要有四种:

(1) SharedPreferences 存储方式。对于简单的数据可以使用该方式存储,以键值对的方式存储一些基本的数据。

(2) 文件存储。将数据以文件的形式存储,可以存储于手机自身的存储器上,也可以存储于 SD 卡上,文件的数据格式可以自己定义。

(3) SQLite 数据库存储。SQLite 是轻量级嵌入式数据库引擎,它支持 SQL 语言,并且只利用很少的内存就能获得很好的性能。此外它还是开源的,任何人都可以使用。在 Android 软件开发中,SQLite 数据库是较为复杂数据存储的首选。

(4) 网络存储数据。可将应用程序的数据存放在相关的网络服务器上。

3. Android 中的四大组件

Android 四大基本组件分别是 Activity 组件、BroadcastReceiver(广播接收者)、Service(服务)和 ContentProvider(内容提供者)。

(1) Activity(组件)。

在 Android 应用程序中,一个 Activity 通常就是一个单独的屏幕,它上面可以显示一些控件,也可以监听并对用户的事件做出响应。因此,可以说 Activity 是 Android 软件开发的基础。进行 Android 应用程序的开发时,需要显示很多界面让用户进行交互,而 Activity 就负责管理这些界面。多个 Activity 之间可以直接跳转或者通过 Intent 进行通信。

(2) BroadcastReceiver(广播接收者)。

广播机制是 Android 操作系统中一个非常有用的消息机制。例如当手机电池电量变化的时候,Android 系统会发出广播告诉其他应用程序电池电量发生了变化,如果 Android 不发出广播,突然因电量不足造成的关机可能会使很多应用来不及保存数据,致使数据受损。有了广播机制,在电池的电量过低可能要关机的时候,Android 系统就会发布“手机电量耗尽,即将关机”的消息给其他应用程序,应用程序在接收到广播以后,就会进行相应的操作,防止突然关机造成损失。

BroadcastReceiver 就是用来接收 Android 系统或 Android 应用中其他组件发出的广播的一个组件,广播接收者没有用户界面,但却实现了 Android 操作系统与应用程序之间或应用程序与应用程序之间的直接通信。当开发者需要获取 Android 操作系统或者其他程序的一些状态时,就可以通过广播接收者来进行。

(3) Service(服务)。

Service 也是 Android 系统中的一种重要组件,它跟 Activity 是同级别的。但 Activity 展现的内容都会以界面的形式呈现出来,用户可以看得见,而 Service 是在后台运行的,它没有可视化界面。Service 可以在后台为 Activity 准备相关的数据,也可以在后台维持用户的操作。例如,我们可以通过服务组件在后台播放音乐,同时在前台运行不同的 Android 程序界面,也可以通过服务组件在后台获取网络上的数据而不会阻塞前台用户的交互活动。

(4) ContentProvider(内容提供者)。

ContentProvider 组件可以让一个应用程序将自己的数据集提供给其他应用程序使用。其他应用程序也可以通过 ContentResolver 类从该内容提供者那里获取或修改数据。内容提供者组件主要应用于多个应用程序间共享数据的场景,例如通讯录应用程序中的数据被多个应用程序使用。

4. Android 网络编程

如今脱网运行的 Android 应用程序已不常见,所以网络编程是 Android 软件开发中必须掌握的技术。现在一般的软件公司都会封装自己的联网结构,但是基本的联网还是要熟悉。

第2章 Android 开发环境搭建

学习目标

- 掌握 Android 开发环境的搭建方法;
- 掌握 Android SDK 目录结构。

2.1 Android 开发环境搭建

工欲善其事，必先利其器。Android 软件开发技术作为一项新型的计算机技术，在进行开发学习之前首先要搭建其对应的开发环境。

2.1.1 搭建环境工具介绍

搭建 Android 开发环境所需的工具软件有 JDK、Eclipse、Android SDK、ADT 插件等，这些工具都可以从相关网络下载。为了方便读者，在本书前言中添加了所有软件的下载地址，可以下载课本所需工具与代码的合集，下载后解压即可看到“开发所需工具”的文件夹，文件夹中提供了相关的软件，读者可直接安装。考虑到 64 位 Windows 7 操作系统已经普及，本书提供的所有软件均为 64 位 Windows 7 环境下的软件。

1. JDK

JDK 是 Java 语言的软件开发工具包，主要用于移动设备、嵌入式设备上的 Java 应用程序。Android 开发语言是基于 Java 的，所以要做 Android 的开发必须安装 JDK，并且对 JDK 的版本有一定的要求，必须是 JDK 1.5 以上的版本。JDK 1.5 以前的版本 Android 不支持。如果你的机器上已经安装了 JDK，可以通过在 Dos 命令行下输入 `java-version` 命令来查看版本信息。综上所述，不管是做 Java 开发，还是做 Android 开发，都必须在电脑上安装 JDK。

2. Eclipse

Eclipse 是一个开放源代码的、基于 Java 的可扩展开发平台。就其本身而言，它只是一个框架和一组服务，用于通过插件构建开发环境。Eclipse 的插件环境吸引了众多公司和组织为它开发面向各种程序设计需求的 Eclipse 插件。Eclipse 也附带了一个标准的插件集，包括 Java 开发工具 JDT，使得开发者能够在标准的 Eclipse 工具中完成一些普通的 Java 开发工作。但 Eclipse 不仅仅是 Java 开发工具，还可以配合众多令人眼花缭乱的插件，完全满足从企业级 Java 应用到手机终端应用游戏的开发。Eclipse 开发平台支持普通 Java 程序、