ORACLE

Mc Graw Hill Education

完全更新至JDK 7

Java: A Beginner's Guide, Fifth Edition

新手学Java 7编程(第5版)

马上就创建、编译和运行Java程序

[美] Herbert Schildt 著
石 磊 译

清华大学出版社

新手学 Java 7 编程

(第 5 版)

[美] Herbert Schildt 著

石 磊 译

清华大学出版社

北 京

Herbert Schildt
Java: A Beginner's Guide, Fifth Edition
EISBN: 978-0-07-160632-5

北京市版权局著作权合同登记号 图字：01-2012-2228

图书在版编目(CIP)数据

新手学 Java 7 编程(第 5 版)/(美)斯切尔特(Schildt, H.) 著；石磊 译. —北京：清华大学出版社，2012.9
书名原文：Java: A Beginner's Guide, Fifth Edition
ISBN 978-7-302-29541-9

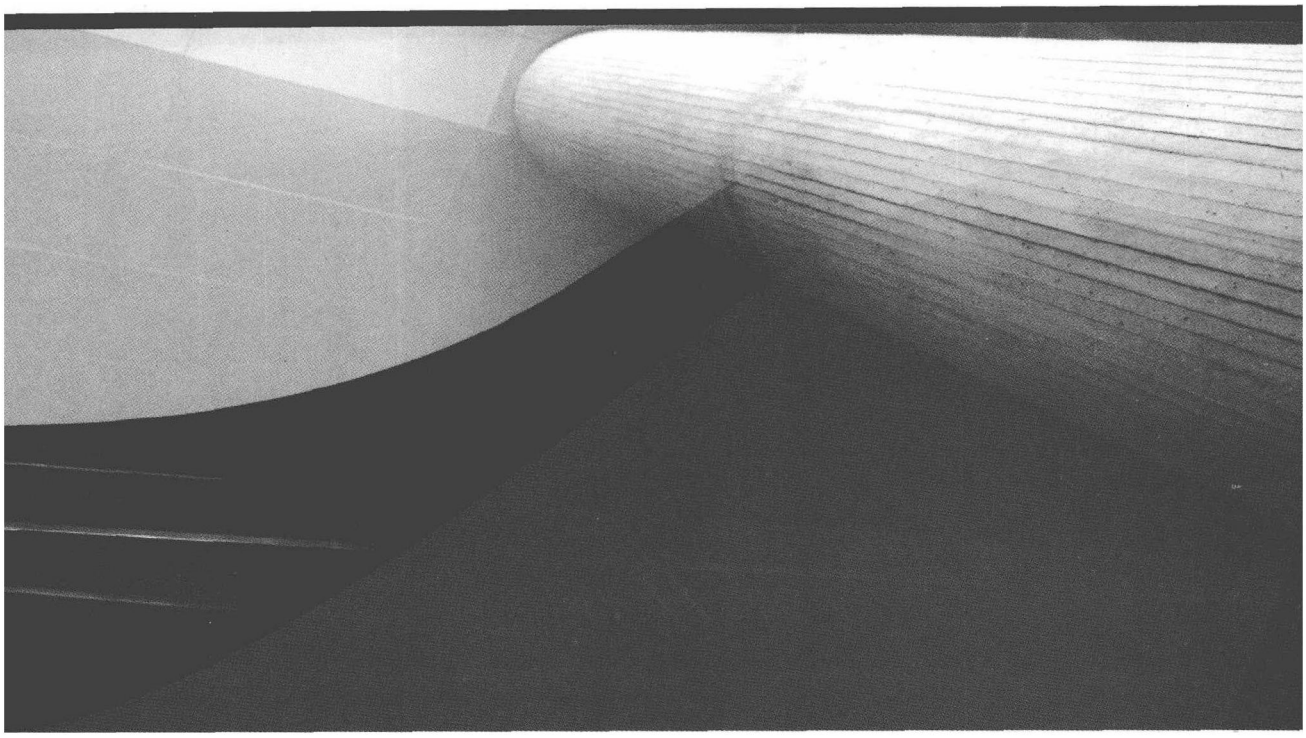
Ⅰ. ①新… Ⅱ. ①斯… ②石… Ⅲ. ①JAVA 语言—程序设计 Ⅳ. ①TP312

中国版本图书馆 CIP 数据核字(2012)第 170529 号

责任编辑：王 军 刘伟琴
装帧设计：牛艳敏
责任校对：成凤进
责任印制：王静怡

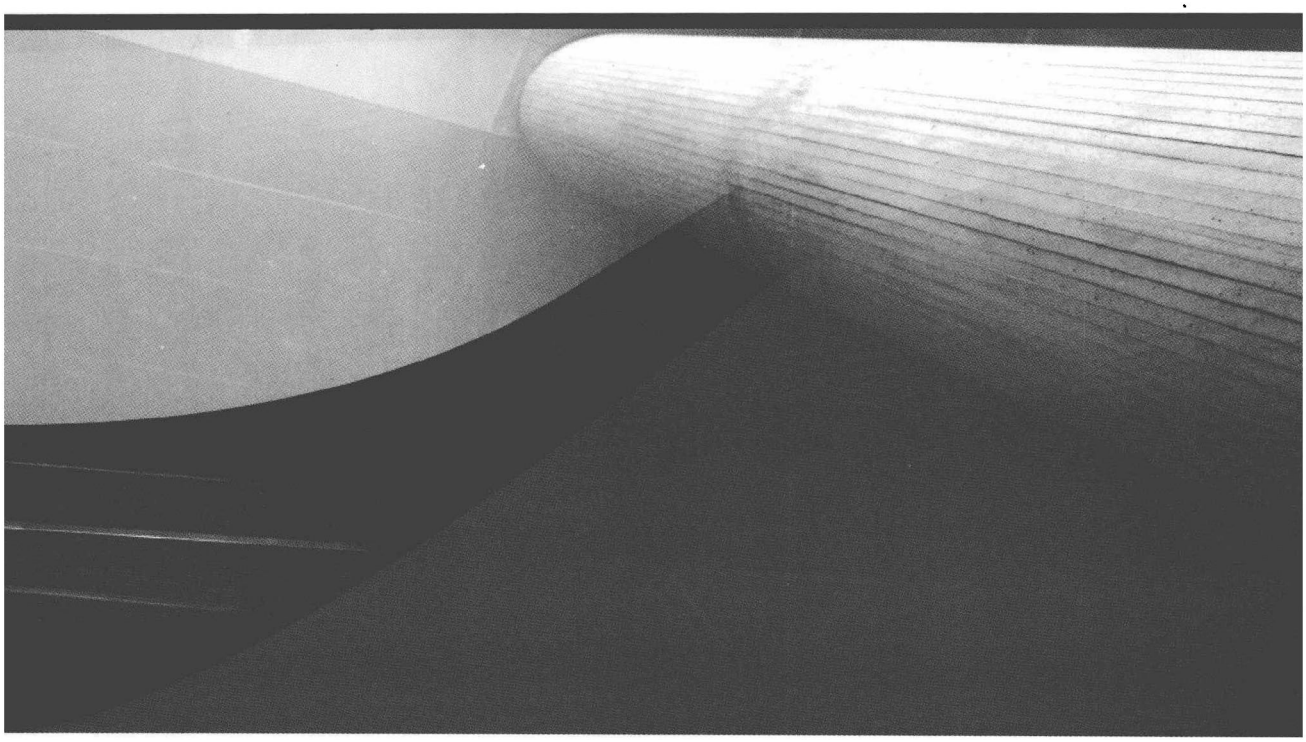
出版发行：清华大学出版社
网 址：http://www.tup.com.cn，http://www.wqbook.com
地 址：北京清华大学学研大厦 A 座 邮 编：100084
社 总 机：010-62770175 邮 购：010-62786544
投稿与读者服务：010-62776969，c-service@tup.tsinghua.edu.cn
质 量 反 馈：010-62772015，zhiliang@tup.tsinghua.edu.cn
印 装 者：北京嘉实印刷有限公司
经 销：全国新华书店
开 本：185mm×260mm **印 张**：34 **字 数**：786 千字
版 次：2012 年 9 月第 1 版 **印 次**：2012 年 9 月第1 次印刷
印 数：1～4000
定 价：59.00 元

产品编号：047197-01



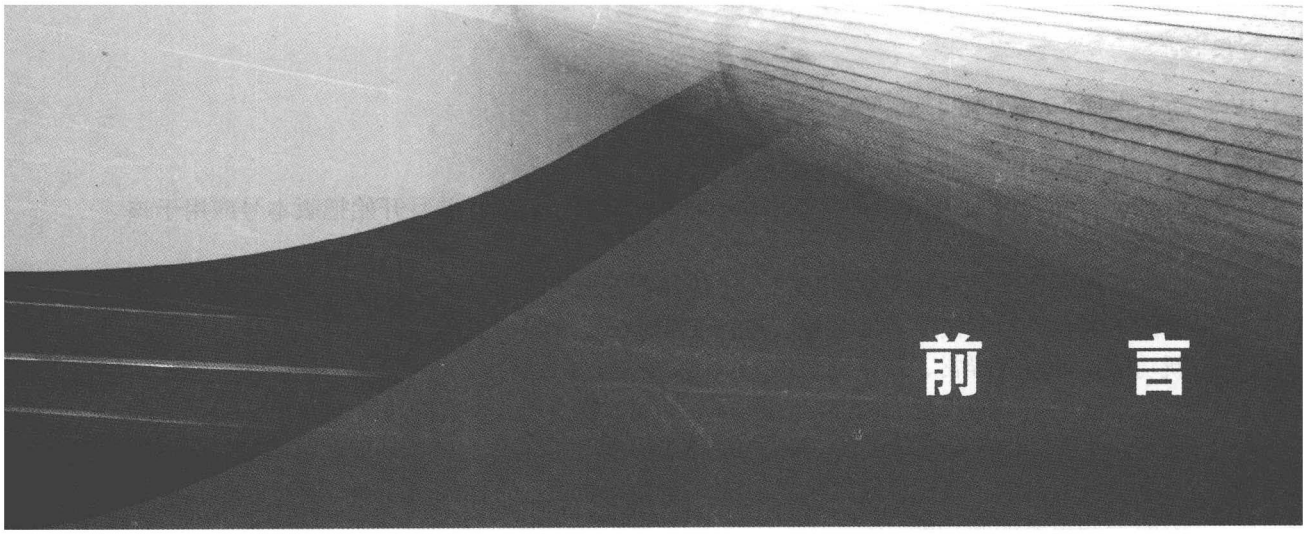
作 者 简 介

畅销书作家 Herbert Schildt 是 Java、C++、C 和 C#语言的权威。他撰写的程序设计图书在全世界销售了数百万册，并被翻译成了多种语言。他撰写了众多关于 Java 的图书，包括 *Java: The Complete Reference*、*Herb Schildt's Java Programming Cookbook*、*Swing: A Beginner's Guide* 和 *The Art of Java*。他的其他一些畅销书还包括 *C++: The Complete Reference*、*C#: The Complete Reference* 和 *C: The Complete Reference*。虽然他对计算机的方方面面都很感兴趣，但是主要关注点是计算机语言，包括编译器、解释器和机器人控制语言。他对语言的标准化也有浓厚的兴趣。Schildt 获得了伊利诺伊大学的学士和硕士学位。他的咨询中心的号码为(217)586-4683。他的网站为 www.HerbSchildt.com。



技术编辑简介

Danny Coward 博士从 1997 年开始一直对 Java 平台贡献自己的力量。他就职于 Sun 公司时，是 Java EE 组的创建者之一。他是 Java Community Process Executive Committee 的成员，并一直是 Java 平台的各个版本——Java SE、Java ME 和 Java EE——的主要贡献者。他还组建了最初的 JavaFX 团队。



前　言

作为首屈一指的 Internet 语言，Java 是最重要、也是最流行的计算机程序设计语言之一。今天要想成为一名专业的 Web 开发者，就必须掌握 Java。因此，如果你将来要从事 Web 应用程序设计工作，那么 Java 是正确的选择。因为 Android 程序设计使用了 Java，所以 Java 也成为了智能手机变革的一部分。简单来说，Java 程序设计是现代计算世界很多方面的基础。

本书正是为了帮助你学习 Java 程序设计基础而编写的。本书采用循序渐进的教学方法，安排了许多示例、自测题和编程练习。本书不需要读者具备编程经验，而是从最基本的基础知识，从如何编译并运行一个 Java 程序开始讲起。然后讨论了构成 Java 语言核心的关键字、功能和结构。还介绍了 Java 的一些最重要的高级功能，如多线程编程和泛型。本书最后介绍了 Swing。学完本书后，读者将会牢固地掌握 Java 编程精髓。

值得说明的是，本书只是你学习 Java 的起点。Java 不仅仅是一些定义语言的元素，它还包括了扩展的库和工具来帮助开发程序。要想成为顶尖的 Java 程序员，就必须掌握这些领域。读者在学习完本书之后，就有了足够的知识来继续学习 Java 的其他方面。

0.1　Java 的发展历程

只有少数几种编程语言对程序设计带来过根本性的影响。其中，Java 的影响由于迅速和广泛而格外突出。可以毫不夸张地说，1995 年 Sun 公司发布的 Java 1.0 给计算机程序设计领域带来了一场变革。这场变革迅速地把 Web 转变成了一个高度交互的环境，也给计算机语言的设计设置了一个新标准。

多年以来，Java 不断地发展、演化和修订。和其他语言加入新功能的动作迟缓不同，Java 一直站在计算机程序设计语言的前沿，部分原因是其变革的文化，部分原因是它所面对的变化。Java 已经做过或大或小的多次升级。

第一次主要的升级是 Java 1.1 版，这次升级比较大，加入了很多新的库元素，修订了处理事件的方式，重新配置了 1.0 版本的库中的许多功能。

第二个主要的版本是 Java 2，它代表 Java 的第二代，标志着 Java 的“现代化”的到来。Java 2 第一个发布的版本号是 1.2。Java 2 在第一次发布时使用 1.2 版本号看上去有些奇怪。原因在于，该号码最初指 Java 库的内部版本号，后来就泛指整个版本号了。Java 2

被 Sun 重新包装为 J2SE（Java 2 Platform Standard Edition），并且开始把版本号应用于该产品。

Java 的下一次升级是 J2SE 1.3，它是 Java 2 版本首次较大的升级。它增强了一些已有的功能，并且紧凑了开发环境。J2SE 1.4 进一步增强了 Java。该版本包括一些重要的新功能，如链式异常、基于通道的 I/O，以及 assert 关键字。

Java 的下一版本是 J2SE 5，它是 Java 的第二次变革。以前的几次 Java 升级提供的改进虽然重要，但都是增量式的，而 J2SE 5 却从该语言的作用域、功能和范围等方面提供了根本性的改进。为了帮助你理解 J2SE 5 的修改程度，下面列出了本书要讨论的 J2SE 5 中的一些主要的新功能：

- 泛型
- 自动装箱/自动拆箱
- 枚举
- 增强型 for-each 形式的 for 循环
- 可变长度实参
- 静态导入
- 注释

这些项目都是重要的升级，每一个项目都代表了 Java 语言的一个重要改进。其中，泛型、增强型 for 循环和可变长度实参引入了新的语法元素；自动装箱和自动拆箱修改了语法规则；注释增加了一种全新的编程注释方法。

这些新功能的重要性反映在使用的版本号“5”。从版本号的变化方式看，这一版本的 Java 应该是 1.5。由于新功能和变革如此之多，常规的版本号升级（从 1.4 到 1.5）无法标识实际的变化，所以 Sun 决定使用版本号 5，以强调发生了重要的改进。因此，当前的版本叫做 J2SE 5，开发工具包叫做 JDK 5。但是，为了维持和以前的一致性，Sun 决定使用 1.5 作为内部版本号，也叫做开发版本号。J2SE 5 中的“5”叫做产品版本号。

之后发布的 Java 版本是 Java SE 6，Sun 再次决定修改 Java 平台的名称，把“2”从版本号中删除了。因此，Java 平台现在的名称是 Java SE，官方产品名称是 Java Platform, Standard Edition 6，对应的 Java 开发工具包叫做 JDK 6。和 J2SE 5 一样，Java SE 6 中的“6”是指产品的版本号，内部的开发版本号是 1.6。

Java SE 6 建立在 J2SE 5 的基础之上，做了进一步的增强和改进。Java SE 6 并没有对 Java 语言本身添加较大的功能，而是增强了 API 库，添加了多个新包，改进了运行时环境。它在漫长的生命周期(Java 术语)经历了一些更新，添加了一些升级功能。总之，Java SE 6 进一步巩固了 J2SE 5 建立的领先地位。

Java 的最新版本是 Java SE 7，对应的 Java 开发工具包叫做 JDK 7。其内部版本号是 1.7。Java SE 7 是 Oracle 收购 Sun Microsystems(2009 年 4 月开始进行，2010 年 1 月完成)之后发布的第一个主版本。Java SE 7 包含许多新功能，对语言和 API 库做了许多增强，还升级了 Java 运行时系统来支持非 Java 语言。

对本书而言，Java SE 7 添加的最重要的功能是在 Project Coin 中开发的那些功能。Project Coin 的目的是确定把对 Java 语言所作的很多小改动包含到 JDK 7 中。虽然这些

新功能被称为“小”改动，但是它们的效果对代码产生了巨大的影响。事实上，对于许多程序员来说，这些改动可能是 Java SE 7 中最重要的新功能。下面列出了本书中介绍的新语言特性：

- 现在 String 可以控制一个 switch 语句。
- 二进制整数字面值。
- 在数值字面值中使用下划线。
- 新增了一种叫做 try-with-resources 的 try 语句，它支持自动资源管理(例如，当不再需要文件流时，现在可以自动关闭它们)。
- 构造泛型实例时，通过菱形运算符使用类型推断。
- 增强了异常处理，可以使用一个 catch 捕获两个或更多个异常(多重捕获)，并且对重新抛出的异常可以进行更好的类型检查。

可以看到，虽然 Project Coin 功能被视为是小改动，但是“小”这个词实在不能体现它们带来的好处。特别是，try-with-resources 语句对大量代码的编写方式会产生深远的影响。

本书的内容已经过完全更新，对新功能、更新和增强做了细致介绍，以反映 Java SE 7 的特色。

0.2 本书的组织结构

本书采用教程式的组织结构，每一章都建立在前面的基础之上。本书共分 15 章，每一章讨论一个有关 Java 的主题。本书的特色就在于它包含了许多便于读者学习的特色内容。

- **关键技能与概念** 每一章都首先介绍一些该章中要介绍的重要技能。
- **自测题** 每一章都有自测题，测试读者学习到的知识。答案在附录 A 中提供。
- **专家问答** 每一章中都穿插一些“专家问答”，以一问一答的形式介绍补充知识和要点。
- **编程练习** 每一章中都包含 1～2 个编程练习，帮助读者将学习到的知识应用到实践中去。很多这样的练习都是实际的示例，可以用作自己的程序的起点。

0.3 本书不需要读者具备编程经验

本书假定读者没有任何编程经验。如果读者没有编程经验，阅读本书是正确的选择。如果读者有过一些编程经验，在阅读本书时可以加快速度。但是要记住，Java 在几个重要的地方与其他一些流行的计算机语言不同，所以不要急于下结论。因此，即使读者是经验丰富的程序员，也仍然建议仔细阅读本书。

0.4 本书需要的软件环境

要想编译和运行本书提供的所有程序，需要获得 Oracle 提供的最新版本的 Java Developers Kit(JDK)，在撰写本书时，其版本为 JDK 7，这是 Java SE 7 使用的 JDK 版本。本书在第 1 章介绍如何获得 Java JDK。

如果读者使用早期版本的 Java，如 Java 5，也仍然可以阅读本书，只是无法编译和运行使用了 Java 新功能的程序。

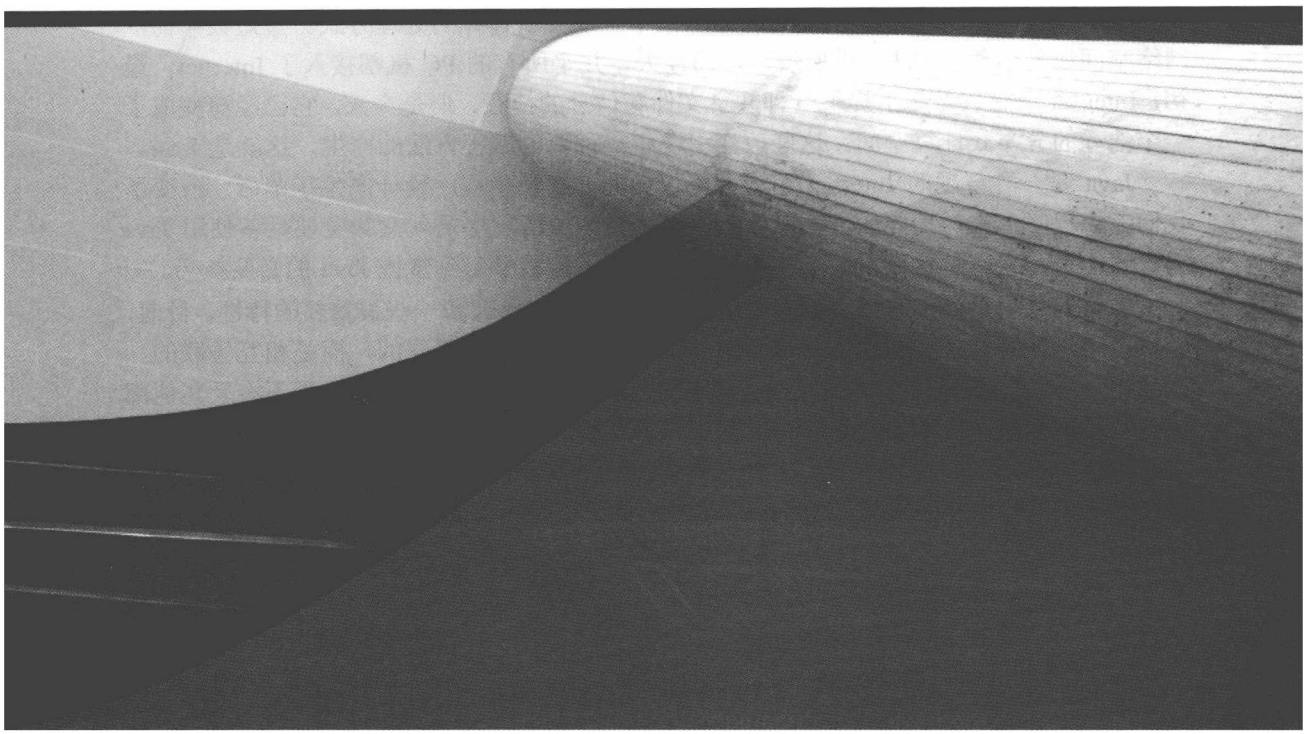
0.5 不要忘记 Web 上的代码

本书所有示例和编程项目的源代码都可以免费从 Web 网址 www.oraclepressbooks.com 获得。

0.6 特别感谢

特别感谢本书的技术编辑 Danny Coward。他提出了宝贵的建议和意见，对此我十分感谢。

目　录



第1章

Java 基础

关键技能与概念

- 了解 Java 的历史和基本原理
- 理解 Java 对 Internet 的贡献
- 理解字节码的重要性
- 了解 Java 的术语
- 理解面向对象程序设计的基本原理
- 创建、编译和运行一个简单的 Java 程序
- 使用变量
- 使用 if 和 for 控制语句
- 创建代码块
- 理解如何定位、缩进和终止语句
- 了解 Java 关键字
- 理解 Java 标识符的规则

Internet 和 World Wide Web 的兴起从根本上改变了计算的处理方式。短短数年前，网络空间还是由孤立的 PC 机所统治，而今天，几乎所有的 PC 机都接入了 Internet。最初，Internet 本身只是用于提供一种共享文件和信息的捷径，但是今天，它已经演变成了一个浩瀚的分布式计算空间。这些改变引发了一种新的编程方法的产生，这就是 Java。

Java 是一种卓越的 Internet 语言，不仅如此，它还使程序设计产生了革命，改变了我们考虑程序形式与功能的方式。今天，要成为职业的程序员就意味着要具备使用 Java 编程的能力，它就是这么重要。在本书的课程中，你将学习到掌握 Java 的必要技巧。

本章的目的是向你介绍 Java，包括它的历史、设计原理和一些最重要的特性。目前，学习程序设计语言最大的难点是语言的各部分之间不是相互孤立的，而是相互关联的。这种相互关联性在 Java 中尤为突出。事实上，只讨论 Java 的一个方面，而不涉及其他部分是非常困难的。为了帮助克服这一困难，本章对 Java 的几个特性进行了简单的概述，其中包括 Java 程序的基本形式、一些基本的控制结构和运算符。对于这些内容我们并不进行深入讨论，而只是关注一下 Java 程序共有的一些概念。

1.1 Java 的起源

驱使计算机语言革新的因素有两个：即程序设计技术的改进和计算环境的改变。Java 也不例外。在大量继承 C 和 C++的基础之上，Java 还增加了反映当前程序设计技术状态的功能与精华。针对在线环境的蓬勃发展，Java 为高度的分布式体系结构提供了流水线程序设计功能。

Java 是 1991 年由 Sun Microsystems 公司的 James Gosling、Patrick Naughton、Chris Warth、Ed Frank 和 Mike Sheridan 共同构想的成果。这个语言最初名为“Oak”，于 1995 年更名为“Java”。多少有些让人吃惊的是，设计 Java 的最初动力并不是源于 Internet，而是为了开发一种独立于平台的语言，使其能够用于创建内嵌于不同家电设备，如烤箱、微波炉和遥控器的软件。你可能想到了，不同类型的 CPU 都可以作为控制器使用。麻烦在于当时多数的计算机语言都是针对一个特定的目标设计的，例如 C++。

虽然任何类型的 CPU 或许都能编译 C++程序，然而这需要 CPU 有一个完整的 C++编译器。而开发编译器的成本很高，并且很耗时。为了找到更好的解决方法，Gosling 和其他人尝试开发一种可移植的跨平台语言，使该语言生成的代码可以在不同环境下的不同 CPU 上运行。这一努力最终导致了 Java 的诞生。

大概就在即将设计出 Java 细节的时候，另一个对 Java 的形成有更重要影响的因素出现了。这第二个动力就是 World Wide Web。如果 Web 没有在 Java 即将成型的时候问世，那么它可能会成为对消费类电子产品的程序设计而言有用但却晦涩的语言。然而随着 Web 的出现，以及 Web 对可移植语言的需求，Java 被推到了计算机语言设计的前端。

多数程序员在其工作不久就了解到可移植程序既令人期待，也让人难以捉摸。虽然在有了程序设计学科的时候就有了对创建高效可移植(平台独立)程序的需要，但是它还是让位于其他一些更为迫切的问题。Internet 和 Web 的出现使原有的可移植性问题重新摆上了桌面。因为，Internet 毕竟是一个由许多类型的计算机、操作系统和 CPU 组成的多样化的分布式空间。

曾经恼人心绪，却没那么重要的问题也就成为了要亟待解决的问题。到 1993 年，Java 设计团队的成员发现，在创建嵌入式代码时经常遇到的问题同样也出现在创建的 Internet 代码中。了解到这一点以后，Java 的重点从消费类电子产品转移到了 Internet 程序设计。因此，尽管开发独立于体系结构的程序设计语言的初衷提供了起初的星星之火，然而却是 Internet 最终促成了 Java 的燎原之势。

1.1.1　Java 与 C 和 C++的关系

Java 与 C 和 C++直接相关。Java 继承了 C 的语法，Java 的对象模型是从 C++改编来的。Java 与 C 和 C++的关系之所以重要，是出于以下几个原因。

第一，许多程序员都熟悉 C/C++语法。这样对于他们而言，学习 Java 就简单了。同样，Java 程序员学习 C/C++也是很简单的。

第二，Java 设计者并没有重复工作。相反，他们进一步对已经成功的程序设计范式进行了提炼。现代程序设计始于 C，而后过渡到 C++，现在则是 Java。通过大量的继承，Java 提供了一个强大的、可以更好利用已有成果的、逻辑一致的程序设计环境，并且增加了在线环境需求的新功能。然而，最重要的一点或许在于，由于它们的相似性，C、C++和 Java 为专业程序员定义了一个统一的概念架构。程序员从其中一种语言转为另一种语言时，不会遇到太大的困难。

C和C++的核心设计原理之一就是程序员的控制。Java也继承了这一原理。除了Internet环境施加的约束以外，Java 为程序员提供了完全的控制。如果程序编得好，就会体现出来，而如果不好，也会体现出来。换句话说，Java 并不是一种教学式语言，它是为专业程序员准备的语言。

Java 还有一个与 C 和 C++共有的属性：它是由真正的程序员设计、测试和修改的。它与设计者的需求和经验紧密结合。因此，再没有比这更好的方法来创建如此一流的专业程序设计语言了。

因为 Java 与 C++的相似性，特别是它们对面向对象程序设计的支持，有些程序员可能会将 Java 简单地看做“C++的 Internet 版”。然而，这种观点是错误的。因为 Java 在实际应用以及基本原理上与 C++有显著的不同。尽管 Java 受到 C++的影响，但是它绝不是 C++的增强版。例如，Java 不提供对 C++的向上或向下兼容。当然，Java 与 C++的相似是十分明显的，如果你是一名 C++程序员，那么在使用 Java 时会有驾轻就熟的感觉。另外，Java 不是为替代 C++而设计的，而是为了解决一系列特定问题而设计的。C++则是用来解决另一个不同系列的问题的。两者将在未来几年中共存。

1.1.2　Java 与 C#的关系

在 Java 问世以后几年，Microsoft 开发出了 C#。C#与 Java 密切相关。事实上，C#的许多功能都是直接从 Java 改编来的。Java 和 C#共享相同的 C++语法风格，都支持分布式程序设计，使用相同的对象模型。它们之间当然也有不同之处，但就整体感觉而言，两者极为相似。这就意味着，如果你已经了解了 C#，那么学习 Java 就很简单；反之，如果你将来要学的是 C#，那么现在学到的有关 Java 的知识也会对你将来有所帮助。