

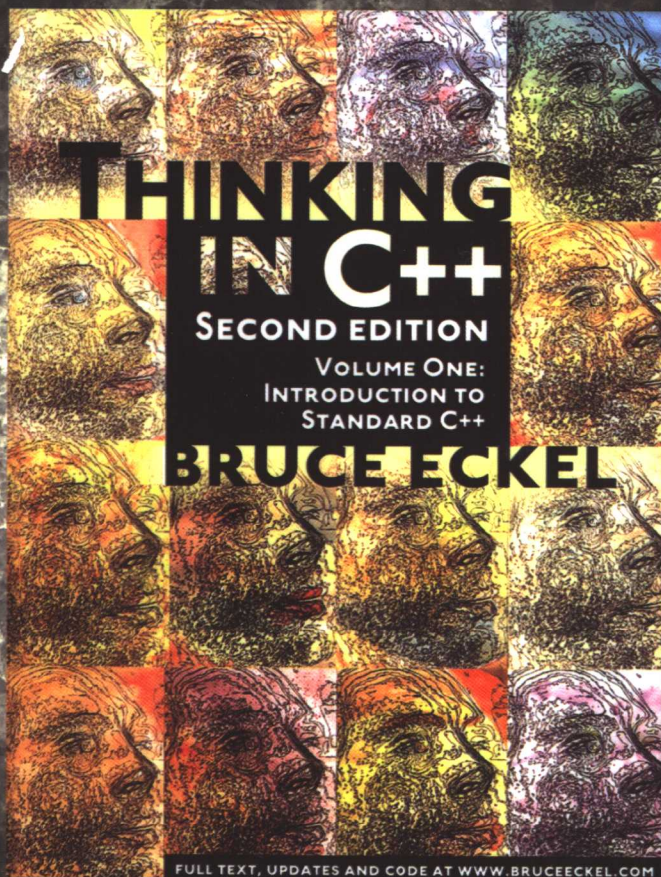
计 算 机 科 学 丛 书

第2版

C++ 编程思想

第1卷：标准C++导引

(美) Bruce Eckel 著 刘宗田 袁兆山 潘秋菱 等译



Thinking in C++

Second Edition

Volume One: Introduction to Standard C++



机械工业出版社
China Machine Press

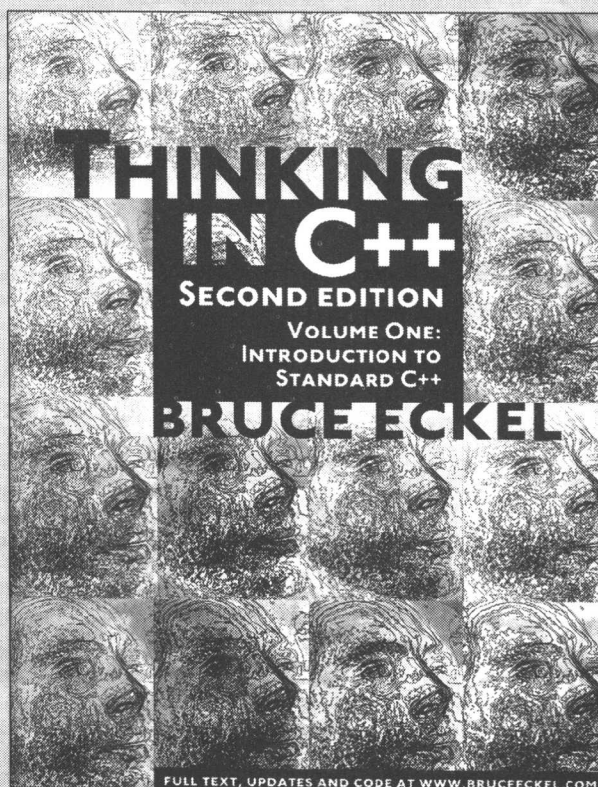
Prentice
Hall

计 算 机 科 学 丛 书

第2版

C++ 编程思想

(美) Bruce Eckel 著 刘宗田 袁兆山 潘秋菱 等译



Thinking in C++

Second Edition

Volume One: Introduction to Standard C++



机械工业出版社
China Machine Press

本书第1版荣获美国“软件开发”杂志评选的1996年图书震撼大奖 (Jolt Award), 中文版自2000年推出以来, 经久不衰, 获得了读者的充分肯定和高度评价。

第2版与第1版相比, 在章节安排上有以下改变。增加了两章: “对象的创建与使用”和“C++中的C”。前者与“对象导言”实际上是第1版“对象的演化”一章的彻底重写, 增加了近几年面向对象方法和编程方法的最新研究与实践的丰硕成果; 后者的添加使不熟悉C的读者可以直接使用本书。删去了四章: “输入输出流介绍”、“多重继承”、“异常处理”和“运行时类型识别”, 删去的内容均为C++中较复杂的主题, 作者将它们连同C++标准完成后增加的一些内容放到本书的第2卷中, 使本书的第1卷内容显得更加集中, 可以供不同程度的读者选择阅读。需要强调的是, 第2版的改变不仅体现在这些章节的调整上, 更多的改变体现在每一章的字里行间, 包括例子的调整和练习的补充。与众不同的精心选材和认真推敲的叙述使得第2版更趋成熟。

本书是C++领域内一本权威的著作, 书中的内容、讲授方法、例子和练习既适合课堂教学, 又适合读者自学。无论是高等院校计算机及相关专业的学生, 还是业界的从业人员, 以及广大的计算机爱好者, 都可从阅读本书中获益。

Bruce Eckel: Thinking in C++ (Second Edition), Volume One: Introduction to Standard C++ (ISBN 0-13-979809-9).

Copyright © 2000 by Bruce Eckel, MindView, Inc.

All rights reserved.

Chinese simplified language edition published by China Machine Press.

Copyright © 2002 by China Machine Press.

版权所有, 侵权必究。

本书版权登记号: 图字: 01-2000-4303

图书在版编目(CIP)数据

C++编程思想(第2版): 第1卷: 标准C++导引/(美)埃克尔(Eckel, B.)著; 刘宗田等译. -北京: 机械工业出版社, 2002.9

(计算机科学丛书)

书名原文: Thinking in C++ (Second Edition), Volume One: Introduction to Standard C++ ISBN 7-111-10807-8

I. C… II. ①埃… ②刘… III. C语言-程序设计 IV. TP312

中国版本图书馆CIP数据核字(2002)第062698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: 姚 蕾

北京牛山世兴印刷厂印刷·新华书店北京发行所发行

2002年9月第1版第1次印刷

787mm × 1092mm 1/16 · 32.25印张

印数: 0 001-10 000册

定价: 59.00元(附光盘)

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，不仅肇划了研究的范畴，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、研究及收藏。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。

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，我们的联系方式如下：

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

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

本书为1996年软件开发图书震撼大奖 (Jolt Award) 得主。

“这本书是一项巨大的成就。你的书架上早就该有这本书了。讲述输入输出流的这一章是我至今见过的对这个主题最全面最易懂的叙述。”

—— *《Doctor Dobbs Journal》* 杂志的资深编辑Al Stevens

“Eckel的作品是惟一一本能如此清晰地叙述以面向对象方法构造程序的书籍。这本书也是一本优秀的C++入门指南。”

—— *《Unix Review》* 杂志的编辑Andrew Binstock

“Bruce继续用他对C++的洞察力让我惊叹。《C++编程思想》是他思想的精华荟萃。如果你需要C++中难题的清晰解答，就去买这部杰作吧。”

—— *《The Tao of Objects》* 一书的作者Gary Entsminger

“《C++编程思想》耐心而系统地探讨了C++的一些重要问题，例如内联、引用、运算符重载、继承和动态对象，还有一些更深奥的主题，例如模板的合理使用、异常和多继承等。所有这些内容，连同Eckel本人的关于对象和程序设计的观点，完美地编织为一体，成为每个C++开发者案头必备之书。如果你正在用C++开发软件，《C++编程思想》应当是你必须拥有的一本书。”

—— *《PC Magazine》* 的特邀编辑Richard Hale Shaw

读者评论

好书……了不起的作品!

——**Andrew Schulman**, “**Doctor Dobbs Journal**” 杂志

一本绝对的必备之书。我的书架上最有用的、最可信赖的书。

——**TUG Lines**

这是一本对程序员真正有用的书籍。

——**IEEE Computer**

令人耳目一新。

——**PJ Plauger**, “**Embedded Systems Programming**” 杂志

Eckel成功了, …这本书是如此清新可读。

——**Unix World**

绝对应当是你购书的首选。

——**C Gazette**

一本关于C++的绝妙参考书。

——**Michael Brandt**, 高级分析员/程序员, 澳大利亚悉尼

在HRB系统公司的项目中, 我们称你的书为“答案书”, 对于这个项目, 它是我们的C++圣经。

——**Curt Snyder**, HRB系统公司

你的书真是伟大之作, 你还使它可以从Web上免费获得, 我的感激实在是难以言表。这是我已经见到的最透彻和最有用的C++书籍。

——**Russell Davis**

……这是我所仅见的讲述C++复杂知识(以及面向对象程序设计的基础)还能如此好读的书。

——**Gunther Schulz**, KawaiiSoft

我喜欢你书中的例子。这里有我从未想到过的东西(还有一些内容, 我真是不知道你是怎么写出来的)!

——**Rich Herrick**, 洛克希德-马丁联邦系统公司(Owego, NY)高级副软件工程师

这是一本奇书。只要我有问题, 我就在线查阅这本书。可以说是屡试不爽。能够读到如此水准的书, 真是一种幸福。

——**Wes Kells**, 计算机工程学生, SLC Kingston

你是一个无价的源泉，我真的喜欢你的书、邮件列表……。我所做的几乎每个项目都得益于你的真知灼见。

——**Justin Voshell**

这正是我一直在寻找的C++书籍。

——**Thomas A. Fink, Trepp公司经理**

你的书非常权威，并且易于阅读。我对我的同事说你就是C++的K & R。

——**Mark Orlassino, 高级设计工程师, Harmon工业公司(Hauppauge, NY)**

当我第一次开始学习C++时，你的书《C++编程思想》是黑暗隧道中的指引明灯。我尽力提高我的C++技能，而在此过程中《C++编程思想》是我得以持续提高的坚实基础。

——**Peter Tran, 高级系统分析员 (IM), 原康柏计算机公司**

在我学习C++过程中，这本书是最好的综合参考书。大多数书透彻地解释了某些主题，而对于另一些主题则挂一漏万。而《C++编程思想》第2版当仁不让，有问必答。

——**Thomas Michel**

我藏书甚丰，但却没有一本令我满意。我渴求一本好的关于模板和STL的书。后来读了你的书，惊叹不已。你告诉我们如何用模板和STL写C++，一点都不拖泥带水。你所做的正是我期望C++社团（也就是C++的下一代作者）能够做到的。作为一名作者，我被你的写作和阐释才能所深深折服。你阐明了前人未能充分恰当说明的问题，你的方法源于你对所述素材的深入细致的研究。你对明智的情况和问题所在提出了质疑。在我的书架上，它是一本重要书籍，与Petzold放在一起。

——**Christian Gross, 顾问/导师, cgross@eusoft.com**

我认为你的书非常、非常、非常好。我将它与书店里的其他书比较，发现当我学习STL时，是你的书真正教会了我C++基础知识。这是非常好的学习经验，无论是一气呵成还是循序渐进。我认为你的书出奇的好，叙述方式非常容易理解。

——**Jeff Meininger, 软件开发员, boxybutgood.com**

你的书是我迄今看到的里面最好的。快点买吧，这样我们大家就有了优秀的和“可靠的”参考书。赶快！赶快！为了这样一本优秀书籍，这样做是值得的。

——**Steve Strickland, Live Minds (a Puzzle business)**

（来自Usenet）与其他大多数C++作者不同，Eckel专职教授C++和Java课程，他能从大量新手的学习反馈中受益，他的书反映了这一点。他的书不只是写如何用C++/Java编程，而且讲述了如何理解这些语言的意图和它们中蕴含的思想。Eckel也是一个我所读过的继Jeff

Duntemann以后的最好的技术作者。他的书非常清楚，容易阅读。不要被它们的大部头吓倒，实际上，每一本都能在21天之内读完。

——**Randy Crawford, MRJ Technology Solutions (Fairfax VA)**

你的作品广受好评，感谢你帮助我很好地理解了C++和Java。

——**Barry Wallin, 数学/计算机教师, Rosemount高中(Rosemount, MN)**

我应当感谢你的书《C++编程思想》，毋庸置疑，这是我迄今读到的这方面最好的书。

——**Riccardo Tarli, 软件工程师, 意大利R&D TXT Ingegneria Informatica**

我已经读了你的《Java编程思想》和《C++编程思想》，两者都是各自领域最好的书。

——**Ratnakarprasad H. Tiwari, Mumbai, 印度**

“调试技巧”一节相当有用，以至于我将它打印出来长期保留。我认为这一节应当是在第一个或第二个程序设计问题之后任何入门课的必须部分。

——**Fred Ballard, Synectics公司**

你的书实在是关于C++知识的珍宝。我觉得你总能先给出好的概括，然后解释具体细节。

——**Raymond Pickles, 美国海军研究实验室雷达部天线分部, 华盛顿特区**

我作为一个内科医生和计算机科技人员，需要花费大量的时间从书本和杂志中提取信息。我的经验是，一个好的作者能使困难的概念变得容易理解。如果要对这方面的图书打分，你是我认为的最高分数的技术作家之一。祝你常出优秀作品。

——**Declan O'Kane博士, 英国莱斯特**

对于我的二年级C++课程，《C++编程思想》是我要经常参考的书，我要求我的学生认真学习它。我经常提到关于运算符重载这一章。书中的例子或代码本身就很有价值，超过这本书定价的许多倍。许多书和开发环境假设所论述的语言只在Windows环境上应用，因此能找到和使用一本以C++为纯粹内容的书是重要的，这样我就能培养我的学生从事专门领域，例如嵌入式系统、网络等，这些都要求真正的深入理解。

——**Robert Chase教授, Sweet Briar学院**

我认为它是C++的绝佳介绍，特别对于像我这样的长期的业余者，我常常“知其然”，但很少能够“知其所以然”，TIC2真是天赐之物。

——**Tony Likhite, 系统管理员/数据库分析员, Together网络公司**

读了这本书的前80页之后，我感到比读成堆的该主题的书对OOP有了更深刻的理解。多谢！

——**Rick Schneewind**

译者序

作为译者，我有幸组织翻译了《C++编程思想》第1版。在这之前，我仅仅耳闻这是一本别具特色的畅销书，至于如何别具特色，如何得以畅销，并不十分清楚。在第1版的翻译过程中，我逐渐领悟了Eckel编写技巧的真谛。在第1版中文版的译者序中，我曾这样总结他的技巧：“其内容、讲授方法、选用例子和跟随的练习，别具特色。原书作者不是按传统的方法讲解C++的概念和编程方法，而是根据他自己过去学习C++的亲身体会，根据他多年教学中从他的学生的学习中发现的问题，用一些非常简单的例子和简练的叙述，阐明了在学习C++中特别容易混淆的概念。特别是，他经常通过例子引导读者从C++编译实现的汇编代码的角度反向审视C++的语法和语义，常常使读者有‘心有灵犀一点通’的奇特效果，这在以往的C++书中并不多见。”

《C++编程思想》第1版的中文版自2000年1月第1次印刷以来，在中国市场上的畅销势头经久不衰。这充分说明了这本书在中国读者心目中的地位。

Eckel致力于计算机程序设计语言教学数十年，而且是全心全意地从事这项工作，这本身就是难能可贵的，是他成功的根本原因。另外，他的成功还有赖于他的精益求精的精神，这不仅表现在第1版的与众不同的精心选材和认真推敲的叙述方面，也体现在第2版与第1版的不同点上。

表面上，第2版与第1版并无太多的变化，但是通过分析，可以看出，其中的任何变化都是经过深思熟虑的。从章节上看，最大的区别是增加了两章和去掉了四章。增加的两章分别是“对象的创建与使用”和“C++中的C”。前者与“对象导言”实际上是第1版的“对象的演化”一章的彻底重写，增加了近几年面向对象方法和编程方法的最新研究与实践的丰硕成果。后者的添加不仅使不熟悉C的读者直接使用这本书成为可能，而且C本身就是C++的组成部分，这是C++得以成功的主要原因之一。删去的四章是“输入输出流介绍”、“多重继承”、“异常处理”和“运行时类型识别”。这四章属于C++中较复杂的主题，作者将它们连同C++标准完成后又增加的一些内容放到这本书的第2卷中。这样就使得这本书的第1卷内容更加集中，一般的读者可以不被这些复杂内容所困扰，而需要这些复杂知识的读者可以阅读这本书的第2卷。

实际上，第2版的改变不仅仅在于这些章节的调整，更多的改变体现在每一章中，包括例子的调整和练习的补充。这本书更成熟了。

受机械工业出版社华章公司计算机编辑部委托，我又承担起《C++编程思想》第2版的翻译组织任务。翻译这样的成功之作，既是机遇，更是压力。有如此众多的读者阅读我们翻译的作品，无论如何这是令人高兴的事情。诚然，吸引读者的魅力来源于原作，而不是我们的翻译技巧，但是能将如此光辉灿烂的作品变成中文版本，奉献给中国的读者，这其中毕竟融入了我们的心血，而且，第1版中文版的畅销，已经充分证明，并未因我们翻译水平的限制而黯淡了原作的光芒，这对我们已经够宽慰的了。然而，百万双眼睛在阅读这本书的同时也在审视我们的翻译水平，这就足以使我们诚惶诚恐的了。翻译在某种意义上是再创作的过程，

读者见仁见智。为求更多的满意，我们只有尽力而为。由于时间和水平限制，翻译错误在所难免，恳请读者指正。

参加第2版翻译和审校工作的人员包括：刘宗田、韩冬、蔡烈斌、袁兆山、潘秋菱、许东、李航、肖苑、刘璐、姜桂华、张卿、邵坤、陈慧琼、何允如、贾亮、童朝柱、邢大红、潘飏、刘莹、姜川、冯鸿等。

感谢为本书第1版和第2版中文版作出贡献的所有朋友。感谢关心和支持本书翻译出版的广大读者。

刘宗田
2002.8.10

前 言

像任何人类语言一样，C++提供了一种表达思想的方法。如果这种表达方法是成功的，那么当问题变得更大和更复杂时，该方法将会明显地表现出比其他方法更容易和更灵活的优点。

1

不能只把C++看做是语言要素的一个集合，因为有些要素单独使用是没有意义的。如果我们不只是用C++语言编写代码，而是用它思考“设计”问题，那么必须综合使用这些要素。而且，为了以这种方法理解C++，我们必须了解使用C的问题和一般的编程问题。本书讨论的是编程问题、为什么这些编程问题会成为要解决的问题以及用C++解决编程问题所采用的方法。因此，在每一章中所解释的一组语言要素，都建立在C++语言解决某一类特殊问题所用方法的基础之上。以这种方式，我希望一点一点地引导读者，从掌握C开始，直到读者使用C++变成自己的母语思维方式。

我将始终坚持一种观点：读者应当在头脑中建立一个模型，以便逐步理解这种语言，直到炉火纯青的程度。如果读者遇到难题，他可以将问题纳入这个模型，推导出答案。我将努力把已经印在我脑海中的见解传授给读者，正是这些见解，使得我能开始“用C++进行思考”。

0.1 第2版中的新内容

本书是第1版的彻底重写，反映了C++标准最终完成所带来的C++的所有改变，也反映了自从第1版写完后我又学习到的内容。我已经检查并重写了第1版中的全部文字，在这个过程中，我删去了一些过时的例子，修改了一些现有的例子，并增加了一些新的例子和新的练习。我对第1版的内容进行了大规模的重新整理和重新编排，以便反映新出现的更好的工具和我对人们如何学习C++的进一步理解。为方便没有C背景知识的读者能阅读本书后面的章节，在第2版增加了一章，简要地介绍C概念和基本的C++特征。本书配套的CD ROM包含了一份课堂讨论的材料，这是为了理解C++（或Java）所必需的C概念的介绍，这是由Chuck Allison为我的公司（MindView, Inc.）创建的，称为“Thinking in C: Foundations for Java and C++”，它介绍了转向C++或Java所必需的C的知识。其中省略了C程序员必须每天面对，而C++和Java语言能让我们避免的繁琐的内容（在Java中，甚至可能消除这些繁琐的工作）。

2

因而，对于“第2版与第1版相比有何不同”这个问题的简要回答是：不同之处不在于版本号是新的，而是进行了重写，有的地方读者甚至无法认出原来的例子和材料。

0.1.1 第2卷的内容是什么

C++标准增加了一些重要的新库，例如String、在标准C++库中的容器和算法，以及模板中的新的复杂性。这些新增的内容和其他更高级的主题被放进本书的第2卷，包括多重继承、异常处理、设计模式和建立和调试稳定系统等内容。

0.1.2 如何得到第2卷

就像当前你手上的这本书一样,《C++编程思想》第2卷完全可以从我的网站 www.BruceEckel.com 上下载。

这个站点还包括这两本书的源代码,以及有关MindView公司提供的CD ROM上其他课堂讨论材料的更新和信息,其中包括:公开课堂讨论、内部培训、辅导课和演示。

0.2 预备知识

在本书第1版中,我假定读者已经学习了C,并至少具有自如阅读的水平。我的重点放在简化我认为比较困难的部分:C++语言。第2版增加了一章,快速地介绍C,并在光盘上提供“Thinking in C”的课堂讨论材料,但是即使如此,我仍然假设读者具有一定的程序设计经验。
3 另外,正如读者可以通过读小说而直接地学会许多新词一样,读者也可以从在本书后面的文字中学习有关于C的大量知识。

0.3 学习C++

我希望本书的读者有和我进入C++时相同的情况:作为一个C程序员,对于编程持有实在而执着的态度。但糟糕的是,我的背景和经验是在硬件层的嵌入式编程方面。在那里,C常常被看做高层语言,它对于位操作是低效率的。后来我发现,自己甚至不是一个好的C程序员,平时总是掩盖了对`malloc()`和`free()`、`setjmp()`和`longjmp()`结构以及其他“复杂”概念的无知,当开始触及这些主题时就竭力回避,而不是努力去获取新的知识。

在我开始致力于学习C++时,当时惟一像样的书是Stroustrup夫子自道式的“专家指南”^①,因此我只好靠自己弄清基本概念。这引出了我的第一本关于C++的书^②,这本书基本上就是直接把我头脑中的经验倒出来而写成的。它的构思是作为读者的指南,引导程序员同时进入C和C++。这本书的两个版本^③都收到了读者的热情反响。

几乎就在《Using C++》出版的同时,我开始讲授这门语言。讲授C++已经变成了我的职业。自1989年以来,在授课时我看到了世界各地听众昏昏欲睡的样子、茫然不知的面容和困惑不解的表情。当我对一些人数不多的人群进行内部培训时,在练习过程中又发现了某些问题。
4 即便那些面带微笑和会心点头的学生,实际上对许多问题也还是糊涂的。通过开创和多年主持“软件开发会议”的C++和Java系列专题,我发现,我和其他讲演者都有一种倾向,即过快地向听众灌输了过多的主题。后来,我做了一些努力,通过区别对待不同层次的听众和提供相关资料的方法,尽量吸引听众。也许这是过分的要求,但是因为我是一个抵触传统教学的人(对于大部分人而言,我相信这种抵触源于厌倦),所以希望我通过努力,使每一个人都能跟得上教学进度。

有一段时间,我编写了大量的教学演示。这样,我结束了通过实验和重复方式进行学习(在设计C++程序的过程中,这也是一项很有用的技术)的阶段。最后,从我多年的教学经验中总结出来的所有内容,形成了一门课程。在课程中,我用一系列分离的、易于理解的步骤

^① Bjarne Stroustrup, *The C++ Programming Language*, Addison-Wesley, 1986 (第1版)。

^② *Using C++*, Osborne/McGraw-Hill 1989。

^③ *Using C++ and C++ Inside & Out*, Osborne/McGraw-Hill 1993。

并采用实地课堂讨论的形式解决学习中的问题（理想的学习情况），并在每次课后面跟随着练习。读者能从www.BruceEckel.com找到我的公开课堂讨论，还可以学习我已经制作成CD ROM的课堂讨论材料。

本书的第1版是作为两学年制课程编写的，并且书中的内容已经在许多不同的课堂讨论上通过了多种形式的检验。我从每次课堂讨论上收集反馈意见，不断地修改和调整内容，直到我感觉到它已经成为一本很好的教材为止。但这本书不仅仅是课堂讨论的分发教材，而且我在其中放入了尽可能多的信息，在结构上使得它能引导读者顺利地通过当前主题和进入下一个主题。另外，这本书也适合于自学读者，能帮助他们尽快地掌握这门新的编程语言。

0.4 目标

在这本书中，我的目标是：

1) 以适当的进度介绍内容。每次将学习向前推进一小步，因此读者能很容易地在继续下一步学习之前消化每个已学过的概念。

5

2) 尽可能使用简短的例子。当然，这有时会妨碍我解决“现实世界”的问题。但是，我发现，当初学者能够掌握例子的每个细节，而不受问题的领域所影响时，他们通常会更有兴趣进行学习。另外，在课堂情况下能达到的接受能力，对代码的长短也有严格的限制。为此，我有时会受到使用“玩具例子”的批评，但是我甘愿承受这一批评，因为这样更有利于取得某些教学法上的效果。

3) 仔细安排描述内容的顺序，不让读者看到还没有揭示的内容。当然，这不是总能做到的；如果出现了这种情况，我将会给出简明的介绍性的描述。

4) 只把对于理解这门语言比较重要的东西介绍给读者，而不是介绍我知道的所有内容。我相信，不同信息的重要性是不同的。有些内容是95%的程序员不需要知道的，这些东西只会迷惑人们，增加他们对该语言复杂性的感觉。举一个C语言的例子，如果我们记住运算符优先表（我是记不住的），我们就可以写更漂亮的代码。但是，如果一定要这样做，反而会使代码的读者或维护者糊涂。所以可以忘掉优先级，当不清楚时使用括号。我们对于C++中的某些内容也可以采取同样的态度，因为我认为这些内容对于写编译器的人更重要，而对于程序员就不是那么重要。

5) 保持每一节的内容充分集中，使得授课时间以及两个练习之间的间隔时间不长。这不仅能使听众保持活跃的和在课堂讨论中精力集中，而且会使他们有更大的成就感。

6) 帮助读者打下坚实的基础，使得他们能充分地理解面对的问题，从而可以顺利地转向学习更困难的课程和书籍（特别是这本书的第2卷）。

6

7) 我尽力不用任何特定厂商的C++版本，因为对于学习编程语言，我不认为特定实现的细节像语言本身一样重要。大部分厂商的文档只适合于他们自己的特定实现。

0.5 各章概要

C++是一个在已有文法上面增加了新的不同特征的语言（因此，它被认为是混合的面向对象的语言）。由于很多人走了学习弯路，因此我们已经开始探索C程序员转移到C++语言特征的方法。因为这是过程型训练思想的自然延伸，所以我决定去理解和重复相同的道路，并通过引出和回答一些问题来加速这一进程，这些问题是当我学习该语言时遇到的和听众在

听我的课时提出来的。

设计这门课时，我始终注意一件事：精练C++语言的学习过程。听众的反馈意见帮助我认识到哪些部分是很难学习的和需要额外解释的。在这个领域中，我曾经雄心勃勃，一次讲解包括了太多的内容。通过讲解过程，我知道了，如果包括大量新特征，就必须对它们全部作出解释，而且学生也特别容易混淆这些新特征。因此，我努力一次只介绍尽可能少的特征，理想的情况是每章一次只介绍一个主要概念。

本书的目标是只在每一章中讲授一个概念，或只讲授一小组相关的概念，用这种方法，不会依赖于其他的特征。这样，在进入下一章的学习之前，学生可以对自己的当前知识融会贯通。为了实现这个目标，我把一些C特征留到后面的章节去介绍，甚至放在比我希望的还要往后的地方介绍。这样做的好处是读者不会因为看到了许多未解释的C++特征被使用而困惑，因此，对该语言的介绍将是和缓的，并且将反映出读者自己消化这些特征时将会采用的方式。

7

下面是本书各章内容的简要说明。

第1章 对象导言。当项目对于维护而言变得太大和太复杂时，就产生了“软件危机”。按程序员们的说法，“我们无法完成那些项目，即便能完成，它们也太昂贵了”。这引出了一些问题，在本章中我将讨论这些问题，并且讨论面向对象程序设计（OOP）的思想和如何运用这一思想解决软件危机问题。这一章引导读者了解OOP的基本概念和特征，介绍分析和设计过程。另外，在这一章中，我还将阐述采用这种语言的好处，提出关于如何转入C++语言领域的建议。

第2章 对象的创建与使用。这一章解释用编译器和库建立程序的过程。它介绍了本书中的第一个C++程序，显示程序如何构造和编译，然后介绍标准C++中的可用对象的基本库。在结束这一章时，我们就对如何用流行的对象库编写C++程序有一个深刻的领会。

第3章 C++中的C。这一章详细综述在C++中使用的C的特征和一些只在C++中使用的特征，还介绍在软件开发领域通用的“制作”工具，并且用它建立了本书中的所有例子（本书的源代码在www.BruceEckel.com中可找到，包含了对每章的makefile）。第3章假设读者已经具有某种过程型程序设计语言的坚实基础，例如Pascal和C语言或者甚至某种形式的Basic（只要读者已经用这种语言编写了大量的代码，特别是函数）。如果读者发现这一章有些难，就应当首先看本书附带的CD ROM上的“Thinking in C”的课堂讨论材料（亦可在www.BruceEckel.com下载）。

8

第4章 数据抽象。C++的大部分特征都围绕着创建新数据类型的能力。这不仅可以提供优质代码组织，而且可以为更强大的OOP能力奠定基础。读者将可以看到如何用将函数放入结构内部的简单过程来实现这一思想，并可以看到如何具体地完成这样的过程和创建什么样的代码。读者还能学会组织代码成为头文件和实现文件的最好方法。

第5章 隐藏实现。通过说明结构中的一些数据和函数是**private**（私有的），可以把它们设置为对于这个新结构类型的用户是不可见的。这意味着能够把下层实现和客户程序员看到的接口隔离开来，这样就容易改变具体实现，而不影响客户代码。另外，C++还引入关键字**class**作为描述新数据类型的更具吸引力的方法，而单词“对象”的意思并不神秘，它是一种美妙的变量。

第6章 初始化与清除。C语言的最通常的一类错误是由于变量未初始化而引起的。C++的构造函数使得程序员能保证他的新数据类型（即“他的类的对象”）的变量总是能被恰当地

初始化。如果他的对象还需要某种方式的清除，他可以保证这个清除动作总是由C++的析构函数来完成。

第7章 函数重载与默认参数。C++可以帮助程序员建立大而复杂的项目。这时，可能会引进使用相同函数名的多个库，还可能会在同一个库中选择具有不同含义的相同的名字。C++采用函数重载使这一问题容易解决。重载允许当参数表不同时重用相同的函数名。默认参数通过自动为某些参数提供默认值，使我们能用不同的方式调用同一个函数。

第8章 常量。本章讨论了**const**和**volatile**关键字，它们在C++中有另外的含义，特别是在类的内部。我们将学习对指针定义使用**const**的含义。本章还说明**const**的含义在类的内部和外部有何不同，以及如何在类的内部创建编译时常量。

9

第9章 内联函数。预处理宏省去了函数调用开销，但是也排除了有价值的C++类型检查。内联函数具有预处理宏和实际函数调用的所有好处。这一章深入地研究了内联函数的实现和使用。

第10章 名字控制。在程序设计中，创建名字是基本的活动，而当项目变大时，名字的数量是无法限制的。C++允许在名字创建、可视性、存储代换和连接方面有大量的控制。这一章将说明如何在C++中用两种技术控制名字。第一，用关键字**static**控制可视性和连接，研究它对于类的特殊含义。另一种在全局范围内更有用的控制名字的技术是C++的**namespace**（名字空间）特征，它允许把全局名字空间划分为不同的区域。

第11章 引用和拷贝构造函数。C++指针的作用和C指针一样，而且具有更强的C++类型检查的好处。C++还提供了另外的处理地址的方法：继Algol和Pascal之后，C++使用了“引用”，允许当程序员使用平常的符号时由编译器来处理地址操作。读者还会遇到拷贝构造函数，它通过按值传送控制将对象传送给函数或从函数中返回的方式。最后，本章还将解释C++指向成员的指针。

第12章 运算符重载。这个特征有时被称为“语法糖 (syntactic sugar)”。由于运算符也可以像函数调用那样使用，这使得程序员在运用类型的语法时更加灵活。在这一章中，读者将了解到，运算符重载只是不同类型的函数调用，并且将学会如何写自己的运算符重载，学会处理参数、类型返回以及确定一个运算符是成员还是友元时的一些易混淆的情况。

10

第13章 动态对象创建。一个空中交通系统将处理多少架飞机？一个CAD系统将需要多少种造型？在一般的程序设计问题中，我们不可能知道程序运行所需要的对象的数量、生命期和类型。在这一章中，我们将学习C++的**new**和**delete**如何漂亮地通过在堆上安全地创建对象而解决上述问题。我们还将看到，**new**和**delete**如何用不同的方法重载，从而使我们能控制如何分配和释放存储。

第14章 继承和组合。数据抽象允许程序员从零开始创建新的类型。通过组合和继承，程序员可以用已存在的类型创建新的类型。用组合方法，程序员可以以老的类型作为零件组装成新的类型；而用继承方法，程序员可以创建已存在类型的一个更特殊的版本。在这一章中，读者将学习这一文法，学习如何重定义函数，以及理解构造和析构对于继承和组合的重要性。

第15章 多态性和虚函数。单靠读者自己，可能要花九个月的时间才能发现和理解OOP的这一基石。通过一些小而简单的例子，读者可以看到如何通过继承创建一个类型族，看到在这个类型族中如何通过公共基类对这个族中的对象进行操作。关键字**virtual**使程序员能够按类属处理这个族中的所有对象，这意味着大块代码将不依赖于特殊类型的信息，因此，程