

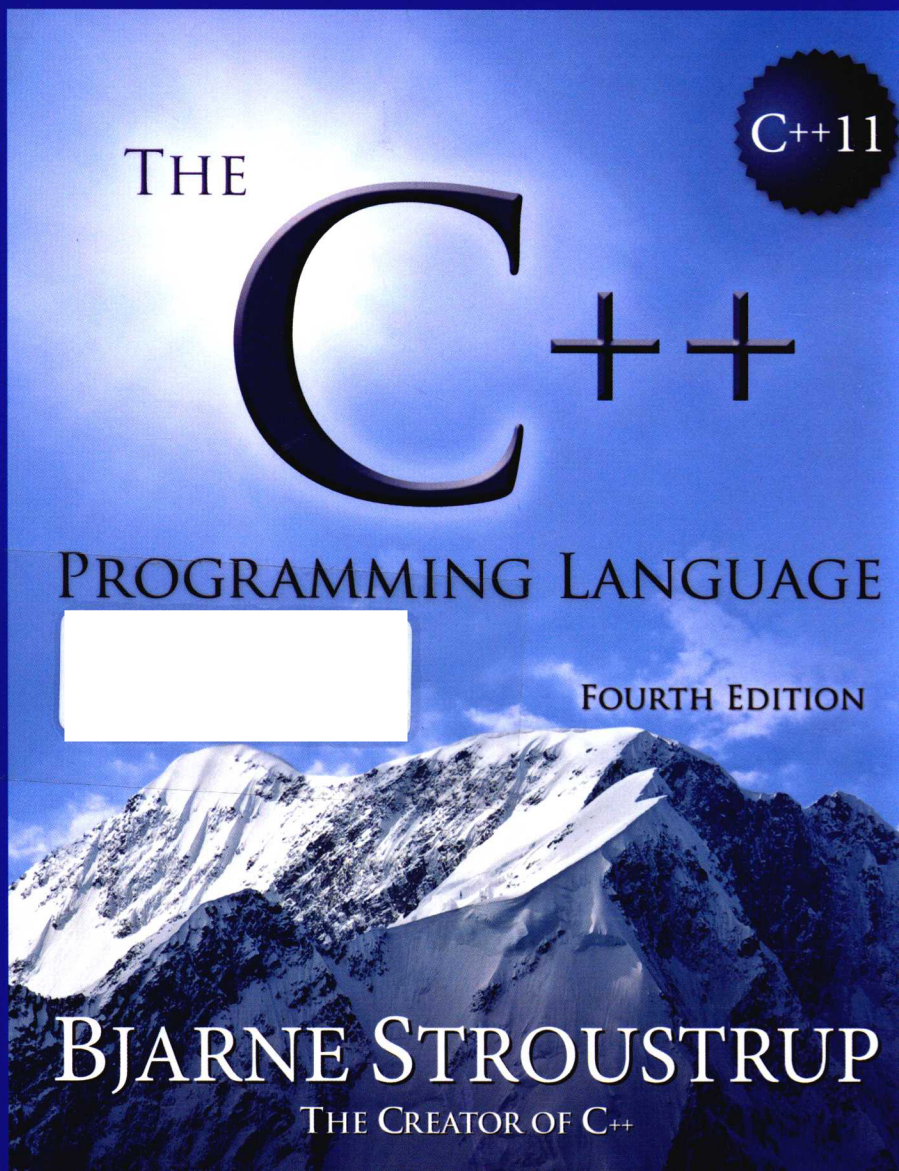
经 典 原 版 书 库

C++程序设计语言

第4部分：标准库

[美] 本贾尼·斯特劳斯特鲁普 著
Bjarne Stroustrup

(英文版·第4版)



经 典 原 版 书 库

C++程序设计语言

第4部分：标准库

(英文版·第4版)

The C++ Programming Language

(Fourth Edition)



[美] 本贾尼·斯特劳斯特鲁普 著
Bjarne Stroustrup



机械工业出版社
China Machine Press

图书在版编目 (CIP) 数据

C++ 程序设计语言 (第4部分:标准库) (英文版·第4版) / (美) 斯特劳斯特鲁普 (Stroustrup, B.) 著. —北京: 机械工业出版社, 2016.1

(经典原版书库)

书名原文: The C++ Programming Language, Fourth Edition

ISBN 978-7-111-52487-8

I. C… II. 斯… III. C 程序—程序设计—英文 IV. TP312

中国版本图书馆 CIP 数据核字 (2015) 第 304114 号

本书版权登记号: 图字: 01-2015-5194

Authorized Adaptation from the English Language edition, entitled *The C++ Programming Language, Fourth Edition* (ISBN 978-0-321-56384-2) by Bjarne Stroustrup, Copyright © 2013 by Pearson Education, Inc.

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage retrieval system, without permission from Pearson Education, Inc.

English language adaptation edition published by Pearson Education Asia Ltd., and China Machine Press Copyright © 2016.

English language adaptation edition is manufactured in the People's Republic of China and is authorized for sale only in People's Republic of China excluding Taiwan, Hong Kong SAR and Macau SAR.

本书英文影印版由 Pearson Education Asia Ltd. 授权机械工业出版社独家出版。未经出版者书面许可, 不得以任何方式复制或抄袭本书内容。

仅限于中华人民共和国境内 (不包括中国香港、澳门特别行政区和中国台湾地区) 销售发行。

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。

出版发行: 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: 迟振春

责任校对: 董纪丽

印刷: 中国电影出版社印刷厂

版次: 2016 年 1 月第 1 版第 1 次印刷

开本: 186mm × 240mm 1/16

印张: 27.5

书号: ISBN 978-7-111-52487-8

定价: 89.00 元

凡购本书, 如有缺页、倒页、脱页, 由本社发行部调换

客服热线: (010) 88378991 88361066

投稿热线: (010) 88379604

购书热线: (010) 68326294 88379649 68995259

读者信箱: hzsj@hzbook.com

版权所有·侵权必究

封底无防伪标均为盗版

本书法律顾问: 北京大成律师事务所 韩光 / 邹晓东

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中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,还揭示了学术的源变,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的现状下,美国等发达国家在其计算机科学发展发展的几十年间积淀和发展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到积极的推动作用,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公司较早意识到“出版要为教育服务”。自1998年开始,我们就将工作重点放在了遴选、移译国外优秀教材上。经过多年的不懈努力,我们与Pearson, McGraw-Hill, Elsevier, MIT, John Wiley & Sons, Cengage等世界著名出版公司建立了良好的合作关系,从他们现有的数百种教材中甄选出Andrew S. Tanenbaum, Bjarne Stroustrup, Brain W. Kernighan, Dennis Ritchie, Jim Gray, Alfred V. Aho, John E. Hopcroft, Jeffrey D. Ullman, Abraham Silberschatz, William Stallings, Donald E. Knuth, John L. Hennessy, Larry L. Peterson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、研究及珍藏。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,有的还专程为其书的中译本作序。迄今,“计算机科学丛书”已经出版了近两百个品种,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。其影印版“经典原版书库”作为姊妹篇也被越来越多实施双语教学的学校所采用。

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。随着计算机科学与技术专业学科建设的不断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将步入一个新的阶段,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。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,我们的联系方式如下:

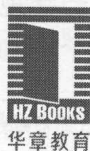
华章网站: www.hzbook.com

电子邮件: hzjsj@hzbook.com

联系电话: (010) 88379604

联系地址: 北京市西城区百万庄南街1号

邮政编码: 100037



华章科技图书出版中心

前言

所有计算机科学问题
都可以通过引入一个新的间接层次来解决，
那些已有过多间接层次的问题除外。

——David J. Wheeler

与 C++98 标准相比，C++11 标准令我可以更清晰、更简洁而且更直接地表达我的想法。而且，新版本的编译器可以对程序进行更好的检查并生成更快的目标程序。因此，C++11 给人的感觉就像是一种新语言一样。

在本书中^①，我追求完整性 (completeness)。我会介绍专业程序员可能需要的每个语言特性和标准库组件。对每个特性或组件，我将给出：

- 基本原理：设计这个特性（组件）是为了帮助解决哪类问题？在其设计之下的原理是什么？它有什么根本的局限？
- 规范：它该如何定义？我将以专业程序员为目标读者来选择内容的详略程度，对于要求更高的 C++ 语言研究者，有很多 ISO 标准的文献可供查阅。
- 例子：当单独使用这个特性或与其他特性组合使用时，如何用好它？其中的关键技术和习惯用法是怎样的？在程序的可维护性和性能方面是否有一些隐含的问题？

多年来，无论是 C++ 语言本身还是它的使用，都已经发生了巨大改变。从程序员的角度，大多数改变都属于语言的改进。与之前的版本相比，当前的 ISO C++ 标准 (ISO/IEC 14882-2011，通常称为 C++11) 在编写高质量代码方面无疑是一个好得多的工具。但是它好在哪里？现代 C++ 语言支持什么样的程序设计风格和技术？这些技术靠哪些语言特性和标准库特性来支撑？精炼、正确、可维护性好、性能高的 C++ 代码的基本构建单元是怎样的？本书将回答这些关键问题。很多答案已经不同于 1985、1995 或 2005 等旧版本的 C++ 语言了：C++ 在进步。

C++ 是一种通用程序设计语言，它强调富类型轻量级抽象的设计和使用。C++ 特别适合开发资源受限的应用，例如可在软件基础设施中发现的那些应用。那些花费时间学习高质量代码编写技术的程序员将会从 C++ 语言受益良多。C++ 是为那些严肃对待编程的人而设计的。人类文明已经严重依赖软件，编写高质量的软件非常重要。

目前已经部署的 C++ 代码达到数十亿行，因此程序稳定性备受重视——很多 1985 年和 1995 年编写的 C++ 代码仍然运行良好，而且还会继续运行几十年。但是，对所有这些应用程序，都可以用现代 C++ 语言写出更好的版本；如果你墨守陈规，将来写出的代码将会是低质量、低性能的。对稳定性的强调还意味着，你现在遵循标准写出的代码，在未来几十年中会运行良

^① 由于篇幅很长，本书影印版分为两册。原书第 1~3 部分为上册，第 4 部分为下册。——编辑注

好。本书中所有代码都遵循 2011 ISO C++ 标准。

本书面向三类读者：

- 想知道最新的 2011 ISO C++ 标准都提供了哪些新特性的 C++ 程序员。
- 好奇 C++ 到底提供了哪些超越 C 语言的特性的 C 程序员。
- 具备 Java、C#、Python 和 Ruby 等编程语言背景，正在探寻“更接近机器”的语言，即更灵活、提供更好的编译时检查或是更好性能的语言的程序员。

自然，这三类读者可能是有交集的——一个专业软件开发通常掌握多门编程语言。

本书假定目标读者是程序员。如果你提问“什么是 for 循环？”或是“什么是编译器？”，那么本书现在还不适合你，我向你推荐我的另一本书《C++ 程序设计原理与实践》，这本书适合作为程序设计和 C++ 语言的入门书籍。而且，我假定读者是较为成熟的软件开发。如果你提问“为什么要费力进行测试？”或是“所有语言基本都是一样的，给我看语法就可以了”，或者确信存在一种适合所有任务的完美语言，那么本书也不适合你。

相对于 C++98，C++11 提出了哪些改进和新特性呢？适合现代计算机的机器模型会涉及到大量并发处理。为此，C++11 提供了用于系统级并行编程（如使用多核）的语言和标准库特性。C++11 还提供了正则表达式处理、资源管理指针、随机数、改进的容器（包括哈希表）以及其他很多特性。此外，C++11 还提供了通用和一致的初始化机制、更简单的 for 语句、移动语义、基础的 Unicode 支持、lambda、通用常量表达式、控制类缺省定义的能力、可变参数模板、用户定义的字面值常量和和其他很多新特性。请记住，这些标准库和语言特性的目标就是支撑那些用来开发高质量软件的程序设计技术。这些特性应该组合使用——将它们看作盖大楼的砖，而不应该相互隔离地单独使用来解决特定问题。计算机是一种通用机器，而 C++ 在其中起到了重要作用。特别是，C++ 的设计目标就是足够灵活和通用，以便处理那些连它的设计者都未曾想象的未来的难题。

致谢

除了本书上一版致谢提及的人之外，我还要感谢 Pete Becker、Hans-J. Boehm、Marshall Clow、Jonathan Coe、Lawrence Cowl、Walter Daugherty、J. Daniel Garcia、Robert Harle、Greg Hickman、Howard Hinnant、Brian Kernighan、Daniel Krüger、Nevin Liber、Michel Michaud、Gary Powell、Jan Christiaan van Winkel 和 Leor Zolman。没有他们的帮助，本书的质量要差得多。

感谢 Howard Hinnant 为我解答很多有关标准库的问题。

Andrew Sutton 是 Origin 库的作者，模板相关章节中很多模拟概念的讨论都是基于这个测试平台的。他还是矩阵库的作者，这是第 29 章的主题。Origin 库是开源的，在互联网上搜索“Origin”和“Andrew Sutton”就能找到。

感谢我指导的研究生毕业设计班，他们从“导学章节”中找出的问题比其他任何人都多。

假如我能遵照审阅人的所有建议，毫无疑问会大幅度提高本书的质量，但篇幅上也会增加数百页。每个专家审阅人都建议增加技术细节、进阶示例和很多有用的开发规范；每个新手审

阅人（或教育工作者）都建议增加示例；而大多数审阅人都（正确地）注意到本书的篇幅可能过长了。

感谢普林斯顿大学计算机科学系，特别感谢 Brian Kernighan 教授，在我利用部分休假时间撰写此书时给予我热情接待。

感谢剑桥大学计算机实验室，特别感谢 Andy Hopper 教授，在我利用部分休假时间撰写此书时给予我热情接待。

感谢我的编辑 Peter Gordon 以及他在 Addison-Wesley 的出版团队，感谢你们的帮助和耐心。

Bjarne Stroustrup

于得克萨斯大学城

第3版前言

去编程就是去理解。

——Kristen Nyggard

我觉得用 C++ 编程比以往更令人感到愉快。在过去这些年里，C++ 在支持设计和编程方面取得了令人振奋的进步，针对其使用的大量新技术已经被开发出来了。然而，C++ 并不只是好玩。普通的实际程序员在几乎所有种类和规模的开发项目上，在生产率、可维护性、灵活性和质量方面都取得了显著的进步。到今天为止，C++ 已经实现了我当初对它的期望中的绝大部分，还在许多我原来根本没有梦想过的工作中取得了成功。

本书介绍的是标准 C++^①以及由 C++ 所支持的关键性编程技术和设计技术。与本书第 1 版所介绍的那个 C++ 版本相比，标准 C++ 是一个经过了更仔细推敲的更强大的语言。各种新的语言特征，如名字空间、异常、模板，以及运行时类型识别，使人能以比过去更直接的方式使用许多技术，标准库使程序员能够从比基本语言高得多的层面上起步。

本书第 2 版中大约有三分之一的内容来自第 1 版。这个第 3 版则是重写了比例更大的篇幅的结果。它提供的许多东西是大部分有经验的程序员也需要的，与此同时，本书也比它的以前版本更容易供新手入门。C++ 使用的爆炸性增长和作为其结果的海量经验积累使这些成为可能。

一个功能广泛的标准库定义使我能以一种与以前不同的方式介绍 C++ 的各种概念。与过去一样，本书对 C++ 的介绍与任何特定的实现都没有关系；与过去一样，教材式的各章还是采用“自下而上”的方式，使每种结构都是在定义之后才使用。无论如何，使用一个设计良好的库远比理解其实现细节容易得多。由于这些情况，在假定读者已经理解了标准库的内部工作原理之前，就可以利用它提供许多更实际、更有趣的例子。标准库本身也是程序设计实例和设计技术的丰富源泉。

本书将介绍每种主要的 C++ 语言特征和这个标准库，它是围绕着语言和库功能组织起来的。当然，各种特征都将在使用它们的环境中介绍。也就是说，这里所关注的是将语言作为一种设计和编程的工具，而不是语言本身。本书将展示那些使 C++ 卓有成效的关键性技术，讲述为掌握它们所需要的那些基本概念。除了专门阐释技术细节的那些地方之外，其他示例都取自系统软件领域。另一本与本书配套出版的书《带标注的 C++ 语言标准》(*The Annotated C++ Language Standard*)，将给出完整的语言定义，所附标注能使它更容易理解。

本书的基本目标就是帮助读者理解 C++ 所提供的功能将如何支持关键性的程序设计技术。这里的目标是使读者能远远超越简单地复制示例并使之能够运行，或者模仿来自其他语言的程序设计风格。只有对隐藏在语言背后的思想有了很好的理解之后，才能真正掌握这个语言。如

① ISO/IEC 14882, C++ 程序设计语言标准。

果有一些具体实现的文档的辅助，这里所提供的信息就足以对付具有挑战性的真实世界中的重要项目。我的希望是，本书能帮助读者获得新的洞察力，使他们成为更好的程序员和设计师。

致谢

除了第1版和第2版的致谢中所提到的那些人之外，我还要感谢 Matt Austern、Hans Boehm、Don Caldwell、Lawrence Cowl、Alan Feuer、Andrew Forrest、David Gay、Tim Griffin、Peter Juhl、Brian Kernighan、Andrew Koenig、Mike Mowbray、Rob Murray、Lee Nackman、Joseph Newcomer、Alex Stepanov、David Vandevoorde、Peter Weinberger 和 Chris Van Wyk，他们对第3版各章的初稿提出了许多意见。没有他们的帮助和建议，这本书一定会更难理解，包含更多的错误，没有这么完全，当然也可能稍微短一点。

我还要感谢 C++ 标准化委员会的志愿者们，是他们完成了规模宏大的建设性工作，才使 C++ 具有它今天这个样子。要罗列出每个人会有一点不公平，但一个也不提就更不公平，所以我想特别提及 Mike Ball、Dag Brück、Sean Corfield、Ted Goldstein、Kim Knuttila、Andrew Koenig、José Lajoie、Dmitry Lenkov、Nathan Myers、Martin O’Riordan、Tom Plum、Jonathan Shopiro、John Spicer、Jerry Schwarz、Alex Stepanov 和 Mike Vilot，他们中的每个人都在 C++ 及其标准库的某些方面直接与我合作过。

在这本书第一次印刷之后，许多人给我发来电子邮件，提出更正和建议。我已经在原书的结构里响应了他们的建议，使后来出版的版本大为改善。将本书翻译到各种语言的译者也提供了许多澄清性的意见。作为对这些读者的回应，我增加了附录 D 和附录 E。让我借这个机会感谢他们之中特别有帮助的几位：Dave Abrahams、Matt Austern、Jan Bielawski、Janina Mincer Daszkiewicz、Andrew Koenig、Dietmar Kühl、Nicolai Josuttis、Nathan Myers、Paul E. Sevinc、Andy Tenne-Sens、Shoichi Uchida、Ping-Fai(Mike) Yang 和 Dennis Yelle。

Bjarne Stroustrup

于新泽西默里山

第2版前言

前路漫漫。

——Bilbo Baggins

正如在本书的第1版中所承诺的，C++ 为满足其用户的需要正在不断地演化。这一演化过程得益于许多有着极大的背景差异，在范围广泛的应用领域中工作的用户们的实际经验的指导。在第1版出版后的六年中，C++ 的用户群体扩大了不止百倍，人们学到了许多东西，发现了许多新技术并通过了实践的检验。这些技术中的一些也在这一版中有所反映。

在过去六年里所完成的许多语言扩展，其基本宗旨就是将 C++ 提升为一种服务于一般性的数据抽象和面向对象程序设计的语言，特别是提升为一个可编写高质量的用户定义类型库的工具。一个“高质量的库”是指这样的库，它以一个或几个方便、安全且高效的类的形式，给用户提供了一个概念。在这个环境中，安全意味着这个类在库的使用者与它的供方之间构成了一个特殊的类型安全的界面；高效意味着与手工写出的 C 代码相比，这种库的使用不会给用户强加明显的运行时间上或空间上的额外开销。

本书介绍的是完整的 C++ 语言。从第1章到第10章是一个教材式的导引，第11章到第13章展现的是一个有关设计和软件开发问题的讨论，最后包含了完整的 C++ 参考手册。自然，在原来版本之后新加入的特征和变化已成为这个展示的有机组成部分。这些特征包括：经过精化后的重载解析规则和存储管理功能，以及访问控制机制、类型安全的连接、const 和 static 成员函数、抽象类、多重继承、模板和异常处理。

C++ 是一个通用的程序设计语言，其核心应用领域是最广泛意义上的系统程序设计。此外，C++ 还被成功地用到许多无法称为系统程序设计的应用领域中。从最摩登的小型计算机到最大的超级计算机上，以及几乎所有操作系统上都有 C++ 的实现。因此，本书描述的是 C++ 语言本身，并不想试着去解释任何特殊的实现、程序设计环境或者库。

本书中给出的许多类的示例虽然都很有用，但也还是应该归到“玩具”一类。与在完整的精益求精的程序中做解释相比，这里所采用的解说风格能更清晰地呈现那些具有普遍意义的原理和极其有用的技术，在实际例子中它们很容易被细节所淹没。这里给出的大部分有用的类，如链接表、数组、字符串、矩阵、图形类、关联数组等，在广泛可用的各种商品和非商品资源中，都有可用的“防弹”和/或“金盘”版本。那些“具有工业强度”的类和库中的许多东西，实际上不过是在这里可以找到的玩具版本的直接或间接后裔。

与第1版相比，这一版更加强调本书在教学方面的作用。然而，这里的叙述仍然是针对有经验的程序员，并努力不去轻视他们的智慧和经验。有关设计问题的讨论有了很大的扩充，作为对读者在语言特征及其直接应用之外的要求的一种回应。技术细节和精确性也有所增强。特

别是，这里的参考手册表现了在这个方向上多年的工作。我的目标是提供一本具有足够深度的书籍，使大部分程序员能在多次阅读中都有所收获。换句话说，这本书给出的是 C++ 语言，它的基本原理，以及使用时所需要的关键性技术。欢迎欣赏！

致谢

除了在第 1 版前言的致谢里所提到的人们之外，我还要感谢 Al Aho、Steve Buroff、Jim Coplien、Ted Goldstein、Tony Hansen、Lorraine Juhl、Peter Juhl、Brian Kernighan、Andrew Koenig、Bill Leggett、Warren Montgomery、Mike Mowbray、Rob Murray、Jonathan Shopiro、Mike Vilot 和 Peter Weinberger，他们对第 2 版的初稿提出了许多意见。许多人对 C++ 从 1985 年到 1991 年的开发有很大影响，我只能提及其中几个：Andrew Koenig，Brian Kernighan，Doug McIlroy 和 Johathan Shopiro。还要感谢参考手册“外部评阅”的许多参与者，以及在 X3J16 的整个第一年里一直在其中受苦的人们。

Bjarne Stroustrup

于新泽西默里山

第 1 版前言

语言磨砺了我们思维的方式，
也决定着我們思考的范围。

——B. L. Whorf

C++ 是一种通用的程序设计语言，其设计就是为了使认真的程序员工作得更愉快。除了一些小细节之外，C++ 是 C 程序设计语言的一个超集。C++ 提供了 C 所提供的各种功能，还为定义新类型提供了灵活而有效的功能。程序员可以通过定义新类型，使这些类型与应用中的概念紧密对应，从而把一个应用划分成许多容易管理的片段。这种程序构造技术通常被称为数据抽象。某些用户定义类型的对象包含着类型信息，这种对象就可以方便而安全地用在那种对象类型无法在编译时确定的环境中。使用这种类型的对象的程序通常被称为是基于对象的。如果用得好，这些技术可以产生出更短、更容易理解，而且也更容易管理的程序。

C++ 里的最关键概念是类。一个类就是一个用户定义类型。类提供了对数据的隐藏，数据的初始化保证，用户定义类型的隐式类型转换，动态类型识别，用户控制的存储管理，以及重载运算符的机制等。在类型检查和表述模块性方面，C++ 提供了比 C 好得多的功能。它还包含了许多并不直接与类相关的改进，包括符号常量、函数的在线替换、默认函数参数、重载函数名、自由存储管理运算符，以及引用类型等。C++ 保持了 C 高效处理硬件基本对象（位、字节、字、地址等）的能力。这就使用户定义类型能够在相当高的效率水平上实现。

C++ 及其标准库也是为了可移植性而设计的。当前的实现能够在大多数支持 C 的系统上运行。C 的库也能用于 C++ 程序，而且大部分支持 C 程序设计的工具也同样能用于 C++。

本书的基本目标就是帮助认真的程序员学习这个语言，并将它用于那些非平凡的项目。书中提供了有关 C++ 的完整描述，许多完整的例子，以及更多的程序片段。

致谢

如果没有许多朋友和同事持之以恒的使用、建议和建设性的批评，C++ 绝不会像它现在这样成熟。特别地，Tom Cargill、Jim Coplien、Stu Feldman、Sandy Fraser、Steve Johnson、Brian Kernighan、Bart Locanthi、Doug McIlroy、Dennis Rechie、Larry Rosler、Jerry Schwarz 和 Jon Shopiro 对语言发展提供了重要的思想。Dave Presotto 写出了流 I/O 库的当前实现。

此外，还有几百人对 C++ 及其编译器的开发做出了贡献：给我提出改进的建议，描述所遇到的问题，告诉我编译中的错误等。我只能提及其中的很少几位：Gary Bishop, Abdreu Hume, Tom Karzes, Victor Milenkovic, Rob Murray, Leonie Rose, Brian Schmilt 和 Gary Walker。

许多人在本书的撰写过程中为我提供了帮助，特别值得提出的是 Jon Bentley、Laura Eaves、Brian Kernighan、Ted Kowalski、Steve Mahaney、Jon Shopiro，以及参加 1985 年 7 月 26~27 日俄亥俄州哥伦布贝尔实验室 C++ 课程的人们。

Bjarne Stroustrup

于新泽西默里山

Contents

Part I: Introductory Material[⊖]

1. Notes to the Reader	1
2. A Tour of C++: The Basics	3
3. A Tour of C++: Abstraction Mechanisms	37
4. A Tour of C++: Containers and Algorithms	59
5. A Tour of C++: Concurrency and Utilities	87
	111

Part II: Basic Facilities

	133
6. Types and Declarations	135
7. Pointers, Arrays, and References	171
8. Structures, Unions, and Enumerations	201
9. Statements	225
10. Expressions	241
11. Select Operations	273
12. Functions	305
13. Exception Handling	343
14. Namespaces	389
15. Source Files and Programs	419

Part III: Abstraction Mechanisms

	447
16. Classes	449
17. Construction, Cleanup, Copy, and Move	481
18. Overloading	527
19. Special Operators	549
20. Derived Classes	577
21. Class Hierarchies	613
22. Run-Time Type Information	641
23. Templates	665

⊖ 由于篇幅很长，本书影印版分为两册。原书第1~3部分为上册，第4部分为下册。——编辑注

24. Generic Programming	699
25. Specialization	721
26. Instantiation	741
27. Templates and Hierarchies	759
28. Metaprogramming	779
29. A Matrix Design	827

Part IV: The Standard Library 857

30. Standard Library Summary	859
31. STL Containers	885
32. STL Algorithms	927
33. STL Iterators	953
34. Memory and Resources	973
35. Utilities	1009
36. Strings	1033
37. Regular Expressions	1051
38. I/O Streams	1073
39. Locales	1109
40. Numerics	1159
41. Concurrency	1191
42. Threads and Tasks	1209
43. The C Standard Library	1253
44. Compatibility	1267

目 录

第一部分 介绍	1
第 1 章 致读者	3
第 2 章 C++ 概览：基本特性	37
第 3 章 C++ 概览：抽象机制	59
第 4 章 C++ 概览：容器和算法	87
第 5 章 C++ 概览：并发和工具	111
第二部分 基本特性	133
第 6 章 类型和声明	135
第 7 章 指针、数组和引用	171
第 8 章 结构、联合和枚举	201
第 9 章 语句	225
第 10 章 表达式	241
第 11 章 其他操作选讲	273
第 12 章 函数	305
第 13 章 异常处理	343
第 14 章 命名空间	389
第 15 章 源文件和程序	419
第三部分 抽象机制	447
第 16 章 类	449
第 17 章 构造、清理、拷贝和移动	481
第 18 章 重载	527
第 19 章 特殊运算符	549
第 20 章 派生类	577
第 21 章 类层次	613
第 22 章 运行时类型信息	641

目 录

第 23 章	模板	665
第 24 章	泛型程序设计	699
第 25 章	特例化	721
第 26 章	实例化	741
第 27 章	模板和层次	759
第 28 章	元程序设计	779
第 29 章	一个矩阵设计	827

第四部分 标准库

857

第 30 章	标准库概览	859
第 31 章	STL 容器	885
第 32 章	STL 算法	927
第 33 章	STL 迭代器	953
第 34 章	内存和资源	973
第 35 章	工具	1009
第 36 章	字符串	1033
第 37 章	正则表达式	1051
第 38 章	I/O 流	1073
第 39 章	区域设置	1109
第 40 章	数值	1159
第 41 章	并发	1191
第 42 章	线程和任务	1209
第 43 章	C 标准库	1253
第 44 章	兼容性	1267