Professional C++, 4th Edition

C++高级编程

(第4版)

[美] 马克·葛瑞格尔(Marc Gregoire) 著
徐志超 曹 瑜 译

清华大学出版社

C++高级编程

(第4版)

[美] 马克 · 葛瑞格尔(Marc Gregoire) 著

徐志超　曹瑜　　译

清华大学出版社

北　京

Professional C++, 4th Edition
Marc Gregoire
EISBN：978-1-119-42130-6

图书在版编目(CIP)数据

C++高级编程：第4版 / (美)马克·葛瑞格尔(Marc Gregoire) 著；徐志超，曹瑜 译. —北京：清华大学出版社，2019
书名原文：Professional C++, 4th Edition
ISBN 978-7-302-52631-5

Ⅰ. ①C…　Ⅱ. ①马…　②徐…　③曹…　Ⅲ. ①C++语言—程序设计　Ⅳ. ①TP312.8

中国版本图书馆 CIP 数据核字(2019)第 046888 号

责任编辑：王　军
装帧设计：孔祥峰
责任校对：成凤进
责任印制：刘海龙

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：清华大学印刷厂
经　　销：全国新华书店
开　　本：190mm×260mm　　印　　张：46.75　　字　　数：1580 千字
版　　次：2019 年 4 月第 1 版　　印　　次：2019 年 4 月第 1 次印刷
定　　价：138.00 元

产品编号：078948-01

作者简介

Marc Gregoire是一位Microsoft Visual C++ MVP，是一名软件工程师和开发人员，是比利时C++用户组的创始人，现供职于Nikon Metrology，负责开发3D激光扫描软件。Marc曾在Siemens 和Nokia Siemens Networks开发关键的2G和3G电信软件。Marc是本书前两版的作者，也曾担任其他多本书籍的技术编辑。

译 者 序

在中国大陆的程序员圈子中，常将C++读作“C加加”；而西方国家的程序员常将C++读作“C plus plus”或“CPP”。C++最初作为C语言的增强版出现，此后不断融入新特性，比如虚函数、运算符重载、多重继承、模板、异常、RTTI、结构化绑定、嵌套的名称空间等。

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，但这门语言却极难完全掌握，即使是经验丰富的C++程序员也未必了解C++中某些很有用的特性。

要成为一名专业的C++程序员，必须扎实理解C++语言的工作原理，了解不同编程方法论和软件开发过程——从设计和编码，到测试、调试等，以便更好地和团队协作。要知道良好设计的重要性，领略面向对象编程的理论，掌握卓越的调试技能，探索可重用的库和常用的设计模式，了解可重用的思想，以大幅提升日常工作效率。

这本讲解C++17的著作将帮助读者全面透彻地掌握C++语言的功能，包罗C++语言的一切，分享真实范例，展现C++17的新工具和功能，详述如何在真实世界中使用C++，揭示最新版C++带来的显著变化，解密C++中鲜为人知的特性，探索编程方法论、可重用的设计模式和良好的编程风格，阐述如何设计可充分利用C++语言功能的高效解决方案。本书深入探讨C++语言功能集的更复杂元素，并讲解避开常见陷阱的技巧。本书提供详尽的编程指南，紧贴实际，是编程人员深入挖掘C++的理想工具。

本书包括5部分。第I部分是C++基础速成教程，第II部分介绍C++设计方法论，第III部分从专业角度分析C++编程技术，第IV部分讲解如何真正掌握C++的高级功能，第V部分重点介绍C++软件工程技术。最后的附录A列出了在C++技术面试中取得成功的指南，附录B列出带注解的参考文献，附录C总结标准的C++头文件，附录D则简要介绍UML。

对于这本经典之作，译者在翻译过程中力求忠于原文，再现原文风格，但是鉴于译者水平有限，错误和失误在所难免，如有任何意见和建议，请不吝指正。除封面署名的译者外，何益枝、吴秀莲、张继春、段治勋、郑玉花、兰翠平、段国鑫也参与了本书的翻译工作。

最后，希望读者通过阅读本书能早日步入C++编程的殿堂，领略C++语言之美！

作者简介

Marc Gregoire 是一名软件工程师，毕业于比利时鲁文大学，拥有计算机科学工程硕士学位。之后，他在鲁文大学获得人工智能专业的优等硕士学位。完成学业后，他开始为软件咨询公司 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 目前担任 Nikon Metrology (www.nikonmetrology.com)的软件架构师；Nikon Metrology 是 Nikon 的分公司，是领先的精密光学仪器和 3D 扫描软件供应商。

Marc 的主要技术专长是 C/C++，特别是 Microsoft VC++和 MFC 框架。他还擅长在 Windows 和 Linux 平台上开发 24×7 小时运行的 C++程序，例如 KNX/EIB 家庭自动化监控软件。除了 C/C++之外，Marc 还喜欢 C#，并且会用 PHP 创建网页。

2007 年 4 月，他凭借 Visual C++方面的专业技能，获得了微软年度 MVP 称号。

Marc 还是比利时 C++用户组(www.becpp.org)的创始人，是 *C++ Standard Library Quick Reference*(Apress)一书的作者，以及多家出版社出版的多本书籍的技术编辑，是 CodeGuru 论坛上的活跃分子(id 为 Marc G)。Marc 还在 www.nuonsoft.com/blog/上维护了一个博客，他热爱旅游和烹饪。

技术编辑简介

Peter Van Weert 是一名软件工程师。他的主要爱好和特长是 C++、编程语言、算法和数据结构。他毕业于比利时勒芬大学，勒芬大学的考试团认定 Peter 在计算机科学方面获得了全优成绩，因此授予他硕士学位。2010 年，他在勒芬大学获得博士学位，他的博士论文围绕基于规则的编程语言(主要是 Java)的有效编译展开。在攻读博士学位期间，他还担任面向对象的分析与设计、Java 编程和声明性编程语言等课程的助教。

完成学业后，Peter 供职于 Nikon Metrology，负责利用 C++开发 3D 激光扫描软件和点云检测软件。在 2017 年，Peter 加盟 Nobel Biocare 的数字化牙科软件研发部门。在他的职业生涯中，Peter 精通 C++开发以及超大型软件库的管理、重构和调试。Peter 也熟悉软件开发过程的所有方面，包括功能和技术需求分析，以及基于敏捷和 Scrum 的项目管理和团队管理。

Peter 是比利时 C++用户组的成员，并且经常在该用户组发表技术演讲。他曾参与撰写以下两本由 Apress 出版的技术书籍：*C++ Standard Library Quick Reference* 和 *Beginning C++ (5th edition)*。

致　谢

在此感谢 John Wiley & Sons 和 Wrox 出版社的编辑和产品团队，感谢他们的支持。特别感谢 Wiley 的执行编辑 Jim Minatel 给我撰写本书的机会，感谢 Wiley 的高级项目编辑 Adaobi Obi Tulton 对这个项目的支持。感谢文稿编辑 Marylouise Wiack 对文字的梳理，保证术语前后一致，以及确保语法正确无误。

特别感谢技术编辑 Peter Van Weert，感谢他一丝不苟地对本书进行审阅。他提出了很多具有建设性的评论和意见，将本书的质量推进到更高水平。

当然，还要感谢我的父母、兄长和妻子给予的支持和关爱，你们的支持对本书的完成至关重要。也衷心感谢 Nikon Metrology 在我完成本项目期间给予的支持。

最后，我要感谢各位亲爱的读者，希望你们通过本书能学习到高级 C++软件开发知识。

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，但这种语言却非常难完全掌握。专业 C++程序员使用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，也未必完全了解 C++中某些很有用的特性。

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语言中的大部分知识，讲解语法并列举示例。本书不遵循这种模式。本书并不讲解语言的大量细节并给出少量真实世界的场景，而是教你如何在真实世界中使用 C++。本书还会讲解一些鲜为人知的让编程更简单的特性，以及区分编程新手和专业程序员的编程技术。

读者对象

就算使用 C++已经多年，也仍可能不熟悉 C++的一些高级特性，或仍不具有使用这门语言的完整能力。也许你编写过实用的 C++代码，但还想学习更多有关 C++中设计和良好编程风格的内容。也许你是 C++新手，想在入门时就掌握“正确”的编程方式。本书能满足上述需求，将你的 C++技能提升到专业水准。

因为本书专注于从对 C++具有基本或中等了解水平蜕变为一名专业 C++程序员的过程，所以本书假设你对该语言具有一定程度的认识。第 1 章涵盖 C++的一些基础知识，可以当成复习材料，但是不能替代实际的语言培训和语言使用手册。如果刚开始接触 C++，但有很丰富的 C、Java 或 C#语言经验，那么应该能从第 1 章获得所需的大部分知识。

不管属于哪种情况，都应该具有很好的编程基础。应该知道循环、函数和变量。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。应该了解一些常见的数据结构(例如队列)以及有用的算法(例如排序和搜索)。不需要预先了解有关面向对象编程的知识——这是第 5 章讲解的内容。

还应该熟悉开发代码时使用的编译器。稍后将简要介绍 Microsoft Visual C++和 GCC 这两种编译器。要了解其他编译器，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。本书贯穿对 C++17 新特性的讨论。这些新的 C++17 特性并不是分散在各章中，而是穿插于全书，在有必要的情况下，所有例子都已更新为使用这些新特性。

本书不仅讲解 C++语法和语言特性，还强调编程方法论、可重用的设计模式以及良好的编程风格。本书讲解的方法论覆盖整个软件开发过程——从设计和编码，到调试以及团队协作。这种方法可让你掌握 C++语言及其独有特性，还能在大型软件开发中充分利用 C++语言的强大功能。

想象一下有人学习了 C++的所有语法但没有见过一个 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，你也能成为一名专业的 C++程序员。你在 C++方面的知识会得到扩充，将接触到鲜为人知和常被误解的语言特性。你还将领略面向对象设计，掌握卓越的调试技能。最重要的或许是，通过阅读本书，你的头脑中有了大量“可重用”思想，可将这种思想贯彻到日常工作中。

有很多好的理由让你努力成为一名专业的 C++程序员，而非只是泛泛了解 C++。了解语言的真正工作原理有助于提升代码质量。了解不同的编程方法论和过程可让你更好地和团队协作。探索可重用的库和常用的设计模式可提升日常工作效率，并帮助避免白费力气地重复工作。所有这些学习课程都在帮助你成为更优秀的程序员，同时成为更有价值的雇员。

本书结构

本书包括 5 部分。

第 I 部分“专业的 C++简介”是 C++基础速成教程，能确保读者掌握 C++的基础知识。在速成教程后，第 I 部分深入讨论字符串和字符串视图的使用，因为字符串在示例中应用广泛。第 I 部分的最后一章介绍如何编写清晰易读的 C++代码。

第 II 部分“专业的 C++软件设计”介绍 C++设计方法论。你会了解到设计的重要性、面向对象方法论和代码重用的重要性。

第 III 部分“专业的 C++编码方法”从专业角度概述 C++技术。你将学习在 C++中管理内存的最佳方式，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。你还会学习这门语言的一些不同寻常之处、输入输出技术、错误处理、字符串本地化和正则表达式的使用，讨论如何实现运算符重载，如何编写模板。这一部分还讲解 C++标准库，包括容器、迭代器和算法。你还会学习 C++标准中的其他一些库，例如处理时间、随机数和文件系统的库。

第 IV 部分“掌握 C++的高级特性”讲解如何最大限度地使用 C++。本书这一部分揭示 C++中神秘的部分，并描述如何使用这些更高级的特性。你将学习如何定制和扩充标准库以满足自己的需求、高级模板编程的细节(包括模板元编程)，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 V 部分“C++软件工程”重点介绍如何编写企业级质量的软件。相关的主题如下：当今编程组织使用的工程实践，如何编写高效的 C++代码，软件测试概念(如单元测试和回归测试)，C++程序的调试技术，如何在自己的代码中融入设计技术、框架和概念性的面向对象设计模式，跨语言和跨平台代码的解决方案，等等。

本书最后是 4 个附录。附录 A 列出在 C++技术面试中取得成功的指南，附录 B 是带注解的参考文献列表，附录 C 总结 C++标准中的头文件，附录 D 简要介绍 UML(Unified Modeling Language，统一建模语言)。

本书没有列出 C++中每个类、方法和函数的参考。Peter Van Weert 和 Marc Gregoire 撰写的 *C++ Standard Library Quick Reference* 是 C++标准库提供的所有重要数据结构、算法和函数的浓缩版。附录 B 列出了更多参考资料。下面是两个很好的在线参考。

```
www.cppreference.com
```

可使用这个在线参考，也可下载其离线版本，在没有连接到互联网时使用。

```
www.cplusplus.com/reference/
```

本书正文中提到“标准库参考资料”时，就是指上述 C++参考资料。

使用本书的条件

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

本书包含 C++17 标准引入的新特性。在撰写本书时，有些编译器还不能完全支持 C++17 的所有新特性。

可使用任意 C++编译器。如果还没有 C++编译器，可下载一个免费的。这有许多选择。例如，对于 Windows，可下载 Microsoft Visual Studio 2017 Community Edition，这个版本免费且包含 Visual C++；对于 Linux，可使用 GCC 或 Clang，它们也是免费的。

下面将简要介绍如何使用 Visual C++和 GCC。可参阅相关的编译器文档来了解更多信息。

Microsoft Visual C++

首先需要创建一个项目。启动 VC++，单击 File | New | Project，在左边的项目模板树中选择 Visual C++ | Win32，再在窗口中间的列表中选择 Win32 Console Application(或 Windows Console Application)模板。在底部指定项目的名称、保存位置，单击 OK。

这会打开一个向导，单击 Next 按钮，选择 Console Application 和 Empty Project，再单击 Finish 按钮。注意，你可能看不到向导，具体取决于使用的 VC++ 2017 版本。相反，将自动创建一个新的项目，其中包含 4 个文件：stdafx.h、stdafx.cpp、targetver.h 和<projectname>.cpp。如果遇到这种情况，而你想要编译源代码文件(取自从配套网站下载的本书源代码压缩文件)，则必须在 Solution Explorer(选择 View | Solution Explorer)中选择这些文件，然后删除它们。

加载新项目后，就会在 Solution Explorer 中看到项目文件列表。如果这个停靠窗口不可见，可选择 View | Solution Explorer。在 Solution Explorer 中右击项目名，再选择 Add | New Item 或 Add | Existing Item，就可以给项目添加新文件或已有文件。

使用 Build | Build Solution 编译代码。没有编译错误后，就可以使用 Debug | Start Debugging 运行了。

如果程序在查看输出之前就退出了，可使用 Debug | Start without Debugging。这会在程序末尾暂停，以便查看输出。

在撰写本书期间，Visual C++ 2017 尚未自动启用 C++17 功能。要启用 C++17 功能，可在 Solution Explorer 窗口中右击项目，然后单击 Properties。在 Properties 窗口中，选择 Configuration Properties | C/C++ | Language，根据使用的 Visual C++版本，将 C++ Language Standard 选项设置为 ISO C++17 Standard 或 ISO C++ Latest Draft Standard。仅当项目至少包含一个.cpp 文件时，才能访问这些选项。

Visual C++支持“预编译的头文件”，这个话题超出了本书的讨论范围。通常而言，如果编译器支持的话，建议使用预编译的头文件。但是，从本书网站下载的源代码文件不使用预编译的头文件，因此，只有禁用这项功能才能使这些代码正确编译。在 Solution Explorer 窗口中右击项目，选择 Properties。在 Properties 窗口中，找到 Configuration Properties | C/C++ | Precompiled Headers，将 Precompiled Header 选项设置为 Not Using Precompiled Headers。

GCC

用自己喜欢的任意文本编辑器创建源代码，保存到一个目录下。要编译代码，可打开一个终端，运行如下命令，指定要编译的所有.cpp 文件：

```
gcc -lstdc++ -std=c++17 -o <executable_name> <source1.cpp> [source2.cpp ...]
```

-std=c++17 用于告诉 GCC 启用 C++17 支持。

例如，可切换到包含代码的目录，运行如下命令来编译第 1 章的 AirlineTicket 示例：

```
gcc -lstdc++ -std=c++17 -o AirlineTicket AirlineTicket.cpp AirlineTicketTest.cpp
```

没有编译错误后，就可以使用如下命令运行了：

```
./AirlineTicket
```

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，但错误总是难免的。如果在本书中找到错误，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 发电子邮件，我们就会检查你提供的信息，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。在这个页面上可查看 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。

源代码

读者在学习本书中的示例时，可以手动输入所有代码，也可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 www.wiley.com/go/proc++4e 下载。

另外，也可进入 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。

还可通过扫描本书封底的二维码来下载源代码。

提示：

由于许多图书的书名都十分类似，因此按 ISBN 搜索是最简单的，本书英文版的 ISBN 是 978-1-119-42130-6。

下载代码后，只需要用自己喜欢的解压缩软件进行解压缩即可。

目　录