



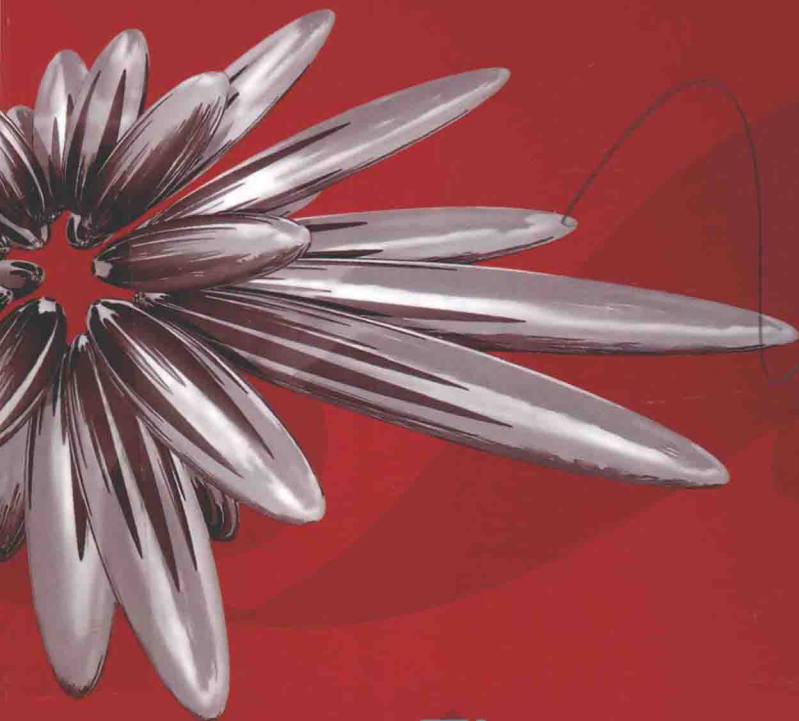
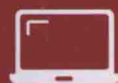
[普通高等教育“十三五”规划教材]

计算机类本科规划教材

基于Android Studio的 应用程序开发教程

◆ 李宁宁 主 编

◆ 郑俊生 张福艳 郭巧丽 副主编



中国工信出版集团



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
<http://www.phei.com.cn>

计算机类本科规划教材

基于 Android Studio 的 应用程序开发教程

李宁宁 主编

郑俊生 张福艳 郭巧丽 副主编

電子工業出版社

Publishing House of Electronics Industry

北京 • BEIJING

内 容 简 介

本书从初学者的角度出发, 基于 Android Studio 开发环境, 循序渐进地针对 Android 应用程序开发进行了介绍。全书共 9 章, 分别为: Android 平台与开发环境, Android 应用程序构成分析, Android 生命周期与通信, 布局和控件, 布局和控件进阶, 系统服务, 数据存储, 后台处理与网络通信, 综合实例设计与分析。从安装环境开始入手, 进入第一个 Android 应用程序的剖析, 最后完成一个基于服务器端和客户端结构的综合案例。本书案例丰富, 每章以项目导学开头, 引入当前章节的内容, 最后通过项目延伸, 引出更深入的需求, 给予读者发挥和实现的空间。

本书可作为高等学校计算机科学与技术、软件工程等专业的教材, 也可供 Android Studio 应用程序开发人员参考。

未经许可, 不得以任何方式复制或抄袭本书之部分或全部内容。
版权所有, 侵权必究。

图书在版编目(CIP)数据

基于 Android Studio 的应用程序开发教程 / 李宁宁主编. — 北京: 电子工业出版社, 2016.8
计算机类本科规划教材
ISBN 978-7-121-29385-6

I. ①基… II. ①李… III. ①移动终端—应用程序—程序设计—高等学校—教材 IV. ①TN929.53

中国版本图书馆 CIP 数据核字 (2016) 第 162037 号

策划编辑: 凌 毅

责任编辑: 凌 毅

印 刷: 北京嘉恒彩色印刷有限责任公司

装 订: 北京嘉恒彩色印刷有限责任公司

出版发行: 电子工业出版社

北京市海淀区万寿路 173 信箱 邮编: 100036

开 本: 787×1 092 1/16 印张: 18.5 字数: 474 千字

版 次: 2016 年 8 月第 1 版

印 次: 2016 年 8 月第 1 次印刷

定 价: 42.80 元

凡所购买电子工业出版社图书有缺损问题, 请向购买书店调换。若书店售缺, 请与本社发行部联系, 联系及邮购电话: (010)88254888, 88258888。

质量投诉请发邮件至 zltz@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

本书咨询联系方式: (010)88254528, lingyi@phei.com.cn。

前 言

Android 是 Google 公司开发的基于 Linux 平台的开源手机操作系统。自诞生以来,经过不断的发展和完善,其功能日益强大,Android 应用程序开发需求量也在不断扩大。而且,由于 Android 采用 Java 语言作为编程基础,更是为 Java 开发人员敞开了大门。

目前,各大高校也感受到 Android 应用开发的市场需求以及互联网势不可挡的应用趋势,逐渐开设一些相关的课程。由于很多院校已经相继为计算机类专业的学生开设了 Java 语言课程,甚至有些学校已经将 Java 语言作为相关专业的第一门编程语言,因此 Android 应用程序开发也随之变得更加轻松和得心应手。本书主要从教学的角度全面介绍 Android Studio 应用开发的核心知识;案例的选取与设计,大多是在真实授课过程中总结和完成的,也是作者们智慧和实践的结晶。

本书共 9 章。

第 1 章 Android 平台与开发环境,开门见山地介绍了 Android 平台和开发环境的搭建,从而为 Android 应用程序开发奠定了基础,然后带领读者完成第一个 Android 应用程序的开发。

第 2 章 Android 应用程序构成分析,在第 1 章的应用程序的基础上,进行了程序深入剖析,使读者掌握 Android 应用程序开发的过程和程序结构,从而可以将第 1 章的程序进行改造和“装修”,变成读者“设计”的程序。

第 3 章 Android 生命周期与通信,讲解了 Android 的生命周期,读者可以由此入手分析各个组件的工作原理,并根据功能需要着手编程。另外,也可以通过 Intent 进行组件的整合,从而构成复杂而庞大的 Android 应用程序。

第 4 章和第 5 章围绕着界面展开了内容介绍。第 4 章介绍了基础的布局和控制件,在此基础上,读者可以开发出基本的界面。本章以常用的登录和注册为例,从界面相关的基础知识到界面优化,实现了比较理想的界面效果,另外也介绍了几个高级控件,提高了编程难度的同时,也为 Android 应用程序开发奠定了一定的高度。第 5 章布局和控制件进阶,引入了目前非常流行的 Fragment 结构,从整体上把握和串联了 Android 界面开发的内容,并通过比较复杂的项目框架,实现目前主流的应用程序效果。

第 6 章系统服务,其中包括服务组件、定时机制、广播组件及通知等,将 Android 应用开发中的系统服务进行整合。

第 7 章数据存储,是 Android 应用开发的重点。本章从最简单的简单存储,到文件存储,再到数据库存储,最后到 ContentProvider 组件,读者可以针对不同的应用场景,选择对应的数据存储方式。

第 8 章后台处理与网络通信,讲解了 Android 应用程序与服务器进行网络通信的原理和实现过程,为移动互联网应用程序开发奠定了基础。而且,本章从原始联网的几种方式延伸到 Volley 框架的使用,循序渐进地将理论以更方便简洁的方式进行实现。

第 9 章综合实例设计与分析,将以上章节的知识点进行整合,实现了一个综合案例。

书中的每一章均通过【项目导学】的形式，引入本章的核心知识点，从而完成【项目实现】；为了发挥读者的个人能动性，通常最后又进行了【项目延伸】，为读者提供更多的发挥空间。本书可作为高等学校计算机科学与技术、软件工程等专业的教材，也可供 Android Studio 应用程序开发人员参考。

本书凝聚了作者们多年的教学经验和总结，由李宁宁担任主编，郑俊生、张福艳、郭巧丽担任副主编。具体编写分工如下：第 1，4，5 章由郑俊生编写；第 2，6，7 章由张福艳编写；第 3，8，9 章由李宁宁编写，郭巧丽负责校稿和审稿。案例是经过大家统一讨论和设计而实现的。此外，参与本书编写和审稿的人员还有杨光、郑纯军、王凯、窦乔、王澜、高志君等。全书最后由李宁宁负责统稿和定稿。

本书配有电子课件、源程序等教学资源，读者可以登录华信教育资源网（www.hxedu.com.cn）免费下载。

由于时间和作者水平有限，书中难免有错误和不妥之处，恳请广大读者特别是同行专家们批评指正。您的任何意见和建议，都将是我們继续改进本书的动力。

作者
2016 年 7 月

目 录

第 1 章 Android 平台与开发环境	1	第 3 章 Android 生命周期与通信	43
1.1 Android 应用开发概述	1	3.1 生命周期	44
1.1.1 Android 发展史与现状	2	3.1.1 概述	44
1.1.2 Android 基本架构	3	3.1.2 Activity 生命周期的方法	45
1.1.3 Android 组件	5	3.1.3 LogCat	47
1.2 开发环境	7	3.2 Android 组件间的通信	51
1.2.1 JDK 安装与配置	7	3.2.1 Intent 对象	51
1.2.2 Android Studio	7	3.2.2 Intent 过滤器	52
1.2.3 Eclipse	11	3.2.3 Intent 传递数据	57
1.2.4 Android Studio 的优势	12	3.2.4 获取 Activity 返回值	57
1.2.5 Genymotion 模拟器	13	3.3 项目实现——启动页	59
1.2.6 DDMS	14	习题 3	61
1.3 第一个 Android 程序	15	第 4 章 布局和控件	62
1.3.1 创建项目	15	4.1 基础控件	63
1.3.2 运行项目	20	4.1.1 TextView 控件	63
1.3.3 项目分析	20	4.1.2 EditText 控件	64
习题 1	22	4.1.3 Button 控件及监听事件	65
第 2 章 Android 应用程序构成分析	23	4.1.4 登录实例	67
2.1 目录结构分析	23	4.2 常用布局	68
2.2 Android 应用程序结构解析	27	4.2.1 线性布局	68
2.2.1 资源文件	27	4.2.2 相对布局	72
2.2.2 代码文件	30	4.2.3 其他布局	74
2.3 AndroidManifest.xml 文件	30	4.3 常用控件	74
2.4 应用程序运行分析	34	4.3.1 RadioGroup 和 RadioButton	75
2.4.1 AndroidManifest.xml 修改	34	4.3.2 Checkbox 控件	75
2.4.2 资源文件修改	36	4.3.3 ImageButton 控件	76
2.4.3 语言国际化	37	4.3.4 ImageView 控件	76
2.5 Android 的基本组件	38	4.3.5 Spinner 控件	77
2.5.1 Activity	38	4.3.6 注册实例	77
2.5.2 BroadcastReceiver	40	4.4 控件和布局优化	82
2.5.3 Service	41	4.4.1 Shape	82
2.5.4 ContentProvider	41	4.4.2 Selector	82
习题 2	42	4.4.3 登录布局和背景优化	84

4.5 高级控件	89	7.3 数据库存储	171
4.5.1 TextInputLayout	89	7.3.1 SQLite 简介	171
4.5.2 ListView	89	7.3.2 手动建库	172
4.5.3 RecyclerView 和 CardView	100	7.3.3 代码建库	177
习题 4	104	7.3.4 数据操作	179
第 5 章 布局和控制件进阶	106	7.3.5 第三方工具——xUtils	188
5.1 Fragment	107	7.4 ContentProvider	194
5.1.1 Fragment 的产生与介绍	107	7.4.1 ContentProvider 简介	194
5.1.2 Fragment 的生命周期	107	7.4.2 访问系统 ContentProvider	196
5.1.3 静态使用 Fragment	108	习题 7	201
5.1.4 动态使用 Fragment	113	第 8 章 后台处理与网络通信	202
5.2 Toolbar 和对话框	121	8.1 概述	203
5.2.1 Toolbar	121	8.2 后台线程	204
5.2.2 DialogFragment	122	8.2.1 子线程	204
5.3 Android 项目框架综合实例	127	8.2.2 Handler 消息传递机制	205
5.3.1 新建项目	128	8.2.3 异步任务	210
5.3.2 底部导航	129	8.3 获取网络数据资源	214
5.3.3 任意数量内导航	129	8.3.1 通过 URL 获取网络资源	214
5.3.4 固定数量内导航	136	8.3.2 通过 URLConnection 获取 网络资源	216
习题 5	141	8.3.3 通过 HTTP 获取网络资源	218
第 6 章 系统服务	142	8.3.4 Eclipse 下的 Tomcat 安装与配置	218
6.1 服务	142	8.3.5 登录案例	221
6.1.1 Service 的原理和用途	142	8.4 JSON	223
6.1.2 使用线程	152	8.4.1 概述	223
6.2 Timer 和 Alarm	156	8.4.2 Google Gson 简述	228
6.2.1 Timer	156	8.5 网络通信框架 Volley	231
6.2.2 Alarm	158	8.5.1 通过 Volley 获取 JSON 数据	233
6.3 Broadcast 组件	158	8.5.2 通过 Volley 加载图片资源	233
6.3.1 静态注册	159	8.6 项目实施——天气预报	238
6.3.2 动态注册	161	习题 8	243
6.4 Notification	162	第 9 章 综合实例——校园订餐 App	244
6.4.1 Notification 简介	162	9.1 功能介绍	244
6.4.2 PendingIntent	162	9.1.1 需求分析	244
6.4.3 Notification 的实现	163	9.1.2 开发环境搭建	244
习题 6	164	9.2 服务器端运行	245
第 7 章 数据存储	165		
7.1 简单存储	166		
7.2 文件存储	169		

9.2.1	数据库设计	245
9.2.2	服务器端运行配置	247
9.2.3	参数接口	248
9.3	客户端设计	258
9.4	客户端开发	258
9.4.1	登录和注册	264
9.4.2	店铺和菜谱列表	266
9.4.3	菜谱详情和购买	270

9.4.4	收藏	273
9.4.5	搜索	275
9.4.6	我	277
9.5	程序签名与打包	280

附录 A	校园订餐 App 服务器端配置	284
------	-----------------------	-----

参考文献		288
------	-------	-----

第1章 Android 平台与开发环境

Android 是一个优秀的开源手机平台，在智能手机市场的占有率排名第一，而且远超其他平台。Android 软件人才的需求也会越来越大，作为程序员或即将成为程序员的你们，马上加入 Android 应用开发阵营中来吧！通过本章的学习可以让读者对 Android 平台的发展、现状、基本框架有初步的了解，掌握安装、配置 Android 开发环境的步骤，理解 Android SDK 和 ADT（Android Developer Tools，Android 开发工具）的用途。通过创建第一个 Android 程序，进一步理解 Android 系统。在此基础上，读者可以尽情地开启 Android 编程之旅。

本章的学习目标 { 重点 { (1) JDK 环境变量的配置
(2) Android Studio 使用
难点 Gradle 配置

【项目导学】

Android 平台开发环境的安装和配置是学习 Android 开发的第一步，环境配置完成后，读者可以创建第一个 Android 项目。开发过程中使用 Android Virtual Device 虚拟机 AVD，虚拟机启动界面如图 1-1 所示，项目效果如图 1-2 所示。

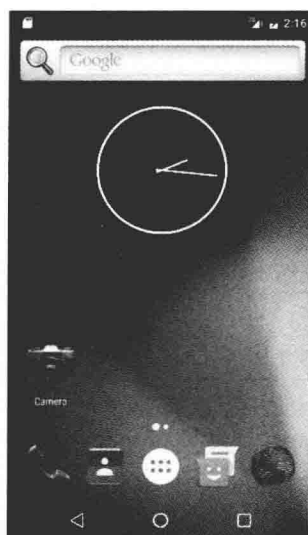


图 1-1 启动界面效果

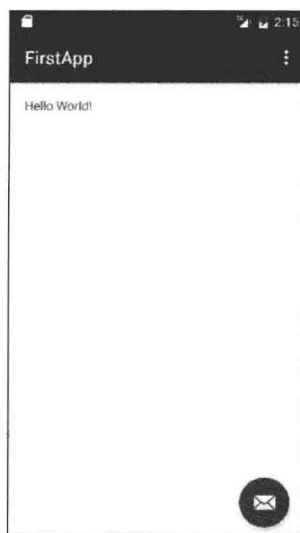


图 1-2 项目运行效果

1.1 Android 应用开发概述

随着 Android 系统和智能设备的迅猛发展，Android 已经成为全球范围内具有广泛影响力

的操作系统。Android 系统已经不仅仅是一款手机的操作系统，它越来越广泛地被应用于平板电脑、可穿戴设备、电视、数码相机等设备上，Android 开发人才需求倍增，从长远上看，Android 软件人才的需求也会越来越大。

从 2007 年 11 月 5 日谷歌公司正式向外界展示了这款名为 Android 的操作系统至今，Android 已经经历了多个版本的更新，直到本书创作之初，最新的版本为 Android 6.0 **Marshmallow**，代号“棉花糖”。

从 Android 6.0 开始，Android 系统有了一个质的飞跃，本书将专注讲解 Android 6.0 及以上版本的开发。Android 的各个版本之间大部分 API 都是向下兼容的，对于一些少部分的 API，也提供了向下兼容包。

1.1.1 Android 发展史与现状

2003 年 10 月，Andy Rubin 等人创建了与 Android 系统同名的 Android 公司，并组建了 Android 开发团队，最初的 Android 系统是一款针对数码相机开发的智能操作系统，之后被 Google 公司收购，从此 Android 取得了长足的发展，迅速占领了智能手机的市场份额。

自 Android 系统首次发布至今，Android 经历了很多次版本更新，表 1-1 列出了 Android 系统的不同版本的发布时间及对应的版本号。

表 1-1 Android 系统的版本号

Android 版本	代号
Android 1.1	
Android 1.5	Cupcake（纸杯蛋糕）
Android 1.6	Donut（炸面圈）
Android 2.0/2.1	Eclair（长松饼）
Android 2.2	Froyo（冻酸奶）
Android 2.3	Gingerbread（姜饼）
Android 3.0/3.1/3.2	Honeycomb（蜂巢）
Android 4.0	Ice Cream Sandwich（冰淇淋三明治）
Android 4.1	Jelly Bean（果冻豆）
Android 4.2	Jelly Bean（果冻豆）
Android 5.0	Lollipop（棒棒糖）
Android 6.0	Marshmallow（棉花糖）

从 Android 1.5 版本开始，Android 系统已经成为一个智能操作系统，Google 开始将 Android 系统的版本以甜品的名字命名。随着 Android 系统近年来的快速普及与发展，越来越多的厂商加入 Android 的阵营，根据 Gartner2015 年第四季的调查，Android 在智能手机市场的占有率为 80.7%，排名第 2 的苹果 iOS 为 17.7%。

因为 Android 系统发展迅速，版本众多，搭载 Android 系统各个版本的设备在现如今的市场上，并没有得到很好的统一，均有一定的占有率。图 1-3 是 Google 公司公布的 Android 各个版本的市场占有率，如图 1-3 所示。

从图 1-3 可知，在市面上占有率最高的为 Android 4.x，在构建 Android 应用时，采用新的版本和技术的同时，一定要考虑 Android 4.x 的兼容性问题。

Android 系统是基于 Linux 的智能操作系统，2007 年 11 月，Google 与 84 家硬件制造商、软件开发商及电信运营商组建开发手机联盟，共同研发改良 Android 系统。随后 Google 以 Apache 开源许可证的授权方式，发布了 Android 的源代码。也就是说，Android 系统是完全公开并且免费的，Android 系统的快速发展，也与它的公开免费有很大关系。

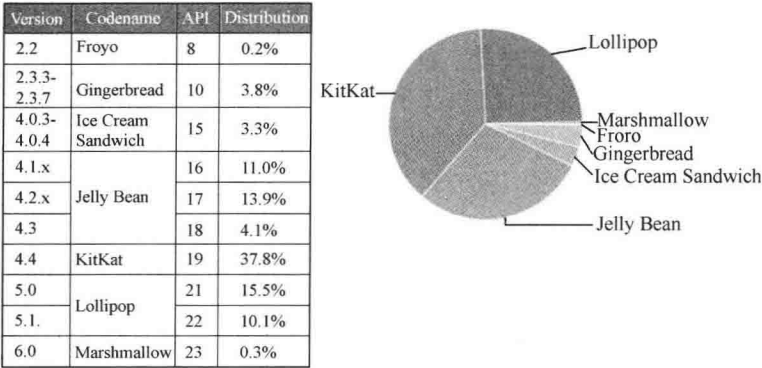


图 1-3 各版本市场占有率

1.1.2 Android 基本架构

Android 分为 5 个层，从高层到低层分别是应用程序层（Applications）、应用程序框架层（Application Framework）、系统运行库层（Libraries）、运行环境层（Android Runtime）和 Linux 核心层（Linux Kernel），如图 1-4 所示。

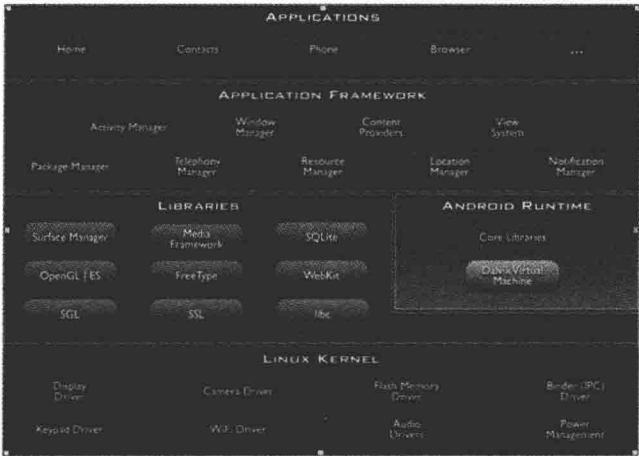


图 1-4 Android 基本架构

Applications、Application Framework 和 Android Runtime 的 Core Libraries 是 Java 程序，Android Runtime 的 Dalvik Virtual Machine 代码为运行 Java 程序而实现的虚拟机，Libraries 部分为 C/C++语言编写的程序库，Linux Kernel 部分为 Linux 内核和驱动。在 Application Framework 之下，由 C/C++的程序库组成，通过 JNI（Java Native Interface，Java 本地调用）完成从 Java 到 C 的调用。

1. 应用程序

所有的应用程序都是使用 Java 语言编写的，每一个应用程序由一个或者多个活动组成，

活动必须以 Activity 类为超类，活动类似于操作系统上的进程，但是活动比操作系统的进程要更为灵活，与进程类似的是，活动可以在多种状态之间进行切换。

利用 Java 的跨平台性质，基于 Android 框架开发的应用程序可以不用编译运行于任何一台安装有 Android 系统的平台，这一点正是 Android 的精髓所在。

2. 应用程序框架

应用程序的架构设计简化了组件的重用。任何一个应用程序都可以发布它的功能块，任何其他的应用程序都可以使用其所发布的功能块（不过得遵循框架的安全性限制），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。

隐藏在每个应用后面的是一系列的服务和系统，其中包括：

- 丰富而又可扩展的视图（Views），可以用来构建应用程序，它包括列表（Lists）、网格（Grids）、文本框（Text Boxes）、按钮（Buttons），还有可嵌入的 Web 浏览器；
- 内容提供者（Content Providers），使得应用程序可以访问另一个应用程序的数据（如联系人数据库），或者共享它们自己的数据；
- 资源管理器（Resource Manager），提供非代码资源的访问，如本地字符串、图形和布局文件（Layout Files）；
- 通知管理器（Notification Manager），使得应用程序可以在状态栏中显示自定义的提示信息；
- 活动管理器（Activity Manager），用来管理应用程序生命周期并提供常用的导航回退功能。

3. 系统运行库

（1）程序库

Android 包含一些 C/C++ 库，这些库能被 Android 系统中不同的组件使用。它们通过 Android 应用程序框架为开发者提供服务。

核心库主要包括基本的 C 库及多媒体库，以支持各种多媒体格式、位图和矢量字体、2D 和 3D 图形引擎、浏览器、数据库支持。

此外包括一个硬件抽象层，Android 并非所有的设备驱动都放在 Linux 内核里面，有一部分在用户空间实现。这么做的主要原因是可以避开 Linux 所遵循的 GPL 协议，一般情况下，如果要将 Android 移植到其他硬件去运行，只需要实现这部分代码即可，包括显示器驱动、声卡、相机、GPS、GSM 等。

（2）Android 运行库

Android 包括一个核心库，该核心库提供了 Java 编程语言核心库的大多数功能。

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。

Dalvik 虚拟机执行 (.dex) 的是 Dalvik 可执行文件，该格式文件针对小内存使用做了优化。

同时虚拟机是基于寄存器的，所有的类都经由 Java 编译器编译，然后通过 SDK 中的“dx”工具转化成 dex 格式由虚拟机执行。

Dalvik 虚拟机依赖于 Linux 内核的一些功能，比如线程机制和底层内存管理机制。

4. 运行时环境

Android 应用程序编写主要使用 Java 语言，而 Java 语言要想运行，需要一个虚拟机。

5. Linux 内核

Android 的核心系统服务依赖于 Linux 2.6 内核，如安全性、内存管理、进程管理、网络协议栈和驱动模型。

Linux 内核也同时作为硬件和软件栈之间的抽象层。另外，Linux 内核还对其做了部分修改，主要涉及以下两部分修改。

(1) Binder

IPC (Inter Process Communication)：提供有效的进程间通信，虽然 Linux 内核本身已经提供了这些功能，但 Android 系统很多服务都需要用到该功能，所以对进程间通信进行了重装封装。

(2) 电源管理

为手持设备节省能耗。

Android 应用开发主要关注应用程序层的开发，使用 Java 语言编写 Android 应用，一般包含 3 个部分，如表 1-2 所示。

表 1-2 Java 语言编写 Android 应用的部分

部分	作用
Java 语言	Java 语法
Java 虚拟机	为了实现一次编译到处可以运行的原则，Java 在编译连接以后，不产生目标机器语言，而是采用了 Java bytecode 这种 Java 公用指令，需要一个虚拟机来执行该指令
库	提供一些常用的库

Android 是以 Linux 操作系统为核心，并针对手机进行了专门的优化，例如电源管理、进程调度等。Linux 提供了操作系统最基本的功能。

1.1.3 Android 组件

Android 系统中有著名的 4 大组件：Activity、Service、BroadcastReceiver、ContentProvider。一个商业的 Android 应用程序，通常由多个基本的组件联合组成。这 4 大组件在使用时均需要在清单文件 AndroidManifest.xml 中进行注册，否则不予使用。本节将对这些组件进行简单的介绍，使读者对 Android 应用开发的内容有一个大致的认识，详细内容请参看本书 2.5 节。

1. 活动 (Activity)

Activity 是 Android 应用中最直接与用户接触的组件，它负责加载 View 组件，使其展现给用户，并保持与用户的交互。所有的 Activity 组件均需要继承 Activity 类，这是一个 Content 的间接子类，包装了一些 Activity 的基本特性。

View 组件是所有 UI 组件、容器组件的基类，也就是说，它可以是一个布局容器，也可以是一个布局容器内的基本 UI 组件。View 组件一般通过 XML 布局资源文件定义，同时 Android 系统也对这些 View 组件提供了对应的实现类。如果需要通过某个 Activity 把指定的 View 组件显示出来，调用 Activity 的 setContentView()方法即可，它具有多个重载方法，可以传递一个 XML 资源 ID 或者 View 对象。

例如：

```
LinearLayout layout=new LinearLayout(this);
setContentView(layout);
```

或者：

```
setContentView(R.layout.main);
```

Activity 为 Android 应用提供了一个用户界面，当一个 Activity 被开启之后，它具有自己的生命周期。Activity 类也对这些生命周期提供了对应的方法，如果需要对 Activity 各个不同的生命周期作出响应，可以重写这些生命周期方法实现。对于大多数商业应用而言，整个系统中包含了多个 Activity，在应用中逐步导航跳转开启这些 Activity 之后，会形成 Activity 的回退栈，当前显示并获得焦点的 Activity 位于这个回退栈的栈顶。

2. 服务 (Service)

Service 主要用于在后台完成一些无须向用户展示界面的功能实现。通常位于系统后台运行，它一般不需要与用户进行交互，因此 Service 组件没有用户界面展示给用户。Service 主要用于完成一些类似于下载文件、播放音乐等无须用户界面与用户进行交互的功能。

与 Activity 组件需要继承 Activity 类相似，Service 组件同样需要继承 Service 类，Service 类也是 Context 的间接子类，其中包装了一些 Service 的专有特性。一个 Service 被运行起来之后，它将具有自己独立的生命周期，Service 类中对其各个不同的生命周期提供了对应的方法，开发人员可以通过在 Service 中重写 Service 类中这些生命周期方法，来响应 Service 各个生命周期的功能实现。

3. 广播接收器 (BroadcastReceiver)

BroadcastReceiver 同样也是 Android 系统中的一个重要组件，BroadcastReceiver 代表了一个广播接收器，用于接收系统中其他组件发送的广播，并对其进行响应或是拦截广播的继续传播。

广播是一个系统级的消息，当系统环境发生改变时会发送一些广播供对应的程序进行接收响应，例如：接收到一条短信、开机、关机、插上充电器、插上耳机、充电完成等，均会发送一条广播供需要监听此类广播的应用进行响应。除了一些系统事件的广播，开发人员也可以自定义广播内容。但是大部分情况下，开发应用时主要用于接收系统广播并对其进行响应，很少需要发送自定义的广播。

使用 BroadcastReceiver 组件接收广播非常简单，只需要实现自己的 BroadcastReceiver 子类，并重写 onReceive() 方法，就能完成 BroadcastReceiver。而这个 BroadcastReceiver 对什么广播感兴趣，则需要对其进行另行配置。

4. 内容提供者 (ContentProvider)

Android 系统作为一个智能操作系统，需要系统中运行的应用程序都必须是相互独立的，各自运行在自己的 Dalvik 虚拟机实例中。在正常情况下，Android 应用之间不能进行实时的数据交换，而考虑到有些应用的数据需要对外进行共享，Android 系统提供了一个标准的数据接口 ContentProvider，通过应用提供的 ContentProvider，可以在其他应用中对这个应用暴露出来的数据进行增、删、改、查。

为应用程序暴露数据接口非常简单，只需要继承 ContentProvider 类，并且实现 insert()、delete()、update()、query() 等方法，使外部应用可对本应用的数据进行增、删、改、查。

5. 意图 (Intent)

虽然 Intent 并不是 Android 应用的组件，也无须专门在清单文件中配置，但是它对于 Android 应用的作用非常大。除了 ContentProvider 之外，其他组件的启动，均需要通过 Intent 进行指定。Intent 不仅可以明确指定一个 Android 组件进行启动，还可以提供一个标准的行为，再由 Android 系统配合意图过滤器来选定启动指定组件来完成任务。而 Intent 在开启组件的过程中，也可以进行各个组件间的数据传递。

1.2 开发环境

1.2.1 JDK 安装与配置

JDK (Java Development Kit) 是 Java 语言的软件开发工具包 (SDK)，主要用于移动设备、嵌入式设备上的 Java 应用程序。

1. JDK 下载与安装

首先下载 JDK，官方下载地址 <http://www.oracle.com/technetwork/Java/javase/downloads/index.html>，选择自己电脑系统的对应版本即可，如图 1-5 所示。

Java SE Development Kit 8u77		
You must accept the Oracle Binary Code License Agreement for Java SE to download this software.		
Thank you for accepting the Oracle Binary Code License Agreement for Java SE; you may now download this software.		
Product / File Description	File Size	Download
Linux ARM 32 Soft Float ABI	77.7 MB	jdk-8u77-linux-arm32-vfp-hflt.tar.gz
Linux ARM 64 Soft Float ABI	74.68 MB	jdk-8u77-linux-arm64-vfp-hflt.tar.gz
Linux x86	154.74 MB	jdk-8u77-linux-i586.rpm
Linux x86	174.92 MB	jdk-8u77-linux-i586.tar.gz
Linux x64	152.76 MB	jdk-8u77-linux-x64.rpm
Linux x64	172.96 MB	jdk-8u77-linux-x64.tar.gz
Mac OS X	227.27 MB	jdk-8u77-macosx-x64.dmg
Solaris SPARC 64-bit (SVR4 package)	139.77 MB	jdk-8u77-solaris-sparcv9.tar.Z
Solaris SPARC 64-bit	99.06 MB	jdk-8u77-solaris-sparcv9.tar.gz
Solaris x64 (SVR4 package)	140.01 MB	jdk-8u77-solaris-x64.tar.Z
Solaris x64	96.18 MB	jdk-8u77-solaris-x64.tar.gz
Windows x86	182.01 MB	jdk-8u77-windows-i586.exe
Windows x64	187.31 MB	jdk-8u77-windows-x64.exe

图 1-5 选择 JDK 版本

双击安装文件进行安装，并选择安装目录，如图 1-6 和图 1-7 所示。

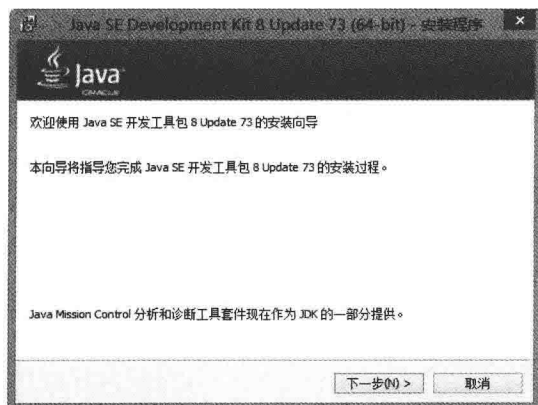


图 1-6 JDK 安装欢迎界面

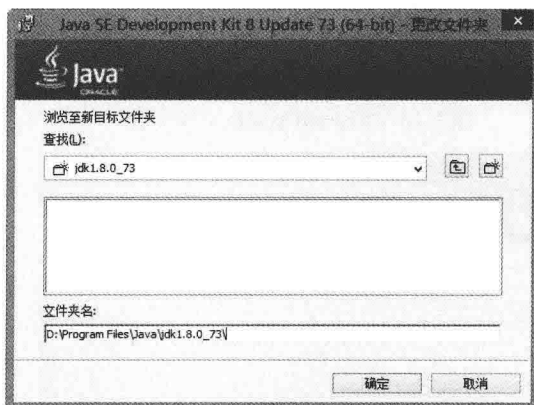


图 1-7 更改安装目录

2. 设置环境变量

首先打开“系统属性”→“高级”选项卡中的“环境变量”按钮，如图 1-8 所示。设置 JAVA_HOME 和 Path 环境变量，如图 1-9 所示。

1.2.2 Android Studio

1. 安装

Android Studio 是编写和调试 Android 应用的工具，安装过程如图 1-10~图 1-14 所示。其中，关于安装路径和 SDK 路径选择，如图 1-12 所示；内存选择推荐的 2GB，如图 1-13 所示。

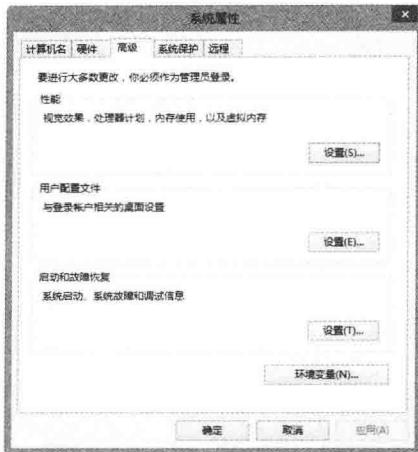


图 1-8 系统属性



图 1-9 环境变量设置



图 1-10 Android Studio 欢迎界面

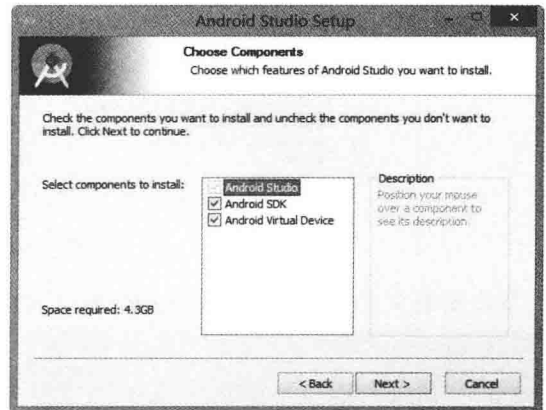


图 1-11 组件选择界面

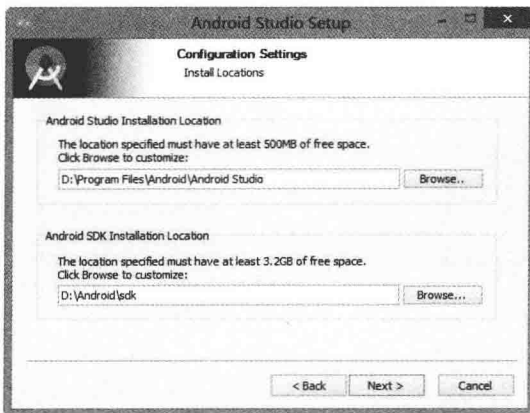


图 1-12 安装路径和 SDK 路径

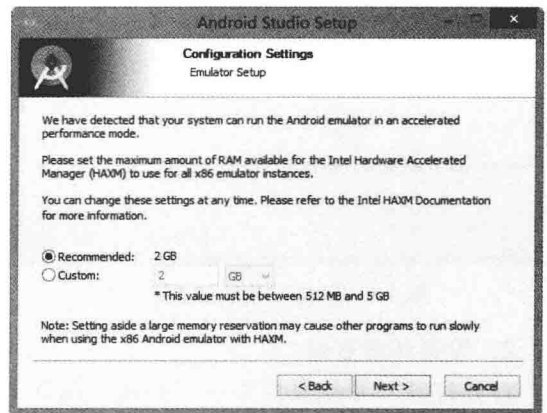
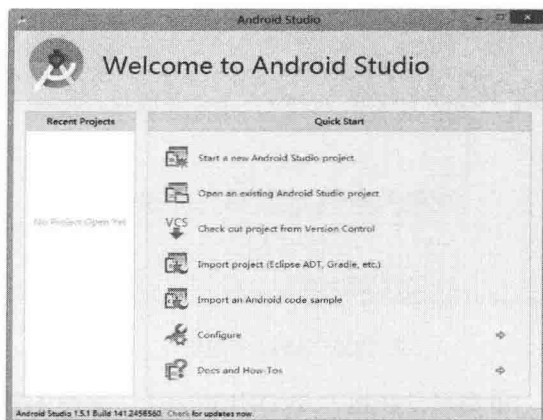


图 1-13 内存设置

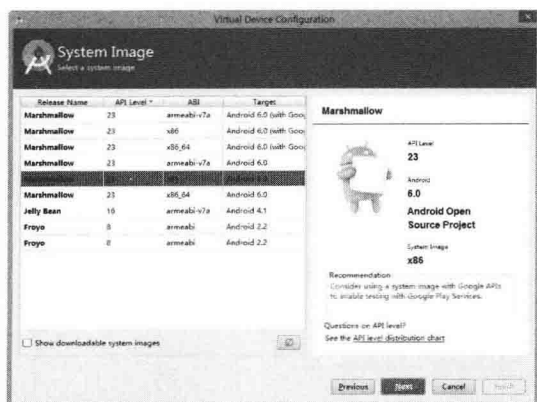
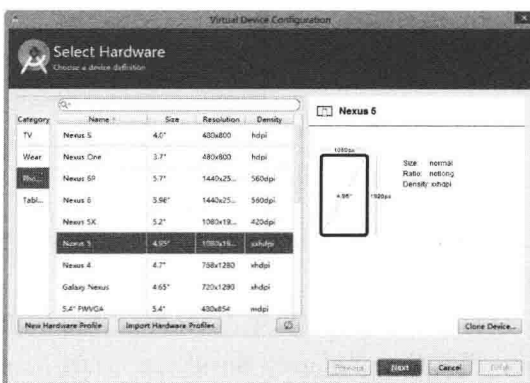
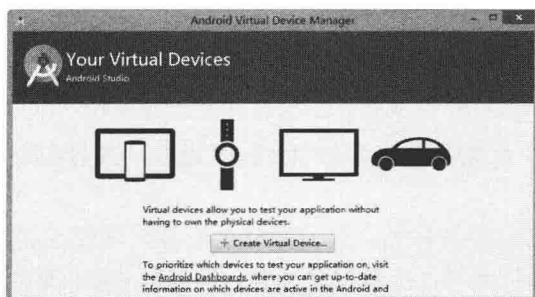
2. AVD Manager

调试和运行 Android 项目可以使用 AVD。AVD 的全称为：Android Virtual Device，就是 Android 运行的虚拟设备，又称作 Android 的模拟器。

下面演示如何在 Android Studio 里创建一个 AVD。



首先打开 AVD 管理界面，单击“Create Virtual Device”按钮，如图 1-16 所示。然后在硬件选择窗口选择相应设备，如图 1-17 所示。在系统镜像窗口选择喜欢的镜像，如图 1-18 所示。最后输入 AVD 名称，如图 1-19 所示。



单击“Finish”按钮，开始进行 AVD 初始化，如图 1-20 所示。创建成功后，可以在管理窗口看到刚刚创建的 AVD，如图 1-21 所示。可以根据不同的需求，创建不同的 AVD。