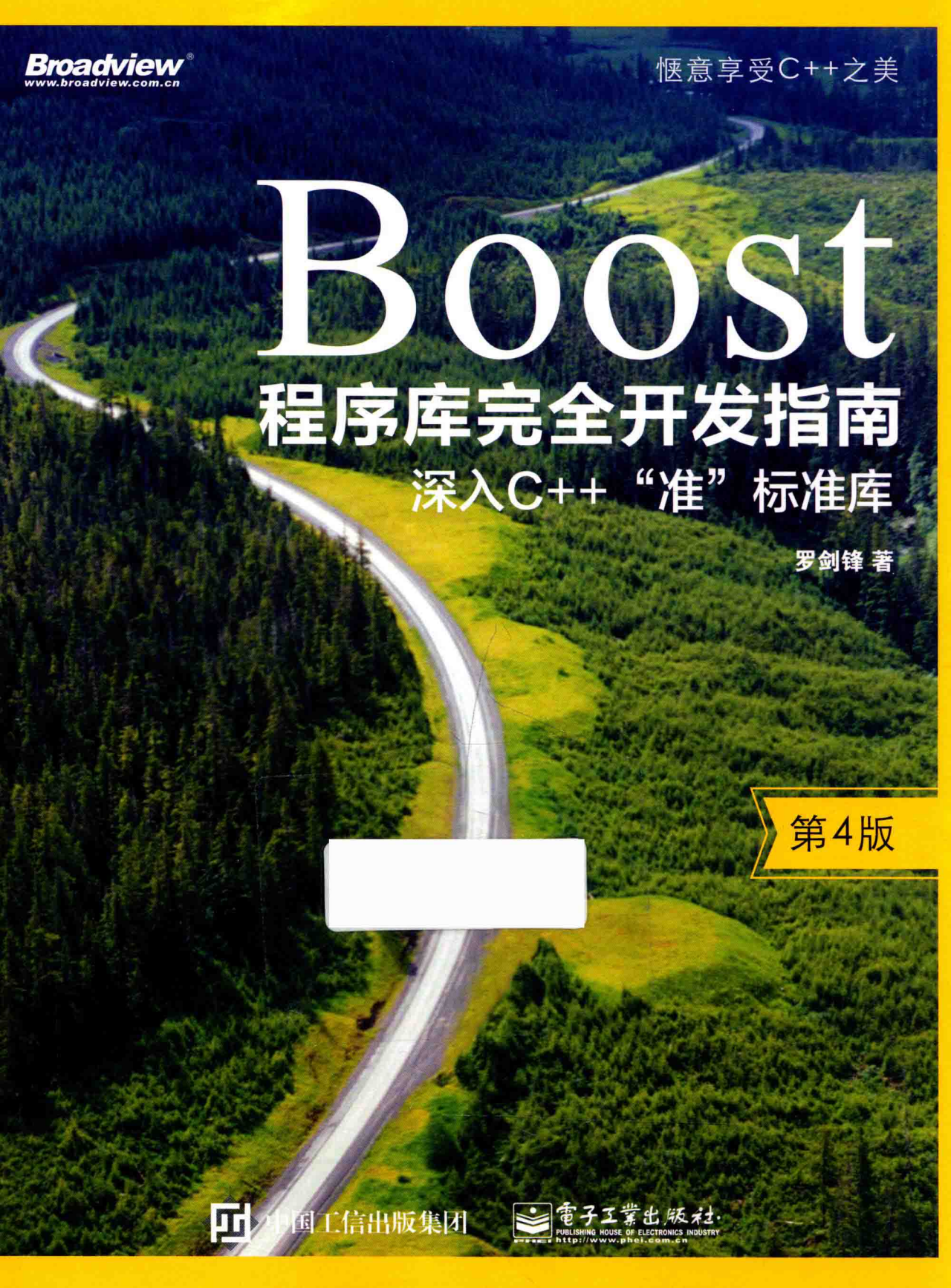
Broadview
www.broadview.com.cn
惬意享受C++之美
Boost
程序库完全开发指南
深入C++“准”标准库
罗剑锋 著
第4版
中国工信出版集团
電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

Boost

程序库完全开发指南

深入C++“准”标准库

（第4版）

罗剑锋 著

電子工業出版社
Publishing House of Electronics Industry
北京•BEIJING

内 容 简 介

Boost 是一个功能强大、构造精巧、跨平台、开源并且完全免费的 C++程序库，有着“C++‘准’标准库”的美誉。

Boost 由 C++标准委员会部分成员所设立的 Boost 社区开发并维护，使用了许多现代 C++编程技术，内容涵盖字符串处理、正则表达式、容器与数据结构、并发编程、函数式编程、泛型编程、设计模式实现等许多领域，极大地丰富了 C++的功能和表现力，能够使 C++软件开发更加简捷、优雅、灵活和高效。

本书基于 2017 年 4 月发布的 Boost1.64 版，介绍了其中的所有 140 余个库，并且结合 C++11/14/17 标准详细、深入地讲解了其中数十个库，同时实现了若干颇具实用价值的工具类和函数，可帮助读者迅速地理解、掌握 Boost 的用法并应用于实际的开发工作。

本书内容丰富、结构严谨、详略得当、讲解透彻，带领读者领略了 C++的最新前沿技术，相信会是每位 C++程序员的必备工具书。

图书在版编目（CIP）数据

Boost 程序库完全开发指南：深入 C++“准”标准库 / 罗剑锋著. —4 版. —北京：电子工业出版社，2017.10
ISBN 978-7-121-32460-4

Ⅰ. ①B… Ⅱ. ①罗… Ⅲ. ①C++语言－程序设计－指南 Ⅳ. ①TP312.8-62

中国版本图书馆 CIP 数据核字（2017）第 195365 号

责任编辑：安 娜
印 刷：三河市鑫金马印装有限公司
装 订：三河市鑫金马印装有限公司
出版发行：电子工业出版社
北京市海淀区万寿路 173 信箱 邮编 100036
开 本：787×980 1/16 印张：36.75 字数：812 千字
版 次：2017 年 10 月第 1 版
印 次：2017 年 10 月第 1 次印刷
印 数：2500 册 定价：99.00 元

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：（010）88254888，88258888。

质量投诉请发邮件至 zlts@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

本书咨询联系方式：010-51260888-819，faq@phei.com.cn。

目录

第 0 章

导读

0.1 关于本书

C++是一种伟大的编程语言，某种程度上它甚至超越了编程语言的境界而升华为一种哲学。

C++又是一种多范式、可扩展的编程语言，支持多种编程风格——基于过程、基于对象、面向对象、函数式、泛型、模板元、自动机，非常自由灵活，易学难精，在不同编程风格间切换时必须小心谨慎以避免失误。

C++标准中出现的 STL（标准模板库）极大地改变了 C++程序员的编程思维，使“泛型”成为了 21 世纪以来程序开发界最流行的词汇之一。而 C++标准委员会成员所设立的 Boost 社区和开发的 Boost 程序库，更将“泛型”等现代 C++编程方法发挥到了极致。

Boost 程序库代表了目前 C++语言最新最前沿的技术，内容博大精深，丝毫不逊于经典的 STL，但同时也令很多人难以摸清门路，不得登堂入室而一窥究竟。有鉴于此，作者根据多年在实际开发中使用 Boost 库的经验并结合最新的 C++标准编写了本书，想为广大 C++程序员了解 C++的最新技术进展尽一份自己的力量。

本书的定位是“指南”（Guide、Introduction），而不是技术手册（Reference）或者使用说明（Manual），能够解答 90%但不是所有 Boost 库相关的问题。但作者尽量做到让本书接近一本参考手册，使读者在阅读时能够脱离计算机，不至于频繁使用鼠标和键盘查询在线帮助或者源代码。

0.2 读者对象

本书适合有一定 C++基础的读者。

阅读本书需要有一定的C++知识，基本要求是熟悉面向对象编程技术，如封装、继承、多态，此外还要理解名字空间、异常、模板[①]、泛型编程等高级特性（但不必非常深入），最好还能够了解设计模式和C++标准库提供的容器、算法等各种组件。

如果读者是C++初学者或还不具备以上所列的知识，建议先阅读附录A推荐书目里的技术书籍，然后再学习本书。如果手头刚好有推荐书目中的一本或两本，则可以一边翻阅这些书籍，一边学习本书。

总之，无论读者是C++哪一层次的用户，现在或将来本书都会给您带来帮助。

0.3 术语与风格

本节列出了书中经常用到的专业术语和编程风格，以期与读者获得阅读的共识。

Boost库并不是一个单一、平面化的程序库，而是有着复杂的内部结构，每个“库”可能是由其他许多更小的“库”组成的。因此，本书把程序库中所有组成部分统称为“组件”，“库”（Library）与“组件”（Compoment）这两个术语有时会通用。

namespace这个术语有译作“命名空间”、“名称空间”、“名字空间”，本书称作“名字空间”。这只是作者个人习惯而已，如果在阅读过程中给读者造成了小小的困扰，还请谅解。

在使用template定义模板类或者模板函数时，本书统一使用typename而不是常见的class，因为typename能够更清楚地向代码阅读者表明这是一个类型参数，而不一定是一个类（class）。但例外的是书中列出Boost源代码，会尽量保持其原始形式。

在命名函数或者类时，本书遵循C++标准库和Boost的惯例，均采用小写形式，单词间以下画线分隔，如demo_class、rand_bytes()，但并不要求读者也必须遵循这种命名方式，通常自己编写的类使用大写字母开头的单词命名会更好。

在for循环递增变量、指针或者迭代器对象的时候，本书统一使用++操作符的前置用法（++i），而不是后置用法（i++），因为前者不需要返回一个临时对象，执行效率更高。

“未定义行为”一词经常用来指代某些操作可能导致的不正确结果，如使用已失效的迭代

① 模板类和模板函数一般形式是class_or_func<T>，例如vector<int>。<>在编程语言中通常被用于大小比较，因此其形式对于初学者很难接受，但这种形式具有良好的可读性，<>可以被读作英文of，比如vector<int>可以读作vector of int，清晰地表明了类/函数和它的模板类型的关系，这样可以减少一部分对尖括号的不适应感，作者经常这么做，读者也可以试一下，很有效。

器、错误地使用指针，等等。对于“未定义行为”的一个较好（但不太精确）的定义是：程序在开发人员面前运行正常，在测试人员面前运行正常，但在老板或者最终用户面前运行时崩溃了。读者应当小心并尽量避免“未定义行为”，它是代码中的“定时炸弹”，如果它在调试的过程中爆炸了，那通常是最好的结果，因为它明确地告诉了我们代码存在问题。

0.4 语言标准

现在 C++有四个版本的国际标准：C++98、C++11、C++14 和 C++17[①]，本书主要使用 C++11（ISO/IEC 14882:2011），不含数字标识的“C++标准”一词通常就是指这个标准，但在涉及 C++某些语言特性时可能会明确标明具体的版本，有时还会以“C++11.x.y.z”的形式标明所引用 C++标准文档的章节号。

书中使用较多的新语言特性有四个：

- nullptr：强类型的空指针；
- auto/decltype：自动推导表达式类型；
- for：基于范围的新式循环，形式更加优雅；
- lambda 表达式：又称闭包（closure），能够非常便捷地定义函数对象，通常的形式是[](...){...}，其中[]是捕获列表，()是函数参数，{}是函数体。

C++标准中定义的函数库本书称为“C++标准库”或者“STL”，但严格意义上 STL 与标准库并不等价，STL 只是标准库中的一个（很大的）子集，这么称呼有时候只是为了行文上的方便。

一般情况下使用 C++标准库都必须包含相应的头文件并且加上“using namespace std”语句，但标准库已经成为了 C++软件开发的基础设施，应用得非常频繁，因此书中的代码示例片断一般会将之略去。但有的情况下为了特别强调，会加上 std 名字空间前缀，如 std::vector。读者可以认为书中所有代码都默认包含了如下的头文件：

```
#include <string>                    //标准字符串类
#include <iostream>                  //标准输入输出
#include <vector>                    //标准向量容器
#include <set>                       //标准集合容器
#include <map>                       //标准映射容器（关联数组）
#include <algorithm>                 //标准算法
using namespace std;                 //打开标准库的名字空间
```

① 实际上还应该有一个 C++03 标准，但因为它与 C++98 的差异很小，故本书忽略之。

为使读者对C++标准能够有更多的了解，作者编写了一份简要介绍，作为本书附录供参考。

0.5 本书的结构

Boost库组件繁多，相互关联也较多，如何排列其顺序是作者面临的一个颇为棘手的问题。

Boost官方提供了两种基本方式：一种是按照组件的字母顺序，另一种是按照功能用途分类顺序，但这两种方式都不是组织本书结构的最佳手法。经反复考量，作者决定以难易度和实用程度对Boost库组件分类排序，采用由浅入深循序渐进的方式，先介绍较简单易用且实用程度高的库，然后逐步深入，介绍用法复杂的库，以期帮助读者尽快掌握Boost的使用方法。

对于每个Boost组件，本书通常首先简要介绍其功能，然后说明其头文件和编译方法（如果需要编译的话），列出类的声明概要，再使用例子讲解详细用法和注意事项，涉及其他Boost库组件时则以交叉引用的方式指明参考章节，最后是对该库的总结。

本书共分16章，各章的内容简介如下：

■ 第1章：总论

简要介绍Boost的历史、特点和获取方式，以及本书的开发环境和如何编译安装Boost。

■ 第2章至第13章

第2章至第13章分门别类、由浅入深地介绍Boost库的各个组件，占据了本书的大部分篇幅，也是读者需要仔细阅读的内容。其中既包括如timer、noncopyable等简单的小工具，也包括test、thread、asio等用法复杂且功能强大的组件，Boost 1.64版全部140余个库在本书中都可以找到阐述。

■ 第14章：设计模式

本章结合之前介绍的Boost库组件简要论述了推荐书目[1]中的23个设计模式和4个其他常用模式，以及Boost库使用这些设计模式的方法，从设计模式的抽象层次来加深理解Boost库。

■ 第15章：结束语

本章简单展望了Boost今后的发展，介绍其他可与Boost互为补充的开源C/C++库，并对如何做一个好的程序员提出了自己的见解。

- 附录

书末的附录也很有价值，列出了作者认为值得阅读的 C/C++经典书籍——它们也是作者编写本书时的案头必备参考资料，还有就是 C++标准简述和编程经验谈。

这些内容被放在附录部分并不是因为它们不重要，而仅仅是因为它们与全书的主题关联不大，但很值得阅读，读者也许会从中发现一些很有用的东西。

0.6 如何阅读本书

本书是一本介绍程序库的书籍，其中的很多组件是彼此独立的，所以在阅读完第 0 章（即本章）和第 1 章后，可以随意愿自由阅读其他章节。

读者既可以按照书的物理顺序循序渐进地逐页阅读，也可以查阅目录，然后直接跳到感兴趣的章节。不过对于大多数读者来说，作者还是推荐第一种方式，因为这可能会是学习效率更高、学习曲线更平滑的方式。

涉及编程语言，尤其是 C++，书中不可避免地会出现大量的代码片断，有的还可能很长，希望读者阅读时能有足够的耐心。这些代码都是精心编制的范例，将会指导读者如何使用 Boost 编写正确、高效的代码。

阅读时还需要注意一点：本书并不包含 Boost 库的所有接口说明，如果在阅读过程中对某些地方有疑虑，请参考 Boost 说明文档或者直接阅读 Boost 实现代码。

0.7 本书的源码

为了更好地方便读者利用本书学习研究 Boost 程序库，作者在 GitHub 网站上发布了书内所有示例程序的源代码，读者可以根据需要自取。[①]

本书源码的 GitHub 项目地址是：

```
https://github.com/chronolaw/boost_guide.git
```

那么，欢迎来到 Boost 的世界。

① 有了 GitHub，再也不需要“附赠光盘”了，笑。