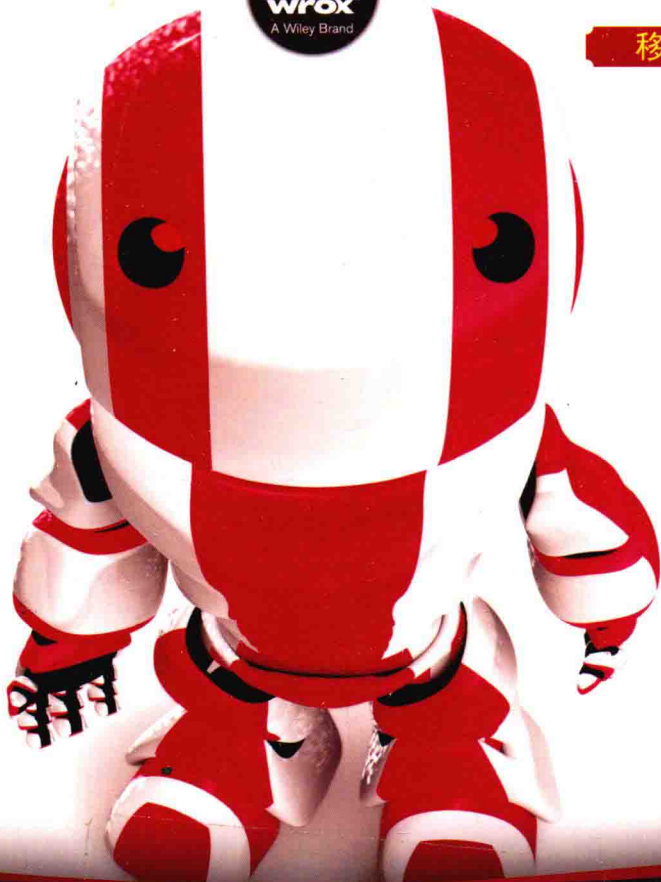




移动开发经典丛书



Expert Android Studio

Android Studio 高级编程

[美] Murat Yener
Onur Dunder
任强 许剑勇

著
译



清华大学出版社

移动开发经典丛书

Android Studio 高级编程

[美] Murat Yener 著
Onur Dunder
任强 许剑勇 译

清华大学出版社

北 京

Murat Yener, Onur Dunder
Expert Android Studio
EISBN: 978-1-119-08925-4

Copyright © 2016 by John Wiley & Sons, Inc., Indianapolis, Indiana
All Rights Reserved. This translation published under license.
Trademarks: Wiley, the Wiley logo, Wrox, the Wrox logo, Programmer to Programmer, and related trade dress are trademarks or registered trademarks of John Wiley & Sons, Inc. and/or its affiliates, in the United States and other countries, and may not be used without written permission. Android is a registered trademark of Google, Inc. All other trademarks are the property of their respective owners. John Wiley & Sons, Inc., is not associated with any product or vendor mentioned in this book.

本书中文简体字版由 Wiley Publishing, Inc. 授权清华大学出版社出版。未经出版者书面许可，不得以任何方式复制或抄袭本书内容。

北京市版权局著作权合同登记号 图字：01-2017-1722

Copies of this book sold without a Wiley sticker on the cover are unauthorized and illegal.

本书封面贴有 Wiley 公司防伪标签，无标签者不得销售。
版权所有，侵权必究。侵权举报电话：010-62782989 13701121933

图书在版编目(CIP)数据

Android Studio 高级编程/(美) 穆拉特·尤内尔(Murat Yener), (美) 奥努尔·邓达尔(Onur Dunder) 著;
任强, 许剑勇 译. —北京: 清华大学出版社, 2017
(移动开发经典丛书)
书名原文: Expert Android Programming
ISBN 978-7-302-47350-3
I. ①A… II. ①穆… ②奥… ③任… ④许… III. ①移动终端—应用程序—程序设计 IV. ①TN929.53
中国版本图书馆 CIP 数据核字(2017)第 124251 号

责任编辑: 王 军 韩宏志
装帧设计: 牛静敏
责任校对: 曹 阳
责任印制: 王静怡

出版发行: 清华大学出版社
网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>
地 址: 北京清华大学学研大厦 A 座 邮 编: 100084
社 总 机: 010-62770175 邮 购: 010-62786544
投稿与读者服务: 010-62776969, c-service@tup.tsinghua.edu.cn
质 量 反 馈: 010-62772015, zhiliang@tup.tsinghua.edu.cn

印 装 者: 三河市金元印装有限公司
经 销: 全国新华书店
开 本: 185mm×260mm 印 张: 24.5 字 数: 596 千字
版 次: 2017 年 7 月第 1 版 印 次: 2017 年 7 月第 1 次印刷
印 数: 1~4000
定 价: 59.80 元

产品编号: 067857-01

译者序

随着移动互联网的浪潮来袭，越来越多的开发人员投身于 Android 应用开发。在 Android Studio 问世之前，一直使用 Eclipse 和 ADT 插件来开发 Android 应用。为提高开发效率并更快地使用 Android 新功能，Google 公司推出了 Android Studio 开发工具。Google 同时声明，中止为 Eclipse ADT 插件以及 Android Ant 编译系统提供支持，这意味着 Android Studio 将成为 Android 开发唯一官方指定 IDE。因此当清华大学出版社的编辑约我们翻译此书时，我们欣然接受；希望本书能成为一盏明灯，帮助大家渡过最初的难关，通过新的开发工具释放出你们的全部潜能。

任何一门技艺都离不开“术”、“器”、“道”。如果把 Android 技术比作“术”，开发及设计思想比作“道”，那么过程中所需要的各种工具就是“器”。市场上已有不少关于 Android 开发的“术”和“道”的书籍，但关于“器”的书籍却凤毛麟角。要把开发技术提升到更高阶段，“术”、“器”、“道”缺一不可。古语云：工欲善其事，必先利其器。本书旨在帮助读者更好地了解和使用开发工具，收到事半功倍的学习效果。

本书首先介绍 Android Studio 的安装步骤，接着详细介绍 Android 应用开发、布局设计、构建系统、模块管理、版本控制等主题。还涵盖了调试和测试、NDK 开发、持续集成、编写自己的插件以及如何使用各硬件厂商的第三方工具等内容。本书既是一本操作指南，也可用作读者桌边常备的工具书。

在翻译过程中，清华大学出版社的编辑们不厌其烦地帮助我们、指导我们，他们为本书的翻译投入了巨大热情并付出了很多心血，没有他们本书不可能顺利付梓。感谢《Android 4 高级编程(第 3 版)》的译者余建伟给予的帮助和鼓励。感谢同事陈家伟、林启明和冯俊，他们给了我许多鼓励和帮助。感谢我的家人给我营造了一个安静的环境，让我能够专心于翻译。

对于这本经典之作，译者本着“诚惶诚恐”的态度，在翻译过程中力求“信、达、雅”，但是鉴于译者水平有限，错误和失误在所难免，如有任何意见和建议，请不吝指正。感激不尽！本书的第 1、3、5、7、9、11 章节由任强翻译，第 2、4、6、8、10、12、13 章节由许剑勇翻译。

最后，希望读者通过阅读本书能早日掌握 Android Studio 及相关工具，把你们的奇思妙想变成丰富多彩的 Android 应用！

许剑勇

序 言

“拙匠总怪工具差。”

小时候，当我们兄弟还生活在树木茂密的英格兰城郊时，父亲就坚信这句谚语。我们总是找一些借口来逃避那些要求我们做的真心为数不多的家务事，把不能完成的责任归结于手上那无法还嘴的工具。这句谚语就是父亲对于我们借口的标准回复，它立即否决了我们的抗议。

随着年龄的增长，我从这句谚语中学到的智慧远不止如何让小伙子们完成他们的家务活。这句话的意思不只是你要确保有合适的工具来完成手头的任务，更深层的含义是如何有效地使用工具的知识才是生产力的关键。如果考虑一下勤杂工工具包里的标准工具，相信大多数人都清楚锤子不是去除木板上螺丝钉的最佳工具，但随着工具变得更加复杂和精致，这些区别就变得不那么清晰了。

在 Android 开发世界里，事实上的标准开发工具是 Android Studio，尤其是因为它是谷歌非常积极开发的产物，与 Android 同属一个组织。自 2013 年 5 月早期预览版本发布以来，功能集已经变得相当充实可靠，并继续以令人惊讶的速度增长。考虑到 Android Studio 建立在 IntelliJ IDEA 的基础之上，而 IntelliJ IDEA 已经是功能极其丰富的开发环境，那么就应该很清楚，用 Android Studio 与锤子和螺丝刀做任何类比毫无意义。与其把 Android Studio 比作独立工具，还不如把它比作整个工具箱，其中包含大量独立工具，有时可以单独使用，有时可以一起使用，一旦有效使用就可以简化并加快开发任务，其中包括我们都讨厌的极其平淡的或重复的劳动！

现代软件开发不仅是简单地编写代码，在 Android 开发上尤其如此。应用的主要逻辑可能是用 Java 编写的。你也有一些资源(多数基于 XML)，如矢量图片(它将 SVG 路径数据转换为 XML)、构建文件(groovy/gradle 文件)和测试用源代码(Java 及测试领域特定语言，如 Espresso、Fest 或 Hamcrest)。开始考虑框架后就会变得更复杂，有些框架(如 Rx)会改变你的语法和代码流程，甚至还有越来越强劲的第二语言，如 Kotlin。掌握所有这一切可能很难。某些基础功能使得在不同的组件、语言、框架和方言之间进行“语境转换(context switch)”变得容易得多，如代码高亮和预编译等可以在代码中显示错误。但是我们已经习惯了这些，几乎没有注意到它们，因为它们变成了我们的第二天性，语境转换本身就成了第二天性。

使用手边的工具直到它们成为第二天性，这点很重要，但先决条件是，知道有什么工具以及如何有效地使用它们。这就是本书的来源。Murat 和 Onur 提供了 Android Studio 的指南，其中许多方面无论对于新手还是经验丰富的 Android Studio 用户来说都极具价值。

Mark Allison

作者简介

Murat Yener 是一位代码极客、开源贡献者、Java Champion 以及 Android 开发专家，他在 Intel 从事 Android 开发工作。他还是 *Professional Java EE Design Patterns* (2015 年 Wrox 出版) 一书的作者。Murat 在 Android、Java、Web、Java EE、OSGi 应用开发方面经验丰富，同时还从事教学及指导工作，他也是一位 Eclipse 贡献者，是 Eclipse Libra 项目的初始贡献者之一。

从 2009 年起，Murat 开始担任 GDG Istanbul 的用户组组长，组织和参加各项活动，并在活动上演讲。他还是 DroidCon、JavaOne 和 Devvxx 等重大会议的常驻演讲者。

领英: www.linkedin.com/in/muratyener

推特: @yenerm

博客: www.devchronicles.com

Onur Dundar 最初为 Intel 公司的软件应用工程师，他在 MeeGo 和 Android 等移动平台的平板和手机上从事开发工作，其后在 IoT 平台(诸如 IPTV、Intel Galileo 和 Edison)上从事开发。他还是 *Home Automation with Intel Galileo* (2015 年 Packt 出版) 一书的作者。Onur 还推出了他的 Android 应用开发培训课程，因此热切地希望出版此书。

领英: www.linkedin.com/in/odundar

推特: @odunculuk

亚马逊作者页: www.amazon.com/Onur-Dundar/e/B00V0VOIGA

技术编辑简介

Xavier Hallade 是法国 Intel 软件与服务集团的一名应用工程师。他长期致力于 Android 架构、类库和应用的广泛开发。Xavier 是一位 Android 开发专家，关注于 Android NDK，在 Android 社区积极做出贡献，撰写文章，在 Stack Overflow 上帮助开发者，在全世界范围与开发者开展对话。

致 谢

首先感谢与我共同执笔的 Onur Dundar, 感谢他的辛勤工作以及所付出的努力, 最终使我们能按时完成本书。没有他, 本书恐怕远不及现在的一半。

我要感谢我的团队和同事, 特别是 John Wei 和 Sunil Tiptur Nataraj, 他们为我提供了时间和灵活性来完成此书, 感谢 Angus Yeung 的支持。感谢我的 Google Developer Relations 联系人: Uttam Tripathi、Martin Omander 和 Baris Yesugey, 感谢其他以各种方式支持我的人。我想要感谢 Alex Theedom, 在我深陷 Android 时帮我回复 Java EE 相关问题。任何言辞都不足以感谢 Jim Minatel 和 Tom Dinse, 他们耐心地处理了所有琐事, 使我们远离压力。当然, 还要感谢 Wrox/Wiley 的每一位, 没有他们的帮助本书无法顺利付梓。

我要特别感谢以下三位重要人士, 他们对我的职业生涯有重大影响。

首先, 感谢我的父亲 Mustafa Yener, 在我年幼时他为我提供了第一台电脑 C64, 当时我所希望的是一辆轨道赛车。就是在那台电脑上, 我写了人生中的第一段代码。

其次, 我要感谢我的论文导师 Mahir Vardar, 在早期指导我即将开始的职业生涯。

最后, 感谢我的人生导师和朋友 Naci Dai, 他教会了我成为专业软件开发人员所需要知道的一切。

——Murat Yener

我要感谢 Murat Yener 带我经历这次旅程, 让我通过撰写本书分享乐趣。凭借 Murat 的经验和知识, 我们为开发人员和工程师提供了这本优质书籍。

我要向 Oktay Ozgun 表示感谢, 感谢他在我职业生涯和生活中所给予的智慧和建议。通过他的指导使我成为了一名充满激情的工程师以及更友善的人。

我还要感谢 Arda Yurdakul 教授, 她慷慨提供了经验和指导, 鼓励我更多地了解软件和计算机并成为了一名更优秀的工程师。

感谢 Cem Ersoy 教授、Alper Sen 教授和 Hande Alemdar 博士在我就读 Bogazici 大学的最后一年里提供的巨大帮助。我还要感谢 Steve Cutler、Andrew John、Brendan LeFoll、Todor Minchev、Peter Rohr、Rami Radi、Alex Klimovitski 和 Marcel Wagner, 感谢他们的大力支持, 感谢他们帮助我学习并在我的职业生涯中提供机会。

感谢 Gurkan Kok 教授为本书提供素材。

感谢所有 Wiley/Wrox 出版社的编辑们, 感谢他们对本书的帮助。

最后, 特别感谢 Ali Caglar Ozkan 在我的第一本书出版后给予的鼓励(并拍摄了很棒的照片), 还要感谢 Aytac Yurdakurban 和 Engin Efecik 支持我, 使我对写书保持兴趣。

——Onur Dundar

前言

无论你多么擅长编写代码，但若不具备集成开发环境(IDE)方面的相关知识，你将面临许多障碍。本书介绍 Google 用于开发 Android 应用的官方工具——Android Studio。每章都关注一个特定的主题，逐步从基本 IDE 用法扩展到高级知识，如编写插件来扩展 IDE 的功能。

本书读者对象

本书旨在为任何级别的开发人员提供经验。无论你是 Android 新手，还是经验丰富的 Android 开发人员(之前使用过基于 Eclipse 的 ADT)，本书都将引导你达到一个技术新高度，这样就可以利用 Android Studio 的工具释放你真实的开发潜能。

本书内容

本书不仅涵盖开发 Android 应用必备的 Android Studio 特性，也涉及整个开发周期的相关主题。以下是部分 Android Studio 的基本或扩展能力的主题示例：

- 使用 Git 共享代码并管理版本
- 使用 Gradle 管理构建
- 通过测试保持代码的可维护性和零 bug
- 通过持续集成控制整个构建和测试周期
- 为 Android Studio 编写插件扩展其功能或添加期望的定制特性
- 使用 Android Studio 及第三方工具来加快开发进度

本书结构

每章都会关注某个特定的 Android Studio 相关话题或附带的工具，阐述它的必要性以及如何使用或配置。有些章提供代码示例来演示用例或为该章的主题提供示例。

- **第 1 章“新手入门”**：安装和设置开发环境。创建可以运行项目的模拟器。
- **第 2 章“Android Studio 基础知识”**：首先使用 Android Studio 创建一个新项目，构建你的项目，将项目迁移到 Android Studio 上。

- 第 3 章“使用 Android Studio 开发 Android 应用”：Android Studio 项目结构。如何使用 Asset、XML 文件和 Android Manifest 文件。创建并使用模块。
- 第 4 章“深入 Android Studio”：深入了解 Android Studio，介绍菜单、编辑器、视图和快捷方式。讨论如何使用 Live 模板和重构功能，如何构建项目并为 APK 签名。
- 第 5 章“使用 Android Studio 创建布局”：如何在 Android Studio 中使用布局。介绍 UI 预览和 UI 开发工具。分析如何管理外部依赖关系，如何使用和组织 Asset。
- 第 6 章“Android 构建系统”：如何有效地使用和配置 Gradle，如何编写 Gradle 插件。
- 第 7 章“多模块项目”：在项目中添加模块。如何创建并使用 Phone/Tablet、Library、Wear、TV、Glass、Auto 和 Cloud 模块。
- 第 8 章“调试和测试”：使用 ADB 调试 Android 代码。详细了解 Android 设备监视器、Android 虚拟设备、Lint 以及如何测试代码。
- 第 9 章“使用版本控制系统：Git”：如何分享项目，使用 Git 启用版本控制。
- 第 10 章“持续集成”：使用持续集成服务器，使构建、测试及发布自动化。
- 第 11 章“在 Android Studio 中使用 Android NDK”：安装 Android NDK，通过 Android NDK 构建用 C/C++ 编码的项目。
- 第 12 章“编写自己的插件”：编写自己的插件来扩展 IntelliJ 平台的功能。与 UI、编辑器交互，添加你自己的操作。
- 第 13 章“第三方工具”：其他可以帮助和加快开发生命周期的附加工具。

为什么你需要本书

任何现代计算机，只要操作系统支持 Android SDK 和 Android Studio，都足以运行 Android Studio、构建 Android 应用并运行本书中的示例。你需要在操作系统上安装适当的 Android SDK、Android Studio 和 Java 虚拟机(JVM)。某些章需要安装额外的工具或框架，如 Android NDK。可在第 1 章中找到有关所需硬件要求的更多信息。

撰写本书的原因

2007 年 11 月，谷歌发布了 Android SDK 预览版，允许开发人员开始使用这个新的手机操作系统。大约两年后，2009 年 10 月，Google 向公众发布了作为 Eclipse 插件组的 ADT(Android Developer Tools)。

作为 Google I/O 2009 大会的与会者之一，我(Murat)很幸运拥有了一台 Android 设备；我可能是最早一代的开发人员，下载 ADT 插件并在 Eclipse 上安装使用。在过去这些年里，我们怀着同样的激情下载和尝试新的 ADT 版本以及与之一同发布的新东西。

当时，我是一名 Eclipse 贡献者，知道如何编写自己的插件，扩展 IDE 的功能，实现我需要的行为和功能。所以每当 ADT 发布时，我就会越发兴奋地要去看工具能完成什么。

2013 年 5 月，大约是在我们与 ADT 爱恨交织了四年后，同样是在 Google I/O 大会上，Google 发布了 Android Studio，并很快成为支持 Android 开发的官方 IDE。ADT 虽然并不完美，但我们非常熟悉它。像其他许多开发人员一样，我们知道所有的快捷键，知道它是如何工作的，知道当某些功能不工作时该如何解决，以及如何让项目结构化。更重要的是，我们可以编写自己的插件，或者检查 ADT 插件找到出错的原因。随着 Android Studio 的发布，我们突然有了一个新的平台，但对其所知甚少。

在一段时间里我们拒绝切换到 Android Studio，但最后还是做了一次尝试。突然间，Android 这个我们熟悉的平台，仿佛变成了陌生人。因为 IntelliJ 和 Gradle 带来的变化，新的项目结构也大相径庭。采用 IntelliJ 后，我们决定遵循 IntelliJ 快捷键，而不是使用 IntelliJ 的 Eclipse 快捷键映射，这使得情况变得更糟。我们几乎不能搜索到一个文件或一段代码，无法通过菜单导航，通过右击创建文件，甚至生成一些基本的 getter 和 setter 方法。我们从 ADT 专家变成了 Android Studio 初学者。

终于，我们受够了！我们都是经验丰富的开发人员，但与 Android Studio 的斗争并不能彰显我们的技能。所以我们开始关注 IntelliJ 话题，把 IntelliJ 快捷键小抄钉在我们的分隔间里，阅读 IntelliJ 插件代码，并强迫自己在日常工作中使用 Android Studio。

本书是我们自己在这条艰难的道路上学会走路的经验总结。当我们从 Eclipse 的 ADT 切换到基于 IntelliJ 的 Android Studio 时，我们自己所需要的正是本书。我们相信，任何开发人员，无论是 Android 新手，还是曾经使用 ADT 的经验丰富的老手，都会发现本书有助于拓展他(或她)的工具知识，这些工具能切实地支持编码技巧。

下面引用我前一本书的合著者 Alex Theedom 的一句话，“我们写的每一章都有这样一个目标：写我们自己想读的内容。”我们带着 Onur(荣誉)追寻相同目标的结果，就是此时你手里的书。

我们希望你阅读本书时，能够感受到我们撰写时的快乐。

注意：请务必阅读我们的博客 <http://www.devchronicles.com/2016/06/expert-android-studio-book-updates.html>，查看在 Google I/O 2016 大会上公布的变化。

源代码

在读者学习本书中的示例时，可以手工输入所有代码，也可以使用本书附带的源代码文件。本书使用的所有源代码都可以从本书合作网站 www.wiley.com/go/expertandroid 下载。打开网页，接着单击本书详细页面上的 Download Code 链接，就可以获得本书所有源代码。

此外，可以访问 www.tupwk.com.cn/downpage，输入中文书名或 ISBN 下载源代码，也可心直接扫描封底的二维码下载。

注意：由于许多图书的书名都很类似，因此按照 ISBN 搜索是最简单的，本书英文版的 ISBN 是 978-1-119-08925-4。

在下载了代码后，只需用自己喜欢的解压缩软件对它进行解压缩即可。另外，也可以进入 <http://www.wrox.com/dynamic/books/download.aspx> 上的 Wrox 代码下载页，查看本书和其他 Wrox 图书的所有代码。

勘误表

尽管我们已经尽了各种努力来保证文章或代码中不出现错误。但是错误总是难免的，如果你在本书中找到了错误，例如拼写错误或代码错误，请告诉我们，我们将非常感谢。通过勘误表，可以让其他读者避免挫折，当然，这还有助于提供更高质量的信息。

要在网站上找到本书英文版的勘误表，可以登录 <http://www.wrox.com>，通过 Search 或书名列表查找本书。然后在本书的详细页面上，单击 Book Errata 链接。在这个页面上可以查看到 Wrox 编辑已提交和张贴的所有勘误项。完整的图书列表包括每本书的勘误表，网址是 www.wrox.com/misc-pages/booklist.shtml。

如果你没在 Book Errata 网页找到你发现的错误，请访问 www.wrox.com/contact/techsupport.shtml，在那里完成表单将你发现的错误发送给我们。我们将会检查该信息，一旦确实就会发布到本书的勘误表页面，并在本书的后续版本中修复该问题。

p2p.wrox.com

要与作者和同行讨论，请加入 p2p.wrox.com 上的 P2P 论坛。这个论坛是一个基于 Web 的系统，便于你张贴与 Wrox 图书相关的消息和相关技术，与其他读者和技术用户交流心得。该论坛提供了订阅功能，当论坛上有新的消息时，它可以给你发送感兴趣的论题。Wrox 作者、编辑和其他业界专家都会到这个论坛上来讨论问题。

在 <http://p2p.wrox.com> 上，有许多不同的论坛，它们不仅有助于阅读本书，还有助于开发自己的应用程序。要加入论坛，可以遵循下面的步骤：

- (1) 进入 p2p.wrox.com，单击 Register 链接。
- (2) 阅读使用协议，并单击 Agree 按钮。
- (3) 填写加入该论坛所需要的信息和自己希望提供的其他信息，单击 Submit 按钮。
- (4) 你会收到一封电子邮件，其中的信息表述了如何验证账户，完成加入过程。

注意：不加入 P2P 论坛也可以阅读消息，但要张贴自己的消息，就必须加入该论坛。

加入论坛后，就可以张贴新消息，响应其他用户张贴的消息。可以随时在 Web 上阅读消息。如果要让该网站给自己发送特定论坛中的消息，可以单击论坛列表中该论坛名旁边的 [Subscribe to this Forum](#) 图标。

关于使用 Wrox P2P 的更多信息，可阅读 P2P FAQ，了解论坛软件的工作情况以及 P2P 和 Wrox 图书的许多常见问题。要阅读 FAQ，可以在任意 P2P 页面上单击 FAQ 链接。

目 录

第 1 章 新手入门	1
1.1 Windows、Mac OS X 及 Linux 平台的系统要求	1
1.2 安装 Java	3
1.2.1 为 Windows 操作系统安装 Java	4
1.2.2 为 Mac OS X 安装 Java	6
1.2.3 为 Linux 安装 Java	7
1.3 安装 Android Studio	8
1.3.1 为 Microsoft Windows 10 安装 Android Studio	9
1.3.2 为 Mac OS X 安装 Android Studio	11
1.3.3 为 Linux 安装 Android Studio	13
1.4 首次启动 Android Studio	15
1.5 独立 SDK 的安装	16
1.6 小结	17
第 2 章 Android Studio 基础知识	19
2.1 创建新的示例项目	19
2.1.1 使用不同的 SDK 版本	23
2.1.2 Android 项目结构	23
2.1.3 构建并运行项目	30
2.2 Android 模拟器	31
2.2.1 安装 HAXM	31
2.2.2 创建 Android 虚拟设备	32
2.2.3 使用 ADB	35
2.3 导入 Eclipse 项目	36
2.4 小结	38

第 3 章 使用 Android Studio 开发 Android 应用	39
3.1 Android 项目	40
3.1.1 创建新的 Android 项目	40
3.1.2 创建一个多目标设备的项目	44
3.1.3 启动 Android 应用	47
3.2 Android Activity	49
3.2.1 Intent 事件处理	50
3.2.2 将模板 Activity 添加到 Android 项目	51
3.2.3 添加 Blank Activity	51
3.2.4 Android Fragment	55
3.2.5 添加 Tabbed Activity	57
3.3 Android Service	61
3.4 为 Android 项目添加 Asset	64
3.4.1 添加 Images Asset	65
3.4.2 添加 Sound Asset	67
3.4.3 添加 Video Asset	67
3.5 将 XML 文件添加到 Android 项目	68
3.6 Android Manifest 文件	68
3.7 Android Module	70
3.8 小结	71
第 4 章 深入 Android Studio	73
4.1 Android Studio 菜单项	73
4.1.1 Android Studio 菜单	74
4.1.2 File 菜单	75
4.1.3 Edit 菜单	76
4.1.4 View 菜单	77
4.1.5 Navigate 菜单	78

- 4.1.6 Code 菜单 79
- 4.1.7 Analyze 菜单 80
- 4.1.8 Refactor 菜单 80
- 4.1.9 Build 菜单 81
- 4.1.10 Run 菜单 82
- 4.1.11 Tools 菜单 82
- 4.1.12 Version Control System
菜单 82
- 4.1.13 Window 菜单和 Help
菜单 83
- 4.2 Android Studio 快捷键 83
- 4.3 Android Studio 工具视图 84
 - 4.3.1 Messages 工具 86
 - 4.3.2 Android Studio 项目结构
视图 86
 - 4.3.3 Favorites 工具 88
 - 4.3.4 Android Monitor 工具 88
 - 4.3.5 Structure 窗口 89
 - 4.3.6 Android Model 视图 90
 - 4.3.7 Gradle 和 Gradle Console
窗口工具 90
 - 4.3.8 Run 窗口 91
 - 4.3.9 Debug 窗口 91
 - 4.3.10 Event Logs 工具 91
 - 4.3.11 Terminal 窗口 92
- 4.4 Android Studio 编辑器 92
 - 4.4.1 代码辅助 92
 - 4.4.2 注释掉代码块 92
 - 4.4.3 移动代码块 93
 - 4.4.4 编辑器中的导航 93
 - 4.4.5 重构 93
- 4.5 Android Studio LiveTemplates 94
 - 4.5.1 插入 Live Template 95
 - 4.5.2 创建 Live Templates 96
- 4.6 Android Studio 的代码重构 98
- 4.7 在 Android Studio 中为
Android 应用创建一个
签名密钥 102

- 4.8 在 Android Studio 中构建
APK 104
- 4.9 小结 104
- 第 5 章 使用 Android Studio 创建
布局 107
 - 5.1 使用 Android Studio 创建
布局 108
 - 5.1.1 添加新的布局文件 108
 - 5.1.2 Android Studio 中的布局
设计结构 110
 - 5.2 布局预览 112
 - 5.2.1 布局渲染选项 112
 - 5.2.2 预览虚拟设备视图 113
 - 5.2.3 在不同的 Android SDK 版本
上预览 114
 - 5.2.4 选择主题 114
 - 5.3 使用 Android Studio 设计
布局 115
 - 5.4 管理资源 119
 - 5.4.1 使用 String 119
 - 5.4.2 使用 Style 120
 - 5.4.3 使用 Dimen 121
 - 5.4.4 使用 Color 122
 - 5.4.5 使用 Drawable 122
 - 5.5 使用布局工具 123
 - 5.5.1 翻译 123
 - 5.5.2 Activity Association 125
 - 5.6 Asset 管理 126
 - 5.7 小结 131
- 第 6 章 Android 构建系统 133
 - 6.1 使用 Gradle 134
 - 6.2 使用 Gradle 管理依赖关系 136
 - 6.2.1 外部依赖 137
 - 6.2.2 本地依赖 138
 - 6.2.3 遗留 Maven 依赖 139
 - 6.3 Gradle 的 Android 插件 140
 - 6.3.1 为 Gradle 配置 Android
插件 140

6.3.2 构建配置	140	7.10.3 导入 JAR/AAR 包	176
6.3.3 构建任务	141	7.11 从项目里删除模块	176
6.3.4 flavor	141	7.12 小结	177
6.3.5 ProGuard	142	第 8 章 调试和测试	179
6.3.6 自动化测试	144	8.1 调试 Android 代码	180
6.4 Gradle 插件	144	8.1.1 Android Debug Bridge	180
6.4.1 编写自己的 Gradle 插件	144	8.1.2 无线调试	181
6.4.2 扩展 Gradle 的 Android 插件	145	8.1.3 开始调试	182
6.5 小结	146	8.2 Android 监视器	189
第 7 章 多模块项目	147	8.2.1 使用 logcat	192
7.1 将模块添加到 Android 项目	148	8.2.2 使用内存监视器	193
7.2 Phone&Tablet 模块	149	8.2.3 使用 CPU 监视器	196
7.3 Android Library 模块	151	8.2.4 使用 GPU 监视器	197
7.4 Java Library 模块	154	8.2.5 使用网络监视器	197
7.5 Android Wear 模块	156	8.3 Android 设备监视器	198
7.5.1 运行及调试 Android Wear 模块	157	8.4 Android 虚拟设备扩展控件	203
7.5.2 构建支持 Android Wear 的 APK	159	8.5 使用 Lint	206
7.6 Android TV 模块	159	8.6 测试 Android 代码和应用 UI	207
7.6.1 运行和调试 Android TV 模块	162	8.6.1 单元测试	212
7.6.2 构建 Android TV 模块的 APK	164	8.6.2 集成测试	214
7.7 Glass 模块	165	8.6.3 UI 测试	217
7.7.1 运行和调试 Glass 模块	166	8.6.4 性能测试	223
7.7.2 构建 Glass 模块的 APK	167	8.6.5 性能测试任务	224
7.8 Android Auto 模块	167	8.7 小结	226
7.9 Google Cloud 模块	169	第 9 章 使用版本控制系统: Git	227
7.9.1 添加 New Google Cloud 模块	169	9.1 Git 介绍	228
7.9.2 运行和调试 Google Cloud 模块	172	9.1.1 理解 Git	228
7.10 导入模块	173	9.1.2 安装 Git	230
7.10.1 导入 Gradle 项目	173	9.2 使用 Git	231
7.10.2 导入 Eclipse ADT 项目	174	9.3 使用 GitHub 客户端	232
		9.4 在 Android Studio 中使用 Git	240
		9.5 Git 流	247
		9.6 小结	257
		第 10 章 持续集成	259
		10.1 是持续集成介绍	259
		10.2 使用持续集成服务器集成 Android 项目	260

10.3	安装 Jenkins	261
10.4	创建构建作业	267
10.5	发布管理	273
10.6	小结	280
第 11 章	在 Android Studio 中使用 Android NDK	281
11.1	Android NDK 介绍	282
11.2	Android Studio 中的 NDK 集成	282
11.2.1	在 Linux 上安装 Android NDK	283
11.2.2	在 Windows 10 上安装 Android NDK	284
11.2.3	在 Mac OS X 上安装 Android NDK	285
11.3	Android Studio 中的 Android NDK 项目	286
11.3.1	导入 NDK 示例项目	287
11.3.2	迁移现有的 NDK 项目	292
11.3.3	构建 Android NDK 项目	296
11.4	Android NDK 项目的发布与部署	297
11.5	小结	300
第 12 章	编写自己的插件	301
12.1	IntelliJ IDEA 插件架构	302
12.1.1	Action	302
12.1.2	线程	309
12.1.3	文件系统	310
12.1.4	项目和组件	311
12.1.5	编辑器	320
12.1.6	向导	323
12.2	打包和分发	328
12.3	小结	329

第 13 章	第三方工具	331
13.1	Android Studio 插件	331
13.2	Intel 的 Android 软件工具	340
13.2.1	Intel System Studio	340
13.2.2	Intel C++ Compiler	343
13.2.3	Intel IPP	345
13.2.4	Intel TBB	346
13.2.5	Intel VTune Amplifier	346
13.2.6	Intel GPA	348
13.3	Intel INDE	348
13.3.1	Intel Tamper Protection Toolkit	349
13.3.2	Intel Multi-OS Engine	349
13.3.3	Intel Context Sensing SDK	351
13.4	Qualcomm Android 软件工具	353
13.4.1	适用于 Android 的 Snapdragon LLVM Compiler	354
13.4.2	Qualcomm Adreno GPU SDK	354
13.4.3	Qualcomm FastCV Computer Vision SDK	355
13.4.4	适用于 Android 的 Snapdragon SDK	359
13.4.5	Qualcomm AllPlay Click SDK	362
13.4.6	Qualcomm Profilers	364
13.5	NVIDIA 软件工具	366
13.6	小结	369

第 1 章

新手入门

本章主要内容:

- Android Studio 的系统要求
- Microsoft Windows、Mac OSX 以及 Linux 上的 Java 安装说明
- Microsoft Windows、Mac OSX 以及 Linux 上的 Android Studio 安装说明

在本章中，你将完成开发环境的设置，这样就可以使用 Android Studio 开始 Android 开发。为此，本章将介绍 Android Studio 的安装步骤以及系统要求。

尽管 Android 操作系统是基于 Linux 的，但 Android SDK 及其工具可用于所有主要的操作系统，因此你可以在所工作的操作系统上设置开发环境。本书自始至终使用 Mac OS 作为主要环境，然而我们也会介绍 Linux 和 Windows 的设置。

1.1 Windows、Mac OS X 及 Linux 平台的系统要求

要使用 Android Studio，你的开发环境必须满足最低系统要求。本节将列出 Windows、Mac OS X 以及 Linux 的最低要求。

Microsoft Windows

- Microsoft Windows 10/8/7/Vista/2003 (32 或 64 位)
- 最低 2GB RAM 内存，推荐 4GB RAM 内存
- 400MB 硬盘空间
- 至少 1GB 用于 Android SDK、模拟器系统镜像及缓存
- 最低 1280 × 800 屏幕分辨率
- Java Runtime Environment (JRE) 6 或更高
- Java Development Kit (JDK) 7