滇一西一学一术一文一丛






IBM Robocode 人工智能机器人研究

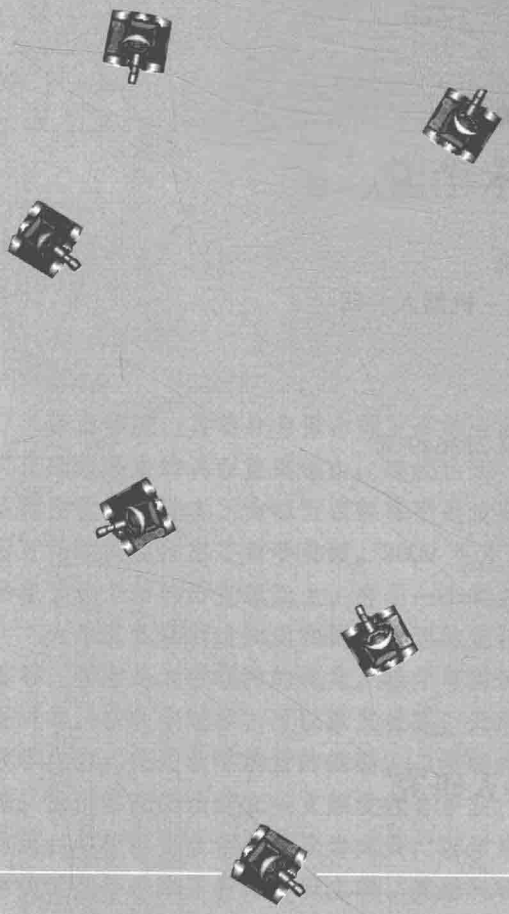
本书从IBM Robocode 一个半成品的游戏平台入手，通过对游戏平台底层的研究，挖掘出它的工作原理、规则、参数，撰写了IBM Robocode 机器人的开发设计思路、过程，设计、开发出一些人工智能游戏算法——智能机器人瞄准、移动、躲避等，总结出一套用游戏学计算机编程的教学方法。

本书分为10章和1个附录，从IBM Robocode 人工智能机器人的环境、概要，简单机器人、高级机器人、智能机器人设计过程，智能机器代码重构、瞄准算法、移动算法、躲避算法，到IBM Robocode 人工智能机器人的一些其他知识，最后附录部分是IBM Robocode 人工智能机器人的一些API 的解释。

王善发 著

云南大学出版社
Yunnan University Press

滇西学术文丛



IBM Robocode
人工智能机器人研究

王善发 著

云南大学出版社
Yunnan University Press

图书在版编目（CIP）数据

IBM Robocode 人工智能机器人研究 / 王善发著. --
昆明：云南大学出版社，2013
（滇西学术文丛. 第 7 辑）
ISBN 978 - 7 - 5482 - 1689 - 6

Ⅰ. ①Ⅰ… Ⅱ. ①王… Ⅲ. ①人工智能 - 机器人 - 研究 Ⅳ. ①TP242

中国版本图书馆 CIP 数据核字（2013）第 213649 号

IBM Robocode 人工智能机器人研究

王善发 著

策划编辑：徐 曼
责任编辑：石 可
封面设计：刘 雨
出版发行：云南大学出版社
印　　装：昆明市研汇印刷有限责任公司
开　　本：787mm × 1092mm 1/16
印　　张：16.5
字　　数：412 千
版　　次：2013 年 9 月第 1 版
印　　次：2013 年 9 月第 1 次印刷
书　　号：ISBN 978 - 7 - 5482 - 1689 - 6
定　　价：40.00 元

社　　址：云南省昆明市翠湖北路 2 号云南大学英华园内
邮　　编：650091
电　　话：0871 - 65033244 65031071
网　　址：http：//www. ynup. com
E - mail：market@ ynup. com

本书由“保山学院学术出版基金”资助出版

《滇西学术文丛》总序

蒋永文

保山学院，其前身为保山师范高等专科学校，地处气候宜人、风景秀丽、历史悠久、文化底蕴厚重的滇西重镇保山，建校已30余年，为滇西区域培养了上万名中小学教师和各行业优秀建设者，为祖国西部特别是边疆少数民族地区教育事业的繁荣昌盛与经济社会的可持续发展作出了重要贡献。2009年4月，学校被教育部批准为保山学院。这使我们站在了一个新的历史起点上，有了一个更为广阔的发展空间。

大学肩负着创造知识和传播知识的重任。学术是支撑大学的精髓，学科是构筑大学的基石，学者是大学精神的化身。教学与科研相统一是大学的基本理念。科研和教学是彼此促进的，在教学过程，可以激发灵感，开阔思路，发现研究课题。而研究成果又可以丰富教学内容，促进教学质量的提高，二者相得益彰。为了给滇西地区提供更好的高等教育资源，保山学院必须建立一支热爱教育事业，业务素质过硬，高水平、高质量的教师队伍，为此，学校以重点学科建设为龙头，以形成科研特色，增强科研实力，提高效益为目标。学校近几年采取了资助科研立项、奖励科研成果、出版学术论文等措施，不断提高广大教师的教学水平和科研水平，已收到了较好的效果。为更好地为广大教师提供出版学术论著的园地，学校决定继续出版《滇西学术文丛》，出版学术水平较高的著作。相信《滇西学术文丛》的出版，一定会对保山学院科学研究的深入、学科建设和学科带头人、骨干教师的培养产生积极的影响。

辽阔的天空，允许大鹏展翅翱翔，也允许小鸟上下蓬蒿；广袤的大地，允许参天大树生长，也允许无名小草成长。我们是小鸟，我们是小草，这套丛书，远非成熟完美，作者水平也还需要不断提高。我们期待着批评和指教。相信我们会做得越来越好。

2013年5月

前　言

是否有一种软件环境能让大家边学边玩？通过玩游戏实现我们征服程序世界的梦想？IBM Robocode 人工智能机器人就具有这些功能。它以竞技游戏、比赛、任务驱动、寓教于乐的方式进行学习和思维训练。打破了传统教育的模式，利用游戏的互动性和操作性，让使用者在玩游戏、学编程、学设计中运用知识，提高能力。IBM Robocode 把游戏风潮变成了教学工具，为全世界的开发者实现这个愿望。

IBM Robocode 为使用者提供了一个虚拟机器人的制作环境和竞技环境。首先，使用者通过利用数学、物理等相关学科知识自己编程建立一个虚拟机器人。虚拟机器人通过纯代码编写方式编写机器人控制代码来体现。编写好的机器人控制代码经过编译以后，就可以在虚拟机器人运行平台竞技环境下和其他的机器人战斗。在这个过程中，为了取得胜利，你需要不断学习程序设计和机器人策略算法，通过学到的程序不断完善自己的机器人。在学习的同时感到了快乐，可是在娱乐的同时你发现自己目前的知识储备不能给你更高的支持，所以为了得到更多的快乐，你又需要投入学习，让你在“玩—学习”的循环过程中不断提高自己程序设计和策略算法设计的水平。

学语言必须做项目，在“做中学”基本上已经是大家学习编程时的一种共识，而编写一个具有智能的机器人就是做一个比较大的项目。国外早已经引入“入门即项目”的创新意识教学。比如美国斯坦福大学就是用机器人教 Java 语言。本书研究的 IBM Robocode 人工智能机器人更有趣，能让学习者从解决问题的角度入手，来学习语法。另外，利用它也能研究高深的算法领域，真正的由浅入深。

IBM Robocode 人工智能机器人打破了传统计算机程序设计教学方式的无趣、枯燥。让学习者在设计自己的机器人的同时，学会 Java 编程语言，在与全世界 IBM Robocode 爱好者的积极挑战中，不断改进完善自己的机器人，在达到胜利的同时不知不觉牢牢地掌握了所用的程序设计语言，甚至包括数学、物理、人工智能等各种知识。

只用几十行代码，就能立刻创造出一个简单但完整的机器人，将它装入 IBM Robocode 引擎中，立即就能参加战斗，这就是 IBM Robocode 魅力所在。你可以不停地修改你的程序，设计新射击模型、躲避模型、移动模型，当你打败了那些示例机器人之后，你还可以在网上找到其他程序员编写的水平更高的机器人，与它们比试一下，提高自己的智能机器人编程水平。

把 IBM Robocode 人工智能机器人这个游戏接入课程教学是因为它上手很容易，也可以作为深入研究各种顶级算法的工具。本书从 IBM Robocode 一个半成品的游戏平台入手，通过对游戏平台底层的研究，挖掘出它的工作原理、规则、参数，撰写了 IBM Robocode 机器人的开发设计思路、过程，设计、开发出一些人工智能游戏算法：智能机器人瞄准、移动、躲避等算法，总结出一套用游戏学计算机编程的教学方法。

本书分为10章和1个附录，主要包括IBM Robocode人工智能机器人的环境、概要，简单机器人、高级机器人、智能机器人设计过程，智能机器人代码重构、瞄准算法、移动算法、躲避算法，IBM Robocode人工智能机器人的一些其他知识等。

本书为云南省教育厅2013年科研基金项目“IBM Robocode人工智能机器人研究”（项目编号：2013Y051）的成果之一。

王善发

2013年5月

目　录

第1章 IBM Robocode 人工智能机器人的环境——Java 概述

IBM Robocode 人工智能机器人是使用 Java 编写的。要进行 IBM Robocode 人工智能机器人坦克游戏开发与运行，首先要安装 Java 语言的 JDK 开发与运行环境，接着再安装 IBM Robocode 人工智能机器人开发与运行环境（IBM Robocode 人工智能机器人引擎），因此，在讲述 IBM Robocode 之前首先要概要讲述 Java 基础。

Java 是当今世界最重要、也是使用最广泛的计算机语言之一。而且，在多年之前它就已经拥有这一荣誉。与其他一些计算机语言随着时间的流逝影响逐渐减弱不同，Java 随着时间的推移反而变得更加强大。从首次发布开始，Java 就跃到了 Internet 编程的前沿。后续的每一个版本都进一步巩固了这一地位。如今，Java 依然是开发基于 Web 的应用程序的最佳选择。此外，Java 还是智能手机变革的推手，Android 编程采用的就是 Java 语言。在现实世界中，很多应用都是使用 Java 开发的，Java 真的非常重要！

Java 程序设计语言是目前在实际软件项目开发中主流的编程语言之一，它具有与平台无关性以及网络编程等特点。随着当今信息技术的飞速发展和 Internet 的普及，Java 语言作为一种简单易学的面向对象语言受到了越来越多的软件开发人员的青睐。从 Java 程序设计语言推出至今，在十几年的时间里，其不断更新和发展，在越来越多的领域中发挥着重要的作用。

Java 作为目前的主流面向对象程序设计语言之一，因其面向对象、跨平台、支持多线程和分布式等特点在 Web 应用程序开发、网络编程、手机游戏等各个方面都得到了广泛应用，并且受到了越来越多程序设计者的青睐。

1.1 Java 语言产生及其特点

在经历了以大型机为代表的集中计算模式和以 PC 为代表的分散计算模式之后，计算机网络的出现使得计算模式进入了网络计算时代。网络计算模式的一个特点是计算机是异构的，即计算机的类型和运行的操作系统可能各不相同，各种电子设备使用的嵌入式系统的硬件体系和操作系统也是不一样的。网络计算模式的另一个特点是代码可以通过网络在各种计算机之间迁移，这就迫切需要一种跨平台的编程语言，使得用它编写的程序在网络中的各种计算机上能够正常运行，Java 语言就在这种需求下应运而生。

1.1.1 Java 历史和演变

为了完全地理解 Java，必须理解创建它的背后原因、促使其成型的动力以及它所继承的思想。与以前所有成功的计算机语言一样，Java 是一个混合物，它是由大量继承自其他

编程语言的特性中的最优元素，以及 Java 为完成自身特殊使命所必需的创新性概念联合组成的。本章将介绍 Java 出现的背景、创建 Java 的原因、是什么原因使 Java 如此重要，以及多年来 Java 的演变过程。

尽管 Java 已经变得与 Internet 的在线环境密不可分，但是 Java 首先并且首要的仍然是一种语言，记住这一点很重要。计算机语言的创新与发展取决于以下两个基本原因：

- 适应环境和用途的变化
- 实现编程艺术的完善与提高

在后面将会看到，Java 的发展就是由这两个因素驱动的，而且这两个因素的驱动程度几乎相同。

Java 与 C++ 相关，C++ 是 C 的直接后代。Java 的大量特性就是从这两种语言继承的。Java 继承了 C 的语法，许多面向对象特性则受 C++ 的影响。实际上，Java 的一些特性来自它的前辈，或受其影响。而且，Java 的创建基于过去几十年来计算机编程语言的改良和发展过程。由于这些原因，本节将回顾促使 Java 产生的一系列事件和动力。我们将会看到，语言设计的每次革新，都是为了解决之前语言不能解决的基本问题，Java 也不例外。

1. 现代编程语言的诞生：C 语言

C 语言的诞生震惊了计算机界。不应当低估它的影响，因为它从根本上改变了编程的方式和思路。C 语言的产生是人们对结构化、高效率（在创建系统程序时能够取代汇编代码）高级语言需求的直接结果。正如我们知道的，当设计一种计算机语言时，经常需要进行取舍，例如权衡下面这些因素：

- 易用性与功能
- 安全性与效率
- 稳定性与可扩展性

在 C 语言以前，程序员通常需要在品质不同的各种计算机语言之间进行选择。例如，尽管可以使用 FORTRAN 为科学计算应用程序编写出相当高效的程序，但是对于编写系统代码它不是很好。再比如，尽管 BASIC 易于学习，但它的功能不是很强大，并且由于缺少结构化设计，很让人怀疑是否可以将其应用于大型程序。汇编语言可以生成非常高效的代码，但是它不易于学习，使用效率低。而且，调试汇编代码可能相当困难。

另外一个复杂的问题是，早期的计算机语言，例如 BASIC、COBOL 以及 FORTRAIN，没有遵循结构化设计原则。反而，它们依赖于 GOTO 作为程序控制的主要手段。因此，使用这些语言更容易编写出“意大利面条式的代码”——大量混乱的跳转语句和条件分支语句，使程序实际上很难理解。而类似 Pascal 的语言虽然是结构化的，但是它们不是针对高效率而设计的，并且没有提供使它们能够应用于大范围编程领域所需要的特性（特别是在确定标准 Pascal 语言时，实际上并没有考虑将它用于系统级代码）。

因此，在 C 语言出现以前，没有哪种语言能够解决这些矛盾。但是对这样一种语言的需要是迫切的。到了 20 世纪 70 年代早期，计算机革命开始出现，并且对软件的需求快速增长，超出了程序员的能力。为了尝试创建出一种更好的计算机语言，学术界为此付出了大量的努力。但是，促使 C 语言诞生的第二个因素，也许是最重要的因素正在出现。计算机硬件最终变得非常普遍，达到了发生变化的临界状态。计算机不再被锁起来，程序员第一次可以真正地随意使用他们的计算机，从而可以随意地进行尝试，并且程序员还可

以开始创建他们自己的工具。在 C 语言诞生前夕，计算机语言向前飞跃发展的舞台已经具备。

C 语言是由 Dennis Ritchie 在运行 UNIX 操作系统的 DEC PDP－11 机器上发明并首次实现的，它是老式 BCPL 语言不断发展的结果，BCPL 语言是由 Martin Richards 开发。BCPL 语言对 Ken Thompson 发明的 B 语言产生了影响，B 语言导致了在 20 世纪 70 年代对 C 语言的开发。多年来，由 UNIX 操作系统提供的标准成为 C 语言事实上的标准，并且在 Brian Kernighan 和 Dennis Ritchie 编写的 The C Programming Language（Prentice－Hall，1978）一书中得到了描述。1989 年 12 月，当美国国家标准学会（American National Standards Institute，ANSI）制定的 C 语言标准被采纳后，C 语言被正式标准化。

C 语言的诞生被许多人认为是现代计算机语言时代开始的标志，它成功地综合了早期计算机语言曾经非常麻烦的矛盾特性，从而使 C 语言成为功能强大、高效率、结构化的语言，并且相对容易学习。C 语言还具有另外一个几乎是在无形中产生的特性：它是程序员的语言。在 C 语言诞生之前，计算机语言通常要么是作为学术实验而设计，要么是由官方委员会设计。而 C 语言不同，它是由真正从事编程工作的程序员设计、实现和开发，反映了程序员进行实际编程工作的方法。C 语言的特性经过实际使用该语言的人们不断提炼、测试、思考、再思考，成为广大程序员最喜欢使用的语言。确实，C 语言迅速吸引了许多狂热的追随者。于是，C 语言被程序员广泛采用并被迅速接受。总之，C 语言是由程序员设计并由他们使用的一种语言。正如即将看到的 Java 继承了这一传统。

2. C 语言下一个阶段：C++

从 20 世纪 70 年代晚期到 80 年代早期，C 语言成为主要的计算机编程语言，并且在今天仍然被广泛使用。既然 C 语言是一种成功并且有用的语言，有人可能会好奇为什么还需要其他语言呢。答案是复杂性。纵观编程的历史，正是程序复杂性的不断增加驱动了管理复杂性的更好方式的需要。C++ 是对这一需求的响应。

自从发明计算机以来，编程方式发生了很大的变化。例如，计算机刚出现时，编程是通过面板用手工打孔的方法输入二进制机器指令实现。对于那些只有几百行指令的程序，这种方法可以工作。随着程序的增长，引入了汇编语言，通过使用机器指令的符号化表示，程序员可以编写更大、更复杂的程序。随着程序的不断增长，出现了高级语言，为程序员提供了更多用于处理复杂性的工具。

当然，第一种广泛使用的高级语言是 FORTRAN。虽然 FORTRAN 迈出了令人印象深刻的第一步，但是它很难开发出条理清晰且易于理解的程序。20 世纪 60 年代诞生了结构化编程（structured programming）。这种编程方法被 C 语言这类语言采用。通过使用结构化编程语言，程序员第一次能够比较容易地编写出相对复杂的程序。但是，即使是使用结构化编程方法，一旦项目达到一定的规模，它的复杂性就会超出程序能够管理的范围。到了 20 世纪 80 年代早期，许多项目超出了结构化方法的极限。为了解决这一问题，发明了一种新的编程方法，称为面向对象编程（Object－Oriented Programming，简称 OOP）。OOP 是一种编程方法论，通过使用继承、封装和多态来帮助组织复杂的程序。

通过分析可以看出，尽管 C 语言是世界上最伟大的编程语言之一，但是它处理复杂性的能力也是有一定限度的。一旦程序规模超过特定的临界点，就会变得非常复杂以至于难以从整体上进行把握。虽然根据程序的自身特征以及程序员的不同，发生这种情况的准

确界限会有所不同，但总是存在这样一个门槛，一旦超过这个门槛，程序就变得难以管理。C++添加了能够突破这一界限的特征，允许程序员理解并管理更大的程序。

C++语言是由 Bjarne Stroustrup 于 1979 年发明，当时他在位于美国新泽西州 Murray Hill 的 Bell 实验室工作。Stroustrup 最初将这种新语言称为“带类的 C”。但是，在 1983 年他将名称修改为 C++。C++通过添加面向对象特征对 C 语言进行了扩展。因为 C++构建于 C 语言的基础之上，所以它包含了 C 语言的全部特征、特性以及优点。这是 C++作为一种语言能够成功的关键原因。发明 C++语言不是试图创建一种全新的编程语言，相反，它是对已经取得极大成功的 C 语言的改进。

3. Java 出现的时机

到了 20 世纪 80 年代末 90 年代初，使用面向对象编程的 C++占据了编程语言的主导地位。确实，程序员好像一度找到了完美的语言。因为 C++既支持面向对象编程模式，又具有 C 语言的高效率以及风格优点，它确实是一种可以用于创建各种程序的语言。然而，就像过去一样，推动计算机语言向前演变的力量又一次在酝酿。在短短的几年中，万维网（World Wide Web）和 Internet 达到了临界规模。这一事件又将会促成编程的另一场革命。

1.1.2 Java 诞生

Java 是由 James Gosling、Patrick Naughton、Chris Warth、Ed Frank 和 Mike Sheridan 于 1991 年在 Sun 公司构想出来的。开发第一个版本花费了 18 个月。这种语言最初称为 Oak，在 1995 年被命名为 Java。从 1992 年秋 Oak 的最初实现到 1995 年春 Java 语言的公开发布，许多人对 Java 的设计和改进做出了贡献。Bill Joy、Arthur van Hoff、Jonathan Payne、Frank Yellin 和 Tim Lindholm 是主要贡献者，他们的奉献使 Java 的最初原型逐渐成熟。

让人惊奇的是，Java 的最初推动力不是 Internet！相反，主要动机是对平台独立（即体系结构中立）语言的需要，这种语言可用于开发能够嵌入各种消费类电子设备（例如微波炉、遥控器）的软件。大家可能已经猜到了，许多不同类型的 CPU 被用作控制器。使用 C 和 C++语言（以及大部分其他语言）的麻烦是，它们被设计为针对特定的目标进行编译。尽管能够为各种类型的 CPU 编译 C++程序，但是这需要一个完整的以该 CPU 为目标的 C++编译器。问题是创建编译器很耗费时间，所以需要一种更容易并且更经济的解决方案。在寻找这样一种方案的尝试过程中，Gosling 和其他人一起开始开发一种可移植的、平台独立的语言，可以使用这种语言生成在不同环境下运行于各种 CPU 之上的代码。他们的努力最终导致了 Java 的出现。

在 Java 的细节被开发出来的同时，第二个并且最终也更加重要的因素出现了，它在 Java 的未来中扮演了关键的角色，第二个动力当然是万维网。假如 Web 的形成和 Java 的出现不在同一时间，那么 Java 虽然仍会有用，但可能只是一种用于为消费类电子产品编写代码的没有名气的语言。然而，随着万维网的出现，Java 被推到计算机语言设计的最前沿，因为 Web 也需要可移植的程序。

大部分程序员在职业生涯的早期就知道，可移植程序既让人向往又让人逃避。尽管人们对创建高效、可移植（平台独立）程序的探索，几乎和编程自身的历史一样久远，但它总是让位于其他更为紧迫的问题。此外，因为在那时计算机界已经被 Intel、Macintosh

和 UNIX 这三个竞争阵营垄断，大多数程序员都在其中的某个领域内工作，所以对可移植性编码的迫切需求降低了。但是，随着 Internet 和 Web 的出现，古老的可移植性问题又出现了。毕竟，Internet 是由各种各样的、分布式的系统构成，这些系统使用各种类型的计算机、操作系统和 CPU。尽管许多类型的平台都依附于 Internet，但是用户仍然希望他们能够运行相同的程序。这曾经是一个令人烦恼、但是优先级较低的问题，现在已经变成了必须解决的问题。

在为嵌入式控制器编写代码时经常遇到的可移植性问题，在尝试为 Internet 编写代码的过程中也出现了。到了 1993 年，这个问题对于 Java 设计小组的成员而言已经变得很明显了。实际上，最初针对解决小范围问题而设计的 Java，也可以应用于更大范围的 Internet。这一认识导致 Java 的关注点由消费类电子产品转移到了 Internet 编程。因此，虽然对体系结构中立的编程语言的需求提供了最初的思想火花，但最终是 Internet 成就了 Java 的成功。

正如在前面提到的，Java 从 C 和 C++ 继承了许多特性，这是有意为之。Java 设计人员清楚，使用与 C 语言类似的语法以及模仿 C++ 面向对象特性，可以使 Java 语言对于众多经验丰富的 C/C++ 程序员更具吸引力。除了表面类似外，Java 还借鉴了帮助 C 和 C++ 成功的其他一些特性。首先，Java 的设计、测试和不断改进是由真正从事编程工作的人员完成的，它是扎根于设计人员的需要和经验的一种语言，因此 Java 是程序员的语言。其次，Java 结构紧凑并且逻辑上协调一致。最后，除了 Internet 强加的那些约束外，Java 为程序员提供了完全的控制权，如果程序编写得好，程序本身就能反映出来；如果程序编写得不好，程序本身也能反映出来。因为存在这样的区别，所以 Java 不是一种用于培训的语言，而是一种针对专业程序员的语言。

因为 Java 与 C++ 之间的相似性，可能会简单地将 Java 看作“Internet 版的 C++”。但是，如果这么认为，将会是很大的错误。Java 无论是在实践上还是在理论上都与 C++ 有着很大的区别。虽然 Java 深受 C++ 的影响，但它不是 C++ 的增强版。例如，Java 与 C++ 既不向上兼容，也不向下兼容。当然，Java 与 C++ 之间的相似性还是很明显的。并且，如果是一位 C++ 程序员的话，会感觉 Java 很熟悉。另外一点：设计 Java 的目的并不是取代 C++。Java 是针对解决特定的一系列问题而设计的。Java 和 C++ 将会长期共存。

正如前面提到的，计算机语言的发展取决于两个因素：适应环境的变化以及实现编程艺术的提高。促使 Java 发展的环境变化是对平台独立程序的需求，Internet 上的分布式系统天生就需要平台独立的程序。同时，Java 也体现了编程方式的变化。例如，Java 增强并改进了 C++ 使用的面向对象编程，增加了对多线程的支持，提供了简化 Internet 访问的库。总之，并不是 Java 的某个单一特征，而是整体上作为一种语言，使它如此非凡。Java 是对新出现的高度分布计算领域需求的完美响应。Java 对于 Internet 编程的意义，就如同 C 语言对系统编程一样：它们都是改变世界的革命性力量。

1.1.3　Java 与 C#

在计算机语言开发领域，人们会继续感受到 Java 的影响和力量。许多创新性的特征、结构以及概念，已经成为所有新语言的基准组成部分。

Java 影响力的最重要例子可能是 C#。C#是由 Microsoft 创建的用于支持 .NET Frame-