

Android程序设计：第2版

作者：Zigurd Mednieks; Laird dornin ; G.Blake Meike; Masumi Nakamura

O'Reilly精品图书系列

Android程序设计：第2版

Programming Android,Second Edition

(美)梅德尼克斯 (Mednieks,Z.) 等著

祝洪凯 李妹芳 译

ISBN: 978-7-111-46758-8

本书纸版由机械工业出版社于2014年出版，电子版由华章分社（北京华章图文信息有限公司）全球范围内制作与发行。

版权所有，侵权必究

客服热线：+ 86-10-68995265

客服信箱：service@bbbvip.com

官方网址：www.hzmedia.com.cn

新浪微博 @研发书局

腾讯微博 @yanfabook

目录

[O'Reilly Media, Inc. 介绍](#)

[前言](#)

[第一部分 工具和基础](#)

[第1章 Android SDK的安装和系统需求](#)

[Android SDK的安装和系统需求](#)

[小试牛刀：确认安装是否能够正常工作](#)

[SDK的组件](#)

[保持版本最新](#)

[示例代码](#)

[关于代码阅读](#)

[第2章 Android开发之Java必备基础](#)

[Android正在重塑客户端Java](#)

[Java类型系统](#)

[作用域](#)

[Java编程惯例](#)

[第3章 Android应用的主要组件](#)

[活动、意图和任务](#)

[其他Android组件](#)

[组件生命周期](#)

[静态应用资源和上下文](#)

[Android应用时运行的环境](#)

[扩展Android](#)

[Android中的并发编程](#)

[序列化](#)

[Parcelable](#)

[第4章 应用发布之前的准备工作](#)

[应用签名](#)

[把应用投放到Android市场](#)

[其他发布](#)

[Google地图API密钥](#)

[指定API级别的兼容性](#)

[兼容不同的屏幕类型](#)

[第5章 Android软件开发工具Eclipse](#)

[Eclipse相关的概念和术语](#)

[Eclipse视图和透视图](#)

[Eclipse中的Java编程](#)

[Eclipse和Android](#)

[预防Bug并保持代码整洁](#)

[Eclipse特性和替代品](#)

[第二部分 Android框架](#)

[第6章 构建视图](#)

[Android GUI架构](#)

[组装图形界面](#)

[控制器布局](#)

[菜单栏和活动栏](#)

[视图调试和优化](#)

[第7章 Fragment和跨平台支持](#)

[创建Fragment](#)

[Fragment生命周期](#)

[Fragment管理器](#)

[Fragment事务](#)

[Support Package](#)

[Fragment和布局](#)

[第8章 二维和三维图形的绘制](#)

[自己动手开发部件](#)

[Bling](#)

[第9章 数据管理和数据持久化](#)

[关系数据库概览](#)

[SQLite](#)

[SQL语言](#)

[Android应用中的SQL和数据库为中心的数据模型](#)

[Android数据库类](#)

[Android应用的数据库设计](#)

[使用数据库API：MJAndroid](#)

[第三部分 Android的框架应用](#)

[第10章 运行良好的应用框架](#)

[对生命周期进行可视化](#)

[Fragment生命周期可视化](#)

[Activity类和运行良好的应用](#)

[Application类的生命周期方法](#)

[第11章 构建用户界面](#)

[Top-Level设计](#)

[用户界面的可视化编辑](#)

[从白板开始](#)

[Fragment布局](#)

[折叠和展开可扩展的UI界面](#)

[使得Activity、Fragment、Action Bar和多布局能够一起工作](#)

[其他活动](#)

[第12章 使用内容提供者](#)

[理解内容提供者](#)

[定义提供者的公用API](#)

[编写并集成内容提供者](#)

[文件管理和二进制数据](#)

[Android MVC和内容查看器](#)

[一个完整的内容提供者代码：SimpleFinchVideoContentProvider](#)

[声明自己的内容提供者](#)

[第13章 内容提供者作为RESTful Web服务的Facade](#)

[开发RESTful类型的Android应用](#)

[Network MVC](#)

[Network MVC方式的优点小结](#)

[代码实例：动态列出并缓存YouTube视频内容](#)

[Finch YouTube视频示例的源代码结构](#)

[视频搜索应用的各个步骤](#)

[步骤1：UI获取用户输入](#)

[步骤2：控制器监听事件](#)

[步骤3：控制器调用内容提供者（即模型）的managedQuery](#)

[步骤4：实现RESTful请求](#)

[第四部分 高级话题](#)

[第14章 搜索](#)

[搜索界面](#)

[查询建议](#)

[第15章 地理位置和地图定位](#)

[基于地理位置的服务](#)

[地图定位](#)

[Google Maps Activity](#)

[MapView和MapActivity](#)

[使用MapView](#)

[MapView和MyLocationOverlay初始化](#)

[暂停和恢复MapActivity](#)

[通过菜单按钮控制地图](#)

[通过键盘操作地图](#)

[没有地图的地理位置](#)

[街景（StreetView）](#)

[第16章 多媒体](#)

[音频和视频](#)

[播放音频和视频](#)

[录制音频和视频](#)

[存储的媒体内容](#)

[第17章 传感器、NFC、语音、手势和可访问性](#)

[传感器](#)

[近场通信](#)

[手势输入](#)

[可访问性](#)

[第18章 通信、身份认证、同步和社交媒体](#)

[账户信息](#)

[认证和同步](#)

[蓝牙](#)

[第19章 Android本地开发工具箱](#)

[本地方法和JNI调用](#)

[Android NDK](#)

[NDK提供的本地库文件和头文件](#)

[编译自己定制的库模块](#)

[本地Activity](#)

O'Reilly Media, Inc. 介绍

O'Reilly Media通过图书、杂志、在线服务、调查研究和会议等方式传播创新知识。自1978年开始，O'Reilly一直都是前沿发展的见证者和推动者。超级极客们正在开创着未来，而我们关注真正重要的技术趋势——通过放大那些“细微的信号”来刺激社会对新科技的应用。作为技术社区中活跃的参与者，O'Reilly的发展充满了对创新的倡导、创造和发扬光大。

O'Reilly为软件开发人员带来革命性的“动物书”；创建第一个商业网站（GNN）；组织了影响深远的开放源代码峰会，以至于开源软件运动以此命名；创立了Make杂志，从而成为DIY革命的主要先锋；公司一如既往地通过多种形式缔结信息与人的纽带。O'Reilly的会议和峰会集聚了众多超级极客和高瞻远瞩的商业领袖，共同描绘出开创创新产业的革命性思想。作为技术人士获取信息的选择，O'Reilly现在还将先锋专家的知识传递给普通的计算机用户。无论是通过书籍出版，在线服务或者面授课程，每一项O'Reilly的产品都反映了公司不可动摇的理念——信息是激发创新的力量。

业界评论

“O'Reilly Radar博客有口皆碑。”

——Wired

“O'Reilly凭借一系列（真希望当初我也想到了）非凡想法建立了数百万美元的业务。”

——Business 2.0

“O'Reilly Conference是聚集关键思想领袖的绝对典范。”

——CRN

“一本O'Reilly的书就代表一个有用、有前途、需要学习的主题。”

——Irish Times

“Tim是位特立独行的商人，他不光放眼于最长远、最广阔的视野并且切实地按照Yogi Berra的建议去做了：‘如果你在路上遇到岔路口，走小路（岔路）。’回顾过去Tim似乎每一次都选择了小路，而且有几次都是一闪即逝的机会，尽管大路也不错。”

——Linux Journal

前言

本书旨在帮助你创建良好的能实际使用的Android应用，而不仅仅是一些小的示例应用。

本书适合不同背景的Android编程人员。如果你一直在用Objective-C为iPhone或Mac OS编写应用，你会发现本书介绍的Android工具和Java语言特性等以及Android编程相关内容能够帮助你理解Android移动应用开发。如果你是一名经验丰富的Java编程人员，会发现本书介绍的Android应用架构有助于你在这个新的充满活力的Java客户端应用开发中运用Java专业知识。简而言之，本书面向的是那些在面向对象语言、移动应用、REST应用及类似领域中有一定经验的人。从本书中，可以更深入地了解Android编程，而这些内容远超过入门书或者一些在线的教程所涵盖的内容。

本书组织方式

我们希望能够快速带你掌握本书的内容。本书第一部分将帮你逐步掌握SDK工具，使你能够访问本书的示例代码，通过该部分的SDK介绍，你甚至可以加深你对SDK工具、Java和数据库设计的理解。如果你已经对第一部分所涉及的工具和基础知识非常熟悉，则可以跳过该部分直接进入本书第二部分，第二部分的目的是帮助你奠定构建较大型Android应用的基础。

本书的核心是一个应用实例，它使用Web服务给用户传送信息——这种方式用在很多应用的核心。还介绍了一种应用架构，以及一种使用Android框架类的新方法，它们能够帮助你非常高效地使用Android框架类。可以使用这个应用实例作为框架来创建自己的应用，也可以作为学习Android编程的工具。

本书最后一部分介绍Android在特定应用领域的API：多媒体、地理位置、传感器和通信等，从而帮助你在自己感兴趣的领域掌握Android应用编程。

当你阅读完本书时，希望你所掌握的不仅仅是参考资料知识和对一些例子的实践经验。希望你对如何创建优秀的Android应用也能有自己的思路。

本书约定

本书遵循以下字体体例：

斜体 (*Italic*)

表示新的术语、URL、E-mail地址、文件名和文件扩展名。

等宽字体 (`Constant width`)

用于程序清单及段落中的程序单元，如变量或函数名称、数据库、数据类型、环境变量、声明和关键字等。

等宽粗体 (`Constant width bold`)

显示命令或者其他应该由用户逐字输入的文本。

等宽斜体 (`Constant width italic`)

表示必须根据用户提供的值或者由上下文决定的值进行替代的文本。

注意：这个图标表示提示、建议或一般性注意事项。

警告：表示和附近文字相关的警告和注意事项。

使用本书的示例代码

本书旨在帮助你完成工作。通常来说，你可以在程序和文档中使用本书的代码。除非你使用了本书的大量代码，否则你无需联系我们获取许可。例如，编写一个用到本书中几段代码的程序就不需要获得许可，销售和分发O'Reilly丛书的代码CD-ROM需要获得许可，引用本书的示例代码来解决一个问题不需要获得许可，在你的产品文档中使用本书的大量代码需要获得许可。

我们不要求你引用本书时给出出处，但是如果你这么做，我们会对此表示感谢。出处通常包含标题、作者、出版社和ISBN。例如：“Programming Android, Second Edition, by Zigurd Mednieks, Laird Dornin, G. Blake Meike, and Masumi Nakamura. Copyright 2012 O'Reilly Media, Inc., 978-1-449-31664-8.”

如果你觉得你对本书示例代码的使用超出了这里给出的许可范围，请与我们联系：permissions@oreilly.com。

联系我们

有关本书的任何建议和疑问，可以通过下列方式与我们联系：

美国：

O'Reilly Media, Inc.

1005 Gravenstein Highway North

Sebastopol, CA 95472

中国:

北京市西城区西直门南大街2号成铭大厦C座807室 (100035)

奥莱利技术咨询(北京)有限公司

我们会在本书的网页中列出勘误表、示例和其他信息。可以通过http://oreilly/prog_android_2e访问该页面。

要评论或询问本书的技术问题,请发送邮件到:

bookquestions@oreilly.com

有关我们的书籍、会议、资源中心以及O'Reilly网络,可以访问我们的网站:

<http://www.oreilly.com>

<http://www.oreilly.com.cn>

在Facebook上联系我们: <http://facebook.com/oreilly>

在Twitter上联系我们: <http://twitter.com/oreillymedia>

在You Tube上联系我们: <http://youtube.com/oreillymedia>

致谢

本书的部分内容改编自作者以前出版的《Android Application Development》一书(O'Reilly)。

本书的草稿发布在O'Reilly开放源反馈发布系统(OFPS)上,以便得到读者的反馈,我们希望了解本书是否以及如何满足读者的需求。我们非常感谢参与OFPS的读者,我们感谢他们纠正了我们的错误,提高我们的写作水平。公开审校草稿将成为未来版本中的一部分,我们欢迎您对本书提出的任何意见。

Zigurd Mednieks

我非常感谢我的妻子Terry以及孩子们Majjia和Charles,感谢你们体谅我为了写本书而牺牲了很多陪伴你们的时间。本书之所以得以出版,要归功于我们的代理商Carole Jelen(Waterside Productions)提议出版本书以及Mike Hendrickson(O'Reilly)启动了该项目。编辑Brian Jepson和Andy Oram的组织 and 努力才有了本书的产出。感谢Johan Van der Hoeven,他对本书提出了很多意见,使得本书更准确简洁。感谢所有使用OFPS帮助本书改进的所有热心读者!

Laird Dornin

感谢我的妻子Norah鼓励我参与本书,虽然她对写书要付出的巨大努力并不了解。Acadia旅行、New Hampshire旅行以及无数个夜晚埋头工作,都是美好的回忆。很高兴,本书没有耽误我们更重要的“项目”——美丽的女儿Claire来到了我们身边。感谢编辑Andy以及本书的其他作者给予我这个机会。感谢Larry审查本书,并鼓励我致力于本书的写作。最后,感谢我们的主编Vijay和Johan,你们为改进本书内容指明了方向。

G.Blake Meike

感谢我们的代理商Carole Jelen(Waterside Productions),没有他的提议,本书还只会停留在一个好想法上。感谢编辑Brain Jepson和Andy Oram,处事总是那么优雅。本书还要归功于Johan Van der Hoeven和Vijay Yellapragada的技术审查;还要感谢Sumita Mukherji、Adam Zarenba, O'Reilly团队的其他成员。感谢那些通过O'Reilly的OFPS,给我们提出很多充满智慧的评论并指出错误的热心读者。感谢大家!与其他三位作者共同完成本书实在是我的荣幸。最后,永远感谢我的妻子Catherine,在我风调雨顺的时候给我忠告,在黑暗时期陪我度过困难。当然,本书的工作还在继续。

Masumi Nakamura

感谢我的朋友和家庭对我因为本书需要作出的付出的理解。尤其感谢Jessany,感谢你一直陪伴我这么多年!感谢Brian和Andy为编写和出版本书给了很多宝贵的建议。感谢其他作者带我一起参与本书写作。感谢WHERE公司的所有人对我技术上的热心帮助。最后,感谢本书的读者以及Android开发人员,你们的辛勤工作使得Android成为一个伟大的平台!

第一部分 工具和基础

第一部分主要介绍如下内容：如何安装和使用工具，编写良好的Android代码所需的Java知识，以及如何设计和使用SQL数据库。这些内容对于Android程序中的Android应用模型、持久性系统和关键设计模式的实现是至关重要的。

第1章 Android SDK的安装和系统需求

本章介绍如何安装Android软件开发工具包（SDK）及一些可能用到的相关软件。最后，我们将可以在一个模拟器上运行简单的“Hello World”程序。在Windows、Mac OS X和Linux这几个系统上都可以进行Android应用开发。下面将载入该软件，介绍SDK中的工具并对示例源代码进行说明。

本书（尤其是本章）将介绍各个网站提供的用于创建Android应用的工具的安装和更新说明。寻找相关工具的信息和链接最重要的网站是Android开发者网站：

<http://developer.android.com>

尽管各个环节的细节说明会有一定差异，但这里主要是展示整个安装的过程，并对Android的各个部分及其开发工具进行介绍说明。

Android SDK的安装和系统需求

要安装Android SDK，需要两个系统软件，它们不是Android SDK的一部分：即Java开发工具包（JDK）和Eclipse集成开发环境（IDE）。这两个系统没有作为Android SDK的一部分进行提供是考虑到你可能已在非Android软件的开发中使用它们，或者你的系统中可能已经安装了这两个系统软件，而在一个系统上安装这两个软件的不同版本可能会导致版本冲突。

Android SDK和最近发布的几个版本的JDK和Eclipse IDE都是兼容的。但最好安装这些工具的最新版本。其具体的要求在Android开发者的网站的系统需求（system requirements）页面有介绍：<http://developer.android.com/sdk/requirements.html>。

在Android软件开发中，除了Eclipse之外，还可以使用几款其他IDE。在Android文档中给出了可以使用的其他IDE的信息：<http://developer.android.com/guide/developing/other-ide.html>。之所以在本书中选择Eclipse作为IDE，是因为Eclipse对Android SDK工具和其他插件的支持是最好的，而且Eclipse也是用得最广泛的Java IDE，当然有很多Java程序员更喜欢IntelliJ IDEA。

Java开发工具包（JDK）

如果你的系统已经安装了最新的JDK，就不需要再安装它。JDK提供了一些工具，如Java编译器，IDE和SDK使用这些工具来开发Java程序。JDK还包含Java运行时环境（Java Runtime Environment, JRE），JRE使得如Eclipse这样的Java程序能够在系统上运行。

如果你使用的Macintosh运行的是Android SDK所支持的某个版本的Mac OS X，则在该系统上已经安装了JDK。

如果你是Linux或Windows用户，或者由于某些其他原因，需要从Oracle网站下载安装JDK，可以在这里找到它：<http://www.oracle.com/technetwork/java/javase/downloads/index.html>。

你下载的Windows安装文件是个可执行文件。运行该可执行文件即可安装SDK。

Linux用户需要把下载的JDK文件夹解压到他们的home目录，执行以下步骤安装SDK。这些步骤假定你希望使用当前的Oracle JDK作为默认的Java运行时JDK：

根据系统版本，下载压缩包（archive）或安装包（package）（如果是安装包，通过安装包管理器完成安装，否则继续以下步骤）

```
tar
-xvf archive-name.tar.gz
```

JDK压缩包会解压到./jdk-name目录中。现在，把该JDK目录mv到/usr/lib中：

```
sudo mv ./jdk-name /usr/lib/jvm/jdk-name
```

在Linux环境中，把JDK放到这个位置使得它可以作为配置选项，如果你有其他项目或程序需要其他版本的JRE或JDK，这样做很有用。现在，运行如下命令：

```
sudo update-alternatives --install "/usr/bin/java" "java" \
    "/usr/lib/jvm/jdk-name/bin/java" 1
sudo update-alternatives --install "/usr/bin/javac" "javac" \
    "/usr/lib/jvm/jdk-name.0/bin/javac" 1
sudo update-alternatives --install "/usr/bin/javaws" "javaws" \
    "/usr/lib/jvm/jdk-name/bin/javaws" 1
sudo update-alternatives --config java
```

你会看到如下的输出结果：

对于/usr/bin/java而言，还有另外3种java选项。

选择路径		
优先级		
状态		
*0	/usr/lib/jvm/java-6-openjdk/jre/bin/java	63
默认选择		
1	/usr/lib/jvm/java-6-openjdk/jre/bin/java	63
手动选择		
2	/usr/lib/jvm/java-6-sun/jre/bin/java	63
手动选择		
3	/usr/lib/jvm/jdk1.7.0/jre/bin/java	1
手动选择		

按下enter键保留当前选项[*]，或者按下要选择的序号：

当选择了正在安装的JDK，会看到如下输出：

```
update-alternatives: using /usr/lib/jvm/jdk1.7.0/jre/bin/java to provide
/usr/bin/java (java) in manual mode.
```

为javac选择，命令如下：

```
sudo update-alternatives --config javac
```

为javaws选择，命令如下：

```
sudo update-alternatives --config javaws
```

输出的具体内容取决于系统上安装的Java版本以及JDK的版本号，可能和你在这里看到的有所不同。

对于所有的操作系统，你都可以通过以下命令检查安装的Java版本：

```
java -version
```

输出的Java版本号应该和安装的版本号一致。如果不一致，重新安装，并确保在安装过程中没有弹出错误信息。

Eclipse集成开发环境（IDE）

Eclipse是一个通用的技术平台。它用于创建多种语言的IDE及很多针对特定的SDK定制的IDE。在作为软件开发工具之外，Eclipse还有一些其他用途，如为以Lotus Notes为代表的一些其他应用提供富客户端平台（Rich Client Platform，RCP）。

通常情况下，Eclipse是作为IDE用于编写、测试和调试软件，尤其是Java开发。此外，在很多Java软件开发中还有一些以Eclipse为基础衍生出来的IDE和SDK。具体到本书所要介绍的Android开发，我们可以选用得到广泛应用的Eclipse包和插件来进行开发。下面先来看看如何下载和安装Eclipse包。

Eclipse包可以在<http://www.eclipse.org/downloads>下载。

在这个页面中列出了最常用的Eclipse包。Eclipse包是多个Eclipse模块的组合，这使得Eclipse更适合于某类软件的开发。通常情况下，Eclipse用户需要先从该页面下载某个版本的Eclipse包，然后再安装一些插件进行定制。当你为安装的Eclipse添加Android开发工具（ADT）时，所遵循的正是这个步骤。Android开发者网站的系统需求页面列出了Android开发所需要的基础软件Eclipse的3种选择：

- Eclipse经典版（Eclipse 3.5或更高版本）
- 面向Java开发者的Eclipse IDE
- 面向富客户端平台（RCP）和插件开发者的Eclipse

上述这些版本都可以正常工作，只要不是开发Eclipse插件，选择Eclipse经典版或Java开发版（EE或标准的）都是可以的。笔者使用的一直是Java EE开发版（“EE”表示企业版，即Enterprise Edition），本书截图中的Eclipse就是这个版本。

Eclipse下载网站可以判断出你所用的操作系统的类型并自动为你选出该系统下可用的下载，不过你可能需要指定自己的系统是32位还是64位的。从下载网站下载的文件是一个压缩包（archive）。在安装Eclipse时，首先需要解压这个压缩包，然后把其中的Eclipse文件夹复制到home目录下。在我们的系统上可直接运行的Eclipse可执行程序就在这个目录下。

警告：把Eclipse安装在home目录下（或者另一个目录）是非常重要的做法，尤其是在所用的系统上有多个用户账户的情况下。不要使用系统的软件包管理器。Eclipse安装只是众多Eclipse插件组合中的一组。此外，你很有可能需要进一步对Eclipse进行定制。Eclipse插件和更新的管理独立于系统中的其他软件。

如果你的系统是Ubuntu或其他Linux发行版，则安装Eclipse时不要直接使用系统自带的库。如果你的Eclipse是按照这种方式安装的，则建议删除它并参照本节介绍的步骤重新安装。在Ubuntu中的Eclipse是Ubuntu所基于的Debian库中自带的。使用系统发行版的软件库中的Eclipse安装包来安装不是太常见，因为大多数时候，这些Eclipse的版本会比较老。

为了确认Eclipse安装正确并且支持Eclipse运行的JRE包也存在，可在Eclipse目录下运行其可执行的二进制文件 [\[1\]](#)。可以考虑为该可执行文件创建一个快捷方式，这样可以更快地启动Eclipse。此时应该可以看到如图1-1所示的欢迎界面。

□

图1-1：第一次运行Eclipse时的欢迎界面

没有安装脚本。本书后面会介绍如何配置Eclipse插件，这样Eclipse就可以找到SDK安装路径。Android工具箱的展现形式、需求和功

欢迎访问：电子书学习和下载网站 (<https://www.shgis.cn>)

文档名称：《Android程序设计：第2版》Zigurd Mednieks; Laird dornin ; G.Blake Meike;

请登录 <https://shgis.cn/post/266.html> 下载完整文档。

手机端请扫码查看：

