

Join the discussion @ [p2p.wrox.com](http://p2p.wrox.com)



SECOND EDITION

Professional C++, Second Edition

# C++高级编程 (第2版)



[比] Marc Gregoire  
[美] Nicholas A. Solter 著  
Scott J. Kleper  
侯普秀 郑思遥 译

清华大学出版社

# C++高级编程(第2版)

[比] Marc Gregoire

[美] Nicholas A. Solter 著

Scott J. Kleper

侯普秀 郑思遥 译

清华大学出版社

北京

Marc Gregoire, Nicholas A. Solter, Scott J. Kleper

Professional C++, Second Edition

EISBN: 978-0-470-93244-5

Copyright © 2011 by John Wiley&Sons, Inc., Indianapolis, Indiana

All Rights Reserved. This translation published under license.

本书中文简体字版由 Wiley Publishing, Inc. 授权清华大学出版社出版。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。

北京市版权局著作权合同登记号 图字: 01-2011-6702

本书封面贴有 Wiley 公司防伪标签, 无标签者不得销售。

版权所有, 侵权必究。侵权举报电话: 010-62782989 13701121933

### 图书在版编目(CIP)数据

C++高级编程(第2版)/(比)格莱戈尔(Gregoire, M.), (美)索尔特(Solter, N.A.), (美)凯乐普(Kleper, S.J.)著; 侯普秀, 郑思遥译. —北京: 清华大学出版社, 2012.10

书名原文: Professional C++, Second Edition

ISBN 978-7-302-29897-7

I. ①C… II. ①格… ②索… ③凯… ④侯… ⑤郑… III. ①C语言—程序设计 IV. ①TP312

中国版本图书馆 CIP 数据核字(2012)第 199353 号

责任编辑: 王 军 韩宏志

装帧设计: 牛艳敏

责任校对: 成凤进

责任印制: 张雪娇

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](mailto:c-service@tup.tsinghua.edu.cn)

质 量 反 馈: 010-62772015, [zhiliang@tup.tsinghua.edu.cn](mailto:zhiliang@tup.tsinghua.edu.cn)

印 刷 者: 清华大学印刷厂

装 订 者: 三河市溧源装订厂

经 销: 全国新华书店

开 本: 185mm×260mm 印 张: 59.75 字 数: 1603 千字

版 次: 2012 年 10 月第 2 版 印 次: 2012 年 10 月第 1 次印刷

印 数: 1~4000

定 价: 108.00 元

---

产品编号: 043003-01

# 作者简介



**Marc Gregoire** 是一名软件工程师。他毕业于比利时鲁文的天主教大学，获得计算机科学工程硕士学位。之后，他在该大学获得人工智能的优等硕士学位。完成学业后，他开始为大型软件咨询公司 **Ordina Belgium** 工作。他曾在 **Siemens** 和 **Nokia Siemens Networks** 为大型电信运营商提供有关在 **Solaris** 上运行关键 2G 和 3G 软件的咨询服务。这份工作要求与来自南美、美国、欧洲、中东、非洲和亚洲的国际团队合作。**Marc** 目前在 **Nikon Metrology** 任职，负责开发 3D 扫描软件。

他的主要技术专长是 C/C++，特别是 **Microsoft VC++** 和 **MFC** 框架。除了 C/C++ 之外，**Marc** 还喜欢 **C#**，并且会用 **PHP** 创建网页。除了在 **Windows** 上开发的主要兴趣之外，他还擅长在 **Linux** 平台上开发 24x7 运行的 C++ 程序；例如 **EIB** 家庭自动化监控软件。2007 年 4 月，因为在 **Visual C++** 方面的专业才能他获得了年度 **Microsoft MVP** 称号。**Marc** 还是 **CodeGuru** 论坛的活跃分子(id 为 **Marc G**)，并且为 **CodeGuru** 撰写了一些文章和 **FAQ** 条目。他还编写了一些自由软件和共享软件，并通过他的网站 [www.nuonsoft.com](http://www.nuonsoft.com) 发布。他还在 [www.nuonsoft.com/blog/](http://www.nuonsoft.com/blog/) 维护了一个博客。



**Nicholas A. Solter** 是一名计算机程序员，开发的软件范围很广，包括系统软件、游戏、网络服务和其他很多类型。他在 **Sun Microsystem** 的高可用集群上所做的工作获得了 3 项专利，还就此在国际并行和分布式处理会议上发表了一篇技术论文。在 **Sun** 的时候，他还喜欢参与 **OpenSolaris**，他还是 **OpenSolaris Bible** (**Wiley**, 2009) 的第一作者。现在重新从事 **Web** 开发，他很开心地再次和 **Scott** 一起在 **Context Optional** 工作。

**Nick** 在斯坦福大学学习计算机科学，他在这所大学获得了本科和理科硕士学位，他的主要研究领域是计算机系统。他曾在富勒顿社区大学讲授了一年的 C++ 课程。

**Nick** 和他的妻子和两个孩子生活在美丽的科罗拉多，他在科罗拉多享受着雪上运动的乐趣。



**Scott J. Klerper** 在小学就开始了他的编程生涯，那时他在 **Tandy TRS-80** 上用 **BASIC** 语言编写探险游戏。作为他所在高中的 **Mac** 迷，**Scott** 转向了更高级的语言，并且发布了一些屡获殊荣的共享软件。

**Scott** 加入了斯坦福大学，并且在这所大学获得了本科和计算机科学的理学硕士学位，主要研究领域是人机交互。在上大学的时候，**Scott** 是一门涉及编程入门、面向对象设计、数据结构、**GUI** 框架和小组项目的课程的助教。他之后在斯坦福的一门课程采用这本书作为课本。

毕业后，**Scott** 是几家公司创始团队中的首席工程师。2006 年，**Scott** 与他人合伙创建了 **Context Optional, Inc.**，这是一家提供社会营销技术的市场领先的供应商。

在工作之余，**Scott** 还热衷于在线购物、阅读和弹吉他。

# 技术编辑简介

**Joseph M. Newcomer** 在卡内基梅隆大学(CMU)获得计算机科学博士学位。之后在 CMU 任教，担任 CMU 软件工程学院的一位资深科学家，同时他还是一位独立软件开发者。他连续 16 年获得 C/C++/MFC 领域的 Microsoft MVP。他是两本 Microsoft Windows 编程书籍的合作者：*Win32 Programming*(与 Brent Rector 合著, Addison-Wesley, 1997)和 *Developing Windows NT Device Drivers*(与 Ed Dekker 合著, Addison-Wesley, 1999)。Joseph 是上世纪 70 年代初桌面出版和数据库出版领域的技术先驱之一。

# 致 谢

在此要感谢 John Wiley & Sons & Wrox Press 的编辑和产品团队，感谢他们的支持。特别要感谢 Wiley 的执行编辑 Robert Elliott 让我有机会撰写本书；感谢 Wiley 的高级项目编辑 Adaobi Obi Tulton 对这个项目的管理。

还要感谢本书第一版的作者 Nicholas A. Solter 和 Scott J. Kleper，你们让我这本第二版有了很好的工作基础。

特别感谢技术编辑 Joseph M. Newcomer，感谢他一丝不苟地对本书进行审阅。他提出很多具有建设性的评论和意见，将本书的质量推进到更高的水平。

衷心感谢 Ordina Belgium([www.ordina.be](http://www.ordina.be))，这是我在编写这本书的时候所服务的公司，感谢这家公司给予的慷慨支持。

当然，还要感谢女友 Zulija N. 以及我的父母和兄长对我的支持和关爱，你们的支持对这本书的完成至关重要。

最后，我要感谢各位亲爱的读者的信任；本书也必将使各位读者取得圆满的学习效果。

# 前 言

多年来，C++都是编写性能卓越、功能强大的企业级面向对象程序的事实标准语言。尽管 C++语言已经风靡全球，但是这个语言却非常难完全掌握。专业 C++程序员使用了一些简单但高效的技术，这些技术并未出现在传统教材中；即使是经验丰富的 C++程序员也不了解 C++中的某些很有用的特性。

编程书籍往往重点描述语言的语法，而不是语言在真实世界中的应用。典型的 C++教材在每一章中介绍了语言中的大部分知识，讲解语法并列举示例。本书不遵循这个模式。本书并不是讲解语言的大量细节并给出少量真实世界的场景，而是教您如何在真实世界中使用 C++。本书还会披露一些鲜为人知的特性，使用这些特性可以让编程更简单；还讲解了可重用的编码模式，模式是区分编程新手和专业程序员的标志。

## 本书读者对象

就算您使用了多年的 C++，您仍然有可能不熟悉 C++的一些高级特性，或者仍然不具有使用这个语言的完整能力。也许您编写过实用的 C++代码，但还想学习更多有关使用 C++设计和良好的编程风格的内容。也许您是 C++新手，想在入门的时候就掌握“正确”的编程方式。本书能满足上述需求，能将您的 C++技能提升到专业水准。

由于本书专注于从对 C++具有基本或中等了解水平蜕变为一名专业的 C++程序员的过程，所以本书假设您对这个语言具有一定程度的认识。第 1 章涵盖了 C++的一些基础知识，可以当作复习材料，但是不能替代实际的语言培训和语言使用手册。如果您刚刚开始接触 C++，但有很丰富的 C 语言经验，那么您应该能从第 1 章获得大部分所需的知识。

不管是那种情况，您都应该有很好的编程基础。您应该知道循环、函数和变量。您应该知道如何组织一个程序，而且应该熟悉基本技术，例如递归。您应该了解一些常见数据结构，例如哈希表和队列，以及有用的算法，例如排序和搜索。您不需要预先了解有关面向对象编程的知识——这是第 3 章讲解的内容。

您还应该熟悉开发代码时使用的编译器。这本书没有提供使用具体编译器的指南。请参阅编译器自带的指南。

## 本书主要内容

阅读本书是学习 C++语言的一种方法，通过阅读本书既能提升编码质量，又能提升编程效率。本书贯穿了对 C++11 新特性的讨论。这些新的 C++11 特性并没有分离在几个章节中，而是贯穿于

全书，在有必要的情况下，几乎所有的例子都已经更新为使用这些新特性。



提示：C++11 的完整 ISO 名称为 ISO/IEC 14882:2011(E)，是最新的 C++ 标准。C++11 标准化的过程从本世纪初就开始了。在这个过程中，标准的草案曾被称为 C++0x。

本书不仅讲解 C++ 语法和语言特性，还强调了编程方法学、可重用的设计模式以及良好的编程风格。本书讲解的方法学覆盖了整个软件开发过程——从设计和编码，到测试、调试以及团队合作。这种方法可以让您掌握 C++ 语言及其语言的独特特性，还能够在大型软件开发中充分利用 C++ 语言的强大功能。

想象一下如果有人学习了 C++ 所有的语法但是没有看过一个 C++ 例子的情形。他所了解的知识会让他处于非常危险的境地。如果没有示例的引导，他可能会认为所有源代码都要放在程序的 `main()` 函数中，还有可能认为所有变量都应该为全局变量——这些都不是良好的编程实践。

专业的 C++ 程序员除了理解语法之外，还要正确理解语言的使用方式。他们知道良好设计的重要性、面向对象编程的理论以及使用现有库的最佳方式。他们还开发了大量有用的代码并了解可重用的思想。

通过阅读和理解本书的内容，您也能成为一名专业的 C++ 程序员。您在 C++ 方面的知识会得到扩充，将会接触到鲜为人知的和常被误解的语言特性。您还将领略面向对象的设计，掌握卓越的调试技能。最重要的或许是，通过本书的阅读，您会了解到大量“可重用”思想，并将这种思想贯彻到日常工作中。

有很多好的理由让您努力成为一名专业的 C++ 程序员，而非只是泛泛了解 C++ 的程序员。了解语言的真正工作原理可以提升代码的质量。了解不同的编程方法学和过程可以让您更好地和团队合作。探索可重用的库和常用的设计模式可以提升您日常工作的效率，并帮助您避免重新造轮子。所有这些学习课程都在帮助您成为更优秀的程序员，同时成为更有价值的雇员。尽管这本书不能保证您升职，但是肯定不会有坏处。

## 本书结构

本书的正文部分包括四个部分，另有三个附录。

第 I 部分是一个 C++ 基础速成教程，确保读者掌握 C++ 的基础知识。在速成教程之后，第 I 部分介绍了 C++ 设计方法学。您会了解到设计的重要性、面向对象方法学、代码重用的重要性以及如何编写清晰易读的 C++ 代码。

第 II 部分从专业的角度概述 C++ 技术。您将学习如何创建可重用的类，以及如何利用重要的语言特性，例如继承。您还会学习这个语言的一些不同寻常之处、输入和输出技术、专业级别的错误处理以及字符串和正则表达式的使用。这一部分还讲解了 C++ 标准库，包括容器、迭代器、算法以及如何定制和扩充标准库以满足自己的需求。您还会学习标准中其他一些库，例如处理时间的库和处理随机数的库。

第 III 部分讲解如何最大限度地使用 C++。本书这一部分揭示了 C++ 中神秘的部分，并且描述



了如何使用这些更高级的特性。您会学习在 C++ 中如何恰到好处地管理内存、如何实现高级的运算符重载、如何编写模板以及如何通过多线程编程来充分利用多处理器和多核系统。

第 IV 部分重点介绍如何编写企业级质量的软件。您会学习当今编程组织使用的工程实践；软件测试概念，例如单元测试和回归测试；C++ 程序的调试技术；如何编写高效的 C++ 代码；跨语言和跨平台编程的解决方案；以及如何在代码中综合使用设计技术、框架和概念性的面向对象设计模式。

本书最后三个附录。附录 A 列出在 C++ 技术面试中取得成功的指南(按章分解内容)，附录 B 是带注解的参考文献列表，附录 C 则总结了标准中的 C++ 头文件。在本书网站 [www.wrox.com](http://www.wrox.com) 上的补充材料中可以找到一份实用的 C++ 标准库参考指南。这个参考指南在本书中称为“网站上的标准库参考资源”。

## 使用本书的条件

要使用这本书，您只需要一台带有 C++ 编译器的计算机。这本书只关注 C++ 中的标准部分，没有任何编译器厂商相关的扩展。

本书包含了 C++11 标准引入的新特性。在撰写本书的时候，还没有编译器能支持 C++11 所有的新特性。本书通过运行在 Linux 上的 GCC 4.6 和运行在 Windows 上的 Microsoft Visual C++ 2010 测试所有示例代码。GCC 4.6 支持很多新特性，但并非能支持所有的新特性。使用了以下 C++11 特性的例子不能通过当前的 GCC 4.6 进行测试：

- final 和 override 关键字
- 用户定义的文本
- 类内非静态数据成员的初始化
- 模板别名
- 构造函数委托
- 构造函数继承

本书用于测试的第二个编译器是 Microsoft Visual C++ 2010，然而，这个编译器支持的 C++11 特性比 GCC 4.6 还少。VC++ 2010 中缺少的特性主要包括基于范围的 for 循环、统一初始化、可变参数模板以及线程库等。



本书网站 [www.wrox.com](http://www.wrox.com) 上的附加材料中包含一个额外的附录，这个附录列出了所有的 C++11 特性，以及某个编译器是否支持某项特性的信息。

## 约定

为了帮助您更好地理解正文内容，全书中使用了一些约定。



像这样的框中包含了和周围正文信息相关的重要的、应该牢记的信息。



与当前讨论的内容相关的技巧、提示、小窍门和旁白放在这样的框中。

## 源代码

在读者学习本书中的示例时，可以手动输入所有的代码，也可以使用本书附带的源代码文件。本书使用的所有源代码都可以从本书合作站点 <http://www.wrox.com/>或 <http://www.tupwk.com.cn/download> 上下载。登录到站点 <http://www.wrox.com/>，使用 Search 工具或使用书名列表就可以找到本书。接着单击本书细目页面上的 Download Code 链接，就可以获得所有源代码。



**注释：**由于许多图书的标题都很类似，因此按 ISBN 搜索是最简单的，本书英文版的 ISBN 是 978-0-470-93244-5。

在下载了代码后，只需用自己喜欢的解压缩软件对它进行解压缩即可。另外，也可以进入 <http://www.wrox.com/dynamic/books/download.aspx> 上的 Wrox 代码下载主页，查看本书和其他 Wrox 图书的所有代码。

## 勘误表

尽管我们已经尽了各种努力来保证文章或代码中不出现错误，但是错误总是难免的，如果您在本书中找到了错误，例如拼写错误或代码错误，请告诉我们，我们将非常感激。通过勘误表，可以让其他读者避免受挫，当然，这还有助于提供更高质量的信息。

请给 [wkservice@vip.163.com](mailto:wkservice@vip.163.com) 发电子邮件，我们就会检查您的信息，如果是正确的，我们将在本书的后续版本中采用。

要在网站上找到本书的勘误表，可以登录 <http://www.wrox.com>，通过 Search 工具或书名列表查找本书，然后在本书的细目页面上，单击 Book Errata 链接。在这个页面上可以查看到 Wrox 编辑已提交和粘贴的所有勘误项。完整的图书列表还包括每本书的勘误表，网址是 [www.wrox.com/misc-pages/booklist.shtml](http://www.wrox.com/misc-pages/booklist.shtml)。

## P2P.WROX.COM

要与作者和同行讨论，请加入 [p2p.wrox.com](http://p2p.wrox.com) 上的 P2P 论坛。这个论坛是一个基于 Web 的系统，便于您张贴与 Wrox 图书相关的消息和相关技术，与其他读者和技术用户交流心得。该论坛提供了订阅功能，当论坛上有新的消息时，它可以给您传送感兴趣的论题。Wrox 作者、编辑和其他业界专家和读者都会到这个论坛上来探讨问题。

在 <http://p2p.wrox.com> 上,有许多不同的论坛,它们不仅有助于阅读本书,还有助于开发自己的应用程序。要加入论坛,可以遵循下面的步骤:

- (1) 进入 [p2p.wrox.com](http://p2p.wrox.com), 单击 Register 链接。
- (2) 阅读使用协议, 并单击 Agree 按钮。
- (3) 填写加入该论坛所需要的信息和自己希望提供的其他信息, 单击 Submit 按钮。
- (4) 您会收到一封电子邮件, 其中的信息描述了如何验证账户, 完成加入过程。



**注释:** 不加入 P2P 也可以阅读论坛上的消息, 但要张贴自己的消息, 就必须加入该论坛。

加入论坛后, 就可以张贴新消息, 响应其他用户张贴的消息。可以随时在 Web 上阅读消息。如果要让该网站给自己发送特定论坛中的消息, 可以单击论坛列表中该论坛名旁边的 **Subscribe to this Forum** 图标。

关于使用 Wrox P2P 的更多信息, 可阅读 P2P FAQ, 了解论坛软件的工作情况以及 P2P 和 Wrox 图书的许多常见问题。要阅读 FAQ, 可以在任意 P2P 页面上单击 FAQ 链接。

# 目 录

第 I 部分 专业的 C++ 简介	
第 1 章 C++ 速成	3
1.1 C++ 基础知识	3
1.1.1 小程序的“hello world”	3
1.1.2 名称空间	6
1.1.3 变量	8
1.1.4 运算符	9
1.1.5 类型	12
1.1.6 条件	14
1.1.7 循环	16
1.1.8 数组	18
1.1.9 函数	19
1.2 深入研究 C++	21
1.2.1 指针以及动态内存	21
1.2.2 C++ 中的字符串	24
1.2.3 引用	25
1.2.4 异常	26
1.2.5 const 的多种用法	27
1.3 作为面向对象语言的 C++	28
1.4 标准库	30
1.5 第一个有用的 C++ 程序	31
1.5.1 雇员记录系统	31
1.5.2 Employee 类	32
1.5.3 Database 类	35
1.5.4 用户界面	38
1.5.5 评估程序	40
1.6 本章小结	41
第 2 章 设计专业的 C++ 程序	43
2.1 程序设计概述	43
2.2 程序设计的重要性	44
2.3 C++ 设计的特点	46
2.4 C++ 设计的两个原则	47
2.4.1 抽象	47
2.4.2 重用	48
2.5 重用代码	49
2.5.1 关于术语的说明	50
2.5.2 决定是否重用代码	50
2.5.3 重用代码的策略	52
2.5.4 绑定第三方应用程序	56
2.5.5 开放源代码库	56
2.5.6 C++ 标准库	57
2.6 设计模式以及技巧	58
2.7 设计一个国际象棋程序	58
2.7.1 需求	58
2.7.2 设计步骤	59
2.8 本章小结	63
第 3 章 面向对象设计	65
3.1 过程化的思考方式	65
3.2 面向对象思想	66
3.2.1 类	66
3.2.2 组件	66
3.2.3 属性	67
3.2.4 行为	67
3.2.5 综合考虑	67
3.3 生活在对象世界里	68
3.3.1 过度使用对象	69
3.3.2 过于通用的对象	69
3.4 对象之间的关系	70
3.4.1 “有一个”关系	70
3.4.2 “是一个”关系(继承)	71

3.4.3	“有一个”与“是一个” 的区别	73	5.5.3	使用自定义异常	108
3.4.4	Not-a 关系	75	5.6	格式	109
3.4.5	层次结构	76	5.6.1	关于大括号对齐的争论	109
3.4.6	多重继承	77	5.6.2	关于空格以及圆括号的争论	110
3.4.7	混入类	78	5.6.3	空格以及制表符	110
3.5	抽象	78	5.7	风格的挑战	110
3.5.1	接口与实现	78	5.8	本章小结	111
3.5.2	决定公开的接口	78	<b>第II部分 专业的C++编码方法</b>		
3.5.3	设计成功的抽象	80	<b>第6章 熟悉类和对象</b>		<b>115</b>
3.6	本章小结	81	6.1	电子表格示例介绍	115
<b>第4章 设计可重用代码</b>		<b>83</b>	6.2	编写类	116
4.1	重用哲学	83	6.2.1	类定义	116
4.2	如何设计可重用的代码	84	6.2.2	定义方法	118
4.2.1	使用抽象	84	6.2.3	使用对象	122
4.2.2	构建理想的重用代码	85	6.3	对象的生命周期	123
4.2.3	设计有用的接口	89	6.3.1	创建对象	124
4.2.4	协调通用性以及使用性	92	6.3.2	销毁对象	139
4.3	本章小结	93	6.3.3	对象赋值	140
<b>第5章 编码风格</b>		<b>95</b>	6.3.4	复制以及赋值的区别	142
5.1	良好外观的重要性	95	6.4	本章小结	144
5.1.1	事先考虑	95	<b>第7章 掌握类与对象</b>		<b>145</b>
5.1.2	良好风格的元素	96	7.1	对象的动态内存分配	145
5.2	为代码编写文档	96	7.1.1	Spreadsheet 类	146
5.2.1	使用注释的原因	96	7.1.2	使用析构函数释放内存	147
5.2.2	注释的风格	99	7.1.3	处理复制以及赋值	148
5.2.3	本书的注释	103	7.2	定义数据成员的类型	155
5.3	分解	103	7.2.1	静态数据成员	155
5.3.1	通过重构分解	104	7.2.2	常量数据成员	157
5.3.2	通过设计分解	104	7.2.3	引用数据成员	158
5.3.3	本书中的分解	104	7.2.4	常量引用数据成员	159
5.4	命名	104	7.3	与方法有关的更多内容	159
5.4.1	选择一个恰当的名称	105	7.3.1	静态方法	159
5.4.2	命名约定	105	7.3.2	const 方法	160
5.5	使用具有风格的语言特性	107	7.3.3	方法重载	162
5.5.1	使用常量	108	7.3.4	默认参数	163
5.5.2	使用引用代替指针	108	7.3.5	内联方法	164
			7.4	嵌套类	165

7.5	类内的枚举类型	167	8.6.3	重写方法时的特殊情况	218
7.6	友元	168	8.6.4	子类中的复制构造函数以及 赋值运算符	224
7.7	运算符重载	169	8.6.5	virtual 的真相	225
7.7.1	示例: 为 SpreadsheetCell 实现加法	169	8.6.6	运行时类型工具	228
7.7.2	重载算术运算符	174	8.6.7	非 public 继承	229
7.7.3	重载比较运算符	176	8.6.8	虚基类	230
7.7.4	创建具有运算符重载的类型	177	8.7	本章小结	231
7.8	创建稳定的接口	178	<b>第 9 章</b>	<b>理解灵活而奇特的 C++</b>	<b>233</b>
7.9	本章小结	181	9.1	引用	233
<b>第 8 章</b>	<b>揭秘继承技术</b>	<b>183</b>	9.1.1	引用变量	234
8.1	使用继承构建类	183	9.1.2	引用数据成员	236
8.1.1	扩展类	184	9.1.3	引用参数	236
8.1.2	重写方法	187	9.1.4	引用作为返回值	238
8.2	使用继承重用代码	190	9.1.5	使用引用还是指针	238
8.2.1	WeatherPrediction 类	190	9.1.6	右值引用(仅限 C++11)	241
8.2.2	在子类中添加功能	191	9.2	关键字的疑问	246
8.2.3	在子类中替换功能	192	9.2.1	const 关键字	246
8.3	利用父类	193	9.2.2	static 关键字	250
8.3.1	父类构造函数	193	9.2.3	非局部变量的初始化顺序	254
8.3.2	父类的析构函数	195	9.3	类型以及类型转换	254
8.3.3	使用父类方法	196	9.3.1	typedef	254
8.3.4	向上转型以及向下转型	198	9.3.2	函数指针 typedef	255
8.4	继承与多态性	200	9.3.3	类型别名(仅限 C++11)	256
8.4.1	回到电子表格	200	9.3.4	类型转换	257
8.4.2	设计多态性的电子表格 单元格	200	9.4	作用域解析	261
8.4.3	电子表格单元格的基类	201	9.5	C++11	262
8.4.4	独立的子类	203	9.5.1	统一初始化	262
8.4.5	利用多态性	205	9.5.2	可选函数语法	264
8.4.6	考虑将来	206	9.5.3	空指针文本	265
8.5	多重继承	207	9.5.4	尖括号	265
8.5.1	从多个类继承	207	9.5.5	初始化列表	266
8.5.2	名称冲突以及歧义基类	208	9.5.6	显式转换运算符	266
8.6	有趣而晦涩的继承问题	211	9.5.7	特性	267
8.6.1	修改重写方法的特征	211	9.5.8	用户定义的字面量	268
8.6.2	继承构造函数(仅限 C++11)	215	9.6	头文件	270
			9.7	C 的实用工具	271

9.7.1	变长参数列表	271	11.2.3	本地化	318
9.7.2	预处理器宏	273	11.2.4	智能指针	318
9.8	本章小结	274	11.2.5	异常	318
<b>第 10 章</b>	<b>错误处理</b>	<b>275</b>	11.2.6	数学工具	319
10.1	错误与异常	275	11.2.7	时间工具(仅限 C++11)	319
10.1.1	异常的含义	276	11.2.8	随机数(仅限 C++11)	319
10.1.2	C++中异常的优点	276	11.2.9	编译时有理数运算 (仅限 C++11)	319
10.1.3	C++中异常的缺点	277	11.2.10	元组(仅限 C++11)	319
10.1.4	我们的建议	277	11.2.11	正则表达式 (仅限 C++11)	320
10.2	异常机制	277	11.2.12	标准模板库	320
10.2.1	抛出并捕获异常	278	11.2.13	STL 算法	326
10.2.2	异常类型	281	11.2.14	STL 中还缺什么	333
10.2.3	抛出并捕获多个异常	283	11.3	本章小结	333
10.2.4	未捕获的异常	285	<b>第 12 章</b>	<b>理解容器与迭代器</b>	<b>335</b>
10.2.5	抛出列表	287	12.1	容器概述	335
10.3	异常与多态性	291	12.1.1	元素的需求	336
10.3.1	标准异常体系	291	12.1.2	异常和错误检查	338
10.3.2	在类层次结构中捕获异常	293	12.1.3	迭代器	338
10.3.3	编写自己的异常类	294	12.1.4	C++11 的变化	340
10.3.4	嵌套异常(仅限 C++11)	297	12.2	顺序容器	342
10.4	堆栈的释放与清理	299	12.2.1	vector	342
10.4.1	使用智能指针	300	12.2.2	vector<bool>特化	359
10.4.2	捕获、清理并重新抛出	301	12.2.3	deque	359
10.5	常见的错误处理问题	301	12.2.4	list	360
10.5.1	内存分配错误	301	12.2.5	array(仅限 C++11)	364
10.5.2	构造函数中的错误	304	12.2.6	forward_list(仅限 C++11)	364
10.5.3	构造函数的 function-try-blocks	306	12.3	容器适配器	366
10.5.4	析构函数中的错误	308	12.3.1	queue	366
10.6	综合应用	308	12.3.2	priority_queue	369
10.7	本章小结	312	12.3.3	stack	372
<b>第 11 章</b>	<b>深入探讨标准库</b>	<b>313</b>	12.4	关联容器	373
11.1	编码原则	314	12.4.1	pair 工具类	373
11.1.1	使用模板	314	12.4.2	map	374
11.1.2	使用运算符重载	317	12.4.3	multimap	382
11.2	C++标准库概述	317	12.4.4	set	385
11.2.1	字符串	317	12.4.5	multiset	387
11.2.2	I/O 流	318			

12.5	无序关联容器/哈希表 (仅限 C++11).....	387	13.4.4	排序算法.....	436
12.5.1	哈希函数.....	387	13.4.5	集合算法.....	438
12.5.2	unordered_map.....	388	13.5	算法示例: 审核选民登记.....	440
12.5.3	unordered_multimap.....	391	13.5.1	选民登记审核问题描述.....	440
12.5.4	unordered_set/unordered_ multiset.....	391	13.5.2	auditVoterRolls 函数.....	440
12.6	其他容器.....	391	13.5.3	getDuplicates 函数.....	441
12.6.1	标准 C 风格数组.....	392	13.5.4	测试 auditVoterRolls 函数.....	443
12.6.2	string.....	392	13.6	本章小结.....	443
12.6.3	流.....	393	<b>第 14 章</b>	<b>使用字符串与正则表达式.....</b>	<b>445</b>
12.6.4	bitset.....	393	14.1	动态字符串.....	445
12.7	本章小结.....	397	14.1.1	C 风格字符串.....	446
<b>第 13 章</b>	<b>掌握 STL 算法.....</b>	<b>399</b>	14.1.2	字符串字面量.....	447
13.1	算法概述.....	399	14.1.3	C++ string 类.....	448
13.1.1	find 和 find_if 算法.....	400	14.1.4	原始字符串字面量 (仅限 C++11).....	451
13.1.2	accumulate 算法.....	402	14.2	本地化.....	452
13.1.3	在算法中使用 C++11 的移动 语义.....	404	14.2.1	本地化字符串字面量.....	452
13.2	lambda 表达式(仅限 C++11).....	404	14.2.2	宽字符.....	453
13.2.1	语法.....	404	14.2.3	非西方字符集.....	453
13.2.2	捕捉块.....	406	14.2.4	locale 和 facet.....	455
13.2.3	将 lambda 表达式用作 返回值.....	406	14.3	正则表达式(仅限 C++11).....	457
13.2.4	将 lambda 表达式用作参数.....	407	14.3.1	ECMAScript 语法.....	458
13.2.5	示例.....	408	14.3.2	regex 库.....	463
13.3	函数对象.....	410	14.3.3	regex_match().....	464
13.3.1	算术函数对象.....	410	14.3.4	regex_search().....	467
13.3.2	比较函数对象.....	411	14.3.5	regex_iterator.....	468
13.3.3	逻辑函数对象.....	412	14.3.6	regex_token_iterator.....	469
13.3.4	按位函数对象 (仅限 C++11).....	412	14.3.7	regex_replace().....	472
13.3.5	函数对象适配器.....	413	14.4	本章小结.....	475
13.3.6	编写自己的函数对象.....	419	<b>第 15 章</b>	<b>C++ I/O 揭秘.....</b>	<b>477</b>
13.4	算法详解.....	420	15.1	使用流.....	477
13.4.1	工具算法.....	421	15.1.1	流的含义.....	478
13.4.2	非修改算法.....	422	15.1.2	流的来源和目标.....	478
13.4.3	修改算法.....	428	15.1.3	流式输出.....	479
			15.1.4	流式输入.....	483
			15.1.5	对象的输入输出.....	489
			15.2	字符串流.....	491
			15.3	文件流.....	492



15.3.1	通过 seek()和 tell()在文件中 转移	493	18.1.2	运算符重载的限制	568
15.3.2	将流连接在一起	495	18.1.3	运算符重载的决策	568
15.4	双向 I/O	496	18.1.4	不要重载的运算符	570
15.5	本章小结	497	18.1.5	可重载运算符小结	571
<b>第 16 章</b>	<b>其他库工具</b>	<b>499</b>	18.1.6	右值引用(仅限 C++11)	574
16.1	STD::FUNCTION	499	<b>18.2</b>	<b>重载算术运算符</b>	<b>574</b>
16.2	有理数	501	18.2.1	重载一元负号和一元正号	574
16.3	Chrono 库	503	18.2.2	重载递增和递减运算符	575
16.3.1	持续时间	503	<b>18.3</b>	<b>重载按位运算符和二元逻辑 运算符</b>	<b>577</b>
16.3.2	时钟	507	<b>18.4</b>	<b>重载插入运算符和提取 运算符</b>	<b>577</b>
16.3.3	时点	508	<b>18.5</b>	<b>重载下标运算符</b>	<b>579</b>
16.4	生成随机数	509	18.5.1	通过 operator[]提供只读 访问	582
16.4.1	随机数引擎	510	18.5.2	非整数数组索引	583
16.4.2	随机数引擎适配器	512	<b>18.6</b>	<b>重载函数调用运算符</b>	<b>583</b>
16.4.3	预定义的引擎和引擎 适配器	512	<b>18.7</b>	<b>重载解除引用运算符</b>	<b>585</b>
16.4.4	生成随机数	513	18.7.1	实现 operator*	586
16.4.5	随机数分布	514	18.7.2	实现 operator->	587
16.5	元组	517	18.7.3	operator->*的含义	588
16.6	本章小结	520	<b>18.8</b>	<b>编写转换运算符</b>	<b>588</b>
<b>第 17 章</b>	<b>自定义和扩展 STL</b>	<b>521</b>	18.8.1	转换运算符的多义性问题	590
17.1	分配器	521	18.8.2	用于布尔表达式的转换	591
17.2	迭代器适配器	522	<b>18.9</b>	<b>重载内存分配和释放运算符</b>	<b>593</b>
17.2.1	反向迭代器	522	18.9.1	new 和 delete 的工作原理	593
17.2.2	流迭代器	524	18.9.2	重载 operator new 和 operator delete	595
17.2.3	插入迭代器	524	18.9.3	重载带有额外参数的 operator new 和 operator delete	597
17.2.4	移动迭代器(仅限 C++11)	525	18.9.4	显式地删除/默认化 operator new 和 operator delete(仅限 C++11)	599
17.3	扩展 STL	527	<b>18.10</b>	<b>本章小结</b>	<b>599</b>
17.3.1	扩展 STL 的原因	527	<b>第 19 章</b>	<b>利用模板编写泛型代码</b>	<b>601</b>
17.3.2	编写一个 STL 算法	527	19.1	模板概述	602
17.3.3	编写一个 STL 容器	530	19.2	类模板	602
17.4	本章小结	564			
<b>第 III 部分 掌握 C++的高级特性</b>					
<b>第 18 章</b>	<b>C++运算符重载</b>	<b>567</b>			
18.1	运算符重载概述	567			
18.1.1	重载运算符的原因	568			