Java

面向对象编程

（第2版）

Helping readers to master OOP(Object-Oriented Programming) with Java and get deep understanding of how the language works

孙卫琴 编著

知名IT技术作家孙卫琴老师倾注六年心血升级打造

业界经典畅销10年图书

以行云流水般的语言诠释Java编程艺术

基于最新Java技术理论紧密结合实际

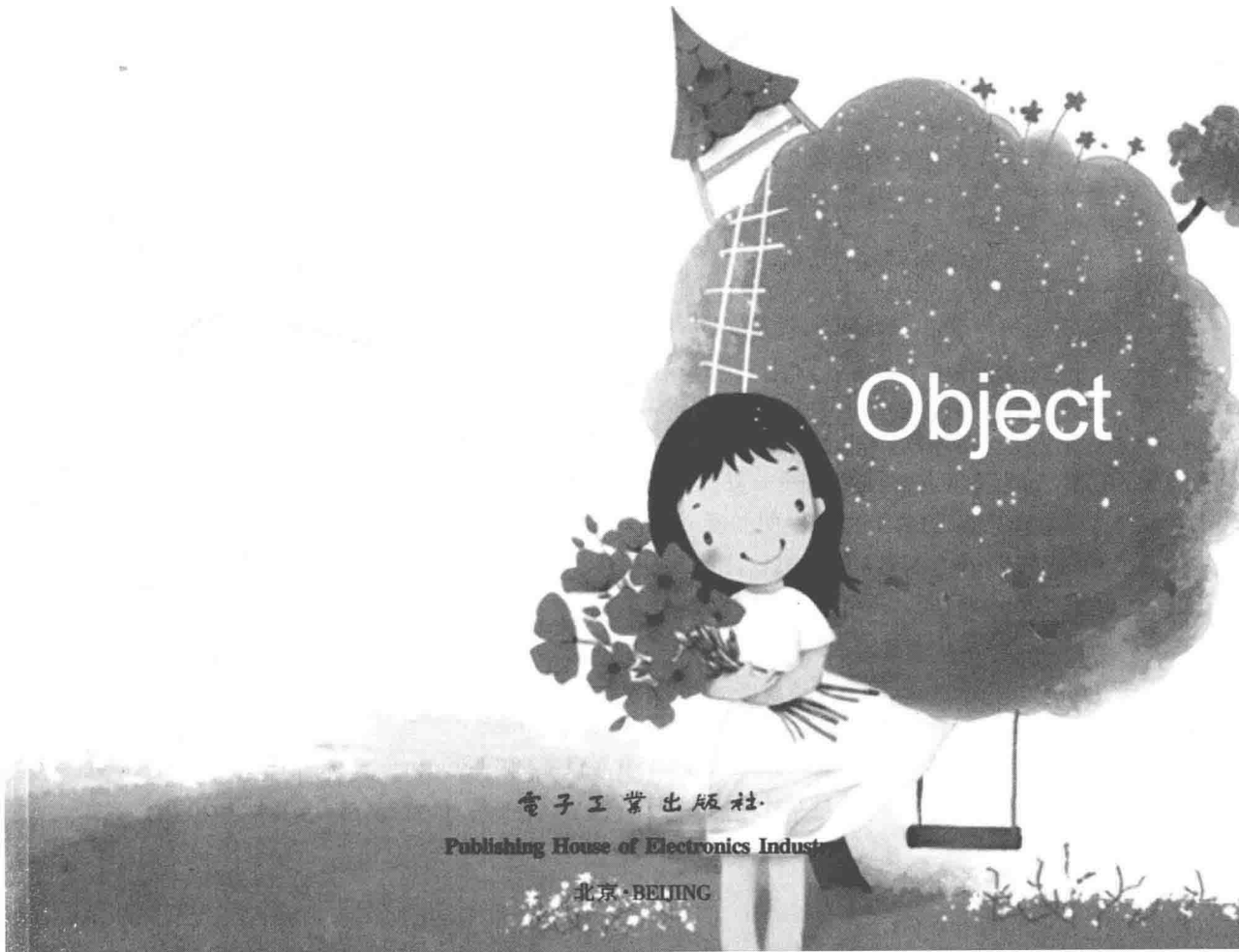
可作为开发指南高校教材和认证辅导教材

中国工信出版集团

電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

Java 面向对象编程（第2版）

孙卫琴 编著



電子工業出版社
Publishing House of Electronics Industry
北京·BEIJING

内 容 简 介

本书采用由浅入深、与实际应用紧密结合的方式，利用大量经典的实例，详细讲解 Java 面向对象的编程思想、编程语法和设计模式，介绍常见 Java 类库的用法，总结优化 Java 编程的各种宝贵经验，深入阐述 Java 虚拟机执行 Java 程序的原理。本书的实例均基于最新的 JDK8 版本。本书内容包括：面向对象开发方法概述、第一个 Java 应用、数据类型和变量、操作符、流程控制、继承、Java 语言中的修饰符、接口、异常处理、类的生命周期、对象的生命周期、内部类、多线程、数组、Java 集合、泛型、Lambda 表达式、输入与输出（I/O）、图形用户界面、常用 Swing 组件、Java 常用类和 Annotation 注解。本书的最大特色是以 6 条主线贯穿生书：面向对象编程思想、Java 语言的语法、Java 虚拟机执行 Java 程序的原理、在实际项目中的运用、设计模式和性能优化技巧。另外，本书还贯穿了 Oracle 公司的 OCJP（Oracle Certified Java Programmer）认证的考试要点。

书中实例源文件及思考题答案的下载网址为：www.javathinker.net/download.htm。

本书适用于所有 Java 编程人员，包括 Java 初学者及资深 Java 开发人员。本书还可作为高校的 Java 教材，以及企业 Java 培训教材，也可作为 Oracle 公司的 OCJP 认证的辅导教材。

图书在版编目（CIP）数据

Java 面向对象编程 / 孙卫琴编著. -- 2 版. -- 北京: 电子工业出版社, 2017.1

ISBN 978-7-121-30314-2

Ⅰ. ①J… Ⅱ. ①孙… Ⅲ. ①JAVA 语言－程序设计Ⅳ. ①TP312.8

中国版本图书馆 CIP 数据核字(2016)第 270355 号

责任编辑：姜　伟
特约编辑：刘红涛
印　　刷：三河市华成印务有限公司
装　　订：三河市华成印务有限公司
出版发行：电子工业出版社
　　北京海淀区万寿路 173 信箱　邮编：100036
经　　销：各地新华书店
开　　本：787×1092　1/16　印张：47　字数：1203.2 千字
版　　次：2017 年 1 月第 1 版
印　　次：2017 年 1 月第 1 次印刷
定　　价：89.00 元

参与本书编写的人员有：孙卫琴、张雷、许亮思、张宇客、孟祥、曹文伟、李红军、王坤、曹雅洁、李洪成。

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：（010）88254888，88258888。

质量投诉请发邮件至 zlts@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

本书咨询联系方式：（010）88254161～88254167 转 1897。

Java 自 1996 年正式发布以来，经历了初生、成长和壮大阶段，现在已经成为 IT 领域里的主流编程语言。Java 起源于 Sun 公司的一个名为“Green”的项目，目的是开发嵌入家用电器的分布式软件系统，使电器更加智能化。图 P-1 为参与 Green 项目的开发人员。Green 项目一开始准备采用 C++语言，但是考虑到 C++语言太复杂，而且安全性差，于是决定基于 C++语言开发一种新的 Oak 语言（即 Java 的前身）。

图 P-1　参与 Green 项目的开发人员

Oak 是一种适用于网络编程的精巧而安全的语言，它保留了许多 C++语言的语法，但去除了明确的资源引用、指针算法与操作符重载等潜在的危险特性。并且 Oak 语言具有与硬件无关的特性，制造商只需更改芯片，就可以将烤面包机上的程序代码移植到微波炉或其他电器上，而不必改变软件，这就大大降低了开发成本。

当 Oak 语言成熟时，Internet 也在全球迅速发展。Sun 公司的开发小组认识到 Oak 非常适合 Internet 编程。1994 年，他们完成了一个用 Oak 语言编写的早期的 Web 浏览器，称为 WebRunner，后改名为 HotJava，展示了 Oak 作为 Internet 开发工具的能力。

1995 年，Oak 语言更名为 Java 语言（以下简称为 Java）。Java 的取名有一个趣闻：据说有一天，几位 Java 成员组的会员正在讨论给这个新的语言取什么名字，当时他们正在咖啡馆喝着 Java（爪哇）咖啡。有一个人灵机一动，说就叫 Java，并得到了其他人的赞赏。于是，Java 这个名字就这样传开了。

1996 年，Sun 公司发布 JDK1.0，计算机产业的各大公司（包括 IBM、Apple、DEC、Adobe、Silicon Graphics、HP、Oracel、Toshiba 和 Microsoft 等）相继从 Sun 公司购买了 Java 技术许可证，开发相应的产品。

1998 年，Sun 公司发布了 JDK1.2（从这个版本开始的 Java 技术都称为 Java2）。Java2 不仅兼容于智能卡和小型消费类设备，还兼容于大型服务器系统，它使软件开发商、服务提供商和设备制造商更加容易抢占市场。这一开发工具极大地简化了编程人员编制企业级 Web 应用的工作，把“一次编程，到处使用”的诺言应用到服务器领域。

1999 年，Sun 公司把 Java2 技术分成 J2SE、J2EE 和 J2ME。其中 J2SE 就是指从 1.2 版本开始的 JDK，它为创建和运行 Java 程序提供了最基本的环境。J2EE 和 J2ME 建立在 J2SE 的基础上，J2EE 为分布式的企业应用提供开发和运行环境，J2ME 为嵌入式应用（比如运行在手机里的 Java 程序）提供开发和运行环境。

Java 的公用规范（Publicly Available Specification，PAS）在 1997 年被国际标准化组织（ISO）认定，这是 ISO 第一次破例接受一个具有商业色彩的公司作为公用规范 PAS 的提交者。

2000 年和 2002 年，Sun 公司分别发布了 JDK1.3 和 JDK1.4，它们在原先版本的基础上做了一些改进，扩展了标准雷库，提高了系统性能，还修正了一些 Bug。

2004 年，Sun 公司发布了 JDK5.0，这一版本做了重大的改进，增加了泛型机制、自动装箱/拆箱和 foreach 语句等。这一版本原称为 JDK1.5，在 2004 年的 SunOne 会议后，更名为 JDK5.0。

2006 年，Sun 公司发布了 JDK6，该版本没有在语言方面进行大的改进，主要是改进了运行性能，并增强了类库功能。

随着以操纵海量数据为主的软件系统越来越依赖商业硬件，而不是专用服务器，以 Solaris 操作系统为主打产品的 Sun 公司变得越来越不景气，最终于 2009 年被 Oracle 公司收购。Java 的发展停滞了很长一段时间。直到 2011 年，Oracle 公司发布了 JDK7，做了一些简单的改进。

2014 年，Oracle 公司发布了 JDK8，这一版本也做出了重大的改进，其中比较显著的改进包括：

- 引入能够简化编程的 Lambda 表达式。
- 接口中允许包含提供了实现的默认方法和静态方法。
- 引入功能强大的 Stream API。
- 引入方便实用的处理日期和时间的新的 Date/Time API。
- 引入避免空指针异常的 Optional 类。

总之，面向对象的 Java 语言具备“一次编程，任何地方均可运行”的能力，使其成为服务提供商和系统集成商用以支持多种操作系统和硬件平台的首选解决方案。Java 作为软件开发的一种革命性的技术，其地位已被确定。如今，Java 技术已被列为当今世界信息技术的主流之一。表 P-1 对 Java 的发展历史做了总结。

表 P-1　Java 发展历史

年份	Java 发展历史
1995	Java 语言诞生
1996	JDK1.0 发布，10 个最主要的操作系统供应商申明将在其产品中支持 Java 技术
1997	JDK1.1 发布
1998	JDK1.1 下载量超过 200 万次，JDK1.2（称 Java2）发布，JFC/Swing 技术发布， JFC/Swing 被下载了 50 多万次
1999	Java 被分成 J2SE、J2EE 和 J2ME，JSP/Servlet 技术诞生
2000	JDK1.3 发布, JDK1.4 发布
2001	Nokia 公司宣布到 2003 年将出售 1 亿部支持 Java 的手机，J2EE1.3 发布
2002	JDK1.4 发布，自此 Java 的计算能力有了大幅度提升。J2EE SDK 的下载量达到 200 万次
2003	5.5 亿台桌面机上运行 Java 程序，75%的开发人员将 Java 作为首要开发工具
2004	JDK1.5 发布，这是 Java 语言发展史上的又一里程碑事件。为了表示这个版本的重要性，JDK1.5 更名为 JDK5.0
2005	JavaOne 大会召开，Java 的各种版本被更名，取消其中的数字“2”：J2EE 更名为 Java EE, J2SE 更名为 Java SE，J2ME 更名为 Java ME
2006	JDK6 发布
2009	Sun 公司被 Oracle 公司收购
2011	JDK7 发布
2014	JDK8 发布，对 Java 语言特性做了重大改进，增加了 Lambda 表达式

Java 语言的特点

Java 应用如此广泛是因为 Java 具有多方面的优势。其特点如下：

（1）面向对象。Java 自诞生之时就被设计成面向对象的语言，而 C++语言是一种强制面向对象的语言。面向对象可以说是 Java 最重要的特性，它不支持类似 C 语言那样的面向过程的程序设计技术。Java 支持静态和动态风格的代码重用。

（2）跨平台。对于 Java 程序，不管是 Windows 平台还是 UNIX 平台或是其他平台，它都适用。Java 编辑器把 Java 源程序编译成与体系结构无关的字节码指令，只要安装了 Java 运行系统，Java 程序就可在任意的处理器上运行。这些字节码指令由 Java 虚拟机来执行，Java 虚拟机的解释器得到字节码后，对它进行转换，使之能够在不同的平台运行。

（3）直接支持分布式的网络应用。除了支持基本的语言功能，Java 核心类库还包括一个支持 HTTP、SMTP 和 FTP 等基于 TCP/IP 协议的类库。因此，Java 应用程序可凭借 URL 打开并访问网络上的对象，其访问方式与访问本地文件系统几乎完全相同。在 Java 出现以前，为分布式环境尤其是 Internet 提供动态的内容无疑是一项非常宏伟、难以想象的任务，但 Java 的语言特性却使我们很容易地达到了这个目标。

（4）安全性和健壮性。Java 致力于检查程序在编译和运行时的错误，类型检查帮助检查出许多开发早期出现的错误。Java 支持自动内存管理，这不但让程序员减轻了许多负担，也减少了程序员犯错的机会。Java 自己操纵内存减少了内存出错的可能性。Java 还能够检测数组边界，避免了覆盖数据的可能。在 Java 语言里，指针和释放内存等功能均被抛弃，从而避免了非法内存操作的危险。

以上特点，是 C++语言及其他语言无法比拟的（C++语言尽管也是面向对象的，但并不是严格意义上的面向对象的语言）。单从面向对象的特性来看，Java 类似于 SmallTalk，但其他特性，尤其是适用于分布式计算环境的特性远远超越了 SmallTalk。Java 发展到现在，已经不仅仅是一种语言，可以说是一种技术，这个技术涉及网络和编程等领域。另外，Java 是非常简单、高效的，有调查数据发现：用 C++和 Java 来做一个相同功能的项目，用 Java 写的程序要比用 C++写的程序节省 60%的代码和 66%的时间。可以说，用 Java 语言编程时间短、功能强，编程人员接手起来更容易、更简便。

本书的组织结构和主要内容

本书以 6 条主线贯穿全书：面向对象编程思想、Java 语言的语法、Java 虚拟机执行 Java 程序的原理、在实际项目中的运用、设计模式和性能优化技巧。书的每一章都会围绕若干条主线来展开内容，并且根据全书的布局，合理安排每一章内容的深度。本书主要内容包括：面向对象开发方法概述、第一个 Java 应用、数据类型和变量、操作符、流程控制、继承、Java 语言中的修饰符、接口、异常处理、类的生命周期、对象的生命周期、内部类、多线程、数组、Java 集合、泛型、Lambda 表达式、输入与输出（I/O）、图形用户界面、常用 Swing 组件、Java 常用类和 Annotation 注解。

这本书是否适合您

在如今的 Java 领域，各种新技术、新工具层出不穷，一方面，每一种技术都会不停地升级换代，另一方面，还会不断涌现出新的技术和工具。Java 世界就像小时候玩的万花筒，尽管实质上只是由几个普通的玻璃碎片组成的，但只要轻轻一摇，就会变化出千万种缤纷的图案。Java 世界如此变化多端，很容易让初学 Java 的人有无从下手的感觉。常常会有读者问我这样的问题：

我学了 Java 已经一年多了，现在就只能用 JSP 写点东西，其他的东西实在太多了，我整天学都学不完，很迷茫，不知道该如何有针对性地去学，以找到一份 Java 工作，现在是困死在 Java 里了。

撰写本书，目的之一是为了帮助读者看清 Java 万花筒的本质，从复杂的表象中寻找普遍的规律，深刻理解 Java 的核心思想，只有掌握了普遍的规律与核心思想，才能以不变应万变，轻轻松松地把握 Java 技术发展的新趋势，迅速地领略并且会熟练运用一门新的技术，而不成为被动的追随者，知其然而不知其所以然。

阅读本书，读者对 Java 的领悟将逐步达到以下境界：

- 熟悉 Java 语法，熟练地编译和调试程序。
- 按照面向对象的思想来快速理解 JDK 类库，以及其他第三方提供的类库，通过阅读 JavaDoc 和相关文档，知道如何正确地使用这些类库。
- 按照面向对象的思想来分析问题领域，设计对象模型。
- 在开发过程中会运用现有的一些优秀设计模式，提高开发效率。
- 当一个方法有多种实现方式时，能够从可维护、可重用及性能优化的角度选择最佳的实现方式。
- 理解 Java 虚拟机执行 Java 程序的原理，从而更深入地理解 Java 语言的各种特性和语法规则。

本书的技术支持网站

本书的技术支持网站为：www.javathinker.net。读者可以在该网站交流 Java 技术，提出本书的勘误信息，作者会在本网站为读者答疑。本书中实例源文件及思考题答案的下载网址为：www.javathinker.net/download.htm。

致谢

本书在编写过程中得到了 Oracle 公司在技术上的大力支持，电子工业出版社少儿与艺术分社负责编辑审核。此外，复旦软件学院的戴开宇等老师为本书的编写提供了有益的帮助，JavaThinker.net 网站的网友们为本书的升级提供了许多宝贵建议，在此表示衷心的感谢！尽管我们尽了最大努力，但本书难免会有不妥之处，欢迎各界专家和读者朋友批评指正。

第 1 章　面向对象开发方法概述

一般说来，软件开发都会经历以下生命周期：

- 软件分析：分析问题领域，了解用户的需求。
- 软件设计：确定软件的总体架构，把整个软件系统划分成大大小小的多个子系统，设计每个子系统的具体结构。
- 软件编码：用选定的编程语言来编写程序代码，实现在设计阶段勾画出的软件蓝图。
- 软件测试：测试软件是否能实现特定的功能，以及测试软件的运行性能。
- 软件部署：为用户安装软件系统，帮助用户正确地使用软件。
- 软件维护：修复软件中存在的 Bug，当用户需求发生变化时（增加新的功能，或者修改已有功能的实现方式），修改相应的软件。

为了提高软件开发效率，降低软件开发成本，一个优良的软件系统应该具备以下特点：

- 可重用性：减少软件中的重复代码，避免重复编程。
- 可扩展性：当软件必须增加新的功能时，能够在现有系统结构的基础上，方便地创建新的子系统，不需要改变软件系统现有的结构，也不会影响已经存在的子系统。
- 可维护性：当用户需求发生变化时，只需要修改局部的子系统的少量程序代码，不会牵一发而动全身，修改软件系统中多个子系统的程序代码。

Tips

本章多次使用了系统结构的说法，这里的系统结构是指系统由多个子系统组成，以及子系统由多个更小的子系统组成的结构。

如何才能使软件系统具备以上特点呢？假如能把软件分解成多个小的子系统，每个子系统相对独立，把这些子系统像搭积木一样灵活地组装起来就构成了整个软件系统，这样的软件系统便能获得以上优良特性。软件中的子系统具有以下特点：

- 结构稳定性：软件在设计阶段，在把一个系统划分成更小的子系统时，设计合理，使得系统的结构比较健壮，能够适应用户变化的需求。
- 可扩展性：当软件必须增加新的功能时，可在现有子系统的基础上创建出新的子系统，该子系统继承了原有子系统的一些特性，并且还具有一些新的特性，从而提高软件的可重用性和可扩展性。
- 内聚性：每个子系统只完成特定的功能，不同子系统之间不会有功能的重叠。为了避免子系统之间功能的重叠，每个子系统的粒度在保持功能完整性的前提下都尽可能小，按这种方式构成的系统结构被称为精粒度系统结构。子系统的内聚性会提高软件的可重用性和可维护性。

- 可组合性：若干精粒度的子系统经过组合，就变成了大系统。子系统的可组合性会提高软件的可重用性和可维护性，并且能够简化软件的开发过程。
- 松耦合：子系统之间通过设计良好的接口进行通信，但是尽量保持相互独立，修改一个子系统，不会影响到其他的子系统。当用户需求发生变化时，只需要修改特定子系统的实现方式，从而提高软件的可维护性。

Tips

在精粒度系统结构中，大系统可以分解为多个松耦合的精粒度子系统。与此相反，在粗粒度系统结构中，粗粒度的大系统无法进一步分解，或者只能被分解为有限的几个粗粒度子系统。

图 1-1 显示了一个用积木搭建起来的建筑物系统，有凯旋门，还有小狗狗的家。这些系统中的最小子系统是各种形状的积木（精粒度的子系统），这些积木能够在多个子系统中重用，多个积木能组合成复杂的子系统。当大的系统被拆散或摧毁时，它所包含的小子系统依然有用。

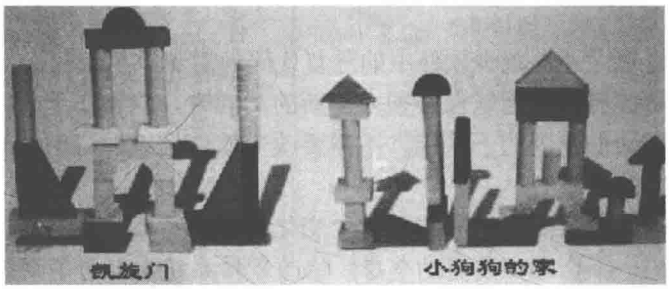


图 1-1　用积木搭建起来的建筑物系统

目前在软件开发领域有两种主流的开发方法：结构化开发和面向对象开发。结构化开发是一种比较传统的开发方法，早期的高级编程语言，如 Basic、C、Fortran 和 Pascal 等，都是支持结构化开发的编程语言。随着软件开发技术的逐步发展，为了近一步提高软件的可重用性、可扩展性和可维护性，面向对象的编程语言及面向对象设计理论应运而生，Java 语言就是一种纯面向对象的编程语言。

本章首先简要介绍结构化的软件开发过程，然后介绍了面向对象的软件开发过程，对面向对象的一些核心思想和概念进行了阐述。本章列举了不少形象的例子，来帮助读者理解面向对象的开发思想，并且以一个画板（Panel）软件系统的例子贯穿整个章节，这个例子分别按照结构化开发方式和面向对象开发方式实现，从而鲜明地对比这两种开发方式对软件的可维护性、可扩展性和可重用性的影响。

本章的思想性和理论性比较强，如果读者已经有一定的面向对象的开发经验，阅读本章时会很顺利。如果读者没有面向对象的开发经验，初次阅读时会感觉比较抽象，在这种情况下，初次阅读时只需了解一些基本概念与核心思想即可，等到阅读完全书后，再来回顾和领悟本章内容。

本章在介绍范例时，展示了部分 Java 程序代码，如果读者对 Java 编程没有任何基