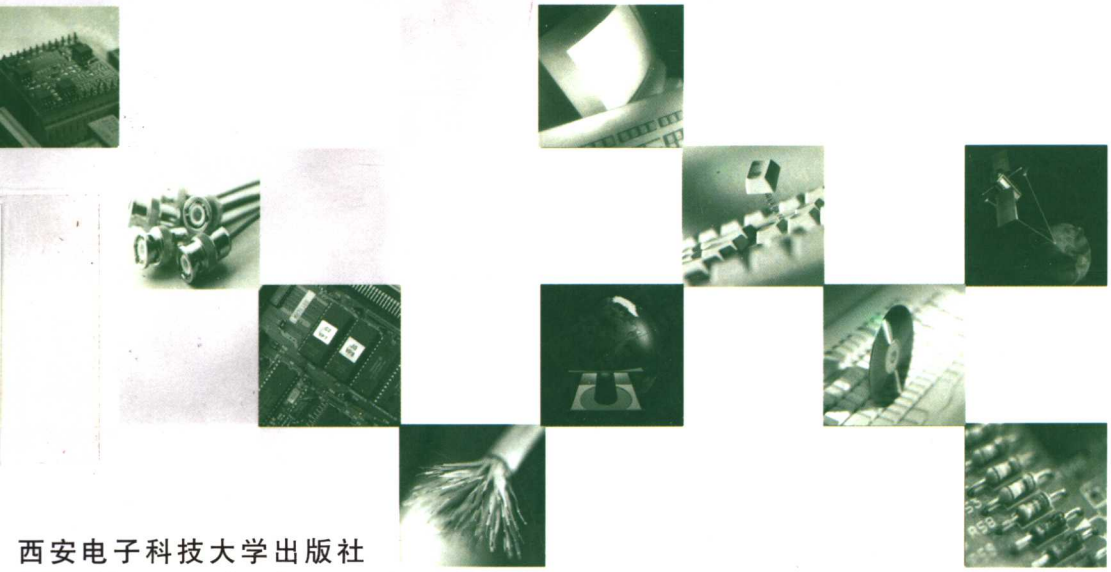
普通高等院校计算机类专业系列教材

Object-Oriented Program Design—Java

面向对象程序设计

——Java (第二版)

张白一 崔尚林 编著



西安电子科技大学出版社
http://www.xduph.com

普通高等院校计算机类专业系列教材

面向对象程序设计
——Java

(第二版)

Object-Oriented Program Design—Java

张白一　崔尚森　编著

西安电子科技大学出版社

2006

内 容 简 介

本书将面向对象的理论与Java语言程序设计技术相结合，意在培养读者正确运用面向对象的思维方法分析问题和解决问题的能力。全书共分16章。前6章主要介绍面向对象的基本理论、原理、技术方法和Java语言基础知识，阐述了面向对象程序设计的基本原则和特点。第7章介绍字符串类。这次修订新增的第8章介绍了在没有指针类型的Java语言中进行链表操作的技术。从第9章开始的以后各章介绍Java的常用标准类库及编程技巧，主要包括GUI设计、Swing组件、异常处理、多线程技术、输入/输出技术、网络编程技术和JDBC数据库应用编程技术等。

本书可作为大专院校相关课程的教材，也可作为对面向对象编程技术和Java语言感兴趣的读者的自学用书。

为方便教学和实践，本书配有光盘一张，其中包括电子教案、示例程序源代码及相关工具软件。

图书在版编目(CIP)数据

面向对象程序设计：Java / 张白一等编著. —2版. 西安：西安电子科技大学出版社，2006.1

(普通高等院校计算机类专业系列教材)

ISBN 7-5606-1605-4

Ⅰ. 面… Ⅱ. 张… Ⅲ. JAVA语言—程序设计—高等学校—教材 Ⅳ. TP312

中国版本图书馆CIP数据核字（2005）第134030号

责任编辑 阎 彬 云立实

出版发行 西安电子科技大学出版社(西安市太白南路2号)

电 话 (029)88242885 88201467 邮 编 710071

http: //www.xduph.com E-mail: xdupfxb@pub.xaonline.com

经 销 新华书店

印刷单位 陕西天意印务有限责任公司

版 次 2003年2月第1版 2006年1月第2版 2006年1月第5次印刷

开 本 787毫米×1092毫米 1/16 印 张 23.5

字 数 548千字

印 数 22 001～26 000册

定 价 32.00元(含光盘)

26.00元

ISBN 7-5606-1605-4/TP・0922

XDUP 1897002-5

普通高等院校计算机类专业系列教材
编审专家委员会名单

第一版前言

面向对象技术引起了程序设计方法学的一场革命，它已经替代面向过程的程序设计技术，成为当今计算机应用开发领域的主流技术。其原因主要在于面向对象技术能够比较客观地模拟现实世界，能够使软件开发人员运用人类认识事物所采用的一般思维方法进行软件开发；其次是在面向对象的程序设计中将数据与操作捆绑在一起，符合现代大规模软件开发的高可靠性和易维护性等方面的要求。计算机网络的发展，要求程序设计语言具有安全性强、可移植性好、与具体的操作平台无关等特性。Java 语言正是为满足这些要求而设计与研发的，并以网络为发展方向，随着网络的发展而兴盛。

1995 年 Java 语言刚一推出，便以其纯面向对象、平台无关性、多线程、高安全性、良好的可移植性和可扩展性等特征，受到了计算机界的普遍欢迎，并得到了广泛的应用和发展。近几年来，Java 的应用已经扩展到各个应用领域，加上各种功能配件的推陈出新，使得 Java 能够满足产品开发的需求，成为网络时代最流行的程序设计语言。利用 Java 来开发软件，具有跨平台、易整合、易扩展的优点。有人预言，不久的将来全世界 90%的程序代码将用 Java 语言书写或改写。

本书是为大专院校的学生及其他对面向对象编程技术和 Java 语言感兴趣的读者编写的，意在培养读者正确使用面向对象的思维方法分析问题和解决问题的能力，让读者学会利用当今最先进的软件开发工具开发软件产品，以适应网络时代社会对人才的需求。根据作者多年来讲授“面向对象程序设计”及其相关课程的经验，本书在内容的取舍上作了精心的选择，确保有一定的深度和广度；在内容的编排上体现了由浅入深、循序渐进的学习规律；在写作风格上立

足于理论与实践相结合，将复杂的面向对象理论融会于众多的实例之中，使读者学会面对一个具体的问题时，能够用面向对象的思维方法分析问题，并利用面向对象的语言编写解决实际问题的计算机程序。本书可作为大专院校相关课程的教材，也可作为对面向对象编程技术和Java语言感兴趣的读者的自学用书。

全书共分15章。第1章介绍Java的特点和运行环境。第2章和第3章讲述程序设计的基本语法规则和程序流程控制。第4章和第5章全面讲述面向对象的理论和程序设计技术，包括类与对象、抽象与封装、消息、继承、多态等诸多概念及其在程序设计中的具体应用。第6章则介绍数组这种在各种程序设计语言中常用的数据结构在Java中的应用技术。从第7章开始介绍Java的常用标准类库及其编程技巧。第7章介绍字符串类。第8～10章讲述Java图形用户界面的设计与编程实现技术，并以最新的Swing GUI组件为主。第11章介绍Java的异常处理。第12章讲述多线程技术。第13章讲述程序的输入与输出技术。第14章和第15章分别介绍Java的网络编程技术和数据库应用编程技术。

要想很好地掌握面向对象技术，学习用Java程序设计语言编写出高质量的程序，先要学习其语法规则，这是编写Java程序的基本功；还要学习使用类库，这是提高编程效率和质量的必由之路，甚至从一定程度上来说，是否能熟练自如地掌握尽可能多的Java类库，决定了一个Java程序员编程能力的高低。此外，计算机学科是注重实践的学科，优秀的软件人员无不经过大量的上机实践的磨练，因此，读者只有在学习书本内容的同时辅以相应的实际练习，才能真正掌握书中介绍的知识和技能。

本书第1～7章由张白一编写，第8～15章由崔尚森编写。在本书的编写过程中，参考了许多相关书籍和网站，得到了许多同仁和同事的支持与帮助，在此我们一并表示感谢。首先应该感谢支持和参与本课程教学的长安大学的同学们，正是他们活跃的思维和永无止境的求知欲帮助作者发现内部试用版的错误，

鞭策作者不断改进与完善书稿。同时，特别感谢赵文静教授在百忙之中仔细审阅了本书全稿，并提出了许多宝贵的修改意见，使本书更趋完善。此外，还要衷心感谢陕西省计算机教育学会、西安电子科技大学出版社以及长安大学的领导和同事们，正是由于他们的大力支持，才使得本书与广大读者见面。

尽管书稿几经修改，但由于作者水平所限，书中难免有疏漏之处，我们热诚地欢迎各位同行和广大读者批评指正。

作　者

2002 年 8 月

第二版前言

Java 语言推出至今已有 20 个年头了，在这 20 年中，尤其是近几年来，网络编程技术的飞速发展使得 Java 语言受到了广泛的欢迎并得到迅速的发展，许多网络软件开发人员将 Java 语言作为首选的开发工具，国内许多高校也相继开设了 Java 编程方面的课程。承蒙读者厚爱，我们于 2002 年编写并出版的《面向对象程序设计——Java》一书已多次印刷，许多读者借助该书学会了 Java 编程，同时也提出了一些宝贵的意见和建议。此外，随着 Java 版本的升级换代，书中的一些内容也需要更新。为适应这种变化，我们对该书进行了修改和补充。

与第一版相比，首先，我们在开发环境上使用了运行于 Windows XP 操作系统上的最新版本 jdk-1_5_0_04 对各个章节进行了修改，对书中所附算法和示例程序全部进行了重新调试和改写，增加了注释与图解等。其次，根据 Java 2D API 的特性，对与 GUI 设计有关的章节进行了彻底的更新。第三，为了使读者对面向对象技术有一个更深入的理解，我们在本书中新增了“链表”一章，介绍了链接存储结构的概念和特点以及在没有指针类型的 Java 语言中进行链表操作的技术。

全书共分 16 章。第 1 章介绍 Java 的特点和运行环境。第 2 章和第 3 章讲述程序设计的基本语法规则和程序流程控制。第 4 章和第 5 章全面讲述面向对象的理论和程序设计技术，包括类与对象、抽象与封装、消息、继承、多态等诸多概念及其在程序设计中的具体应用。第 6 章和第 8 章介绍数组和链表这两种常用的数据结构在 Java 中的应用技术。第 7 章介绍字符串类。从第 9 章开始的以后各章介绍 Java 的常用标准类库及其编程技巧。其中，第 9～11 章以最新的 Java 2D API 和 Swing GUI 组件为主讲述 Java 图形用户界面的设计与编程实现技

术；第 12 章介绍 Java 的异常处理；第 13 章讲述多线程技术；第 14 章讲述程序的输入与输出技术；第 15 章和第 16 章分别介绍 Java 的网络编程技术和数据库应用编程技术。

本书在第一版的基础上，由张白一完成了原第 1～7 章的改写和完善工作，由崔尚森完成了原第 8～15 章(本版为第 9～16 章)的改写和完善工作，新增的第 8 章由张向一编写。

虽然这已是该书的第二版，但因为 Java 开发环境的不断发展及作者水平有限，书中仍难免有疏漏之处，我们热诚地欢迎各位同行和广大读者批评指正。对于本书的各种意见和建议可直接发送 E-mail 到 byzhang@chd.edu.cn。

作　者

2005 年 10 月

目　录

第1章 Java 系统环境概述

Java 语言是美国加州 Sun Microsystem 公司于 1995 年正式推出的纯面向对象(object-oriented，简称 OO)的程序设计语言。由于它很好地解决了网络编程语言中的诸多问题，因此一经推出，便受到了计算机界的普遍欢迎和接受，并得到了广泛的应用和发展，成为目前网络时代最为流行的程序设计语言。

面向对象的编程语言使程序能够比较直观地反映客观世界的本来面目，并且使软件开发人员能够运用人类认识事物所采用的一般思维方法进行软件开发，是当今计算机领域中软件开发和应用的主流技术。所有面向对象的程序设计语言都支持对象、类、消息、封装、继承、多态等诸多概念，而这些概念是人们在进行软件开发、程序设计的过程中逐渐提出来的。因此，要弄清面向对象及其相关概念，就得先从程序设计语言的发展谈起。

1.1 编程语言的发展

自从 1946 年第一台电子计算机问世以来，人们一直在探索着自然语言与计算机语言之间的映射问题。我们知道，人类的任何思维活动都是借助于人们所熟悉的某种自然语言进行的。若希望借助计算机完成人类的一种思维活动，就需要把用自然语言表达的东西转换成计算机能够理解和执行的形式语言，这便是编程语言或程序设计语言。毫无疑问，电子计算机毕竟是一种机器，它能够理解和执行的编程语言和自然语言之间存在着较大的差距，这种差距被人们称作“语言的鸿沟”。这一鸿沟虽不可彻底消除，但可以使其逐渐变窄。事实上，从计算机问世至今，各种编程语言的发展变迁，其目的就是为了缩小这一鸿沟。图 1.1 引自参考文献[3]，笔者稍作修改，该图展示了从机器语言发展到面向对象的语言使“语言的鸿沟”变窄的情形。

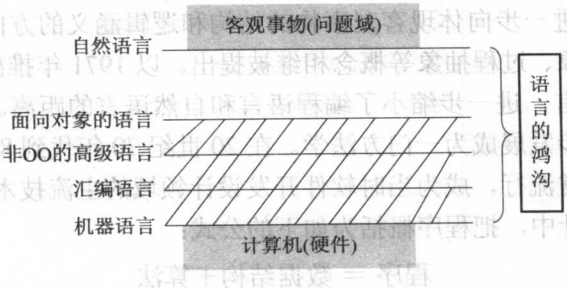


图 1.1　编程语言的发展使“语言的鸿沟”变窄

1.1.1 机器语言

电子计算机是一种机器，这种机器主要由电子元器件构成。对于电子元器件来说，最容易表达的是电流的通/断，或电位的高/低两种状态。因此，在电子计算机问世之初，人们首先想到的是用“0”和“1”两种符号来代表电路的通和断两种状态，这便是最早的编程语言——机器语言。

机器语言是计算机能够理解并直接执行的惟一语言。整个语言只包含“0”和“1”两种符号。用机器语言编写的程序，无论是它的指令、数据还是其存储地址，都是由二进制的“0”和“1”组成的。这种语言离计算机最近，机器能够直接执行它。然而，由“0”和“1”组成的二进制串没有丝毫的形象意义，因此，它离人类的思维最远，“语言的鸿沟”最宽。所以，用机器语言编写程序的效率最低，并且在编写程序时很容易发生错误。

1.1.2 汇编语言

为了克服机器语言的缺陷，人们设想用一些易于理解和记忆的符号来代替二进制码，这便是汇编语言。由于汇编语言用符号构成程序，而这些符号表示指令、数据、寄存器、地址等物理概念，因而，使用汇编语言编程在适合人类形象思维的道路上前进了一步。但是，使用汇编语言编写程序时，编程人员依然需要考虑寄存器等大量的机器细节，即汇编语言仍然是一种与具体机器硬件有关的语言，是一种面向机器的语言，因此，人们也把它称为符号化的机器语言。

1.1.3 高级语言

由于机器语言和汇编语言都离不开具体的机器指令系统，用它们编程时要求程序员必须熟悉所用计算机的硬件特性，因而，用它们编写程序的技术复杂，效率不高，且可维护性和可移植性都很差。为了从根本上摆脱语言对机器的依附，人们经过多年的潜心研究，终于在 1956 年推出了一种与具体机器指令系统无关、表达方式接近自然语言的计算机语言——FORTRAN 语言。在 FORTRAN 语言程序中，采用了具有一定涵义的数据命名和人们容易理解的执行语句，屏蔽了机器细节，使得人们在书写和阅读程序时可以联系到程序所描述的具体事物。所以，人们把这种“与具体机器指令系统无关，表达方式接近自然语言”的计算机语言称为高级语言。高级语言的出现是编程语言发展史上的一大进步，它缩小了编程语言与自然语言之间的鸿沟。

此后，高级语言进一步向体现客观事物的结构和逻辑涵义的方向发展。结构化数据、结构化语句、数据抽象、过程抽象等概念相继被提出。以 1971 年推出的 Pascal 为典型代表的结构化程序设计语言，进一步缩小了编程语言和自然语言的距离。在此后的十几年中，结构化程序设计进一步发展成为一门方法学。在 20 世纪 70 年代到 80 年代，各种结构化程序设计语言及方法非常流行，成为当时软件开发设计领域的主流技术。

在结构化程序设计中，把程序概括为如下的公式：

$$\text{程序} = \text{数据结构} + \text{算法}$$

其中，数据结构是指利用计算机的离散逻辑来量化表达需要解决的问题，而算法则研究如

何高效而快捷地组织解决问题的具体过程。可见，以结构化程序设计为代表的高级语言是一种面向数据/过程的程序设计语言，人们把这类语言也称为面向过程的语言。

面向过程的语言可以精确地用计算机所理解的逻辑来描述和表达待解问题的具体解决过程。然而，它把数据和过程分离为相互独立的实体，使程序中的数据和操作不能有效地组织成与问题域中的具体事物相对应的程序成分，所以它很难把一个具有多种相互关系的复杂事物表述清楚。程序员在编写算法时，必须时刻考虑所要处理问题的数据结构，如果数据结构发生了轻微的变化，那么对处理这些数据的算法也要做出相应的修改，甚至完全重写，否则这个算法就不可再用。因而，用这种程序设计方法编写的软件，其重用性较差。为了较好地解决软件的重用性问题，使数据与程序始终保持相容，人们又提出了面向对象的程序设计方法。

1.1.4　面向对象的语言

面向对象的编程语言(Object-Oriented Programming Language，OOPL)的设计出发点是为了能更直接地描述问题域中客观存在的事物(即对象)以及它们之间的关系。面向对象技术追求的是软件系统对现实世界的直接模拟，是将现实世界中的事物直接映射到软件系统的解空间。它希望用户最大程度地利用软件系统，花费少量的编程时间来解决需要解决的问题。

在面向对象的程序设计语言中，可以把程序描述为如下的公式：

程序＝对象＋消息

面向对象的语言对现实世界的直接模拟体现在下面几个方面：

(1) 对象(object)。只要我们仔细研究程序设计所面对的问题域——客观世界，就可以看到：客观世界是由一些具体的事物构成的，每个事物都具有自己的一组静态特征(属性)和一组动态特征(行为)。例如，一辆汽车有颜色、型号、马力、生产厂家等静态特征，又具有行驶、转弯、停车等动态特征。要把客观世界的这一事实映射到面向对象的程序设计语言中，则需把问题域中的事物抽象成对象，用一组数据描述该对象的静态特征(即属性，在 Java 中称之为数据成员)，用一组方法来刻画该对象的动态特征(即行为)。

(2) 类(class)。客观世界中的事物既具有特殊性又具有共同性。人类认识客观世界的基本方法之一就是对事物进行分类，即根据事物的共同性把事物归结为某些类。考虑一下所有的汽车和一辆汽车之间的关系就很容易理解这一点。OOPL 很自然地用类(class)来表示一组具有相同属性和方法的对象。

(3) 继承(inheritance)。在同一类事物中，每个事物既具有同类的共同性，又具有自己的特殊性。OOPL 用父类与子类的概念来描述这一事实。在父类中描述事物的共性，通过父类派生(derive)子类的机制来体现事物的个性。考虑同类事物中每个事物的特殊性时，可由这个父类派生子类，子类可以继承父类的共同性，又具有自己的特殊性。

(4) 封装(encapsulation)。客观世界中的事物是一个独立的整体，它的许多内部实现细节是外部所不关心的。例如，对于一个只管开车的驾驶员来说，他可能根本不知道他所驾驶的这辆汽车内部用了多少根螺钉或几米导线，以及它们是怎样组装的。OOPL 用封装机制把对象的属性和方法结合为一个整体，并且屏蔽了对象的内部细节。

(5) 关联(association)。客观世界中的一个事物可能与其他事物之间存在某种行为上的联系。例如，一辆行驶中的汽车遇到红色信号灯时要刹车停止，OOPL 便通过消息连接来表示