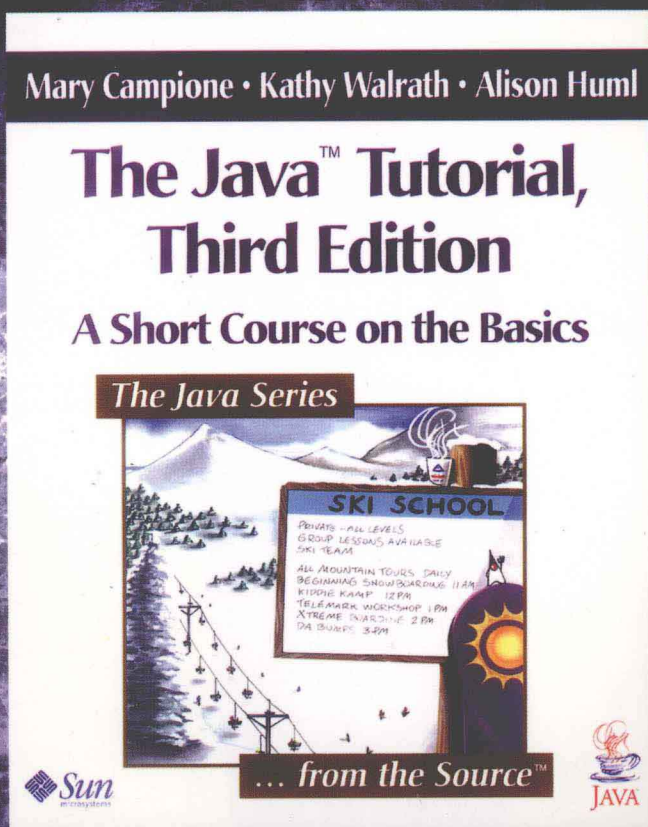


计 算 机 科 学 丛 书

原书第3版

Java语言导学

Mary Campione Kathy Walrath Alison Huml 著 马朝晖 陈美红 译



The Java Tutorial
Third Edition
A Short Course on the Basics



机械工业出版社
China Machine Press

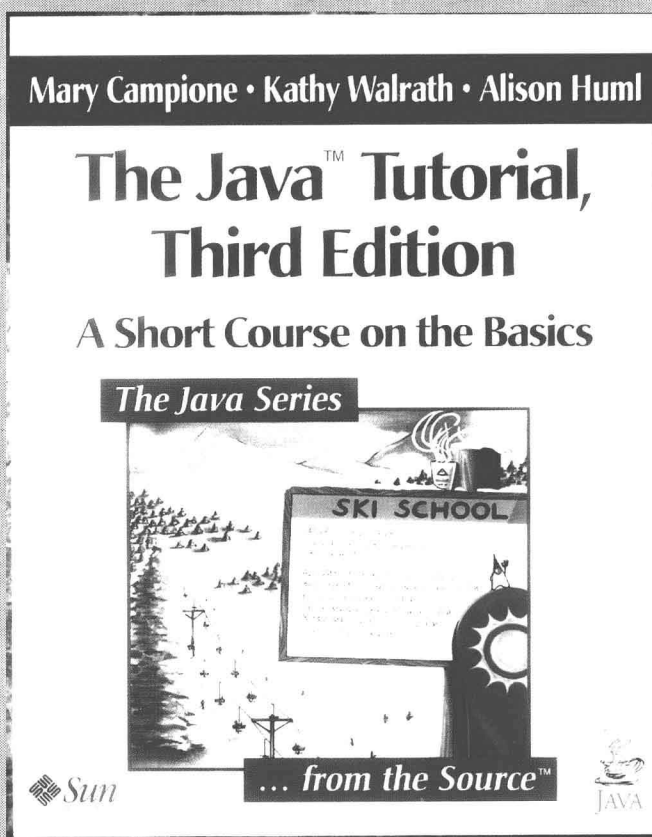


原书第3版

计 算 机 科 学

Java 语言导学

Mary Campione Kathy Walrath Alison Huml 著 马朝晖 陈美红 译



Third Edition
A Short Course on the Basics



机械工业出版社
China Machine Press

本书以面向任务的方法介绍 Java 语言, 覆盖了 Java 语言的基础知识和使用技巧。主要内容
包括: Java 语言基础、对象基础和简单数据对象、类和继承、接口和包、异常处理、线程、输
入/输出、Swing 用户界面等。本书的作者经验丰富, 为初学者指引了易学快用的方法, 精选的
大量实例提供了 Java 最新应用方法 (针对 Java 2 SDK 1.3 版), 并提供了编程问题的解决方案。

本书在国外被广泛采用作为 Java 程序设计语言课程的教材, 既适合高等院校讲授 Java 语
言课程之用, 也可供程序员作为自学和参考指南。

Simplified Chinese edition copyright © 2003 by PEARSON EDUCATION NORTH ASIA LIMITED
and China Machine Press.

Original English language title: *The Java Tutorial, Third Edition: A Short Course on the Basics*, by
Mary Campione, Kathy Walrath and Alison Huml, Copyright 2001. (ISBN 0-201-70939-9)

All rights reserved.

Published by arrangement with the original publisher, Pearson Education, Inc., publishing as Sun
Microsystems, Inc.

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。
版权所有, 侵权必究。

本书版权登记号: 图字: 01 - 2001 - 4772

图书在版编目 (CIP) 数据

Java 语言导学 (原书第 3 版) / 坎皮恩 (Campione, M.), 沃拉瑟 (Walrath, K.), 赫木
尔 (Huml, A.) 著; 马朝晖等译. - 北京: 机械工业出版社, 2003.1

(计算机科学丛书)

书名原文: *The Java Tutorial, Third Edition: A Short Course on the Basics*

ISBN 7-111-09585-5

I .J… II .①坎…②沃…③赫…④马… III .Java 语言 - 程序设计 IV .TP312

中国版本图书馆 CIP 数据核字 (2001) 第 091265 号

机械工业出版社 (北京市西城区百万庄大街 22 号 邮政编码 100037)

责任编辑: 武恩玉

北京牛山世兴印刷厂印刷 · 新华书店北京发行所发行

2005 年 8 月第 2 版第 6 次印刷

787mm × 1092mm 1/16 · 23.75 印张

印数: 16 001-18 000 册

定价: 39.00 元

凡购本书, 如有倒页、脱页、缺页, 由本社发行部调换

出版者的话

文艺复兴以降，源远流长的科学精神和逐步形成的学术规范，使西方国家在自然科学的各个领域取得了垄断性的优势；也正是这样的传统，使美国在信息技术发展的六十多年间名家辈出、独领风骚。在商业化的进程中，美国的产业界与教育界越来越紧密地结合，计算机学科中的许多泰山北斗同时身处科研和教学的最前线，由此而产生的经典科学著作，不仅擘划了研究的范畴，还揭橥了学术的源变，既遵循学术规范，又自有学者个性，其价值并不会因年月的流逝而减退。

近年，在全球信息化大潮的推动下，我国的计算机产业发展迅猛，对专业人才的需求日益迫切。这对计算机教育界和出版界都既是机遇，也是挑战；而专业教材的建设在教育战略上显得举足轻重。在我国信息技术发展时间较短、从业人员较少的现状下，美国等发达国家在其计算机科学发展的几十年间积淀的经典教材仍有许多值得借鉴之处。因此，引进一批国外优秀计算机教材将对我国计算机教育事业的发展起积极的推动作用，也是与世界接轨、建设真正的世界一流大学的必由之路。

机械工业出版社华章图文信息有限公司较早意识到“出版要为教育服务”。自1998年始，华章公司就将工作重点放在了遴选、移译国外优秀教材上。经过几年的不懈努力，我们与Prentice Hall, Addison-Wesley, McGraw-Hill, Morgan Kaufmann等世界著名出版公司建立了良好的合作关系，从它们现有的数百种教材中甄选出Tanenbaum, Stroustrup, Kernighan, Jim Gray等大师名家的一批经典作品，以“计算机科学丛书”为总称出版，供读者学习、研究及度藏。大理石纹理的封面，也正体现了这套丛书的品位和格调。

“计算机科学丛书”的出版工作得到了国内外学者的鼎力襄助，国内的专家不仅提供了中肯的选题指导，还不辞劳苦地担任了翻译和审校的工作；而原书的作者也相当关注其作品在中国的传播，有的还专诚为其书的中译本作序。迄今，“计算机科学丛书”已经出版了近百个品种，这些书籍在读者中树立了良好的口碑，并被许多高校采用为正式教材和参考书籍，为进一步推广与发展打下了坚实的基础。

随着学科建设的初步完善和教材改革的逐渐深化，教育界对国外计算机教材的需求和应用都步入一个新的阶段。为此，华章公司将加大引进教材的力度，在“华章教育”的总体规划之下出版三个系列的计算机教材：针对本科生的核心课程，剔抉外版菁华而成“国外经典教材”系列；对影印版的教材，则单独开辟出“经典原版书库”；定位在高级教程和专业参考的“计算机科学丛书”还将保持原来的风格，继续出版新的品种。为了保证这三套丛书的权威性，同时也为了更好地为学校和老师服务，华章公司聘请了中国科学院、北京大学、清华大学、国防科技大学、复旦大学、上海交通大学、南京大学、浙江大学、中国科技大学、哈尔滨工业大学、西安交通大学、中国人民大学、北京航空航天大学、北京邮电大学、中山大学、解放军理工大学、郑州大学、湖北工学院、中国国家信息安全测评认证中心等国内重点大学和科研机构在计算机的各个领域的著名学者组成“专家指导委员会”，为我们提供选题意见和出版监督。

“国外经典教材”是响应教育部提出的使用外版教材的号召，为国内高校的计算机本科教学度身订造的。在广泛地征求并听取丛书的“专家指导委员会”的意见后，我们最终选定了这20多种篇幅内容适度、讲解鞭辟入里的教材，其中的大部分已经被M.I.T.、Stanford、U.C. Berkley、C.M.U.等世界名牌大学采用。丛书不仅涵盖了程序设计、数据结构、操作系统、计算机体系结构、数据库、编译原理、软件工程、图形学、通信与网络、离散数学等国内大学计算机专业普遍开设的核心课程，而且各具特色——有的出自语言设计者之手、有的历三十年而不衰、有的已被全世界的几百所高校采用。在这些圆熟通博的名师大作的指引之下，读者必将在计算机科学的宫殿中由登堂而入室。

权威的作者、经典的教材、一流的译者、严格的审校、精细的编辑，这些因素使我们的图书有了质量的保证，但我们的目标是尽善尽美，而反馈的意见正是我们达到这一终极目标的重要帮助。教材的出版只是我们的后续服务的起点。华章公司欢迎老师和读者对我们的工作提出建议或给予指正，我们的联系方法如下：

电子邮件：hzedu@hzbook.com

联系电话：(010) 68995265

联系地址：北京市西城区百万庄南街1号

邮政编码：100037

出于对不同读者的需要的考虑，本书不附带提供原英文版的配套光盘。需要配套光盘的读者可以通过以下途径获取：

- 从我们的教学支持站点：<http://www.china-pub.com/computers/main/jiaocai/jiaocai.htm>上下载。

- 通过以下联系方式邮购光盘：

电 话：010 - 68995264

电子邮件：marketing@hzbook.com

地 址：北京市西城区百万庄南街 1 号华章公司

邮政编码：100037

专家指导委员会

(按姓氏笔画顺序)

尤晋元
石教英
张立昂
邵维忠
周克定
郑国梁
高传善
裘宗燕

王 珊
吕 建
李伟琴
陆丽娜
周傲英
施伯乐
梅 宏
戴 葵

冯博琴
孙玉芳
李师贤
陆鑫达
孟小峰
钟玉琢
程 旭

史忠植
吴世忠
李建中
陈向群
岳丽华
唐世渭
程时端

史美林
吴时霖
杨冬青
周伯生
范 明
袁崇义
谢希仁

前 言

自从最初的 Java Development Kit 于 1995 年 5 月发布以来, Sun Microsystems 公司的工程人员为改进和增强 Java 平台付出了艰苦的努力。我们也相应地更新了本书的内容来反映这些工程人员的成果。

本书是第 3 版, 针对 Java 2 Software Development Kit (SDK) 1.3 版。因为你可能不得不使用 Java 平台的较早版本编写或更新代码, 所以本书也适用于 1.2 和 1.1 版。

除了添加了 1.3 版的信息, 我们还增加了问题与练习以帮助你实践所学到的内容。为了帮助初学者避免许多常见的错误, 附录 A 研究了编程问题及其解决方案。每节后面的小结也是本版中新增的。

与第 1 版和第 2 版一样, 本书也以 Sun Microsystems 网站上的 Java 平台在线指南为基础。该网站地址为: <http://java.sun.com/docs/books/tutorial/index.html>。

与在线版本一样, 本书反映了 Java 技术的最新发展。与在线版本不同的是, 本书只关注初、中级程序员需要的 API。一旦你掌握了本书中的资料, 就可以在此网站上研究 Java 平台的其他问题。

我们的目标一直是写一本轻松易学的面向任务的程序员指南, 并且以大量实际例子帮助人们学习编程。

谁应该阅读本书

本书既适合初学者, 也适合有经验的程序员:

- 新程序员通过从头到尾阅读本书可以得到最大的收获, 包括按照第 1 章“起步”中的步骤说明编译和运行自己的第一个程序。
- 有过程式语言 (比如 C) 经验的程序员可能希望从 Java 编程语言的面向对象概念和特性开始学习。
- 有面向对象编程经验的程序员可能希望先学习更高级的内容, 比如关于 applet、基本类或用户界面。

无论你是哪种程序员, 都可以通过本书找到满足自己需要的学习途径。

如何使用本书

本书的设计方式使你既可以通读它, 也可以在主题之间跳转。当另一个地方讨论某主题时, 你会看到到此位置的“链接”, 其中指出了章节号。

本书中使用的所有样例代码都可以从网上和本书附带的光盘上得到。在每章的末尾有一

节“代码样例”，其中的表格指出样例在光盘和网上的位置。

我们尽力使本书跟上最新的技术。要学习更新的信息，请访问以下 URL：

<http://java.sun.com/docs/books/tutorial/books/3e/index.html>

目 录

出版者的话	
专家指导委员会	
前言	
第 1 章 起步	1
1.1 关于 Java 技术	1
1.1.1 Java 编程语言	1
1.1.2 Java 平台	2
1.1.3 Java 技术可以做什么	3
1.2 Java 技术将如何改变我们的生活	4
1.3 第一步 (Win32)	5
1.3.1 检查表	5
1.3.2 创建第一个应用程序	5
1.3.3 创建第一个 applet	8
1.3.4 错误解释 (Win32)	9
1.4 第一步 (UNIX/Linux)	10
1.4.1 检查表	10
1.4.2 创建第一个应用程序	10
1.4.3 创建第一个 applet	13
1.4.4 错误解释 (UNIX/Linux)	14
1.5 第一步 (MacOS)	15
1.5.1 检查表	15
1.5.2 创建第一个应用程序	15
1.5.3 创建第一个 applet	19
1.5.4 错误解释 (MacOS)	20
1.6 分析 HelloWorld	20
1.6.1 对应用程序的解释	20
1.6.2 对 applet 的剖析	23
1.7 问题与练习	26
1.7.1 问题	26
1.7.2 练习	26
1.7.3 解答	27
1.8 代码样例	27
第 2 章 面向对象的编程概念	28
2.1 什么是对象	28
2.2 什么是消息	29
2.3 什么是类	30
2.4 什么是继承	32
2.5 什么是接口	33
2.6 如何将这些概念运用到代码中	34
2.6.1 ClickMe 的源代码和 applet 标记	34
2.6.2 ClickMe applet 中的对象	35
2.6.3 ClickMe applet 中的类	35
2.6.4 ClickMe applet 中的消息	36
2.6.5 ClickMe applet 中的继承	36
2.6.6 ClickMe applet 中的接口	37
2.6.7 API 文档	38
2.7 小结	38
2.8 问题与练习	39
2.8.1 问题	39
2.8.2 练习	39
2.8.3 解答	39
2.9 代码样例	39
第 3 章 语言基础	41
3.1 变量	41
3.1.1 数据类型	42
3.1.2 变量名称	44
3.1.3 作用范围	44
3.1.4 变量初始化	45
3.1.5 final 变量	45
3.1.6 变量的小结	46
3.1.7 问题与练习: 变量	46
3.2 操作符	47
3.2.1 算术操作符	47
3.2.2 关系和条件操作符	51
3.2.3 移动和位操作符	53
3.2.4 赋值操作符	56
3.2.5 其他操作符	57
3.2.6 操作符的小结	58

3.2.7 问题与练习: 操作符	59	4.2.12 问题与练习: 字符和字符串	98
3.3 表达式、语句和代码块	60	4.3 数字	99
3.3.1 表达式	61	4.3.1 数字类	99
3.3.2 语句	62	4.3.2 将字符串转换为数字	101
3.3.3 代码块	63	4.3.3 将数字转换为字符串	102
3.3.4 表达式、语句和代码块的小结	63	4.3.4 对数字进行格式化	103
3.3.5 问题与练习: 表达式、语句和 代码块	63	4.3.5 用定制的格式对数字进行 格式化	104
3.4 流程控制语句	64	4.3.6 高级算术功能	105
3.4.1 while 和 do-while 语句	65	4.3.7 数字小结	109
3.4.2 for 语句	66	4.3.8 问题与练习: 数字	109
3.4.3 if-else 语句	67	4.4 数组	110
3.4.4 switch 语句	68	4.4.1 创建和使用数组	111
3.4.5 异常处理语句	70	4.4.2 对象数组	112
3.4.6 分支语句	71	4.4.3 数组的数组	113
3.4.7 流程控制语句的小结	74	4.4.4 复制数组	115
3.4.8 问题与练习: 流程控制语句	75	4.4.5 数组小结	116
3.5 代码样例	76	4.4.6 问题与练习: 数组	116
第 4 章 对象基础和简单数据对象	78	4.5 代码样例	116
4.1 对象的生存周期	78	第 5 章 类和继承	119
4.1.1 创建对象	79	5.1 创建类	119
4.1.2 使用对象	82	5.1.1 声明类	120
4.1.3 清除不被使用的对象	84	5.1.2 声明成员变量	121
4.1.4 对象小结	85	5.1.3 定义方法	122
4.1.5 问题与练习: 对象	85	5.1.4 为类提供构造器	124
4.2 字符和字符串	86	5.1.5 将信息传递给方法或构造器	125
4.2.1 字符	86	5.1.6 从方法返回值	127
4.2.2 字符串和字符串缓冲区	88	5.1.7 使用 this 关键字	128
4.2.3 创建字符串和字符串缓冲区	88	5.1.8 控制对类成员的访问	128
4.2.4 得到字符串或字符串缓冲区的 长度	90	5.1.9 理解实例和类成员	132
4.2.5 通过索引从字符串或字符串缓冲区 得到字符	90	5.1.10 实例和类成员的初始化	134
4.2.6 在字符串中搜索字符或子字符串	91	5.1.11 创建类的小结	135
4.2.7 比较字符串和部分字符串	93	5.1.12 问题与练习: 创建类	136
4.2.8 操作字符串	94	5.2 管理继承	136
4.2.9 修改字符串缓冲区	95	5.2.1 覆盖和隐藏方法	137
4.2.10 字符串和编译器	96	5.2.2 隐藏成员变量	139
4.2.11 字符和字符串小结	97	5.2.3 使用 super	139
		5.2.4 使用 Object 的后代	140
		5.2.5 编写 final 类和方法	142

5.2.6 编写抽象类和方法	143	7.7 异常的优点	174
5.2.7 管理继承的小结	145	7.7.1 优点 1: 将错误处理代码与“常规” 代码分离	174
5.2.8 问题与练习: 管理继承	145	7.7.2 优点 2: 将错误沿调用堆栈向上 传递	176
5.3 实现嵌套的类	146	7.7.3 优点 3: 对错误类型进行分组和 区分	177
5.3.1 内部类	147	7.8 异常的小结	177
5.3.2 关于嵌套类的其他问题	148	7.9 问题与练习	178
5.3.3 嵌套类的小结	149	7.9.1 问题	178
5.3.4 问题与练习: 嵌套类	149	7.9.2 练习	179
5.4 代码样例	150	7.9.3 解答	179
第 6 章 接口和包	152	7.10 代码样例	179
6.1 编写和使用接口	152	第 8 章 线程: 同时执行多个任务	180
6.1.1 定义接口	153	8.1 什么是线程	180
6.1.2 实现接口	154	8.2 使用 Timer 和 TimerTask 类	182
6.1.3 将接口作为类型使用	154	8.2.1 停止计时器线程	183
6.1.4 接口不能改变	155	8.2.2 重复执行任务	184
6.1.5 接口小结	155	8.2.3 关于 Timer 的更多信息	185
6.1.6 问题与练习: 接口	155	8.3 定制线程的 run 方法	185
6.2 创建和使用包	156	8.3.1 对 Thread 类进行子类化和覆盖 run	185
6.2.1 创建包	157	8.3.2 实现 Runnable 接口	187
6.2.2 命名包	157	8.3.3 决定使用 Runnable 接口	188
6.2.3 使用包成员	158	8.4 线程的生存周期	188
6.2.4 管理源代码文件和类文件	159	8.4.1 创建线程	188
6.2.5 创建和使用包的小结	161	8.4.2 启动线程	189
6.2.6 问题与练习: 创建和使用包	161	8.4.3 使线程不可运行	190
6.3 代码样例	162	8.4.4 停止线程	190
第 7 章 使用异常处理错误	163	8.4.5 isAlive 方法	191
7.1 什么是异常	163	8.5 理解线程优先级	191
7.2 捕获或指定要求	164	8.5.1 400 000 微米线程赛跑	192
7.3 捕获和处理异常	164	8.5.2 自私的线程	193
7.3.1 try 块	165	8.5.3 时间片	193
7.3.2 catch 块	166	8.5.4 线程优先级的小结	195
7.3.3 finally 块	166	8.6 线程的同步	195
7.3.4 结合	167	8.6.1 生产者/消费者例子	195
7.4 指定方法抛出的异常	170	8.6.2 锁定对象	197
7.5 如何抛出异常	171	8.6.3 使用 notifyAll 和 wait 方法	198
7.5.1 throw 语句	171	8.6.4 运行生产者/消费者例子	200
7.5.2 Throwable 类及其子类	172		
7.5.3 创建自己的异常类	172		
7.6 运行时异常——争论	174		

8.6.5 避免饿死和死锁	201	9.8 代码样例	232
8.7 线程的分组	202	第 10 章 Swing 用户界面	234
8.7.1 线程组	202	10.1 SwingAPI 概述	234
8.7.2 在组中显式地创建线程	202	10.2 示例 1: Hello World Swing	235
8.7.3 得到线程的组	202	10.3 示例 2: SwingApplication	236
8.7.4 使用 ThreadGroup 类	203	10.3.1 外观	236
8.8 线程的小结	206	10.3.2 建立按钮和标签	237
8.8.1 线程的包支持	206	10.3.3 处理事件	238
8.8.2 线程的语言支持	206	10.3.4 在组件周围添加边框	239
8.8.3 线程的运行支持	206	10.4 示例 3: CelsiusConverter	240
8.9 问题与练习	207	10.4.1 添加 HTML	241
8.10 代码样例	207	10.4.2 添加图标	241
第 9 章 I/O: 读和写	209	10.5 示例 4: LunarPhases	242
9.1 I/O 流概述	209	10.5.1 复合边框	243
9.1.1 字符流	210	10.5.2 组合框	243
9.1.2 字节流	210	10.5.3 多个图像	244
9.1.3 理解 I/O 超类	211	10.6 示例 5: VoteDialog	244
9.2 使用流	212	10.6.1 单选按钮	245
9.2.1 如何使用文件流	213	10.6.2 对话框	246
9.2.2 如何使用管道流	215	10.7 布局管理	249
9.2.3 如何封装流	216	10.8 线程和 Swing	250
9.2.4 如何联结文件	217	10.8.1 单线程规则	251
9.2.5 操作过滤器流	218	10.8.2 如何在事件调度线程中执行代码	252
9.3 对象的串行化	224	10.9 Swing 组件的可视索引	253
9.3.1 对对象进行串行化	224	10.10 小结	258
9.3.2 为类提供对象串行化	225	10.11 问题与练习	258
9.4 操作随机访问文件	227	10.11.1 问题	258
9.4.1 使用随机访问文件	228	10.11.2 练习	259
9.4.2 为随机访问文件编写过滤器	229	10.11.3 解答	259
9.5 其他问题	231	10.12 代码样例	259
9.6 读和写的小结	231	附录 A 常见问题及其解决方案	261
9.7 问题与练习	231	附录 B 用于 Internet 的 applet	272
9.7.1 问题	231	附录 C 集合	312
9.7.2 练习	232	附录 D 被废弃的线程方法	352
9.7.3 解答	232	附录 E 参考信息	357

第 1 章 起 步

本章将简单介绍 Java 技术。首先，我们解释什么是 Java 平台以及它可以做什么。接下来，一步步地讲解如何在 Win32、UNIX/Linux 或 MacOS 平台^①上编译和运行两个简单的程序。之后，我们看看这两个程序的代码，以便了解它们是如何工作的。本章结尾的问题和练习可以测试和扩展你的知识，最后是一个表格给出本章中使用的代码的下载说明。

Sun Microsystems 公司提供的 Software Development Kit (SDK) 包括运行和编译程序所需的最小的工具集。严谨的开发人员应该使用一个专业的集成开发环境 (Integrated Development Environment, IDE)^②。IDE 的列表参见附录 E。

1.1 关于 Java 技术

到处都在谈论 Java 技术，但是 Java 技术到底是什么？下面两节解释 Java 编程语言和 Java 平台。

1.1.1 Java 编程语言

Java 编程语言是一种高级语言，它具有以下性质^③。

- 简单
- 面向对象
- 分布式
- 解释
- 稳固
- 安全
- 与体系结构无关
- 可移植
- 高性能
- 多线程
- 动态

在大多数语言中，要么编译程序，要么解释程序，才能在计算机上运行它。Java 编程语言的特殊之处在于程序既被编译又被解释。首先，使用编译器将程序翻译为一种称为 Java 字节编码 (bytecode) 的中间语言，这是由 Java 平台上的解释器解释的与平台无关的代码。解释器在计算机上分析并运行每条 Java 字节编码指令。编译只发生一次；而解释在每次执行程序时都发生。图 1-1 表示了此工作过程。

你可以将 Java 字节编码看作用于 Java 虚拟机 (Java Virtual Machine, Java VM) 的机器码指令。每个 Java 解释器，无论是开发工具还是可以运行 applet 的 Web 浏览器，都是一种 Java VM 实现。

Java 字节编码有助于使“一次编写，处处运行”成为可能。你可以在任何有 Java 编译器的平台上将你的程序编译为字节编码。字节编码可以在任何 Java VM 实现上运行。这意

① 如果你使用的平台没有在这里列出，那么怎么办？Sun Microsystems 公司维护一个用于其他平台的第三方端口列表：<http://java.sun.com/cgi-bin/java-ports.cgi>。

② 实际上，Java 2 SDK Standard Edition v.1.3 与一个 IDE 捆绑，Forte for Java Community Edition。本书的光盘中包含这个版本。

③ 这些术语都在 James Gosling 和 Henry McGilton 所著的白皮书“The Java Language Environment”中做了解释，可以在 <http://java.sun.com/docs/white/langenv/index.html> 找到此白皮书。

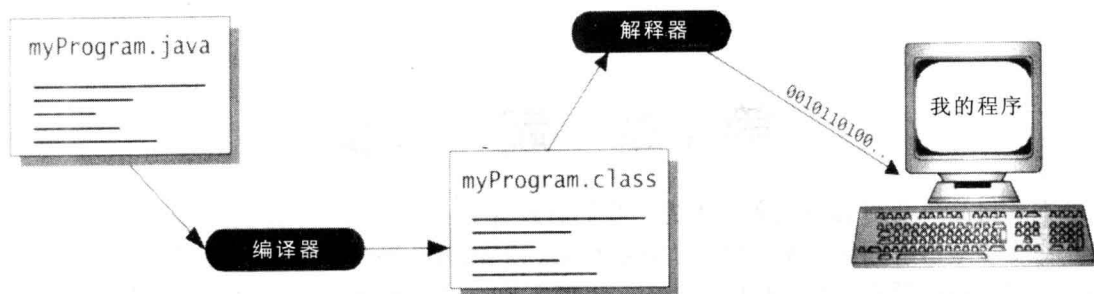


图 1-1 用 Java 编程语言写的程序先被编译，然后被解释

意味着只要计算机上有一个 Java VM，那么用 Java 编程语言写的同样的程序就能够在 Windows 2000、Solaris 工作站或 iMac 上运行，如图 1-2 所示。

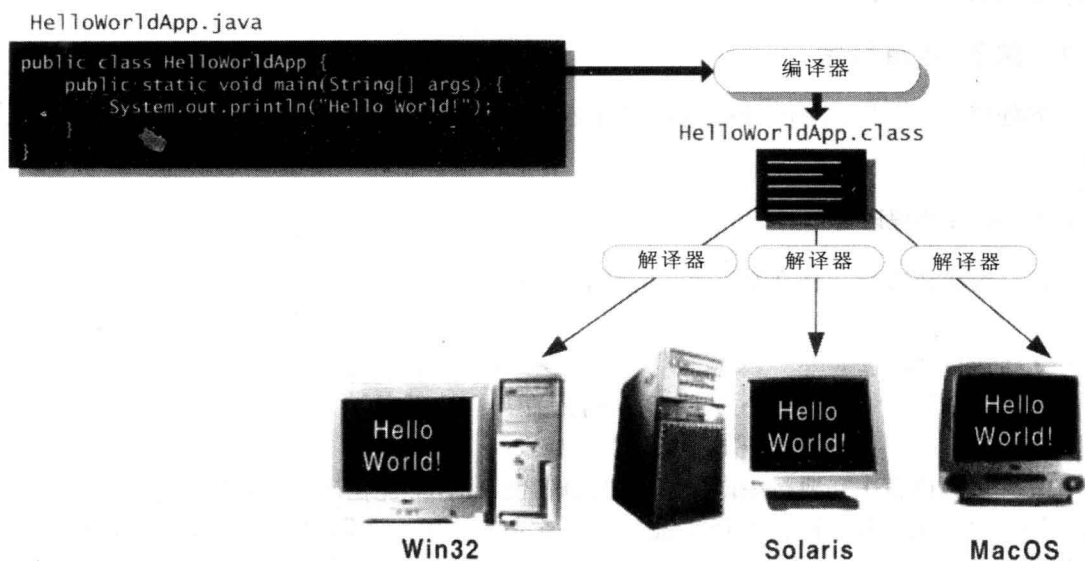


图 1-2 编写一次程序，它可以在几乎任何平台上运行

1.1.2 Java 平台

平台(platform)是程序在其中运行的硬件或软件环境。我们已经提到了一些最流行的平台,比如 Windows 2000、Linux、Solaris 和 MacOS。大多数平台可以被描述为操作系统和硬件的组合。Java 平台与大多数其他平台的不同之处在于,它是一种运行在其他硬件平台上的纯软件平台。

Java 平台有两个组件:

- Java 虚拟机(Java Virtual Machine, Java VM)。
- Java 应用程序编程接口(Java Application Programming Interface, Java API)。

我们已经介绍了 Java VM。它是 Java 平台的基础,已经被移植到了各种硬件平台上。

Java API 是预先建立的软件组件的大型集合,它们提供许多有用的功能,比如图形用户界面(graphical user interface, GUI)部件。Java API 被分组为相关类和接口的库,这些库称为包

(package)。下一节将说明 Java API 中的一些包提供的功能。

图 1-3 描述了一个在 Java 平台上运行的程序。

Java API 和虚拟机将程序与硬件隔离开。

本机代码是编译之后的在特定硬件平台上运行的代码。作为一种与平台无关的环境,Java 平台会比本机代码慢一点儿。但是,智能编译器、良好调整的解释器和即时(just-in-time)字节编码编译器可以将性能提高到接近本机代码的程度,而不影响它的可移植性。

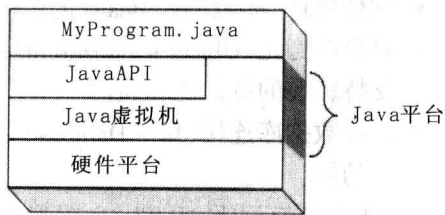


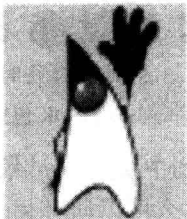
图 1-3 Java API 和 Java VM 将程序与硬件隔离开

1.1.3 Java 技术可以做什么

用 Java 编程语言写的程序最常见的类型是 applet 和应用程序。如果你曾经在 Web 上冲浪,那么可能熟悉 applet。applet 是一种遵从某种约定、可以在支持 Java 的浏览器中运行的程序。要想看看运行中的 applet,请访问本书的在线版本中的以下页面:

<http://java.sun.com/docs/books/tutorial/getStarted/index.html>

在这里,你可以看到一个动画,Java 平台的吉祥物 Duke 正在向你挥手:



但是,Java 编程语言不只用于为 Web 编写可爱、有趣的 applet。通用而高级的 Java 编程语言还是一种强大的软件平台。通过使用丰富的 API,可以编写许多类型的程序。

应用程序是直接在 Java 平台上运行的独立程序。一种称为服务器(server)的特殊应用程序为网络上的客户服务。服务器的例子有 Web 服务器、邮件服务器和打印服务器。

另一种特殊的程序是 servlet。servlet 差不多可以被看作在服务器端运行的 applet。Java servlet 的流行用途是建立交互式的 Web 应用程序,以替代 CGI 脚本。servlet 与 applet 的类似之处在于它们都是应用程序的运行时扩展。但是 servlet 不是在浏览器中运行,而是在 Java Web 服务器中运行,用于配置或调整服务器。

API 如何支持所有这些程序类型呢?它是通过提供大量功能的软件组件包来实现的。Java 平台的每种完整实现都提供了以下功能:

- 基本内容:对象、字符串、线程、数字、输入和输出、数据结构、系统属性、日期和时间等等。
- Applets:由 Java applet 使用的一组约定。
- 连网:URL、传输控制协议(Transmission Control Protocol, TCP)、用户数据报协议(User Datagram Protocol, UDP)套接字和 IP(Internet Protocol)地址。
- 国际化:帮助编写可以针对用户的位置进行本地化的程序。程序可以根据特定的地区

自动调整并以适当的语言显示。

- 安全:包括低层和高层安全性,如电子签名、公共密钥和私有密钥管理、访问控制和证书。
- 软件组件:称为 Java Beans 的组件可以插入现有的组件体系结构。
- 对象串行化(Object serialization):通过远程方法调用(Remote Method Invocation, RMI)支持轻型的持久性和通信。
- Java 数据库连接(Java Database Connectivity, JDBC):提供对各种关系型数据库的统一的访问。

Java 平台还有用于 2D 和 3D 图形、可访问性、服务器、协作、电话、语音、动画等的 API。

图 1-4 说明了 Java 2 SDK 中包含的内容。

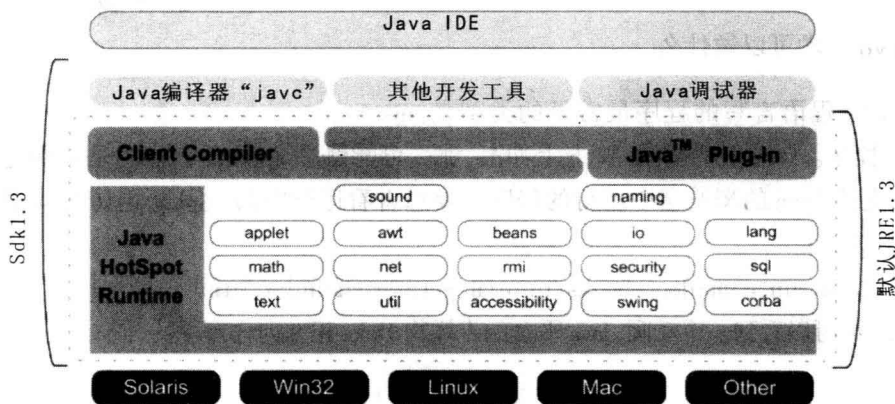


图 1-4 Java 2 SDK, Standard Edition v.1.3。Java 2 运行时环境(Java 2 Runtime Environment, JRE)包含 Java VM、Java 平台核心类和支持文件。Java 2 SDK 包括 JRE 和开发工具,比如编译器和调试器

本书讨论的 Java 编程语言是初级到中级程序员最经常使用的一部分核心 API。如果你需要的其他信息没有包含在本书中,那么可以研究另外两本书:《The JFC Swing Tutorial》和《The Java Tutorial Continued》。这两本书的内容可以在本书附带的光盘中找到,也可以从在线指南中得到:

<http://java.sun.com/docs/books/tutorial/index.html>

1.2 Java 技术将如何改变我们的生活

我们不能保证如果你学习了 Java 编程语言就能够得到荣誉、财富,甚至工作。但是它很可能使你的程序更好,而且你需要付出的努力比使用其他语言时少。我们相信 Java 编程语言会给你带来以下好处:

- 入门迅速:尽管 Java 编程语言是一种强大的面向对象的语言,但是它容易学,特别是对于熟悉 C 或 C++ 的程序员。
- 编写更少的代码:程序度量(比如类的数量、方法的数量等等)的对比说明,用 Java 编程语言写的程序比用 C++ 写的相同程序小四倍。
- 编写更好的代码:Java 编程语言鼓励使用更好的编程方法,而且它的垃圾收集功能有助于避免内存泄露。它的面向对象性质、JavaBeans 组件体系结构以及丰富的 API 使你可以重新使用其他人的代码并减少 bug。