

21世纪高等院校移动开发人才培养规划教材

21Shiji Gaodeng Yuanxiao Yidong Kaifa Rencai Peiyang Guihua Jiaocai

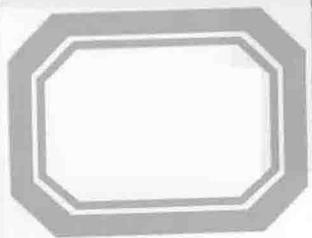
Android移动应用开发 项目教程

李新辉 邹绍芳 主编 陈云志 周昕 吴红娉 副主编

Android Mobile Application Development Tutorial

精心设计7个工作项目，全程贯彻“做中学”理念
每个项目按照任务引入、开发过程、知识拓展编写
提供教学项目源代码、教学录像、课件等资源

 人民邮电出版社
POSTS & TELECOM PRESS



21世纪高等院校移动开发人才培养规划教材

Shiji Gaodeng Yuanxiao Yidong Kaifa Rencai Peiyang Guihua Jiaocai

Android移动应用开发 项目教程

李新辉 邹绍芳 主编

李新辉 邹绍芳 主编 陈云志 周昕 吴红娉 副主编

Android Mobile Application
Development Tutorial

人民邮电出版社

北京

图书在版编目(CIP)数据

Android移动应用开发项目教程 / 李新辉, 邹绍芳主
编. — 北京: 人民邮电出版社, 2014.9
21世纪高等院校移动开发人才培养规划教材
ISBN 978-7-115-35995-7

I. ①A… II. ①李… ②邹… 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(2014)第127247号

内 容 提 要

本书通过精心设计的7个工作项目, 全程贯彻“做中学”理念, 先实践认知, 后理论拓展, 由浅入深, 让读者逐步掌握Android应用程序用户界面布局设计、2D绘图和游戏设计、传感器、网络访问、LBS地图相册开发等技术, 在项目实践过程中理解Android基本开发技术、调试方法、项目重构技巧和规范的编码风格, 掌握开发Android应用程序的方法。

本书项目1阐述Android应用程序开发环境的搭建; 项目2讲解BMI体质指数计算器的开发, 着重阐述基本组件的使用; 项目3讲解ColorCard色卡应用程序的开发, 着重阐述通过灵活布局组件构建动态界面的技术; 项目4讲解PT拼图游戏的开发, 着重阐述2D绘图知识、线程和游戏开发技术; 项目5讲解PhoneSecurity手机防盗器的开发, 着重阐述传感器、短信和电子邮件发送技术; 项目6讲解NewsReader新闻阅读器的开发, 着重阐述ListView等高级组件的使用和XML/JSON数据处理技术; 项目7讲解MapPhotos地图相册的开发, 着重阐述在应用程序中使用地图和相机拍照的技术。本书配有教学视频、习题参考等教学资源, 方便老师教学。

本书可作为各高等院校移动互联网Android开发技术的教材, 也可作为软件开发人员的参考用书。

◆ 主 编 李新辉 邹绍芳

副 主 编 陈云志 周 昕 吴红娉

责任编辑 王 平

责任印制 焦志炜

◆ 人民邮电出版社出版发行 北京市丰台区成寿寺路11号

邮编 100164 电子邮件 315@ptpress.com.cn

网址 <http://www.ptpress.com.cn>

三河市海波印务有限公司印刷

◆ 开本: 787×1092 1/16

印张: 20.75

字数: 535千字

2014年9月第1版

2014年9月河北第1次印刷

定价: 46.00元

读者服务热线: (010)81055256 印装质量热线: (010)81055316

反盗版热线: (010)81055315

广告经营许可证: 京崇工商广字第0021号

前言 PREFACE

Android 是 Google 公司和 OHA (开放手机联盟) 开发的基于 Linux 的开源操作系统, 主要用于智能手机、平板计算机等智能移动设备上。经过短短几年的发展, Android 系统在全球得到了大规模的推广, 除智能手机和平板计算机外, 还渗透到了智能电视、游戏机、可穿戴设备、汽车等领域, 且有“连接一切”的趋势。据不完全统计, Android 系统已经占据了全球智能手机操作系统 80% 以上的市场份额, 中国市场占有率甚至超过 90%。

目前, 大部分高等院校作为高技能型人才的主要培养基地, 大部分院校都开设了计算机类相关专业。随着移动设备在人们日常工作和生活中的普及, 近年来, 不少院校还新增了移动互联网专业, 在设置的课程中基本上都是把 Android 作为主要的移动应用开发技术。与此同时, 市面上已经出现了一些 Android 技术开发的参考书, 但针对本科或高职院校“职业教育过程”的 Android 技术教材不多, 特别是学完之后就能具备实际项目开发能力的实用技术类教材更少。基于这一考虑, 特编写本书。

本书具备如下 4 大特点。

(1) 以 7 个相对独立的工作任务组织内容, 践行“做中学”理念, 进度符合学生认知规律, 内容编排兼顾趣味性、知识性和实用性。

本书设计了 7 个相对独立的工作任务, 除首个任务是搭建 Android 开发环境以外, 其余 6 个工作任务都是经过精心设计的实用项目, 包括体质指数计算器、色卡、拼图游戏、手机防盗器、新闻阅读器和地图相册。在这些项目的开发过程中, 融入了 Android 开发的绝大部分技术, 而且大部分项目都可以在 Google Play 市场找到相类似的产品。

在设计 6 个项目中, 每个项目都是按照“学习提示、任务引入、开发过程、知识拓展、问题实践”5 大步骤推进。

【学习提示】简要概述了本单元的任务目标、技术内容、知识点和技能目标。

【任务引入】阐述了项目开发背景, 并提出待实现的具体功能, 通过程序运行界面截图让学生对开发目标有初步的认识, 然后让学生亲手在手机或模拟器上体验最终完成的项目程序的实际功能, 从而对开发目标有更加直观的理解。

【开发过程】教材中提供了详细的项目设计和开发步骤, 代码也不是直接通篇给出, 而是根据功能模块过程按需提供代码片段。代码片段中包含了较为详尽的注释, 之后通常会进一步解释代码片段的含义。

【知识拓展】集中阐述本项目开发过程中涉及的知识内容, 并适当进行拓展, 完成“先实践体会, 后理论归纳”的自然学习认知过程。

【问题实践】提出了项目开发过程中可以完善或扩展的技术功能, 同时还提供了必要的解答提示或资料建议。问题实践部分的解答提示均以电子文档的形式提供下载。

此外, 项目设计编排兼顾趣味性和实用性。兴趣是最好的老师, 能让学生易于接受就相当于成功了一半。比如, 拼图游戏、手机防盗器这样的 APP (应用程序) 很易被学生理解和喜爱; 新闻阅读器是当前热门 RSS 阅读器的原型, 国内外各大门户 (如新浪、搜狐) 以及传统媒体 (如南方周末等) 都开发了各自的手机新闻阅读器 APP, 选题有一定代表性和实用性; 地图相册则是一个典型的 LBS 运用, 综合使用了地图、相册、拍照、地理定位等技术, 创意

新颖，只要适当扩展即可成为一款实用 APP，如果考虑和云平台结合的话，还可实现一个基于网络的地图相册，甚至还可以加入流行的社交功能等。

(2) 教学过程注重培养学生思考问题的习惯，教学内容兼顾够用原则且适当扩展。

在 6 个具体的项目开发过程中，不是所有待解决的问题都会给出解决方案，而是在适当的场合保留一点让学生能自主解决的问题，让他们独立完成，防止学生“盲目照做”，调动其主观能动性。根据教学实践发现，很多只会跟着做的学生即使在“顺利”完成任务后还是没什么实际收获，所以必须提供一个关联性的动脑区间，这样比较有利于巩固学习效果。

(3) 知识技术兼顾实用性、新颖性和前瞻性。

因为版本变迁的缘故，Android SDK 中提供的很多 API 被标记为“过时”，因此项目中凡是涉及这样的内容都使用了 Android SDK 推荐的实现方法。比如，色卡程序和新闻阅读器本来可以使用 TabActivity 实现，但考虑到 TabActivity 已过时，就改用了其他办法；色卡程序采用的是自主设计布局界面模拟“选项卡”，学生在完成这一项目后能够加深理解一些标准组件的工作机制；新闻阅读器采用了 Fragment 这一重要的技术进行实现；地图相册用到了 Google Map V2 API，没有使用早期的组件实现，对于 Gallery 也是采用 SDK 推荐的 HorizontalScrollView 组件自主实现相册浏览的功能。

本课程的前导课程是 Java 语言基础，很多学生刚开始编写 Android 应用程序时，由于对 Android 开发大量涉及的内部类、匿名类、泛型、线程等 Java 语法现象不熟悉，导致后续学习困难。因此，本书在开发环境搭建环节的拓展部分提供了几个简单的例子让学生复习巩固这些关键 Java 语法的内容。还有，Android SDK 自带的 Sample 系列项目（如 ApiDemo 等），是学习 Android 开发技术的极佳素材，Hierarchy Viewer 和 UiAutomator Viewer 是分析复杂 Android 程序界面布局设计技巧的强大工具，对学生模仿学习一些优秀软件界面的设计是非常有意义的，这类内容在拓展部分也有介绍。

(4) 为学生适应企业级开发做好准备。

学习 Android 开发技术最终是要为企业开发产品服务的，因此项目设计过程中用到的技术大多都不是浅层的。前 3 个项目的设计出于学习目的，后 3 个项目则加深了难度和深度。比如，色卡程序是通过布局组件的灵活运用设计出选项卡外观，地图相册则是完全自定义一个相机界面实现拍照功能。本书对实际开发工作中大量使用的 XML/JSON 数据解析做了详细阐述，还提供了 ListView 下拉刷新组件、侧滑菜单库的开源实现等多种参考资源。

总的来说，本书努力成为一本为学生而写的书，既适合教学，也适合学生自学，通过 7 个项目案例融入 Android 开发用到的大部分技术和知识内容，学生学习到的不是零散的知识点，而是在项目实践过程中理解体会 Android 开发技术，项目案例选择兼顾趣味性和实用性，通过学生的主动参与、综合运用和开发创新，培养学生的实际编程开发能力，提高了学生的学习兴趣。

本书项目开发过程中的代码格式约定如下。

```
public class GameView extends View {
    ...
    private Paint paint;    // 绘制几何图形的画笔
    // 存储所有拼图块的动态数组
    private List<PuzzleCell> puzzCells = new ArrayList<PuzzleCell> ();
    ...
}
```

其中，阴影部分是新增的代码，阴影部分前后通常会保留少许代码片段，可借助这些预留的代码片段确定阴影部分代码的具体位置。

另外，本书还提供了丰富的课程资源，包括教学项目源代码、主要内容教学录像、课件资料、习题答案等，可以到人民邮电出版社教学服务与资源网（<http://www.ptpedu.com.cn>）免费下载使用。

本书参考学时为 90 学时，其中各项目的学时分配推荐如下。

	项 目	推荐学时
1	Android 应用程序开发环境搭建	3
2	BMI 体质指数计算器的开发	9
3	ColorCard 色卡程序的开发	12
4	PT 拼图游戏的开发	18
5	PhoneSecurity 手机防盗器的开发	15
6	NewsReader 新闻阅读器的开发	15
7	MapPhotos 地图相册的开发	18

本书由杭州职业技术学院的李新辉和杭州电子科技大学的邹绍芳担任主编，杭州职业技术学院陈云志、周昕、吴红娉担任副主编，在编写过程中得到了人民邮电出版社的大力指导和支持，杭州电子科技大学徐彤、陈定坝、李丹阳和中国计量学院曹建军、张晓峰、应江娇等同学对本书提出了很多建设性的意见，在此谨向以上单位和人员致以诚挚的谢意。

由于编写组人员技术水平有限，书中难免存在不足之处，恳请广大读者批评指正，任何疑问、宝贵意见和建议请发邮件至 lixinhuixdx@163.com。

Android 移动应用开发项目教程教材编写组

2014 年 4 月

目 录 CONTENTS

项目 1 Android 应用程序开发环境搭建 1

1.1 安装文件准备	1	1.4.2 安装 APK 应用到模拟器	12
1.2 开发环境安装和配置	2	1.4.3 Android 技术架构	13
1.3 Android 应用程序编写	6	1.4.4 Java 语法补充	15
1.4 知识拓展	10	1.5 问题实践	18
1.4.1 在设备上运行应用程序	10		

项目 2 BMI 体质指数计算器的开发 19

2.1 项目引入	19	2.7.1 Activity	35
2.2 BMI 项目准备	20	2.7.2 RelativeLayout	38
2.3 BMI 界面设计	20	2.7.3 AndroidManifest	39
2.4 BMI 功能实现	22	2.7.4 ApiDemo	40
2.5 BMI 重构	26	2.7.5 API Reference	42
2.6 BMI 属性菜单	29	2.8 问题实践	43
2.7 知识拓展	35		

项目 3 ColorCard 色卡程序的开发 44

3.1 项目引入	44	3.6 检索功能实现	69
3.2 ColorCard 项目准备	45	3.7 辨色功能实现	70
3.3 ColorCard 界面设计	46	3.8 知识拓展	77
3.3.1 主界面布局	46	3.8.1 LinearLayout	77
3.3.2 色卡界面布局	50	3.8.2 px/dp/sp	79
3.3.3 检索界面布局	52	3.8.3 Debug	80
3.3.4 辨色界面布局	53	3.8.4 UI Viewer	82
3.4 选项卡切换	54	3.9 问题实践	84
3.5 色卡功能实现	63		

项目4 PT 拼图游戏的开发 85

4.1 项目引入	85	4.9 拼图归位音效	119
4.2 拼图游戏项目准备	86	4.10 游戏进度自动保存	122
4.3 拼图游戏背景显示	89	4.11 知识拓展	128
4.4 拼图游戏界面设计	91	4.11.1 背景音乐	128
4.5 拼图块分割	96	4.11.2 SurfaceView	129
4.6 拼图块触摸和移动	102	4.11.3 游戏中的动画	136
4.6.1 触摸功能实现	102	4.11.4 Android 应用打包	138
4.6.2 移动功能实现	105	4.11.5 游戏引擎	140
4.6.3 移动性能优化	107	4.11.6 给初学者的建议	140
4.7 拼图块吸附与归位	114	4.11.7 连连看/消色块原理	143
4.8 拼图游戏启动动画	119	4.12 问题实践	150

项目5 PhoneSecurity 手机防盗器的开发 151

5.1 项目引入	151	5.7 SIM 卡检测和短信发送	167
5.2 PhoneSecurity 项目准备	152	5.8 电子邮件发送	170
5.3 距离检测与报警	154	5.9 知识拓展	179
5.4 防盗功能实现	157	5.9.1 Service	179
5.5 追回技术分析	160	5.9.2 Broadcast Receiver	182
5.6 手机信息保存	160	5.10 问题实践	184

项目6 NewsReader 新闻阅读器的开发 185

6.1 项目引入	185	6.8 新闻内容查看	215
6.2 NewsReader 项目准备	186	6.9 知识拓展	221
6.3 NewsReader 界面设计	187	6.9.1 Fragment	221
6.3.1 主界面设计	187	6.9.2 HttpClient	224
6.3.2 底部导航栏设计	189	6.9.3 XML/JSON	225
6.4 导航栏切换	193	6.9.4 Notification	226
6.5 新闻获取	198	6.9.5 ListView	228
6.6 RSS 数据源解析	203	6.9.6 Handler/AsyncTask	230
6.7 新闻条目加载	208	6.9.7 Android SDK Source	234
		6.10 问题实践	236

7.1 项目引入	237	7.9.1 GoogleMap	304
7.2 MapPhotos 项目准备	238	7.9.2 Camera	307
7.3 相册条目实现	242	7.9.3 SharedPreferences	308
7.4 地图实现	254	7.9.4 SQLite	309
7.5 相机拍照实现	268	7.9.5 ContentProvider	311
7.6 相册数据保存	281	7.9.6 Intent	314
7.7 地图相册实现	287	7.9.7 Context	318
7.8 图库浏览	296	7.9.8 开发资源参考	319
7.9 知识拓展	304	7.10 问题实践	322

Android 应用程序开发 环境搭建

【学习提示】



- 项目目标：配置 ADT 开发环境，创建 Android 应用程序
- 知识点：Android 发展史；Android 技术架构；Java 关键字（内部类、匿名类、匿名类的对象、泛型、集合元素循环、线程）
- 技能目标：能在 ADT 开发环境中创建 Android 应用程序；在真实设备上运行 Android 应用；安装外部 APK 应用到模拟器上

1.1 安装文件准备

Android 是建立在 Linux 系统基础上的一个面向移动设备的操作系统，它被设计成通过名为 Dalvik 的虚拟机来执行应用程序，Android 支持 Java 语言编写的应用程序。开发 Android 应用程序的过程和普通的 Java 应用程序没有太大区别。当然，使用 Java 语言编写 Android 应用程序时，需要转换为 Dalvik 的虚拟机指令才能在 Android 平台上运行。实际上，Android 同样支持 C/C++ 语言编写的程序，但一般情况下都是使用 Java 进行开发，除非在一些特定的场合才会用到 C/C++ 语言，比如高性能游戏、密集计算、硬件控制或软件移植等。

在开发 Android 程序之前，需要先搭建相应的开发环境。本书所用的 Android 开发环境主要包括 JDK、Android SDK、Eclipse 和 ADT 等软件包，它们都可以通过互联网免费下载。其中，JDK 即 Java Development Kit，是编写 Java 应用程序所必需的开发包，Android SDK 是开发 Android 应用程序所必需的软件集合，Eclipse 则提供了一个功能强大的通用集成开发环境，ADT 是用于在 Eclipse 上开发调试 Android 应用程序的一个功能插件。

(1) 下载 JDK 软件包。找到 [www.oracle.com](http://www.oracle.com/technetwork/java/javasebusiness/downloads/java-archive-downloads-javase6-419409.html) 站点，在其中的页面 <http://www.oracle.com/technetwork/java/javasebusiness/downloads/java-archive-downloads-javase6-419409.html> 中找到相应平台的下载链接进行下载（Win32、Win64、Mac、Linux 等）。本书使用的 32 位和 64 位版本的 JDK 软件分别是 `jdk-6u43-windows-i586.exe` 以及 `jdk-6u43-windows-x64.exe`，尽管目前最新的软件是 JDK 7，但由于 JDK 7 引入了部分新语法现象，这些新的语法现象目前还没有全部获得 ADT 的支持，因此推荐选择 JDK 6，如图 1.1 所示。

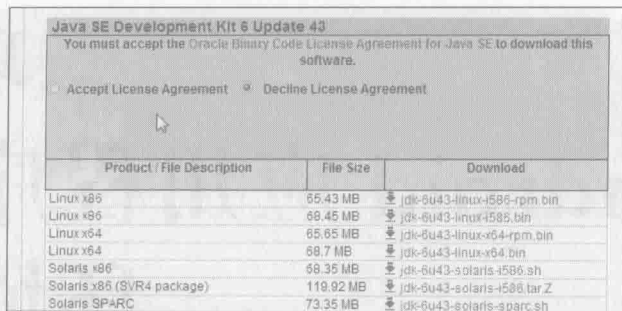


图 1.1 JDK 6 软件包下载

(2) 下载 ADT 集成开发工具。打开 developer.android.com 站点下载 Android SDK 和 ADT 插件。对于 Windows 平台，已经提供了一个集成各种必备插件的 ADT Bundle 下载，它包括 Eclipse、Android SDK 和 ADT 等系列软件包。

在 <http://developer.android.com/sdk/index.html> 页面中，单击右侧 Download the SDK(ADT Bundle for Windows) 链接，选定下载 32 位还是 64 位版本，勾选同意软件协议，这样就开始了下载过程，如图 1.2 所示。

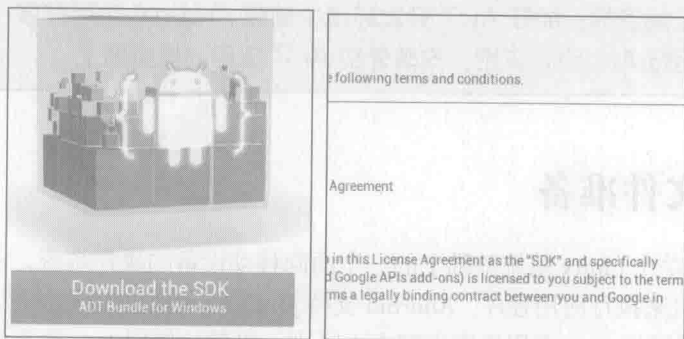


图 1.2 ADT 软件包下载

截至本书出版时，Google 已经发布了支持最新 Android 4.4 版本的 ADT 开发工具。值得注意的是，最新版本 ADT 开发工具新建的 Android 项目结构和源代码，相比之前版本有少许变化。为避免因开发工具版本问题导致的学习障碍，请下载本书编写时使用的 32 位或 64 位版本的 ADT 开发环境压缩包，它们分别是 `adt-bundle-windows-x86-20130219.zip` 和 `adt-bundle-windows-x86_64-20130219.zip`，可以到本书配套教学资源网页下载。



[提示]

1.2 开发环境安装和配置

1. JDK 和 ADT 安装

Android 开发环境的搭建只要两步，即安装 JDK 和 ADT Bundle。

(1) 找到下载的 JDK 安装文件，双击执行以启动安装过程。在本书编写时，安装的是 Windows 7 64 位版本的 JDK 6 软件包。JDK 的安装过程很简单，可根据需要在安装过程的第 2 步指定好安装目录，如图 1.3 所示，其余步骤默认即可。

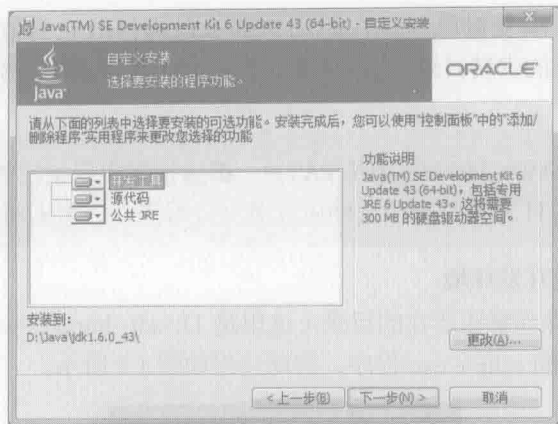


图 1.3 设定 JDK 安装目录

(2) 安装 ADT 集成开发环境。考虑到已下载的 ADT Bundle 已经整合了 Android SDK、Eclipse 和 ADT 插件等软件包，因此只需将下载的文件解压到某个目录中就完成了安装工作。当然，为避免将来出现意外，在指定的目录中最好不要包含汉字或其他特殊字符。

2. 环境变量设置

为方便起见，最好在 Windows 中添加 JAVA_HOME 和 PATH 这两个环境变量，它们会在后续的开发工作中用到。其中 JAVA_HOME 代表 JAVA 运行环境所在的目录，PATH 则提供了可执行命令的搜索目录。比如，在命令提示符窗体中输入 javac 命令时，Windows 会在 PATH 环境变量设置的目录中去寻找对应的可执行程序。

(1) 打开系统属性中的环境变量窗体，如图 1.4 所示。



图 1.4 系统属性-环境变量

(2) 单击环境变量窗体中的“新建”按钮，设定如下两个环境变量。其中，等号左侧是环境变量名，等号右侧是环境变量值。

```

JAVA_HOME=D:\Java\jdk1.6.0_43
PATH=D:\Java\jdk1.6.0_43\bin;D:\adt-bundle-windows\sdk\platform-
tools;D:\adt-bundle-windows\sdk\tools;

```

当然，如果某个环境变量已经存在的话，那么只需修改它的值，否则就要新建它。JAVA_HOME 代表 JDK 的安装目录，PATH 则包含了 JDK 和 Android SDK 可执行程序所在的目录。



无论 JAVA_HOME 还是 PATH，都应根据实际指定的安装目录进行设置。另外，PATH 设置的目录之间应以英文分号隔开，否则所设置的内容将无效。

3. 启动 ADT 集成开发环境

(1) 找到 ADT 软件包解压所在的目录（这里是 D:\adt-bundle-windows 目录），双击执行其中的 eclipse 目录下的 eclipse.exe 程序，启动过程如图 1.5 所示。



图 1.5 ADT 启动过程

(2) 在首次启动 ADT 集成开发环境时，会要求指定一个 Workspace 工作目录，该目录是 ADT 创建项目的默认保存路径。设置好 Workspace 对应的文件夹，勾选“Use this as the default and do not ask again”（使用当前指定路径作为默认的 Workspace 目录，并且不再询问），然后单击“OK”按钮，如图 1.6 所示。

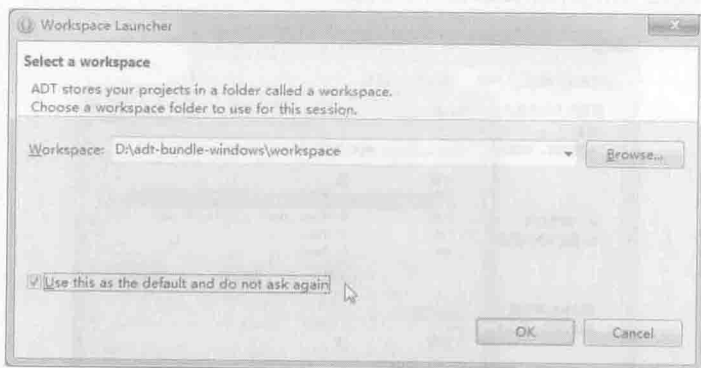


图 1.6 工作目录设置

(3) 进入 ADT 集成开发环境后，首先看到的是欢迎界面。此时，单击左上角“Android IDE”标题右侧的关闭按钮，将欢迎界面关闭即可，如图 1.7 所示。

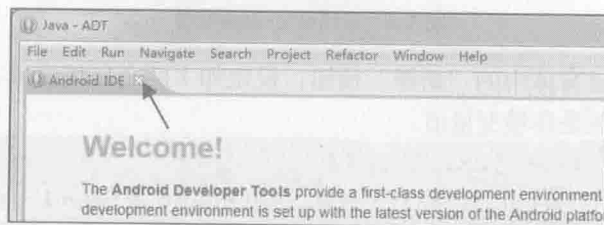


图 1.7 ADT 初始界面

(4) 为方便起见, 最好将 `adt-bundle-windows` 目录下的 `eclipse.exe` 可执行程序创建一个快捷方式放到桌面上, 方法是: 在 `eclipse.exe` 图标上单击鼠标右键, 选择弹出菜单中的“发送到”→“桌面快捷方式”, 然后将桌面上 `eclipse.exe` 快捷方式的名字改为“`android_eclipse`”之类的名字, 如图 1.8 所示。

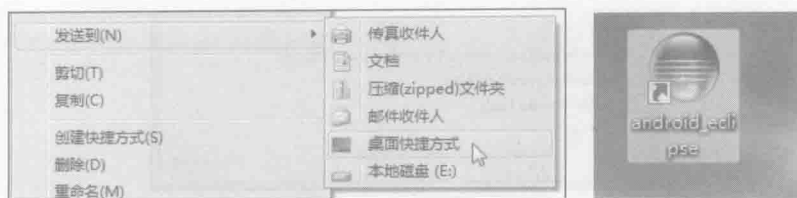


图 1.8 创建 ADT 快捷方式

4. 升级 Android SDK

对于基本的 Android 应用程序开发, 前面 ADT 集成开发环境的安装和配置工作就已经准备好了, 因为 `adt-bundle-windows` 本身已附带一个包含基本内容的 Android SDK。为便于后续的开发工作, 这里阐述一下如何升级 Android SDK 软件包。当然, 这些工作现在也可以不做, 因为本书配套教学资源网页提供下载的 `adt-bundle-windows-x86-20130219.zip` 压缩包中已包含完整内容的 Android SDK。

(1) 单击 ADT 集成开发环境主界面的“Window”菜单中的“Android SDK Manager”项, 如果网络连接正常可用的话, “Android SDK Manager”窗体中将列出当前已安装的 Android SDK 软件包和其他可用版本的 Android SDK 列表, 如图 1.9 所示。

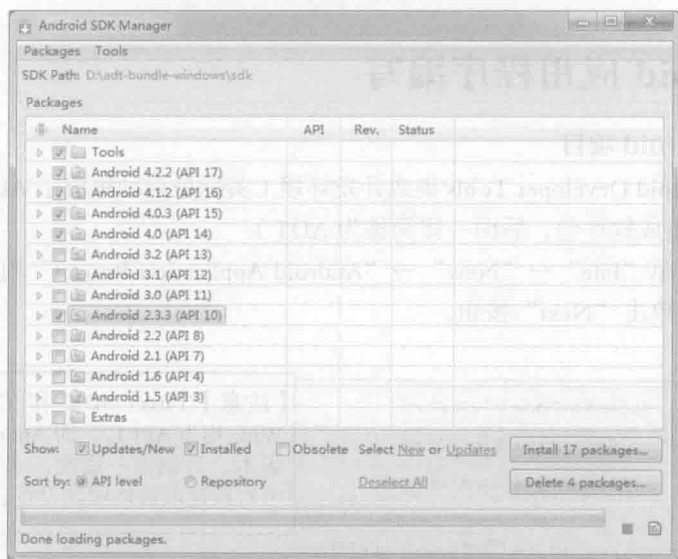


图 1.9 Android SDK 管理器

(2) 勾选图 1.9 中列出的 2.3、4.0、4.1、4.2 等主要版本的 Android SDK, 或者全部勾选, 然后单击“Install x packages”按钮, 此时就会通过网络从 Google 服务器下载这些软件包。视所选软件包的多少和网速快慢, 下载时间可能需要耗费数小时之久, 数据量为 2~5GB。

如果已经可以正常下载 Android SDK 软件包, 下面的第 (3)~(5) 步请直接忽略不做。

(3) 众所周知,如果在“Android SDK Manager”窗体中无法显示可用的 Android SDK 版本完整列表,可试着单击“Android SDK Manager”窗体主菜单“Tools”→“Options”项,勾选“Force https://... Sources to be fetched using http://...”(强制使用 http 协议下载 https 协议的源内容),然后单击“Close”按钮关闭设置窗体,如图 1.10 所示。

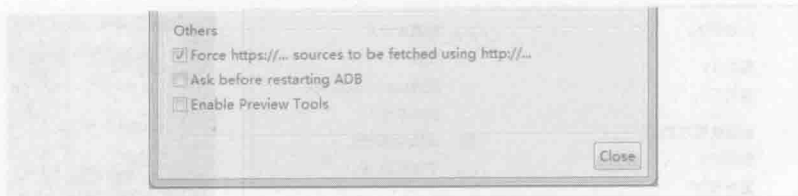


图 1.10 Options 窗体

(4) 单击“Android SDK Manager”窗体主菜单“Packages”→“Reload”,重新加载 Android SDK 版本的列表。

(5) 如果加载仍然失败的话,此时只能通过 HTTP 代理服务器来连接了,最好是使用额外的代理服务器。方法是:在第(3)步的设置窗体中,填写可用的 HTTP 代理服务器和正确端口号,如图 1.11 所示,然后再通过第(4)步的 Reload 重试。

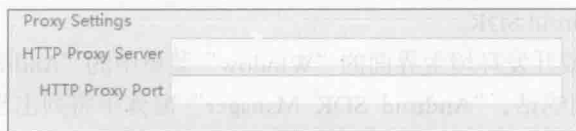


图 1.11 Options 窗体中的代理设置

1.3 Android 应用程序编写

1. 创建 Android 项目

(1) 启动 Android Developer Tools 集成开发环境(实际是启动集成了 ADT 插件、Android SDK 和 Eclipse 的集成软件包,后面一律简称为 ADT)。

(2) 单击主菜单“File”→“New”→“Android Application Project”项,按图 1.12 所示的内容设置,然后单击“Next”按钮。



图 1.12 新建 Android 项目

(3) 在项目设置界面,保持默认勾选项,直接单击“Next”按钮,如图 1.13 所示。

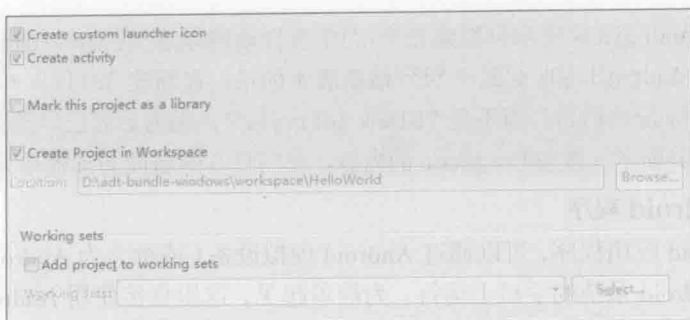


图 1.13 新建项目设置

(4) 在配置程序图标界面，直接按默认设置，单击“Next”按钮进入下一步，如图 1.14 所示。

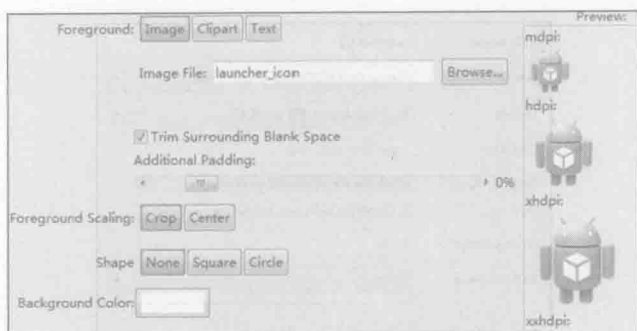


图 1.14 程序图标配置

(5) 在新建 Activity 界面，保持勾选“Create Activity”项，并选中“Blank Activity”以创建一个空白的 Activity，单击“Next”按钮，如图 1.15 所示。

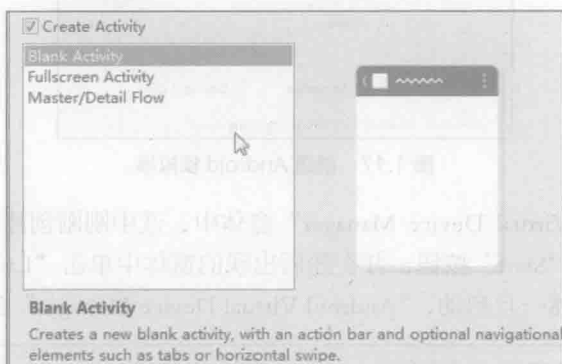


图 1.15 新建的 Activity 类型

(6) 在最后一步，保持默认设置不变，直接单击“Finish”按钮完成项目的创建工作，如图 1.16 所示。

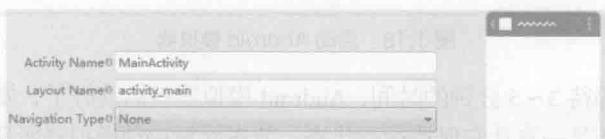


图 1.16 设定 Activity 和布局的名字



[提示]

Android 应用项目创建完毕, ADT 会自动编译这里的 HelloWorld 程序。另外, 如果 Android SDK 已经升级到最新版本的话, 在新建 Activity 界面时应该选择 “Empty Activity” 而不是 “Blank Activity”, 因为后者已经改为使用 Fragment 构建界面了。有关 Fragment 的内容, 在项目 6 的新闻阅读器单元中将予以详述。

2. 运行 Android 程序

要运行 Android 应用程序, 可以通过 Android 虚拟设备 (或称之为 Android 模拟器) 运行, 也可以在一部 Android 系统的手机上运行。为简单起见, 这里直接使用 Android 模拟器, 本书大部分场合都是在模拟器中进行开发工作的。

(1) 单击 ADT 主菜单 “Window” → “Android Virtual Device Manager” 项, 在出现的窗体中单击 “New” 按钮, 然后按如图 1.17 所示的内容进行设置, 最后单击 “OK” 按钮完成 Android 模拟器的创建。

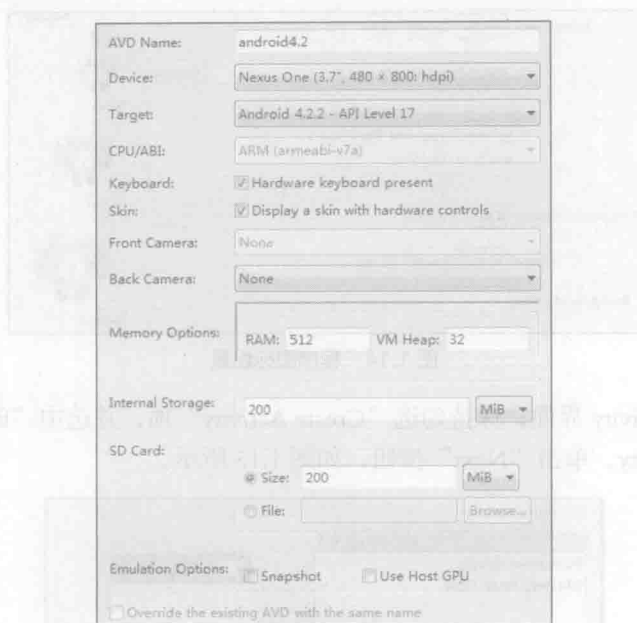


图 1.17 创建 Android 模拟器

(2) 在 “Android Virtual Device Manager” 窗体中, 选中刚刚创建的 android4.2 模拟器, 如图 1.18 所示, 单击 “Start” 按钮, 并在随后出现的窗体中单击 “Launch” 按钮来启动这个模拟器。Android 模拟器一旦启动, “Android Virtual Device Manager” 窗体就可以关闭了。

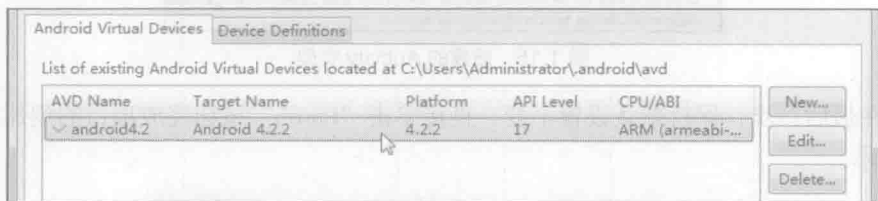


图 1.18 启动 Android 模拟器

不出意外的话, 等待 3~5 分钟的时间, Android 模拟器就启动好了, 如图 1.19 所示。一旦启动, 请不要关闭它, 而是一直让它保持运行状态, 节省反复启动模拟器所耗费的时间。