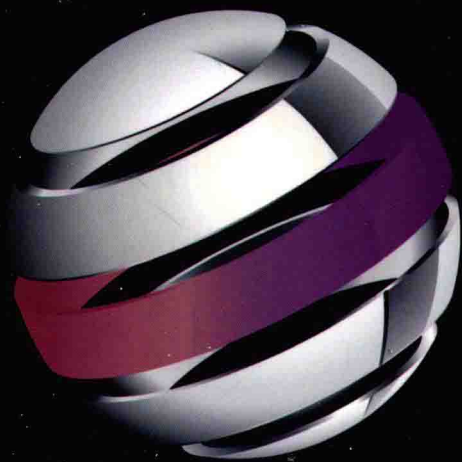


移动开发经典丛书



# Android Studio实战

快速、高效地构建Android应用

Learn Android Studio

Build Android Apps Quickly and Effectively



[美] Adam Gerber 著  
Clifton Craig  
靳晓辉 张文书 译



apress®



清华大学出版社

移动开发经典丛书

# Android Studio 实战

快速、高效地构建

Android 应用

[美] Adam Gerber      著  
Clifton Craig  
靳晓辉 张文书 译

清华大学出版社

北 京

Adam Gerber, Clifton Craig

Learn Android Studio: Build Android Apps Quickly and Effectively

EISBN: 978-1-4302-6601-3

Original English language edition published by Apress Media. Copyright © 2015 by Apress Media.

Simplified Chinese-Language edition copyright © 2016 by Tsinghua University Press. All rights reserved.

本书中文简体字版由 Apress 出版公司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: 01-2016-4648

本书封面贴有清华大学出版社防伪标签,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 实战 快速、高效地构建 Android 应用/(美) 亚当·格伯(Adam Gerber), (美) 克利夫顿·克雷格(Clifton Craig) 著; 靳晓辉, 张文书 译. —北京: 清华大学出版社, 2016  
(移动开发经典丛书)

书名原文: Learn Android Studio: Build Android Apps Quickly and Effectively

ISBN 978-7-302-44153-3

I. ①A… II. ①亚…②克…③靳…④张… 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(2016)第 148557 号

责任编辑: 王 军 李维杰

装帧设计: 牛静敏

责任校对: 成凤进

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](mailto:c-service@tup.tsinghua.edu.cn)

质 量 反 馈: 010-62772015, [zhiliang@tup.tsinghua.edu.cn](mailto:zhiliang@tup.tsinghua.edu.cn)

印 装 者: 北京嘉实印刷有限公司

经 销: 全国新华书店

开 本: 185mm×260mm 印 张: 24.75 字 数: 602 千字

版 次: 2016 年 7 月第 1 版 印 次: 2016 年 7 月第 1 次印刷

印 数: 1~3000

定 价: 59.80 元

产品编号: 067858-01

# 译者序

在当今的移动互联网时代，绿色机器人 Android 可谓家喻户晓。Android 开发自然也随着搭载 Android 平台的智能手机、平板电脑和可穿戴设备的热销而变得异常火热。“工欲善其事，必先利其器”——Android Studio 作为官方推荐的 IDE，已经成为高效 Android 开发者的必备神器。Android Studio 是 Google 公司在 2013 Google I/O 大会上发布的全新 Android 开发 IDE。引用原书中的描述：Android Studio 是 JetBrains 和 Google 合作的产物——它基于 JetBrains 的 IntelliJ 构建，其功能是 IntelliJ 的超集。

互联网是一个比拼效率的行业，而移动互联网更加速了应用开发的节奏。与之相适应的是现代 IDE 的设计也更加注重用户体验，以便让开发者能够将更多的精力放在编写代码上。因此，除了基础的编辑功能之外，智能代码补全、可感知语法的重构、可视化的界面编辑器、调试和性能分析工具、代码管理和工程构建工具已经成为一款优秀 IDE 的标配。如果以此作为衡量标准，那么与其他 Android 开发工具相比，Android Studio 在所有这些方面无疑都更胜一筹。Android Studio 中集成了大量主流的辅助工具，能帮助开发者应对在项目各个阶段可能遇到的问题，可谓提供了 Android 开发的“一站式”服务。

本书结合若干个实际可用的示例，循序渐进地讲解了 Android Studio 的基本用法、Android App 的开发流程(包括设计用户界面、搭建布局、采用 MVC 模式编写代码等)、代码管理工具 Git 和构建工具 Gradle 的使用技巧。本书还简要介绍了 Android 可穿戴设备以及使用 Android Wear SDK 进行开发的方法。本书适合正在考虑从 Eclipse+ADT 向 Android Studio 迁移的 Android 开发者。

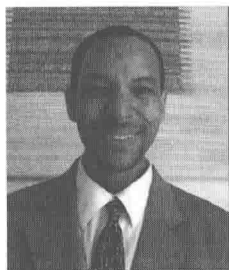
最后，我要在这里感谢清华大学出版社的李阳编辑给予我的信任，将该书的翻译工作交给我，并认真阅读译稿，提出诸多宝贵的修改意见，您的专业和认真给我留下了深刻的印象。本书的翻译和校对工作均是利用业余时间完成的，因此牺牲了不少本应该陪伴家人的时间，在此感谢他们的理解和支持。本书主要由靳晓辉翻译，此外，郭文豪、何孟宇、黄宇轩、李一夫、刘峰、龙伟、贾立克、穆玉伟、乌力吉、张国、张文书、张宇欣、周俊健、朱文等人也参与了本书部分章节的翻译，在此一并对他们表示感谢。由于译者水平有限，加之时间仓促，译文中可能会存在不准确的内容。如果读者在阅读过程中发现谬误和遗漏之处，烦请多多包涵并不吝批评指正。

靳晓辉

# 作者简介



**Adam Gerber** 是 Android Studio 的最早期使用者之一，他使用 Android Studio 开发专业的 Android 应用并将其用于自己在芝加哥大学开设的 Android 应用开发和技术创业等课程的教学。Adam 是“芝加哥创新交流”的成员并从事技术和创业领域的咨询工作。Adam 拥有伊利诺伊大学工业设计本科学位以及位于巴黎的法国国立工艺学院管理科学博士学位。Adam 的邮箱是 [gerber@uchicago.edu](mailto:gerber@uchicago.edu)。



**Clifton Craig** 从事软件工程师的工作已逾 16 载。他的经验涵盖了 J2ME/BlackBerry、Android 和 iOS，以及基于 JEE 的后端系统。他参与过一些备受瞩目的项目，包括 MapQuest 天然气价格门户网站、基于 J2ME 和 Android 的 MapQuest 移动版、基于 iOS 的 MapQuest 移动版以及 Skype 的 iOS 和 Android 版。他维护着一个技术博客 ([cliftoncraig.com](http://cliftoncraig.com))，其中涵盖各种软件主题，从 Android 和 Linux 到 iOS 和 OS X。他有着从军经历而且是一位狂热的自行车迷、虔诚的基督徒以及两个天才小女孩的父亲。

# 技术审稿人简介

Jim Graham 于 1989 年在德克萨斯农工大学获得电子科学系电信专业的本科学位。他在国际通信协会的 1988 期的 ICA 公报上发表过论文(“Fast Packet Switching: An Overview of Theory and Performance”); 在伊利诺伊州芝加哥市 Amoco 公司的网络设计小组做过助理网络工程师; 在佛罗里达州沃尔顿堡海滩的 Tybrin 公司做过高级网络工程师; 在佛罗里达州赫尔伯特训练场的“第 16 特种作业翼情报处”和“美国空军总部特种作战指挥情报处”做过情报系统分析员。他于 2001 年收到来自“第 16 特种作业翼情报处”的正式表扬信。

# 致 谢

讲述像 Android 这样宏大的主题以及像 Android Studio 这样强大的工具需要投入、努力和多人协作。我们想要感谢编辑和技术审稿人——Corbin Collins、Mark Powers 和 Jim Graham。此外，我们还要向其他付出了直接和间接努力的人士致谢。

在大部分写作过程中，Android Studio 都处于 beta 版并且在不断更新。我们的实验和代码示例需要如此频繁地重做，以至于我已经记不清更新的具体次数了。非常感谢我的联合作者——Clifton Craig，他总是大刀阔斧地解决难题。我还想感谢我的家人和朋友，尤其是 Mia Park 一直支持着我度过这段充满挑战但收获颇丰的历程。我还想要感谢 Marilyn Meyers 对我一直以来的信任。非常感谢来自 Apress 的优秀专业团队，他们在编辑方面提供的支持非常重要。

——Adam Gerber

感谢 Onur Cinar 将我介绍给 Apress 的同事们，他们都是和蔼可亲的人。感谢我的联合作者 Adam Gerber，他在整个过程中都保持着积极的态度并且是一位优秀的推动者。也要感谢我最亲密的一些朋友——Juan Carlos Jimenez、Steve O'Sullivan、Nizam Gok 和 Yanxia Zhang，当遇到困难时，他们都不断地支持和鼓励我。在顶尖的科技公司里做全职工作需要不断地平衡工作与生活，而在这中间插入一部技术书籍则是一项艰巨的任务。在这个过程中，我不得不牺牲或忽略掉许多事情。我想对我的经理 Will Camp 和 Aravind Vijayakirithi 说声谢谢，感谢你们容忍我的疏漏并一直给予我指导。最后，我要致意并感谢我的妻子 Altaress，当我面对笔记本电脑陷入沉思时，她一直都在那儿陪着我和孩子们。

——Clifton Craig

# 前言

大约 5.3 亿年前，在那个地质学家称之为“寒武纪大爆发”的时代，包括现存所有门在内的大量物种在短短 100 万年的时间迅速出现——而这对于地质学时间来说仅仅是一眨眼工夫。科学家们对于此现象一直感到非常惊讶，达尔文自己都认为“寒武纪大爆发”发生如此之快，甚至让他的自然进化论产生了疑点。今天，我们正在经历着科技领域的“寒武纪大爆发”。美国劳工统计局预测现在的高中毕业生在其一生中将从事 11 份工作，而这种职业生涯短暂现象主要归因于科技的快速变化<sup>1</sup>。

技术会衍生出更多的技术，而新的技术也正在加速发展。有些新技术几年之后依然会存在，但大多数将会消失。没有比投入时间和精力去掌握一门已经过时或者效用短暂的新技能更糟糕的事情了。我们编写本书是因为相信其中涵盖的工具和技术将会留存下来，而且值得你为之投入。

## 小即是美

摩尔定律是无情的，它指出 CPU 的处理能力大约每 18 个月会翻一倍。在过去几年里，笔记本电脑已经达到了与体积更大的桌面电脑相同的性能。笔记本电脑和平板电脑占据了 2014 年 PC 销售额的 81%<sup>2</sup>，而且销量还会继续增长，相反桌面电脑的销量将会下滑。没有哪个人或组织能够阻止或逆转这种趋势——这是经济规律的力量，是个体选择集聚的结果。笔记本电脑在未来大约 10 年将会是脑力工作者的首选工具。然而，一场悄无声息的革命正在发生，它将会很快推翻全能的笔记本电脑。大约到 2025 年，或者可能更早，我们的智能手机将会达到与笔记本电脑相同的性能——也就是说，更大体积的外形已经不再具备任何性能优势了。最终，我们的移动电脑(Mobile Computer, MC)将用于大多数计算应用，即使是那些你我现在认为只能在笔记本电脑上完成的应用。这场变革是可预见的，与推翻桌面电脑的变革一样确定。同时，你可以期望 MC(换句话说，智能手机或平板电脑)开始发挥出和笔记本电脑一样的功能，包括连接诸如键盘、显示器和鼠标等外围设备。

PC 时代即将结束，但 MC 时代其实更具个性化。不久，一系列新型可穿戴设计即将可用，例如手表、眼镜和鞋子。我们预见在不久后的某一天，我们将会在身上穿着自己的电脑，并在显示器、键盘和鼠标这些外围设备可用的地方连接它们。这将会是一个真正的

---

1 <http://online.wsj.com/news/articles/SB10001424052748704206804575468162805877990>。

2 来源：Forrester Research eReader Forecast, 2010 至 2015(美国)。

个人电脑时代，尽管我们不大可能再这样称呼它了。

## Android 的优势

如果渴望成为一名 Android 开发者，那么你已经做出了极好的选择。在未来的 10 年里，这个不断发展的世界上的数十亿人都会来到线上。对于这些人中的大多数，他们的第一部电脑将会是智能手机，而这些智能手机中的大部分将会搭载 Android 系统<sup>3</sup>。我们的乐观是有充分原因的，因为已经有了大量可供推算的历史数据。Gartner Group 公司指出 2015 年将售出 12.5 亿部 Android 设备<sup>4</sup>。在撰写本书时，Android 独占了超过四分之三的中国市场份额<sup>5</sup>，而中国消费者在移动设备上的投入是惊人的，一些人会在新款移动设备上花费 70% 的月薪，因为互联是参与全球经济的先决条件<sup>6</sup>。在绝对数量上，中国是最大的市场，但我们能够在所有发展中国家观察到类似的趋势。更进一步，由于 Android 操作系统是开源免费的，因此它几乎总是电视游戏机、游戏系统、增强现实系统以及大量其他电子设备厂商的首选。

出于多种原因，Android 将继续巩固其在全球市场的主导地位。Android 的模块化架构允许进行各种各样的配置和定制。Android 设备标配的所有核心应用均可以被任意数量的第三方应用替代，这包括类似电话拨号器、电子邮件客户端、浏览器甚至操作系统导航器等应用。可用的 Android 设备有着各种各样令人惊讶的形状和功能，有 Android 增强现实眼镜、Android 游戏机(Ouya 是最著名的)、Android 手表、各种尺寸的 Android 平板电脑，当然还有 Android 智能手机。

Android 的核心技术毫不逊色于它的主要竞争者。Android 的包容性和开源许可证已经吸引了大量盟友，包括三星——全球最具创新性的公司之一。免费<sup>7</sup>和可定制的操作系统意味着 Android 设备厂商可以专注于向市场上推出优秀的产品，而高度竞争的 Android 设备市场也会持续生产出廉价、高质量且结构上开放的设备。

## Android Studio 是革命性的

作为一名脑力劳动者，工具的选择极其重要。我们总是在寻求可以提高生产效率并能自动完成工作的工具。某些工具有着显而易见的好处，人们会立刻采纳它。Android Studio

3 <http://news.yahoo.com/android-projected-own-smartphone-market-next-fouryears-213256656.html>, <http://www.idc.com/getdoc.jsp?containerId=prUS24302813>。

4 [www.bbc.co.uk/news/technology-25632430](http://www.bbc.co.uk/news/technology-25632430)。

5 报告：Windows Phone 在意大利赶超 iOS，且在欧洲市场份额增加——The Next Web 网(时间未知)。取自 <http://thenextweb.com/insider/2013/11/04/report-windows-phone-overtakes-ios-in-italy-and-makes-progress-in-europe/#!pSdH1>。

6 报告：Windows Phone 在意大利赶超 iOS，且在欧洲市场份额增加——The Next Web 网(时间未知)。取自 <http://thenextweb.com/insider/2013/11/04/report-windows-phone-overtakes-ios-in-italy-and-makes-progress-in-europe/#!pSdH1>。

7 有一点很重要，需要注意，虽然 Google 放弃了 Android 的许可费，但是移动技术增值的整体趋势会提升 Google 的广告收入。

就是一款这样的工具。

2013 年在 Google I/O 预发布 Android Studio 几天之后，我们就接触了它。在那以前，我们在专业领域和教学时均使用 Android 开发者工具(Android Developer Tool, ADT)。ADT 是一个 Android 开发环境，内置于一个称为 Eclipse 的开源集成开发环境(IDE)中。虽然 Android Studio 仍处于早期预发布阶段，但我们已开始专业领域内使用 Android Studio。

Android Studio 是 JetBrains 和 Google 合作的产物。Android Studio 基于 JetBrains 的 IntelliJ 构建，因此其功能是 IntelliJ 的超集。能够使用 IntelliJ 做到的大多数事情，都可以在 Android Studio 中完成。Android Studio 是革命性的，因为它流程化了 Android 开发过程并让 Android 开发比以前更加容易接近<sup>8</sup>。Android Studio 目前是 Android 的官方 IDE。

## Android 工具生态系统

Android 是一个有着自身工具生态系统支持的技术平台。紧随 Android Studio 之后，Git 是 Android 生态系统中下一个最重要的工具。Git 是一款分布式的源代码管理工具，它正在迅速成为标准，不仅是在移动开发领域，而是在整个软件工程领域。我们所有的移动开发项目均使用 Git 来做版本控制，没有例外。Git 足可以用一本书来介绍，但幸运的是，你无须了解 Git 的所有功能就可以熟练地使用它。Android Studio 包含优秀的、全功能的且已集成的 Git 工具，它有着令人印象深刻的图形用户界面。本书涵盖了成为高效 Git 用户所需要了解的特性，如果想要深入掌握关于这个不可或缺工具的知识，我们为你指出了进一步学习所需的资源。

Android 生态系统中的另一个重要工具是 Gradle。Gradle 是一款类似于 Ant 和 Maven 的构建工具，它允许你管理库和库项目、运行仪器测试以及创建条件构建。Android Studio 在库管理方面本身已经很不错了，但 Gradle 使得此项任务更加便捷。与 Git 一样，Gradle 已经完全集成到 Android Studio，而且拥有令人印象深刻的界面，使得用户能够图形化地检查 Gradle 文件并监测 Gradle 构建过程的输出。

## Android 和 Java

如果在没有充分了解 Java 的情况下尝试在 Android Studio 中开发 Android App，那么你将遭遇挫折。出于诸多原因，Java 是一门极其有用且流行的编程语言。或许 Java 流行的最重要原因是其内存托管机制。内存托管意味着程序员不必考虑释放堆内存，也不必担心内存泄漏。在内存托管环境中进行开发的程序员通常更高效，而且程序的运行时错误会更少。和 Java 类似，Android 也是一种内存托管的编程环境。托管内存被证明是一个非常

---

<sup>8</sup> 开发 Android App 需要对 Java 有深入的了解。像 Android 这样强大的系统不会很容易掌握，但使用 Android Studio 会让开发 Android App 的任务变得轻松些。

的思路，使得微软和苹果公司均在各自的移动开发平台中采用了此模型<sup>9</sup>。

## 从 ADT/Eclipse 转型

如果你是一位有经验的 Android 开发者而且习惯使用 ADT 编程，那么你会收获惊喜。幸运的是，所有 SDK 工具(例如 DDMS 和 Hierarchy Viewer)都依然可用，你将发现在 Android Studio 中很容易就可以访问到它们。如果你是一位 ADT 用户，那么你可能需要不断地清理并重新构建项目，以便资源能够与源代码同步(可怕的 R.java 同步错误)。在使用 Android Studio 的这些日子里，我们一直没有被这个问题困扰过。如果你是一位有经验的 ADT 用户，那么为了提升使用 Android Studio 的效率，你将需要学习一些键盘快捷键、熟悉 Gradle 以及适应 Android Studio 的表现逻辑。总而言之，享受 Android Studio 带来的强大功能和乐趣只需付出很小的代价。

## 本书约定

Android Studio 在操作系统之间保持了高度的一致性。事实上，Windows 和 Linux 中的用户界面几乎就是一样的。不过，Mac OS 用户将会发现部分菜单的位置和键盘快捷键是不同的。当涉及需要系统导航的内容时，我们使用 Windows。不过，当给出键盘快捷键时，我们将同时包含 Windows-Linux 和 Mac 快捷键，中间使用竖线分隔(例如，Ctrl+K | Cmd+K)。我们会适时为 Mac 用户给出提示、链接和其他资源。

## 源代码

读者在学习本书中的示例时，可以手动输入所有的代码，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 <http://www.apress.com> 下载，还可访问 [www.tupwk.com.cn/downpage](http://www.tupwk.com.cn/downpage) 来下载源代码。下载代码后，只需要用自己喜欢的解压缩软件进行解压缩即可。

---

<sup>9</sup> Xcode——用于开发 iOS App 的 IDE，最近引入了一项称为“自动引用计数(Automatic Reference Counting)”的特性，使得编译器能够生成自动管理内存的代码。C#就是一种受到 Java 启发的内存托管的编程环境。

# 目 录

|       |                            |    |
|-------|----------------------------|----|
| 第 1 章 | Android Studio 入门          | 1  |
| 1.1   | 在 Windows 上安装 Java 开发工具包   | 1  |
| 1.1.1 | 在 Windows 上下载 JDK          | 2  |
| 1.1.2 | 在 Windows 上执行 JDK 向导       | 3  |
| 1.1.3 | 配置 Windows 环境变量            | 4  |
| 1.2   | 在 Mac 上安装 Java 开发工具包       | 7  |
| 1.2.1 | 在 Mac 上下载 JDK              | 7  |
| 1.2.2 | 在 Mac 上执行 JDK 向导           | 8  |
| 1.2.3 | 在 Mac 上配置 JDK 版本           | 9  |
| 1.3   | 安装 Android Studio          | 10 |
| 1.4   | 创建第一个项目: HelloWorld        | 12 |
| 1.5   | 使用 Android 虚拟设备管理器         | 15 |
| 1.6   | 在 AVD 上运行 HelloWorld       | 16 |
| 1.7   | 在 Andriod 设备上运行 HelloWorld | 17 |
| 1.8   | 小结                         | 19 |
| 第 2 章 | 在 Android Studio 中遨游       | 21 |
| 2.1   | 编辑器                        | 22 |
| 2.1.1 | Editor 选项卡                 | 22 |
| 2.1.2 | 折叠线                        | 23 |
| 2.1.3 | 标记栏                        | 23 |
| 2.1.4 | 工具按钮                       | 23 |
| 2.1.5 | 默认布局                       | 24 |
| 2.2   | 导航工具窗口                     | 24 |
| 2.2.1 | Project 工具窗口               | 25 |

|        |                       |    |
|--------|-----------------------|----|
| 2.2.2  | Stucture 工具窗口         | 26 |
| 2.2.3  | Favorites 工具窗口        | 26 |
| 2.2.4  | TODO 工具窗口             | 27 |
| 2.2.5  | Commander 工具窗口        | 27 |
| 2.3    | 主菜单栏                  | 27 |
| 2.4    | 工具栏                   | 28 |
| 2.5    | 导航栏                   | 28 |
| 2.6    | 状态栏                   | 28 |
| 2.7    | 常用操作                  | 29 |
| 2.7.1  | 选择文本                  | 29 |
| 2.7.2  | 使用 Undo 和 Redo        | 29 |
| 2.7.3  | 找到最近的文件               | 30 |
| 2.7.4  | 遍历最近的导航操作             | 30 |
| 2.7.5  | 剪切、复制和粘贴              | 30 |
| 2.8    | 上下文菜单                 | 31 |
| 2.9    | 获取帮助                  | 32 |
| 2.10   | 使用键盘导航                | 32 |
| 2.10.1 | Select In 命令          | 32 |
| 2.10.2 | Class 命令              | 33 |
| 2.10.3 | File 命令               | 33 |
| 2.10.4 | Line 命令               | 33 |
| 2.10.5 | Related File 命令       | 33 |
| 2.10.6 | Last Edit Location 命令 | 33 |
| 2.10.7 | Type Hierarchy 命令     | 34 |
| 2.10.8 | Declaration 命令        | 34 |
| 2.11   | 查找和替换文本               | 34 |
| 2.11.1 | Find 命令               | 34 |
| 2.11.2 | Find in Path 命令       | 34 |
| 2.11.3 | Replace 命令            | 35 |
| 2.11.4 | Replace in Path 命令    | 35 |
| 2.12   | 小结                    | 35 |

### 第 3 章 在 Android Studio 中编程 .....37

- 3.1 使用代码折叠 .....37
- 3.2 执行代码补全 .....39
- 3.3 注释代码 .....42
- 3.4 使用代码生成 .....42
  - 3.4.1 构造函数 .....43
  - 3.4.2 getter/setter .....44
  - 3.4.3 重载方法 .....44
  - 3.4.4 toString()方法 .....45
  - 3.4.5 代理方法 .....46
- 3.5 插入动态模板 .....47
- 3.6 移动代码 .....48
- 3.7 设计代码风格 .....50
  - 3.7.1 Auto-Indent Lines 选项 .....51
  - 3.7.2 Rearrange Code 选项 .....51
  - 3.7.3 Reformat Code 选项 .....52
  - 3.7.4 Surround With .....52
- 3.8 小结 .....53

### 第 4 章 重构代码 .....55

- 4.1 重命名 .....56
- 4.2 修改签名 .....57
- 4.3 类型迁移 .....58
- 4.4 移动 .....58
- 4.5 复制 .....59
- 4.6 安全删除 .....60
- 4.7 抽取 .....60
  - 4.7.1 抽取变量 .....61
  - 4.7.2 抽取常量 .....61
  - 4.7.3 抽取字段 .....62
  - 4.7.4 抽取参数 .....62
  - 4.7.5 抽取方法 .....63
- 4.8 高级重构 .....65
  - 4.8.1 下推成员和上拉成员 .....65
  - 4.8.2 使用代理代替继承 .....66
  - 4.8.3 封装字段 .....67
  - 4.8.4 封装方法返回值 .....68
  - 4.8.5 使用工厂方法代替构造函数 .....69
  - 4.8.6 将匿名类转换为内部类 .....69

### 4.9 小结 .....70

### 第 5 章 备忘录实验：第 1 部分 .....71

- 5.1 启动新项目 .....73
- 5.2 初始化 Git 仓库 .....74
- 5.3 构建用户界面 .....77
  - 5.3.1 使用可视化设计器 .....78
  - 5.3.2 编辑布局的原始 XML .....78
  - 5.3.3 添加视觉增强效果 .....83
  - 5.3.4 向 ListView 添加条目 .....84
  - 5.3.5 设置操作栏溢出菜单 .....86
- 5.4 持久化备忘录 .....87
  - 5.4.1 数据模型 .....87
  - 5.4.2 SQLite API .....89
- 5.5 小结 .....95

### 第 6 章 备忘录实验：第 2 部分 .....97

- 6.1 添加/删除备忘 .....97
- 6.2 响应用户交互 .....100
- 6.3 提供多选上下文菜单 .....102
  - 6.3.1 兼容较早的 SDK .....104
  - 6.3.2 添加上下文操作模式 .....105
- 6.4 实现添加、编辑和删除 .....107
  - 6.4.1 设计自定义对话框 .....108
  - 6.4.2 将设计转换为代码 .....108
  - 6.4.3 创建自定义对话框 .....110
  - 6.4.4 添加自定义图标 .....113
- 6.5 小结 .....115

### 第 7 章 Git 入门 .....117

- 7.1 安装 Git .....117
- 7.2 忽略文件 .....119
- 7.3 添加文件 .....120
- 7.4 克隆参考 App: Reminders .....121
  - 7.4.1 分叉和克隆 .....121
  - 7.4.2 使用 Git 日志 .....124
  - 7.4.3 分支 .....125
- 7.5 在分支上开发 .....125
  - 7.5.1 Git 提交和分支 .....131
  - 7.5.2 回退在哪里? .....132

|       |                       |     |         |                              |     |
|-------|-----------------------|-----|---------|------------------------------|-----|
| 7.5.3 | 合并                    | 136 | 第 10 章  | 货币实验：第 2 部分                  | 225 |
| 7.5.4 | Git 重置修改历史            | 138 | 10.1    | 定义 MainActivity 的成员          | 225 |
| 7.5.5 | Git 变基                | 142 | 10.2    | 从 bundle 中解压出货币<br>代码        | 226 |
| 7.5.6 | 分离头部                  | 144 | 10.3    | 创建选项菜单                       | 227 |
| 7.5.7 | 相对引用                  | 146 | 10.4    | 实现选项菜单行为                     | 229 |
| 7.5.8 | 在变基时解决冲突              | 148 | 10.5    | 创建 spinner_closed 布局         | 230 |
| 7.5.9 | Git 远端                | 153 | 10.6    | 将 mCurrencies 绑定到选择<br>列表    | 231 |
| 7.6   | 小结                    | 154 | 10.7    | 将选择列表行为代理给<br>MainActivity   | 232 |
| 第 8 章 | 设计布局                  | 155 | 10.8    | 创建偏好管理器                      | 234 |
| 8.1   | Activity              | 155 | 10.9    | 根据给定代码查找位置                   | 235 |
| 8.2   | View 和 ViewGroup      | 156 | 10.10   | 从货币字符串中抽取代码                  | 236 |
| 8.2.1 | 预览面板                  | 157 | 10.11   | 实现共同偏好                       | 237 |
| 8.2.2 | 宽度和高度                 | 159 | 10.12   | 按钮单击行为                       | 239 |
| 8.2.3 | 设计器模式                 | 161 | 10.13   | 保存开发者密钥                      | 240 |
| 8.2.4 | 帧布局                   | 161 | 10.14   | 获取开发者密钥                      | 241 |
| 8.2.5 | 线性布局                  | 164 | 10.15   | CurrencyConverterTask        | 242 |
| 8.2.6 | 相对布局                  | 166 | 10.15.1 | onPreExecute()               | 246 |
| 8.2.7 | 嵌套布局                  | 169 | 10.15.2 | doInBackground()             | 246 |
| 8.2.8 | 列表视图                  | 173 | 10.15.3 | onPostExecute()              | 246 |
| 8.3   | 布局设计指导原则              | 180 | 10.16   | 按钮选择器                        | 247 |
| 8.3.1 | 覆盖各种显示尺寸              | 180 | 10.17   | 启动图标                         | 248 |
| 8.3.2 | 组合在一起                 | 183 | 10.18   | 小结                           | 249 |
| 8.4   | Fragment              | 190 | 第 11 章  | 测试和分析                        | 251 |
| 8.5   | 小结                    | 200 | 11.1    | 创建新的仪器测试                     | 251 |
| 第 9 章 | 货币实验：第 1 部分           | 201 | 11.1.1  | 定义 SetUp()和 TearDown()<br>方法 | 252 |
| 9.1   | Currencies 规范         | 201 | 11.1.2  | 在 MainActivity 中定义<br>回调     | 254 |
| 9.2   | 初始化 Git 仓库            | 205 | 11.1.3  | 定义一些测试方法                     | 255 |
| 9.3   | 修改 MainActivity 的布局   | 207 | 11.1.4  | 运行仪器测试                       | 257 |
| 9.4   | 定义颜色                  | 211 | 11.1.5  | 修改 Bug                       | 258 |
| 9.5   | 为布局应用颜色               | 212 | 11.2    | 使用 Monkey                    | 259 |
| 9.6   | 创建并应用样式               | 213 | 11.3    | 使用分析工具                       | 260 |
| 9.7   | 创建 JSONParser 类       | 216 | 11.3.1  | 检查代码                         | 260 |
| 9.8   | 创建启动界面                | 217 | 11.3.2  | 分析依赖                         | 261 |
| 9.9   | 获取 JSON 格式的活动货币<br>代码 | 220 |         |                              |     |
| 9.10  | 启动 MainActivity       | 223 |         |                              |     |
| 9.11  | 小结                    | 224 |         |                              |     |

|        |                   |     |        |                   |     |
|--------|-------------------|-----|--------|-------------------|-----|
| 11.3.3 | 分析栈轨迹             | 262 | 14.2.2 | 方法跟踪工具            | 324 |
| 11.4   | 小结                | 264 | 14.2.3 | 分配跟踪器             | 325 |
| 第 12 章 | 调试                | 265 | 14.2.4 | 屏幕抓取              | 325 |
| 12.1   | 日志                | 265 | 14.3   | 导航编辑器             | 327 |
| 12.1.1 | 使用 logcat         | 266 | 14.3.1 | 设计用户界面            | 328 |
| 12.1.2 | 写入 Android 日志     | 268 | 14.3.2 | 导航编辑器初步           | 328 |
| 12.2   | 捕捉 Bug!           | 268 | 14.3.3 | 连接 Activity       | 330 |
| 12.2.1 | 使用交互式调试器          | 272 | 14.3.4 | 编辑菜单              | 331 |
| 12.2.2 | 表达式求值             | 275 | 14.4   | 终端                | 333 |
| 12.2.3 | 使用栈轨迹             | 277 | 14.4.1 | 查询设备              | 333 |
| 12.2.4 | 探索交互式调试的工具        |     | 14.4.2 | 安装 APK            | 333 |
|        | 窗口                | 280 | 14.4.3 | 下载文件              | 333 |
| 12.2.5 | 使用断点浏览器           | 281 | 14.4.4 | 上传文件              | 333 |
| 12.2.6 | 条件断点              | 283 | 14.4.5 | 端口转发              | 334 |
| 12.3   | 小结                | 285 | 14.5   | Google 云工具        | 334 |
| 第 13 章 | Gradle            | 287 | 14.5.1 | 创建 HelloCloud 前端  | 335 |
| 13.1   | Gradle 语法         | 288 | 14.5.2 | 创建 Java 后台模块      | 337 |
| 13.2   | IntelliJ 核心构建系统   | 289 | 14.5.3 | 组合在一起             | 339 |
| 13.3   | Gradle 构建概念       | 290 | 14.5.4 | 部署到 App Engine    | 343 |
| 13.3.1 | Gradle Android 结构 | 290 | 14.6   | 小结                | 346 |
| 13.3.2 | 项目依赖              | 291 | 第 15 章 | Android 可穿戴设备实验   | 347 |
| 13.4   | 案例研究: 使用 Gradle 的 |     | 15.1   | 设置可穿戴设备环境         | 347 |
|        | 天气预报项目            | 292 | 15.1.1 | 安装设备驱动程序          | 347 |
| 13.5   | Android 库依赖       | 299 | 15.1.2 | 设置 SDK 工具         | 350 |
| 13.5.1 | Java 库依赖          | 303 | 15.1.3 | 设置可穿戴虚拟设备         | 350 |
| 13.5.2 | 第三方库              | 311 | 15.1.4 | 设置 Android 可穿戴设备  |     |
| 13.6   | 打开较旧的项目           | 313 |        | 硬件                | 353 |
| 13.7   | 小结                | 314 | 15.2   | 创建 MegaDroid 项目   | 353 |
| 第 14 章 | 更多 SDK 工具         | 315 | 15.2.1 | 针对屏幕的优化技术         | 355 |
| 14.1   | Android 设备监视器     | 315 | 15.2.2 | 构建 watch-face 服务  | 356 |
| 14.1.1 | 线程监视器             | 316 | 15.2.3 | 初始化可绘制资源和         |     |
| 14.1.2 | 堆监视器              | 317 |        | 样式                | 358 |
| 14.1.3 | 分配跟踪器             | 318 | 15.2.4 | 管理手表更新            | 359 |
| 14.1.4 | 网络统计              | 319 | 15.2.5 | 绘制界面              | 363 |
| 14.1.5 | 层次查看器             | 320 | 15.3   | 小结                | 367 |
| 14.2   | Android 监视器       | 323 | 第 16 章 | 定制 Android Studio | 369 |
| 14.2.1 | 内存监视器             | 323 | 16.1   | 代码风格              | 370 |
|        |                   |     | 16.2   | 外观、颜色和字体          | 372 |

|                   |     |              |     |
|-------------------|-----|--------------|-----|
| 16.3 键盘映射.....    | 374 | 16.7 插件..... | 378 |
| 16.4 宏.....       | 375 | 16.8 小结..... | 380 |
| 16.5 文件和代码模板..... | 375 |              |     |
| 16.6 菜单和工具栏.....  | 377 |              |     |

# Android Studio 入门

本章带领你安装并配置开发环境，以便你能够跟随本书中的示例和实验进行学习。首先，你将要安装一个重要的前置组件，称为 Java 开发工具包(Java Development Kit, JDK)。接着你将下载并安装 Android Studio，以及构建 Android App 所需的一套软件工具——Android 软件开发包(Software Development Kit, SDK)。我们将为你展示如何使用 New Project Wizard 创建一个名为 HelloWorld 的简单项目。最后，将向你展示如何建立与 Android 虚拟设备(Android Virtual Device, AVD)和 Android 物理设备的连接。到本章结束时，你应该会拥有开始在 Android Studio 中开发 App 所需的所有东西。

## 1.1 在 Windows 上安装 Java 开发工具包

本节主要面向 Windows 用户。如果你是 Mac 用户，那么跳到标题为“在 Mac 上安装 Java 开发包”一节。Android Studio 使用 Java 工具链构建，因此在开始使用 Android Studio 之前，你需要确保已经在自己的电脑上安装了 Java 开发工具包(JDK)。如果你是资深的 Android 或 Java 开发者，那么你的电脑上很有可能已经安装 JDK 了。如果电脑上已经安装了 JDK，而且正在运行着 JDK 1.6 或更高版本，那么可以跳过此节。不过，你可能还是需要下载、安装并配置最新的 JDK。可以从下面的 Oracle 站点下载 JDK：

[www.oracle.com/technetwork/java/javase/downloads/index.html](http://www.oracle.com/technetwork/java/javase/downloads/index.html)

当你到达这个页面时，单击 Java Download 按钮，如图 1-1 所示。