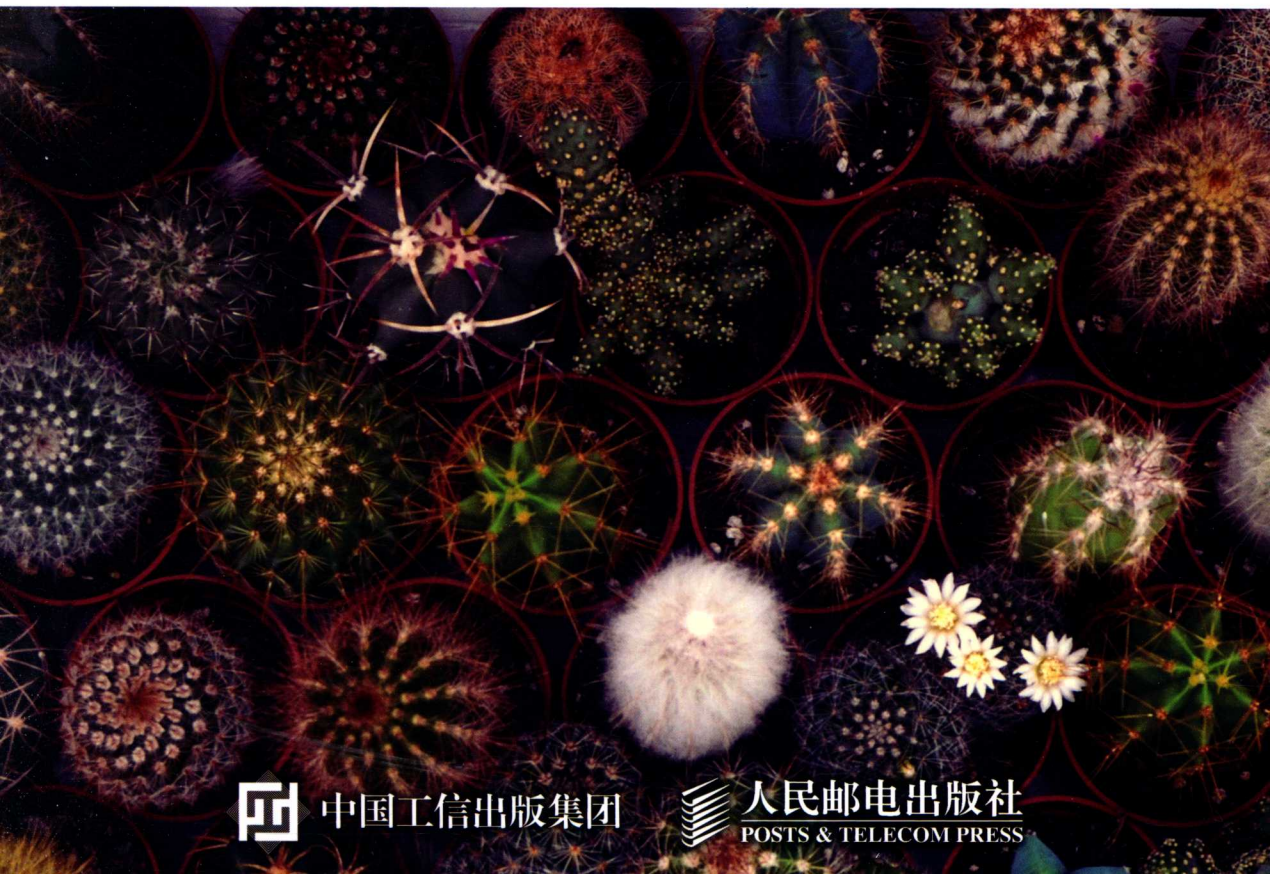
BrainySoftware

Java和Android开发学习指南

（第2版）

[加] Budi Kurniawan 著
李强 译



中国工信出版集团
人民邮电出版社
POSTS & TELECOM PRESS

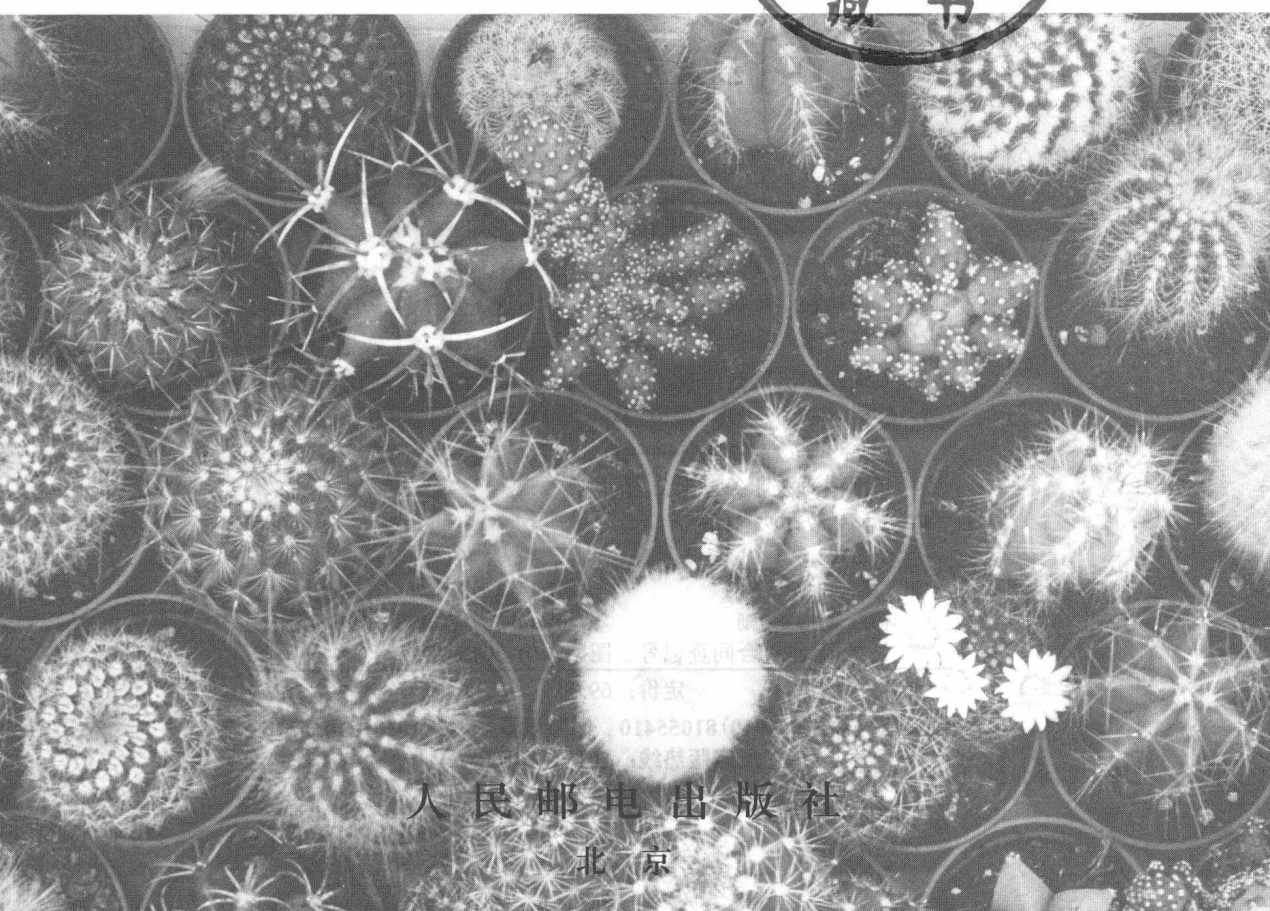
BrainySoftware

Java和Android开发学习指南

（第2版）

安阳师范学院图书馆
藏书

[加] Budi Kurniawan 著
李强 译



人民邮电出版社
北京

图书在版编目（CIP）数据

Java和Android开发学习指南 : 第2版 / (加) 克尼亚万 (Kurniawan,B.) 著 ; 李强译. -- 北京 : 人民邮电出版社, 2016.3
ISBN 978-7-115-41753-4

Ⅰ. ①J… Ⅱ. ①克… ②李… Ⅲ. ①JAVA语言－程序设计②移动终端－应用程序－程序设计 Ⅳ. ①TP312 ②TN929.53

中国版本图书馆CIP数据核字(2016)第032311号

版 权 声 明

Simplified Chinese translation copyright ©2016 by Posts and Telecommunications Press
ALL RIGHTS RESERVED
Java for Android, Second Edition，by Budi Kurniawan
Copyright © 2015 by Brainy Software Inc.

本书中文简体版由作者 **Paul Deck** 授权人民邮电出版社出版。未经出版者书面许可，对本书的任何部分不得以任何方式或任何手段复制和传播。
版权所有，侵权必究。

◆ 著　[加] Budi Kurniawan
译　李　强
责任编辑　陈冀康
责任印制　张佳莹　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：32.5
字数：1047 千字　2016 年 3 月第 1 版
印数：1－3 000 册　2016 年 3 月河北第 1 次印刷
著作权合同登记号　图字：01-2015-7997 号

定价：69.00 元

读者服务热线：(010)81055410　印装质量热线：(010)81055316
反盗版热线：(010)81055315

内 容 提 要

本书是 Java 语言学习指南，特别针对使用 Java 进行 Android 应用程序开发展开了详细介绍。

全书共 50 章。分为两大部分。第 1 部分（第 1 章到第 22 章）主要介绍 Java 语言基础知识及其功能特性。第 2 部分（第 23 章到第 50 章）主要介绍如何有效地构建 Android 应用程序。

本书适合任何想要学习 Java 语言的读者阅读，特别适合想要成为 Android 应用程序开发人员的读者学习参考。

前　言

欢迎阅读本书。

本书是针对那些想要学习 Java 语言，特别是想要进行 Android 应用程序开发的人编写的。本书包含两个部分，第 1 部分主要介绍 Java，第 2 部分介绍如何有效地构建 Android 应用程序。

本书中关于 Java 的内容并非每一项 Java 技术都讲到（在一本书里，无论如何也不可能涵盖所有的内容，这也是为什么大多数 Java 图书都专注于一项技术）。但是，本书介绍了最重要的 Java 编程主题，这些主题是你自学其他技术所必须掌握的。特别是第 1 部分介绍了一名专业的 Java 程序员所必须掌握的 3 个主题：

- Java 编程语言。
- 使用 Java 的面向对象编程（OOP）。
- Java 核心库。

构建一门高效的 Java 课程的难点在于，这 3 个部分是彼此独立的。一方面，Java 是一门 OOP 语言，因此，如果你了解 OOP 的话，其语法很容易学习。另一方面，像继承、多态和数据封装这样的 OOP 功能，最好是和现实世界的例子一起来讲解。遗憾的是，理解现实世界的 Java 编程需要具备 Java 核心库的知识。

由于这种相关性，这 3 个主题并没有分为 3 个独立的部分。相反，介绍一个主题的章节和另一个主题的章节是相互交织的。例如，在介绍多态之前，本书确保你熟悉某些 Java 类，以便可以给出现实世界的例子。此外，如果不能全面理解某种类的话，是不能有效地讲解诸如泛型这样的语言特性的，因此，本书在讨论了支持类之后才介绍泛型。

还有一些情况是，可能会在一个或多个地方找到一个主题。例如，for 语句是一种基本的语言功能，应该在较早的章节中介绍，同时 for 也可以用于遍历一个集合对象，只能在教授了集合框架之后才能介绍这种功能。因此，for 是在第 3 章中初次介绍的，在第 14 章再次介绍。

本书第 2 部分介绍了 Android 框架，以及一个 Java 程序员开发 App 所需要掌握的工具。然后，介绍了进行 Android 编程的基本话题，包括 Android 用户界面、位图和图形处理、动画、音频/视频录制，以及任务同步。

下面的内容从一个较高的高度介绍了 Java，介绍了 OOP 并且简单描述了本书中每章的内容。

Java 语言及其技术

Java 不仅是一种面向对象编程语言，也是一组技术，它使得软件开发更加快速，并且使得应用程序更加健壮和安全。多年来，Java 已经是一种技术选择，因为它提供了如下的一些优点：

- 平台无关性。
- 易于使用。
- 内容丰富的库，加快了应用程序开发。
- 安全性。
- 可伸缩性。
- 广泛的工业支持。

Sun Microsystems 公司于 1995 年引入了 Java，并且 Java 已经成为了通用性的语言，而它最初为人们所熟知，还是作为一种编写 applet 这种较小的、在 Web 浏览器中运行并为静态 Web 添加交互的语言。对于 Java 早期的成功，互联网的发展功不可没。

此为试读，需要完整PDF请访问：www.ertongbook.com

正如我们所说，applet 并不是使 Java 发光的唯一因素，Java 吸引人的另一个因素是平台无关性的承诺，由此而产生的口号是“一次编写，处处运行”。这意味着，你编写的完全一样的程序，将会在 Windows、UNIX、Mac、Linux 以及其他的操作系统上运行。而这是其他一些编程语言所无法做到的。那个时候，C 和 C++还是开发应用程序最常用的语言。Java 一诞生，就抢了它们的风头。

这就是 Java#1.0 版。

1997 年，Java 1.1 发布了，添加了更好的事件模型、Java Beans 以及国际化等重要的功能。

1998 年 12 月，Java 1.2 发布了。在其发布 3 天之后，版本号改为 2，这标志着一次巨大的市场活动的开始，从 1999 年开始，Java 作为“下一代”技术而发售。Java 2 有 4 个版本销售，分别为标准版（J2SE）、企业版（J2EE）、Micro 版（J2ME）和 Java Card（在其品牌名中没有出现 2）。

2000 年发布的下一个版本是 1.3，也就是 J2SE 1.3。两年以后出现版本 1.4，即 J2SE 1.4。2004 年发布了 J2SE 1.5 版。然而，Java 2 的 1.5 版本的名字改为了 Java 5。

2006 年 11 月 13 日，在 Java 6 正式发布前的 1 个月，Sun Microsystems 公司宣布将 Java 开源。Java SE 6 是 Sun 公司邀请外部开发者设计代码并帮助修复 bug 的第一个版本。Sun 公司之前确实也接受非雇员设计的代码，例如 Doug Lea 对于多线程方面的工作，但是，这是 Sun 公司第一次发出公开的邀请。Sun 公司承认自己的资源有限，而外界的贡献者能够帮助他们更快地完成开发任务。

2007 年 5 月，Sun 公司将 Java 源代码作为免费软件开放给了 OpenJDK 社群。随后，IBM 公司、Oracle 公司和 Apple 公司加入了 OpenJDK 社群。

2010 年，Oracle 公司收购了 Sun 公司。

2011 年 7 月发布了 Java 7。2014 年 3 月发布了 Java 8，这都是通过 OpenJDK 进行开源协作的结果。

Java 何以做到平台无关

你可能听到过术语“平台无关性”或“跨平台”，这意味着你的程序可以在多种操作系统上运行。这是对 Java 的流行贡献最大的功能。但是，是什么使得 Java 能够与平台无关呢？

在传统的编程中，源代码编译为可执行的代码。可执行代码只能在它所针对的平台上运行。换句话说，针对 Windows 编写和编译的代码，只能够在 Windows 上运行，针对 Linux 编写的代码，只能够在 Linux 上运行，以此类推，如图 I.1 所示。

图 I.1 传统编程范型

Java 程序则编译为字节码。字节码本身不能运行，因为它不是原生代码。字节码只能够在 Java 虚拟机（JVM）上运行。JVM 是一个原生应用程序，它负责解释字节码。通过使用 JVM 可用在众多的平台上，Sun 公司将 Java 变为了跨平台的语言。如图 I.2 所示，完全相同的字节码，可以在已经开发了 JVM 的任何操作系统上运行。

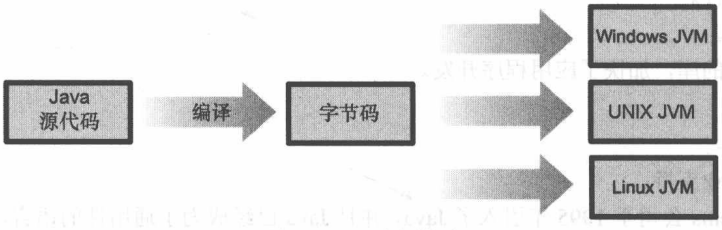


图 I.2 Java 编程模型

当前，JVM 对于 Windows、UNIX、Linux、Free BSD，以及世界上主流操作系统均可用。

JDK、JRE 和 JVM 有何区别

前面提到 Java 程序必须要编译。实际上，任何编程语言都需要编译才可用。编译器是将程序源代码转换为一种可执行格式（字节码、本地代码或者其他形式）的程序。在开始用 Java 编程之前，你需要下载一个 Java 编译器。Java 编译器是名为 javac 的程序，是 Java compiler 的缩写。

尽管 javac 可以把 Java 源代码编译为字节码，但是要运行字节码，你需要一个 Java 虚拟机。此外，由于要使用 Java 核心库中的各种类，还需要下载这些库。Java 运行时环境（Java Runtime Environment，JRE）包含了一个 JVM 和类库。你可能猜到了，针对 Windows 的 JRE 和针对 Linux 的 JRE 是不同的，和针对其他操作系统的 JRE 也不同。

Java 软件以两种形式可用：

- JRE，包括一个 JVM 和核心库。适于运行字节码。
- JDK，包括 JRE 加上一个编译器和其他的工具。这是编译 Java 程序以及运行字节码所需要的软件。

概括起来，JVM 是运行字节码的本地应用程序。JRE 是包含了 JVM 和 Java 类库的环境。JDK 包括 JRE 以及其他工具，包括一个 Java 编译器。

JDK 的第 1 个版本是 1.0。其后的版本是 1.1、1.1、1.2、1.3、1.4、1.5、1.6、1.7 和 1.8。对于较小的发布，在版本号的后面再添加另外一个数字。例如，1.8.1 是 1.8 版本的第一个较小的升级。

JDK 1.8 比 JDK 8 更为知名。包含在一个 JDK 中的 JRE，其版本和 JDK 的版本相同。因此，JDK 1.8 包含了 JRE 1.8。这个 JDK 通常也叫作 SDK（软件开发工具箱）。

除了 JDK，Java 程序员还需要下载说明了核心库中的类、接口和枚举类型的 Java 文档。你可以从提供 JRE 和 JDK 的相同的 URL 来下载文档。

Java 2、J2SE、J2EE、J2ME、Java 8 分别是什么？

Sun Microsystems 公司对于推动 Java 做了很好的工作。其市场策略的一部分是，创造了 Java 2 这个名字，而实际上它是基于 JDK 1.2 的。Java 2 有 3 个版本。

- Java 2 Platform，Standard Edition（J2SE）。J2SE 基本上就是 JDK。它还作为 J2EE 中定义的技术的基础。
- Java 2 Platform，Enterprise Edition（J2EE）。它定义了开发基于组件的多端企业应用程序的标准。功能包括 Web 服务支持和开发工具。
- Java 2 Platform，Micro Edition（J2ME）。它针对在消费设备（例如手机和 TV 机顶盒）上运行的应用程序提供了一个环境。J2ME 包括一个 JVM 和一组有限的类库。

第 5 版中的名称变了。J2SE 变成了 Java Platform，Standard Edition 5（Java SE 5）。此外，J2EE 和 J2ME 中的 2 已经去掉了。企业版当前的版本是 Java Platform，Enterprise Edition 7（Java EE 7）。J2ME 现在叫作 Java Platform，Micro Edition（Java ME，不带版本号）。在本书中，Java 8 通常写作 Java SE 8。

和 Sun 公司作为产品推出的第一个 Java 版本 J2SE 1.4 不同，Java SE 5 及其后的 Java 版本都是一组规范，定义了需要实现的功能。这个软件自身叫作一个参考实现。Oracle、IBM 和其他的公司与 OpenJDK 一起工作，提供了 Java SE 8 参考实现，以及 Java 的下一个版本的参考实现。

Java EE 6 和 Java EE 7 也是一组规范，包括了 servlets、JavaServer Pages、JavaServer Faces、Java Messaging Service 等技术。要开发并运行 Java EE 应用程序，需要一个 Java EE 应用程序服务器。任何人都可以实现一个 Java EE 应用程序服务器，这就解释了为什么市场上有各种应用程序服务器可供使用，包括很多开源的产品。如下是 Java EE 6 和 Java EE 7 应用程序服务器的示例：

- Oracle WebLogic
- IBM WebSphere
- GlassFish
- Jboss
- WildFly
- Apache Geronimo
- Apache TomEE

在如下网址中可以找到完整的列表。

```
http://www.oracle.com/technetwork/java/javaee/overview/compatibility-jsp-136984.html
```

JBoss、GlassFish、WildFly、Geronimo 和 TomEE 都是开源的 Java EE 服务器。它们有各自不同的许可，因此，在决定使用该产品之前要确保先阅读其许可。

JCP 程序

Java 持续成为占有统治地位的技术，这和 Sun 公司的很多策略有关，Sun 公司纳入了其他产业的从事者来确定 Java 的未来。通过这种方式，很多人感到他们拥有 Java。很多大型的公司，例如 IBM、Oracle、Google、Fujitsu 等，都在 Java 中投资颇多，因为它们都可以提出一种技术的规范，并且推动 Java 技术的下一个版本成为它们想要看到的样子。协作的努力采取了 JCP 程序的形式。其 Web 站点的 URL 是 http://www.jcp.org。

JCP 程序所产生的规范，叫作 Java Specification Requests（JSR）。例如，JSR 337 指定了 Java SE 8。

面向对象编程概览

面向对象编程（object-oriented programming，OOP）通过基于现实世界的对象来建模应用程序而起作用。OOP 的 3 大原理是封装、继承和多态。

OOP 的好处是实实在在的。这也是大多数现代编程语言（包括 Java），都采用面向对象范型的原因。我甚至可以引用语言变迁中的两个知名的例子来说明 OOP 所得到的支持：C 语言演变为 C++，而 Visual Basic 升级为 Visual Basic.NET。

下面介绍 OOP 的好处，并且介绍学习 OOP 的容易之处和难处。

OOP 的好处

OOP 的好处包括代码易于维护、代码复用以及扩展能力。下面更为详细地介绍这些好处。

1. 易于维护。现代软件应用程序倾向于变得很大。一个较大的系统可能曾包含数千行的代码。而现在，即便是那些数百万行代码的程序，也不能算是大程序了。C++之父 Bjarne Stroustrup 曾经说过，当系统变得越来越大的时候，就会给开发者带来问题。无论如何，一个较小的程序可以用任何语言编写。即便不是很容易的话，最终也可以让它工作。但是一个较大的程序则完全不同。如果你没有使用良好的编程技术，你刚修改完旧的错误，就会出现新的错误。

之所以出现这种情况，是因为较大的程序中存在相互依赖的情况。当修改了程序中的一部分的内容时，你可能不会意识到这个修改会影响到其他的部分。OOP 很容易让应用程序模块化，并且模块化会降低维护的难度。模块化是 OOP 内在的特性，因为作为对象的模板，一个类自身就是模块化的。好的设计应该允许

一个类包含类似的功能和相关的数据。OOP 中经常使用的一个重要的术语是耦合（coupling），它表示两个模块之间相互作用的程度。各个部分之间的松耦合，使得代码更容易复用，而代码复用正是 OOP 的另一个好处。

2．复用性。复用性表示之前编写的代码，可以由代码的作者或其他需要使用最初代码所提供的相同功能的人重复使用。这并不会令人吃惊，因为 OOP 语言常常带有一组准备好的库。在 Java 中，该语言带有数百个类库或应用程序接口（application programming interfaces，API），都经过了仔细的设计和测试。编写和发布你自己的库也很容易。在编程平台中，支持可复用性是非常吸引人的，因为它缩短了开发时间。

类的可复用性的主要的挑战之一是要为类库创建好的文档。一个程序员有多快才能找到他想要的功能的类？查找这样一个类更快，还是从头开始编写一个新的类更快？好在 Java 核心 API 和扩展 API 带有详尽的文档。

可复用性并不只是适用在编码阶段复用类或其他的类型，当设计一个 OO 系统的应用程序的时候，OO 设计问题的解决方案也可以复用。这些解决方案叫作设计模式（design pattern）。为了使得引用每个解决方案更加容易，需要给每个模式一个名称。可复用的设计模式的最早的目录，可以在 Erich Gamma、Richard Helm、Ralph Johnson 和 John Vlissides 所编写的经典图书《Design Patterns: Elements of Reusable Object-Oriented Software》中找到。

3．可扩展性。每个应用程序都是独特的，它有自己的需求和规范。就可复用性而言，有时候，你不会发现一个已有的类提供了你的应用程序所需的确切功能。然而，你可能会发现有一两个类提供了部分功能。可扩展性意味着，你仍然可以通过扩展它们以满足你的需求，从而使用这些类。你仍然可以节省时间，因为你不必从头开始编写代码。

在 OOP 中，可扩展性是通过继承来实现的。你可以扩展一个已有的类，为其添加一些方法或数据，或者修改你不喜欢的方法的行为。如果你知道一些基本功能在很多情况下都要使用，但是你不想自己的类提供非常具体的功能，你可以提供一个泛型类，随后可以扩展这个类来为应用程序提供具体的功能。

OOP 难吗

Java 程序员需要掌握 OOP。然而，如果你曾经使用一种过程式语言，如 C 或 Pascal 的话，那么，掌握 OOP 很有意义。在这一点上，有坏消息，也有好消息。

先来看坏消息吧。

研究者已经争论过在学校教授 OOP 的最好的方法，一些人认为最好是在介绍 OOP 之前教授过程式编程。在很多课程中，当学生接近其大学时期的最后一年的时候，才教授 OOP 课程。

然而，最近的研究表明，一些具备过程式编程技能的人，其思考范型与那些 OOP 程序员的视角以及解决问题的方式非常不同。当这类人学习 OOP 的时候，他们所面临的最大的挑战是，必须经过范型迁移。也就是说，要花 6～18 个月的时间将思维方式从过程式范型过渡到面向对象范型。另一项研究表明，那些没有学习过过程式编程的学生，则不会认为 OOP 有那么难。

现在来看好消息。

Java 可以算得上最容易学习的 OOP 语言了。例如，你不需要担心指针，不必花费宝贵的时间来解决由于没有成功销毁未使用的对象而导致的内存泄露问题等。最后，Java 带有非常充足的类库，在其早期的版本中，这些类库的 bug 也很少。一旦你了解了 OOP 的边边角角的知识，使用 Java 编程真的很容易。

本书简介

下面我们来看看本书中每一章的内容。

第 1 部分 Java

第 1 章介绍了如何下载和安装一个 JDK，其目标是让你对使用 Java 有一些感觉。此部分内容包括编写一个简单的 Java 程序，使用 javac 工具来编译它，以及使用 java 程序运行它。此外，对于编码惯例给出了一些建议，还介绍了集成开发环境。

第 2 章介绍了 Java 语言的语法。你将会了解诸如字符集、基本类型、变量以及操作符等主题。

第 3 章介绍了 Java 语句，包括 for、while、do-while、if、if-else、switch、break 和 continue 等。

第 4 章是本书中的 OOP 内容的开始。首先介绍了 Java 对象是什么，以及如何存储在内存中。然后，继续讨论类、类成员以及两个 OOP 概念（抽象和封装）。

第 5 章介绍了 Java 核心库中的重要的类，包括 java.lang.Object、java.lang.String、java.lang.StringBuffer、java.lang.StringBuilder、包装类和 java.util.Scanner。这是很重要的一章，因为本章所介绍的这些类是 Java 中最常用的类。

第 6 章讨论了数组，这是 Java 的一种特殊的语言功能，得到了广泛的使用。本章还介绍了操作数组的工具类。

第 7 章讨论了支持代码扩展的 OOP 特性。本章教你扩展一个类、影响子类的可见性、覆盖方法等。

第 8 章介绍了错误处理机制。毫无疑问，错误处理在任何语言中都是一项重要的功能。作为一门成熟的语言，Java 有非常强大的错误处理机制，可以帮助预防 bug。

第 9 章介绍了操作数字时的 3 个问题：解析、格式化和操作。本章介绍了能够帮助你完成这些任务的 Java 类。

第 10 章介绍了接口，它不仅仅是没有实现的类。接口定义了服务提供者和客户之间的一个协议。本章介绍了如何使用接口和抽象类。

第 11 章介绍了多态，并且给出了有用的示例。多态是 OOP 的主要支柱之一。当一个对象的类型在编译的时候还不知道，多态就特别有用。

第 12 章介绍了枚举类型，这是从 Java 5 之后增加的一种类型。

第 13 章介绍了如何使用 Java 8 新添加的日期和时间 API，以及在老的 Java 版本中使用的旧 API。

第 14 章介绍了如何使用 java.util 包的成员来组织对象并操作它们。

第 15 章详细地介绍了泛型，它是 Java 中非常重要的一项功能。

第 16 章介绍了流的概念，并且介绍了如何使用 Java IO API 中的 4 种类型的流来执行输入/输出操作。此外，还介绍了对象序列化和反序列化。

第 17 章介绍了注解，讲解了 JDK 所带有的标准注解、通用注解、元注解和定制注解。

第 18 章介绍了如何在另一个类中编写一个类，以及为什么 OOP 功能很有用。

第 19 章介绍了 Java 中的多线程编程，这不只是专家程序员才能使用的技术了。线程是操作系统分配处理器时间的一个基本处理单位，并且在进程中也可以有多个线程来执行代码。

第 20 章是讨论多线程编程的另外一章，介绍了使得编写多线程程序更加容易的接口和类。

第 21 章介绍了 Java 程序员所能够使用的技术。如今，能够在不同的国家和地区部署的软件应用程序已经很常见了。这样的应用程序需要在设计的时候就牢记国际化。

第 22 章介绍了在网络编程中使用的类。给出了一个示例的 Web 服务器应用程序，以说明如何使用这些类。

第 2 部分 Android

第 23 章介绍了 Android 框架。

第 24 章包含了下载和安装开发 App 的工具的说明。

第 25 章介绍了活动及其生命周期。活动是 Android 编程中的最重要的概念之一。

第 26 章介绍了最为重要的 UI 组件，包括微件、Toast 和 AlertDialog。

第 27 章介绍了如何在 Android 应用程序中布局 UI 组件，以及使用 Android 中可用的内建布局。

第 28 章介绍了如何创建一个监听器以处理事件。

第 29 章介绍了如何向操作栏添加项，以及如何使用它驱动应用程序导航。

第 30 章详细介绍了 Android 菜单。菜单是很多图形化用户界面（GUI）系统中常见的功能，其主要角色是提供某些操作的快捷方式。

第 31 章介绍了 ListView，它会显示可以滚动的列表项并且从一个 ListAdapter 获取其数据源的一个视图。

第 32 章介绍了 GridView 微件，这是和 ListView 类似的一个视图。和 ListView 不同的是，GridView 在栅格中显示其项。

第 33 章介绍了两个重要的主题，它们直接关系到 App 的视觉体验。

第 34 章教你如何操作位图图像。即便你不能编写一个图像编辑器应用程序，本章所介绍的技术也很有用。

第 35 章介绍了如何创建一个定制视图以及在画布上绘制形状。Android SDK 带有很广泛的视图，可以在应用程序中使用它们。如果这些都不符合你的需要，你可以创建一个定制视图并且在其上绘制。

第 36 章介绍了片段，这是可以添加到活动中的组件。片段有自己的生命周期，当其进入生命周期的某个阶段的时候，会调用的相应方法。

第 37 章介绍了如何针对不同的屏幕大小使用不同的布局，例如，手机和平板电脑。

第 38 章介绍了 Android 中最新的动画 API 属性动画，并给出了示例。

第 39 章介绍了如何使用 Preference API 来存储应用程序设置并将其读回。

第 40 章介绍了如何使用 Android 应用程序中的 Java File API。

第 41 章介绍了 Android Database API，可以使用它来连接 SQLite 数据库。SQLite 是每一个 Android 设备所附带的默认的关系数据库。

第 42 章介绍了如何使用内建的 Camera 和 Camera API 来获取静态的图像。

第 43 章介绍了两种方法来为应用程序提供拍摄视频的功能，分别是使用内建的意图和使用 MediaRecorder 类。

第 44 章介绍了如何记录音频。

第 45 章介绍了 Handler 类，可以使用它来调度将来要执行的一个 Runnable。

第 46 章介绍了如何在 Android 中处理异步任务。

第 47 章介绍了如何创建后台服务，即便当启动它们的应用程序已经结束了，它们还会运行。

第 48 章介绍了用于接收广播的另一种 Android 组件。

第 49 章介绍了如何使用 AlarmManager 来调度任务。

第 50 章介绍了另一个应用程序组件类型，它用来封装数据并且跨应用程序共享。

附录

附录 A、附录 B 和附录 C 分别介绍了 javac、java 和 jar 工具。

附录 D 和附录 E 分别给出了 NetBeans 和 Eclipse 的简短教程。

下载程序示例

可以通过如下的链接，从出版商的站点下载本书的配套程序示例。

`http://books.brainysoftware.com`

可以下载为单个的 ZIP 文件，或者将其作为一个 Git 项目导入。

此为试读，需要完整PDF请访问：www.ertongbook.com

作者简介

Budi 编写计算机编程图书有 15 年的经验，以清晰的写作风格著称。他编写的一本 Java 教程，最近被德国斯图加特传媒学院的一组计算机科学教授选作该大学的主教材，他们是在把 Budi 的书与其他类似图书进行比较后做此决定的。

Budi 有 20 年担任软件架构师和开发者的经历，这为他的写作提供了支撑。他为世界各地的很多企业提供咨询服务，包括芬兰的手机厂商、英国的投资银行以及美国和加拿大的创业企业。

Budi 编写过诸如基于 Web 的文档管理软件 CreateData，这是一款商业软件，目前在撰写 Java 虚拟机的一篇研究文章。他编写的图书还包括《How Tomcat Works》《Servlet & JSP: A Tutorial and Struts 2 Design and Programming》等。

目录

此为试读，需要完整PDF请访问：www.ertongbook.com