

Classes and Methods

References

Encapsulation



Addison-Wesley's

CODEMATE

Interfaces

UML

Javadoc

Inheritance

# Absolute Java

## 中文版

### Absolute Java

Exception Handling

Iterators

Stream and File I/O

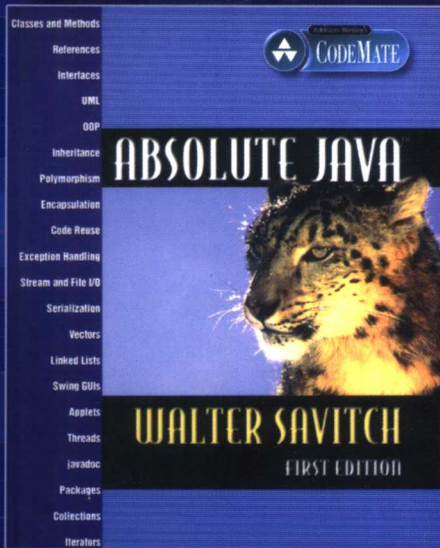
Serialization

Packages

Applets

Swing GUIs

Linked Lists



[美] Walter Savitch 著

陈明 等译



电子工业出版社

Publishing House of Electronics Industry

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

国外计算机科学教材系列

# Absolute Java

Absolute Java

[美] Walter Savitch 著

陈明等译

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

## 内 容 简 介

本书是一本讲授 Java 程序设计的优秀教材和参考书。在内容组织方面, 主要围绕 Java 语言自身的特点展开, 涵盖了 Java 语言的各个方面。书中探讨了多种编程技巧, 系统全面地介绍了 Java 语言, 对继承、多态性及异常处理的阐述尤为详尽。此外, 还探讨了 Java 集和 Java 标准库, 并用独立的章节论述了递归及链表数据结构。

本书的结构合理, 灵活性强, 每章都附有习题及答案, 易于学习、理解和使用。本书开篇的论述简单, 而后逐渐深入, 层层铺开。因此, 即适合于初学者, 也适合于有经验的编程人员。

Simplified Chinese edition Copyright © 2005 by PEARSON EDUCATION ASIA LIMITED and Publishing House of Electronics Industry.

Absolute Java, ISBN: 0321205677 by Walter Savitch. Copyright © 2004. All Rights Reserved.

Published by arrangement with the original publisher, Pearson Education, Inc., publishing as Addison-Wesley.

This edition is authorized for sale only in the People's Republic of China (excluding the Special Administrative Region of Hong Kong and Macau).

本书中文简体字翻译版由电子工业出版社和 Pearson Education 培生教育出版亚洲有限公司合作出版。未经出版者预先书面许可, 不得以任何方式复制或抄袭本书的任何部分。

本书封面贴有 Pearson Education 培生教育出版集团激光防伪标签, 无标签者不得销售。

版权贸易合同登记号 图字: 01-2004-2233

### 图书在版编目 ( CIP ) 数据

Absolute Java/ (美) 赛维斯 (Savitch, W.) 著; 陈明等译. -北京: 电子工业出版社, 2005.6  
(国外计算机科学教材系列)

书名原文: Absolute Java

ISBN 7-121-01304-5

I. A... II. ①赛... ②陈... III. JAVA 语言 - 程序设计 - 教材 IV. TP312

中国版本图书馆 CIP 数据核字 (2005) 第 051658 号

责任编辑: 史 平

印 刷: 北京智力达印刷有限公司

出版发行: 电子工业出版社

北京市海淀区万寿路 173 信箱 邮编: 100036

经 销: 各地新华书店

开 本: 787 × 1092 1/16 印张: 45 字数: 1267 千字

印 次: 2005 年 6 月第 1 次印刷

定 价: 69.00 元 (附光盘 1 张)

凡购买电子工业出版社的图书, 如有缺损问题, 请向购买书店调换; 若书店售缺, 请与本社发行部联系。联系电话: (010) 68279077。质量投诉请发邮件至 zllts@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

## 出版说明

21世纪初的5至10年是我国国民经济和社会发展的关键时期,也是信息产业快速发展的关键时期。在我国加入WTO后的今天,培养一支适应国际化竞争的一流IT人才队伍是我国高等教育的重要任务之一。信息科学和技术方面人才的优劣与多寡,是我国面对国际竞争时成败的关键因素。

当前,正值我国高等教育特别是信息科学领域的教育调整、变革的重大时期,为使我国教育体制与国际化接轨,有条件的高等院校正在为某些信息学科和技术课程使用国外优秀教材和优秀原版教材,以使我国在计算机教学上尽快赶上国际先进水平。

电子工业出版社秉承多年来引进国外优秀图书的经验,翻译出版了“国外计算机科学教材系列”丛书,这套教材覆盖学科范围广、领域宽、层次多,既有本科专业课程教材,也有研究生课程教材,以适应不同院系、不同专业、不同层次的师生对教材的需求,广大师生可自由选择和自由组合使用。这些教材涉及的学科方向包括网络与通信、操作系统、计算机组织与结构、算法与数据结构、数据库与信息处理、编程语言、图形图像与多媒体、软件工程等。同时,我们也适当引进了一些优秀英文原版教材,本着翻译版本和英文原版并重的原则,对重点图书既提供英文原版又提供相应的翻译版本。

在图书选题上,我们大都选择国外著名出版公司出版的高校教材,如Pearson Education培生教育出版集团、麦格劳-希尔教育出版集团、麻省理工学院出版社、剑桥大学出版社等。撰写教材的许多作者都是蜚声世界的教授、学者,如道格拉斯·科默(Douglas E. Comer)、威廉·斯托林斯(William Stallings)、哈维·戴特尔(Harvey M. Deitel)、尤利斯·布莱克(Uyless Black)等。

为确保教材的选题质量和翻译质量,我们约请了清华大学、北京大学、北京航空航天大学、复旦大学、上海交通大学、南京大学、浙江大学、哈尔滨工业大学、华中科技大学、西安交通大学、国防科学技术大学、解放军理工大学等著名高校的教授和骨干教师参与了本系列教材的选题、翻译和审校工作。他们中既有讲授同类教材的骨干教师、博士,也有积累了几十年教学经验的老教授和博士生导师。

在该系列教材的选题、翻译和编辑加工过程中,为提高教材质量,我们做了大量细致的工作,包括对所选教材进行全面论证;选择编辑时力求达到专业对口;对排版、印制质量进行严格把关。对于英文教材中出现的错误,我们通过与作者联络和网上下载勘误表等方式,逐一进行了修订。

此外,我们还将与国外著名出版公司合作,提供一些教材的教学支持资料,希望能为授课老师提供帮助。今后,我们将继续加强与各高校教师的密切联系,为广大师生引进更多的国外优秀教材和参考书,为我国计算机科学教学体系与国际教学体系的接轨做出努力。

电子工业出版社

## 教材出版委员会

- |    |     |   |
|----|-----|---|
| 主任 | 杨芙清 | 北京大学教授<br>中国科学院院士<br>北京大学信息与工程学部主任<br>北京大学软件工程研究所所长 |
| 委员 | 王 珊 | 中国人民大学信息学院院长、教授                                     |
|    | 胡道元 | 清华大学计算机科学与技术系教授<br>国际信息处理联合会通信系统中国代表                |
|    | 钟玉琢 | 清华大学计算机科学与技术系教授<br>中国计算机学会多媒体专业委员会主任                |
|    | 谢希仁 | 中国人民解放军理工大学教授<br>全军网络技术研究中心主任、博士生导师                 |
|    | 尤晋元 | 上海交通大学计算机科学与工程系教授<br>上海分布计算技术中心主任                   |
|    | 施伯乐 | 上海国际数据库研究中心主任、复旦大学教授<br>中国计算机学会常务理事、上海市计算机学会理事长     |
|    | 邹 鹏 | 国防科学技术大学计算机学院教授、博士生导师<br>教育部计算机基础课程教学指导委员会副主任委员     |
|    | 张昆藏 | 青岛大学信息工程学院教授  |

# 译者序

Java语言是适合在分布环境下开发的最优秀的程序设计语言。在1996年，当C++语言在商业和学术方面得到了广泛应用时，Java语言作为一个强有力的竞争者登上了这个舞台。

安全、可移植且支持现代面向对象化结构是Java的主要特点。同C++相比，在编译或运行时Java很容易发现很多常见的程序错误。Java有一个异常机制，要求程序员精确地处理各种错误，并且有一个相对简单的模型用于区别基本数据类型（如int）和用户自定义类型。此外，Java中没有明确的指针类型。

Java是可移植的。例如，不管基本的计算机结构如何，在任何Java实现中，整型数据都有相同的范围值。Java还提供了一个图形用户界面（GUI）工具箱，允许用表单形式处理输入和输出。相对而言，GUI的确是容易使用的。更重要的是，对于Java能实现可移植的GUI。Java的精髓是：“一次性编程，任何地方都可运行。”

Java源程序保存在扩展名为.java的文件中。当地编译器Javac编译源程序并产生class文件，该文件由字节码构成。可移植的中间语言通过运行Java解释器来解译执行字节码（可移植的中间语言）。这个解释器也称做虚拟机。

本书是一本Java程序语言的教程，所包含的内容十分广泛，主要有Java简介、控制台输入/输出、流程控制语句、定义类、数组、继承、多态和抽象类、异常处理、文件和输入/输出、递归、统一建模语言和模式、接口和内部类、链表数据结构、集合和迭代器、Swing、applets等。每章给出自测题答案、程序设计和本章小结。在附录A至附录E中分别列出了关键字、优先关系和结合规则、统一代码字符集、类和接口的总结、ConsoleIn等。本书作者编程经验丰富，叙述透彻而简洁，提供的程序源码清晰流畅，突出了Java语言的编程风格和编程方法。

本书在结构上呈金字塔式，注重实践应用，从实际出发介绍了各种常用概念和方法，避免抽象的理论论述和复杂的公式推导，在典型的算法介绍方面深入浅出，简洁明了。

本书可以作为大专院校计算机专业和相关专业的教材，也可作为从事计算机应用的工程技术人员参考书。

本书由陈明、王国庆、王振武、王浩丽、郭新、廖丽萍、孙修虎、鹿扬、王风霞、耿燕、陈清夷、田丽娜、吴霞、朱新月、王玉亭、陈琳、张琳、刘志德、赵灵秀、任化敏等人翻译并校对。

尽管我们尽了很大的努力，但限于专业和英语水平，译稿中难免出现不妥之处，敬请读者批评指正。

# 前 言

本书是一本Java语言编程方面的教材和参考书。虽然本书含有编程技巧,但不是局限于此,而是围绕Java语言本身的特色展开的。作者写作时,面向的读者主要是不具备很多Java编程经验的本科生。因此,本书适合用做Java初级课程的教材或参考书,也适合作为Java语言的高级计算机课程的教材或参考书。本书面向多层次的读者。开始几章是为初学者而设计,阅读这几章的小结可以使有编程经验的读者快速了解Java的基本语法。后面各章同样简单易懂,但是更适合于那些有能力学习这些高级主题的学生。

本书对Java语言的讲述非常全面,特别是对Java中的继承、多态性及异常处理进行了详细而全面的论述。同时,提供了Java集类和Swing GUI库等Java标准库方面的详细材料,并且用独立的章节论述了递归及链表数据结构。

## 不含非标准的软件

对简单的控制台输入,Java没有标准类。包括我在内的一些作者,在书中为控制台输入提供了一个非标准的Java类,这种做法不受大学教师们的欢迎。本书中不包含这种软件,而是提供了简单的控制台输入类,因为一些教师非常喜欢这样的软件,毕竟它是可选择的。没有编程实例需要这种控制台输入类。许多程序中用到的是在任何Java的安装环境下均适用的标准类。

## 面向对象编程

本书全面概括了Java语言中实现的封装性、继承性及多态性。介绍Swing GUI的章节中提供了事件驱动编程的概述及大量练习。

UML和模式一章提供了与OOP相关的资料。

## 主题讲解的灵活性

本书给教师们很大的灵活性来重新安排教学资料的顺序。本书的特色是能适应各种教师的教学风格,而不是让教师们按照作者个人编著的顺序来授课。考虑到这些,各章的引言里均介绍了学习该章应具备的知识。

## 易于理解

仅仅按合适的顺序讲解正确的内容是不够的。一本书,对教师或专业人员来说仅仅清晰和明确也是不够的。对读者而言,易于理解是很重要的。本书同作者以往写的书一样,友好而易于掌握。

### 总结部分

每个重要的知识点均单独总结出来。作为每一章内容的概述,以及便捷性的参考资料,适合有经验的编程人员进行快速查找。

### 自测练习

每一章中均有大量的自测练习,并且在每章末尾给出这些自测练习的完整答案。

## 其他特征

常见错误, 编程小技巧, 以及带有运行示例的完整程序示例贯穿每一章。最后, 每一章均有小结, 适合作为编程项目系列的学生作业。

## CodeMate

本书已被收入 Addison-Wesley 的 CodeMate, 这是一个在线的专业书编程向导。借助它, 学生不必安装编译器就可以浏览、运行、编辑代码清单及他们书中的编码问题。本书为学生提供从各章选出的编程项目及一些附加项目的在线向导。用这个向导, 学生可以测试自己对每章关键概念的理解程度。

## 支持材料

本书所有用户均可获得下面的材料:

- 访问 Addison-Wesley 的 CodeMate 的途径。(登录名称是 Te\_service, 密码是 6666。)
- 本书中的源代码。
- J2SDK 的免费副本, 是标准版本 1.4 及 Java 2.0 的增强版本。



到 <http://www.aw-bc.com/codemate>, 可以访问 CodeMate 及在线源代码。

下面的资源仅仅对有资格的教师提供。

- 教师访问 Addison-Wesley 的 CodeMate
- 教师的解决问题手册
- 计算机化测试题库
- 幻灯片

## 如何获得 Java

本书附带的 CD 包含一个 J2SDK 的免费副本, 这是标准版本 1.4 及 Java 2.0 的增强版本。该副本包含一个 Java 编译器和一个 IDE (集成开发环境)。然而, 我们发现一个相对简单的 IDE 更适合初学者, 因为它在设计程序时几乎不出现差错。作者推荐 TextPad 环境, 到 <http://www.textpad.com/> 可以下载一个免费的试用 TextPad 副本。

要注意的是, 使用 TextPad 也需要安装前面提到的 Java 2.0 版本。

## 致谢

没有很多人给予的无偿帮助, 就不会有本书。感谢我的编辑, 来自 Addison-Wesley 的 Susan Hartman, 是她首先构思了此书, 她也是该书整个编写过程的灵魂人物。感谢 Beth Paquin, Marilyn Lloyd, Joyce Wells, Patty Mahtani, 以及 Addison-Wesley 的其他人员, 他们不断支持和鼓励, 为本书的编写、审查、修订和最终出版提供了极大的帮助。

特别感谢 Addison-Wesley 的 Patty Mahtani, 以及 Argosy 出版社的 Daniel Rausch, Jen Jackowitz, Kim Cofer 和 Cherry Govoni。正是他们不知疲倦的工作, 才使本书按时出版。

下列人员提出了修正意见和建议, 对本书最后的完成有极大的贡献。我衷心感谢他们(按字母顺序排列):



Jim Adams  
Gerald W. Adkins  
Dr. Bay Arinze  
Prof. Richard G. Baldwin  
Jon Bjornstad  
Adrienne Decker  
Arthur Geis  
Judy Hankins  
Chris Howard  
Eliot Jacobson  
Balaji Janamanchi  
Suresh Kalathur  
Dr. Clifford R. Kettemborough  
Frank Levey  
Xia Lin  
Mark M. Meysenburg  
Hoang M. Nguyen  
Prof. Bryson R. Payne  
W. Brent Seales  
Jeff Six  
Xueqing(Clare) Tang  
Natalie S. Wear  
Dale Welch  
Wook-Sung Yoo

Chandler-Gilbert Community College  
Georgia College & State University  
Drexel University  
Austin Community College  
Gavilan College  
University of Buffalo  
College of DuPage  
Middle Tennessee State University  
DeVry University  
University of California, Santa Barbara  
Texas Tech University  
Boston University  
IT 顾问, 教授  
Manatee Community College  
Drexel University  
Doane College  
Deanze College  
North Georgia College & State University  
University of Kentucky  
University of Delaware  
Governors State University  
University of South Florida  
University of West Florida  
Gannon University

特别感谢 Rick Ord, 他审阅了整本书的草稿, 并且为改进此书提供了详细而很有帮助的建议。他的洞察力对本书的帮助极大。

Walter Savitch  
<http://www-cse.ucsd.edu/users/savitch/>  
wsavitch@ucsd.edu

# 目 录

<b>第 1 章</b>	<b>引言</b> .....	1
1.1	Java 简介 .....	1
1.2	表达式和赋值语句 .....	8
1.3	字符串类 .....	20
1.4	编程风格 .....	28
1.5	小结 .....	32
<b>第 2 章</b>	<b>控制台输入和输出</b> .....	35
2.1	屏幕输出 .....	35
2.2	JOptionPane .....	43
2.3	使用 BufferedReader 的控制台输入 .....	48
2.4	使用 ConsoleIn 进行输入 .....	51
2.5	小结 .....	54
<b>第 3 章</b>	<b>流程控制语句</b> .....	58
3.1	分支语句 .....	58
3.2	布尔表达式 .....	67
3.3	循环 .....	80
3.4	小结 .....	96
<b>第 4 章</b>	<b>定义类 I</b> .....	102
4.1	类的定义 .....	102
4.2	信息的隐藏和封装 .....	123
4.3	重载 .....	130
4.4	构造函数 .....	135
4.5	小结 .....	146
<b>第 5 章</b>	<b>定义类 II</b> .....	153
5.1	静态方法与静态变量 .....	153
5.2	引用与类参数 .....	163
5.3	引用的使用与误用 .....	172
5.4	包与 javadoc .....	185
5.5	小结 .....	192
<b>第 6 章</b>	<b>数组</b> .....	197
6.1	数组概述 .....	197
6.2	数组和引用 .....	204

6.3	数组编程 .....	212
6.4	多维数组 .....	224
6.5	小结 .....	232
<b>第7章</b>	<b>继承 .....</b>	<b>241</b>
7.1	继承基础 .....	241
7.2	封装和继承 .....	253
7.3	使用继承编程 .....	258
7.4	小结 .....	264
<b>第8章</b>	<b>多态和抽象类 .....</b>	<b>269</b>
8.1	多态 .....	269
8.2	抽象类 .....	285
8.3	小结 .....	289
<b>第9章</b>	<b>异常处理 .....</b>	<b>292</b>
9.1	异常处理的基础知识 .....	292
9.2	在方法中抛出异常 .....	312
9.3	更多异常处理的技术 .....	319
9.4	小结 .....	330
<b>第10章</b>	<b>文件输入/输出 .....</b>	<b>335</b>
10.1	文件输入/输出简介 .....	335
10.2	文本文件 .....	336
10.3	File 类 .....	349
10.4	二进制文件 .....	352
10.5	随机访问二进制文件 .....	365
10.6	小结 .....	369
<b>第11章</b>	<b>递归 .....</b>	<b>374</b>
11.1	无返回值的递归方法 .....	374
11.2	有返回值的递归方法 .....	382
11.3	递归的思想 .....	385
11.4	小结 .....	392
<b>第12章</b>	<b>统一建模语言和模式 .....</b>	<b>396</b>
12.1	UML .....	396
12.2	模式 .....	399
12.3	小结 .....	405
<b>第13章</b>	<b>接口和内部类 .....</b>	<b>408</b>
13.1	接口 .....	408
13.2	内部类的简单应用 .....	422

13.3	关于内部类的更多内容 .....	426
13.4	小结 .....	433
<b>第 14 章</b>	<b>链表数据结构 .....</b>	<b>438</b>
14.1	Java 链表 .....	439
14.2	树 .....	465
14.3	小结 .....	471
<b>第 15 章</b>	<b>集合和迭代器 .....</b>	<b>475</b>
15.1	向量 .....	475
15.2	集合 .....	486
15.3	迭代器 .....	499
15.4	小结 .....	504
<b>第 16 章</b>	<b>Swing I .....</b>	<b>508</b>
16.1	事件驱动程序设计技术 .....	508
16.2	按钮、事件和其他 Swing 组件 .....	509
16.3	容器和布局管理器 .....	522
16.4	菜单和按钮 .....	535
16.5	文本字段和文本区域 .....	543
16.6	小结 .....	552
<b>第 17 章</b>	<b>applet .....</b>	<b>559</b>
17.1	HTML 简介 .....	559
17.2	applet 程序设计 .....	566
17.3	HTML 中的 applet .....	573
17.4	小结 .....	576
<b>第 18 章</b>	<b>Swing II .....</b>	<b>579</b>
18.1	窗口监听器 .....	579
18.2	图标和滚动条 .....	584
18.3	Graphics 类 .....	596
18.4	颜色 .....	607
18.5	字体和 drawString 方法 .....	612
18.6	小结 .....	616
<b>第 19 章</b>	<b>Java 永远不会结束 .....</b>	<b>620</b>
19.1	多线程 .....	620
19.2	JavaBeans .....	629
19.3	Java 和数据库连接 .....	630
19.4	小结 .....	631

附录 A	关键字 .....	633
附录 B	优先关系和结合规则 .....	634
附录 C	Unicode 字符集 .....	635
附录 D	类和接口的总结 .....	636
附录 E	ConsoleIn .....	677
索引	.....	684

# 第1章 引 言

## 本章要点

- Java 简介
- 表达式和赋值语句
- 字符串类
- 编程风格

## 简介

本章主要介绍 Java 语言以及用其编写简单程序时应注意的一些细节,包括表达式、赋值和控制输出。Java 关于赋值和表达式的一些详细规则与其他大部分高级程序设计语言非常类似。每一种语言都有其各自的方式来处理字符串和控制输出,因此,即使是有经验的程序员也应该阅读本章的内容。尽管你也许是使用其他语言的一位资深开发者,但也应该至少阅读题为“一个简单的 Java 应用程序示例”的 1.1.4 小节,当然,通读 1.1 节更好。同时,最好通读 1.3 节有关字符串的部分。而且为了了解 Java 中的常量定义和注释,至少应浏览 1.4 节。

## 预备知识

本书自成体系,无需多少预备知识,只要懂得高中代数即可。

## 1.1 Java 简介

这一节我们对 Java 语言进行概述。

### 1.1.1 Java 语言的起源

Java 是一种众所周知的网络编程语言。然而,本书和许多其他著作以及一些程序员们的看法一样,认为 Java 是一种通用的编程语言,适用于大多数应用程序的开发,无论是否与网络有关。Java 的第一个版本虽然不是面向开发网络和桌面应用程序,但 Java 的后续版本最终却朝这个方向发展,且成就非凡。

1991 年, Sun Microsystems 的 James Gosling 领导一个小组开发首版 Java (当时并不叫 Java), Java 的第一个版本是为家用电器(譬如洗衣机和电视机)编程而设计的。尽管是在一个不被看好的应用领域,但设计这样的一种语言也绝非易事。家用电器由许多不同类型的计算机处理器(芯片)控制。Gosling 正在设计的这种语言就是需要在这些不同的处理器上工作。而且,家用电器通常是些廉价产品,因此制造商不愿投入大量的资金来开发复杂的编译器(一个编译器就是将一个程序源代码翻译成处理器能够理解的语言的一个程序)。为了简化为每一类设备编写编译器的任务,开发小组采用了两步的解释处理过程。首先,程序被解释成为一种中间语言,这种语言对所有设备(或所有的计算机)来说都一样。然后,再由一个小的、易编写的和廉价的程序来解释这种中间语言,

使其转化为专用设备或计算机的机器语言。这种中间语言称做Java字节码，简称字节码。由于只有一种中间语言，所以从原程序到中间语言再到机器语言的两步解释中，较困难的那一步的难易程度对于所有的设备（或计算机）而言都是一样的。因而，节省了直接翻译成多种机器语言的大部分费用。这种为家电设计的语言从未受到家电制造商的青睐，然而，如今却发展成为一种广泛应用的编程语言。

为什么称做字节码（byte-code）？code这个词通常指的是一个程序或程序的一部分。一个字节是一个小的存储单位（精确地说是8位）。计算机可读取的信息通常以字节为单位。因此，字节码这个专业术语指的是专供机器，而不是人阅读的程序。

1994年，Sun Microsystems的Patrick Naughton和Jonathan Payne开发了一个浏览器，能够在Internet上运行用Java编写的程序。这个Web浏览器发展成为今天的HotJava，这是Java同Internet连接的开始。在1995年的秋天，Netscape Incorporated使自身的浏览器也能够运行Java程序，其他公司紧随其后，开发相适应的软件。

### 1.1.2 对象和方法

Java是一种面向对象编程的语言，简称为OOP（object-oriented programming）。什么是OOP？我们所处的世界是由许许多多的对象组成的，例如人、车、房屋、街道、计算机、纸张，等等。每一个对象都有某种行为能力，而且每种行为都会对世界上的其他对象产生一定的影响。OOP就是一种编程思想，主张一个程序由很多对象组成，这些对象通过行为彼此影响。

面向对象编程有其独特的术语，对象具体化就是实例。一个对象的行为称做方法。如果一类对象具有相同的类型，通常说它们是同一类。例如，在一个航空模拟程序中，所有的模拟飞机都属于同一个类，类名为airplane。类的所有对象都有同样的方法（或行为），比如起飞、飞往特定地点、降落，等等，然而，所有的模拟飞机并不相同。通过不同的数据（不同的信息）标明它们具有不同的特征。例如，可能某个飞机对象仅有速度和纬度两个参数。

---

### 为什么命名为“Java”？

为一种编程语言命名的现有惯例是根据设计者的喜好，Java也不例外。有关命名为Java的缘由众说不一。尽管有很多种不同的解释，但有一点是明确的，即“Java”这个词与任何有关Java（爪哇）的特性或者历史无关。一个可信的说法是，在为其命名的讨论会上没有达成一致的情形下，开发小组人员一起去喝咖啡时，有了将其命名为“Java”的灵感。

---

倘若你以前用过其他的编程语言，那么用这些语言的术语解释Java对应的术语会很有收益。其他语言中称做过程、方法、函数或者子程序的项目在Java中统称为方法。在Java里，所有的方法都是类的一部分。正如我们将看到的一样，一个Java应用程序是一个具有main方法的类，当运行这类程序时，运行系统自动唤醒main方法（也就是说，自动初始化main行为）。Java应用程序可以视为“常规”的Java程序，除此之外，还有一类称做applet的Java程序。其他的术语与大多数编程语言一样，当涉及到这些术语时，我们会对每一个概念进行解释。

### 1.1.3 applets

有两种Java程序：applets和applications。一个application程序是一个常规程序。尽管applet这个名字听起来好像与苹果有些关系，但其实指的是一个小的Java应用程序，而不是一个苹果。applets和applications几乎相同，不同之处在于applications同其他程序一样可在计算机上运行，而applets

可以运行于 Web 浏览器上, 因此, 可以发送到 Internet 的另一个地址的计算机上运行。applet 总是使用一个类似窗口的接口, 但是, 正如将在第 16 章到第 18 章中所介绍的, 并非使用窗口接口的所有程序都属于 applets。

applet 一般运行于 Web 浏览器, 不过也能够在一个称做 applet viewer 的程序上运行。applet viewer 其实是一个调试助手, 而不是供用户使用的运行 applet 的最终环境。而现在一般情况都是在 applet viewer 下运行, 这样不是很好。Java 具有很多类库设计窗口接口, 不需要与浏览器连接。我们更愿意使用这些库而不是 applets 去编写不需要 Web 浏览器就能运行的程序<sup>①</sup>。在本书中, 我们说明怎样编写针对 applets 的 windowing 程序和无需 Web 浏览器便能运行的程序。事实上, 这两种方法在技术实现上和对类库程序的使用程度上都有很多相同之处。一旦知道如何设计和编写 applets 或者 applications, 那么学习和编写以上两类程序就是轻而易举的事情了。

一个 applet 总是有一个窗口接口, 一个 application 程序可能有一个窗口接口或使用简单的控制 I/O 流。因此, 不要忽视将要开始学习的这些代码。在本书前面的部分, 大多数程序实例都使用简单的控制 I/O 流 (也就是简单的文本 I/O 流)。

#### 1.1.4 一个简单的 Java 应用程序示例

实例 1.1 是一个简单的应用程序, 运行时屏幕将显示运行结果。前面已经说过, 一个 Java 程序其实就是一个具有 main 方法的类。当程序运行时, 也就是通过 main 声明的行为被执行时, main 方法被唤醒。main 方法的主体包含在 {} 中, 因此, 当程序运行时, {} 中的语句被执行 (假如你对类和方法还是不太明白, 这里将进行解释, 请继续阅读)。

下面这一行说明这段程序是一个叫 FirstProgram 的类:

```
public class FirstProgram
{
```

下面一行, 开始定义 main 方法:

```
public static void main(String[] args)
{
```

一个 Java 类定义的具体细节, 还有像 public, static, void 等词的含义将在后续几章做出解释, 到那时, 你会想起开始的这几行。

```
public class FirstProgram
{
public static void main(String[] args)
{
```

下面两行, 是程序将要执行的第一个动作:

```
System.out.println("Hello reader.");
System.out.println("Welcome to Java.");
```

这两行均以 System.out.println 开头, 其功能是将圆括号中的字符串显示在屏幕上。比如, 执行第一句, 将把 Hello reader 显示到屏幕上, 下一行会紧跟上一句执行。由此, 这两行语句的输出结果如下:

```
Hello reader.
```

<sup>①</sup> 一个 applet viewer 的运行实际上还是在后台用浏览器来运行 applet 程序, 只是从观感上似乎是同浏览器无关的单独运行。



```
Welcome to Java.
```

以 `System.out.println` 开头的这几行的意思是“输出圆括号中所显示的内容”，具体为什么这样写在现在还不需要我们关心。但是，在这里先略微介绍一些。

Java 程序是通过对象执行行为的方式运行的，由对象执行的行为称做方法。`System.out` 是进行屏幕输出的一个对象；`println` 是该对象的方法（也就是行为），作用是将圆括号中的内容输出到屏幕上。当一个对象通过方法执行一个行为时，称做唤醒方法（或者调用方法）。在 Java 程序中，调用方法的书写形式是，对象名后紧跟一个“.”运算符，然后是方法名，最后是一对圆括号。括号里的内容可有可无，一般是该方法或行为所需的参数。示例中的几行代码中用到的方法是 `println`，该方法实现屏幕输出。括号中的参数代表要输出的内容。

调用方法有时也称做发送消息给对象，获得这一消息的对象做相应的动作。发送消息这一术语我们很少使用，但它是一些程序员和很多作者惯用的标准术语。

Java 中的变量声明和其他编程语言类似，下面是在实例 1.1 中声明名为变量 `answer` 的语句：

```
int answer;
```

`int` 类型是 Java 语言中整型的一种。因此，该语句的意思为 `answer` 是一个可以存储整型值的变量。

### 实例 1.1 Java 程序实例

```
1 public class FirstProgram
2 {
3     public static void main(String[] args)
4     {
5         System.out.println("Hello reader.");
6         System.out.println("Welcome to Java.");
7
8         System.out.println("Let's demonstrate a simple calculation.");
9         int answer;
10        answer = 2 + 2;
11        System.out.println("2 plus 2 is " + answer);
12    }
```

类 (程序) 名

main 方法

#### 示例对话 1

```
Hello reader.
Welcome to Java.
Let's demonstrate a simple calculation.
2 plus 2 is 4
```

下面这条语句是第一个程序真正做的运算：

```
answer = 2 + 2;
```

在 Java 中，等号用做赋值操作符，它是给等号左边变量赋值的一个指令。在上面的程序中，等号并不意味着 `answer` 等于 `2 + 2`，而是作为一条指令，使得 `answer` 的值成为 `2 + 2`。

最后的程序行为是：

```
System.out.println("2 plus 2 is " + answer);
```

这是一条类似我们前面讨论过的输出语句，但有新的地方。注意，“2 plus 2 is ”字符串后紧跟一个“+”号和变量 `answer`。在这种情况下，“+”号是连接两个字符串的运算符。然而，变量 `answer` 不是一个字符串。对于“+”号来说，假如其中一个操作数是字符串，Java 会自动转换另一个操作数，