附录B 标准库头文件

C++标准库的接口包含 98 个头文件，其中有 21 个表示 C 标准库。要记住源代码中应该包含哪些头文件往往很难，所以这个附录简要描述每个头文件的内容，按照以下 8 类组织：

- C 标准库
- 容器
- 算法、迭代器、范围和分配器
- 通用工具
- 数学工具
- 异常
- I/O 流
- 线程支持库

B.1 C 标准库

C++标准库包含完整的 C 标准库。头文件通常是一样的，除了以下两点：

- 头文件为<cname>而不是<name.h>
- <cname>头文件中声明的所有名称都在 std 名称空间中

注意：
为了后向兼容，如有必要，仍可包含<name.h>。然而，这样会把名字放在全局名称空间而不是 std 名称空间中。另外，<name.h>已不赞成使用。建议避免这种用法。

注意：
C 标准库不保证是可以导入的。因此，使用“#include <cstdio>”而不是“import <cstdio>”。

表 B-1 列出了所有的 C 标准库头文件并总结了最常用的功能。注意建议避免使用 C 功能，而尽量使用等价的 C++功能。

表 B-1 C 标准库头文件

头文件名	内容
<cassert>	assert()宏
<cctype>	字符谓词和操作函数，例如 isspace()和 tolower()
<cerrno>	定义 errno 表达式，它是一个宏，获得某些 C 函数的最后一个错误编号
<cfenv>	支持浮点数环境，例如浮点异常、浮点数取整等
<cfloat>	和浮点数算术相关的 C 风格定义，例如 FLT_MAX
<cinttypes>	定义与 printf()、scanf()和类似函数结合使用的一些宏，还定义一些操作 intmax_t 的函数
<climits>	C 风格的限制定义，例如 INT_MAX。建议改用 C++中对应的<limits>
<clocale>	一些用于本地化的宏和函数，例如 LC_ALL 和 setlocale()。见 C++中对应的< locale>
<cmath>	数学工具，包括三角函数、sqrt()和 fabs()等
<csetjmp>	setjmp()和 longjmp()，绝不要在 C++中使用
<csignal>	signal()和 raise()，避免在 C++中使用
<cstdarg>	处理变长参数列表的宏和类型
<cstddef>	重要的常量，例如 NULL，以及重要的类型，例如 size_t 和 byte。
<cstdint>	定义一些标准的整数类型，例如 int8_t 和 int64_t 等，还包含表示这些类型的最大值和最小值的宏
<cstdio>	文件操作，包括 fopen()和 fclose()。格式化 I/O：printf()、scanf()等系列函数。字符 I/O：getc()、putc()等系列函数。文件定位：fseek()和 ftell()。建议改用 C++流(见稍后的“I/O 流”)
<cstdlib>	随机数操作：rand()和 srand()，从 C++14 开始已不建议使用，而改用 C++ <random>。这个头文件包含 abort()和 exit()函数，应该避免使用这两个函数。C 风格的内存分配函数：calloc()、malloc()、realloc()和 free()。C 风格的排序和搜索函数：qsort()和 bsearch()。字符串到数值的转换函数：atof()和 atoi()等。一组与多字节/宽字符串处理相关的函数
<cstring>	底层内存管理函数，包括 memcpy()和 memset()。C 风格的字符串函数，例如 strcpy()和 strcmp()
<ctime>	时间相关的函数，包括 time()和 localtime()
<cuchar>	定义一些与 Unicode 相关的宏和函数，例如 mbrtoc16()
<cwchar>	宽字符版本的字符串、内存和 I/O 函数
<cwctype>	<cctype>中函数的宽字符版本：iswspace()和 towlower()等

表 B-2 列出了在 C++20 中已被移除的 C 标准库。

表 B-2 C++20 中已被移除的 C 标准库

头文件名	内容
<ccomplex>	只包括<complex>。从 C++17 开始，已不赞成使用，在 C++20 中移除
<ciso646>	在 C 语言中，<iso646.h>文件定义宏 and 和 or 等。在 C++中，这些都是关键字，所以这个头文件为空
<cstdalign>	和对齐相关的宏：__alignas_is_defined，从 C++17 开始已不赞成使用，在 C++20 中移除
<cstdbool>	与布尔类型相关的宏__bool_true_false_are_defined，从 C++17 开始已不赞成使用，在 C++20 中移除
<ctgmath>	只包含<ccomplex>和<cmath>，从 C++17 开始已不赞成使用，在 C++20 中移除

B.2　容器

可在以下 13 个头文件中找到标准库容器的定义，如表 B-3 所示。

表 B-3　标准库容器的定义

头文件名	内容
<array>	array 类模板
<bitset>	bitset 类模板
<deque>	deque 类模板
<forward_list>	forward_list 类模板
<list>	list 类模板
<map>	map 和 multimap 类模板
<queue>	queue 和 priority_queue 类模板
<set>	set 和 multiset 类模板
	span 类模板，严格来说不是容器，而是存储在其他地方的连续元素序列的视图
<stack>	stack 类模板
<unordered_map>	unordered_map 和 unordered_multimap 类模板
<unordered_set>	unordered_set 和 unordered_multiset 类模板
<vector>	vector 类模板和 vector<bool>特例化

每个头文件都包含使用特定容器需要的所有定义，包括迭代器。第 18 章“标准库容器”详细讨论了这些容器。

B.3　算法、迭代器、范围和分配器

表 B-4 中的不同头文件定义可用的标准库算法、迭代器、分配器以及范围库。

表 B-4　标准库算法、迭代器、分配器以及范围库的定义

头文件名	内容
<algorithm>	标准库中大部分算法的原型，例如 min()、max()、minmax()和 clamp()等。详见第 20 章
<bit>	定义 endian 类枚举类型，详见第 34 章，并提供函数原型来执行对位序列的低级操作，如 bit_ceil()、rotl()和 countl_zero()等。参见第 16 章
<execution>	定义与标准库算法一起使用的执行策略类型，详见第 20 章
<functional>	定义内建函数对象、取反器、绑定器和适配器，详见第 19 章
<iterator>	定义 iterator_trait、迭代器标签、iterator、reverse_iterator、插入迭代器(例如 back_insert_iterator)和流迭代器，详见第 17 章
<memory>	定义默认分配器、一些处理容器内未初始化内存的工具函数，以及第 7 章介绍的 shared_ptr、unique_ptr、make_unique()和 make _ shared()
<memory_resource>	定义多态分配器和内存资源，详见第 25 章
<numeric>	一些数值算法的原型，例如 accumulate()、inner_product()、partial_sum()、adjacent_difference()、gcd()和 lcm()等，详见第 20 章
<ranges>	提供范围库的所有功能。详见第 17 章
<scoped_allocator>	可用于内嵌容器的分配器，例如 string 的 vector、map 的 vector

B.4 通用工具

标准库在一些不同的头文件中包含一些通用的工具函数，如表 B-5 所示。

表 B-5 通用的工具函数

头文件名	内容
<any>	定义 any 类，详见第 24 章
<charconv>	定义 chars_format 枚举类、from_chars()函数、to_chars()函数和相关结构体，详见第 2 章“使用 string 和 string_view”
<chrono>	定义 chrono 库，详见第 22 章“日期和时间工具”
<codecvt>	为不同字符编码提供代码转换。从 C++17 开始，已经不赞成使用这个头文件
<compare>	为支持三向比较运算符提供了类和函数。详见第 1 章和第 9 章
<concepts>	提供了标准概念，例如 same_as、convertible_to、integral 和 movable 等。详见第 12 章。
<filesystem>	定义用于处理文件系统的可用类和函数，详见第 13 章
<format>	提供了格式化库所需的全部功能，例如 format()和 format_to()等。详见第 2 章。
<initializer_list>	定义 initializer_list 类，详见第 1 章
<limits>	定义 numeric_limits 类模板，以及大部分内建类型的特例化，详见第 1 章
<locale>	定义 locale 类、use_facet()和 has_facet()模板函数以及 facet 系列函数，详见第 21 章
<new>	定义 bad_alloc 异常和 set_new_handler()函数，以及 operator new 和 operator delete 的所有 6 种原型，详见第 15 章
<optional>	定义 optional 类，详见第 20 章
<random>	定义随机数生成器库，详见第 23 章
<ratio>	定义 Ratio 库，以操作编译时有理数，详见第 22 章
<regex>	定义正则表达式库，详见第 21 章
<source_location>	提供<source_location>类。详见第 14 章
<string>	定义 basic_string 类模板以及 string 和 wstring 的类型别名实例，详见第 2 章
<string_view>	定义 basic_string_view 类模板和类型别名 aliases string_view 和 wstring_view，详见第 2 章
<system_error>	定义错误分类和错误代码
<tuple>	定义 tuple 类模板，作为 pair 类模板的泛化，详见第 24 章
<type_traits>	定义模板元编程中使用的类型 trait，详见第 26 章
<typeindex>	定义 type_info 的简单包装，可在关联容器和无序关联容器中用作索引类型
<typeinfo>	定义 bad_cast 和 bad_typeid 异常。定义 type_info 类，typeid 运算符返回这个类的对象。有关 typeid 详见第 10 章
<utility>	定义 pair 类模板和 make_pair()，详见第 1 章。这个头文件还定义了工具函数 swap()、exchange()、move()和 as_const()等
<variant>	定义 variant 类，详见第 24 章
<version>	提供与使用的 C++标准库相关的实现相关信息，并公开所有标准库特征测试宏。详见第 16 章

B.5　数学工具

C++提供了一些数值处理功能，如表 B-6 所示，这些功能没有在本书中详细讨论。

表 B-6　数值处理功能

头文件名	内容
<complex>	定义处理复数的 complex 类模板
<numbers>	提供了一些数学常量，例如 pi、phi 和 log2e 等
<valarray>	定义 valarray 类，以及处理数学矢量和矩阵的相关类和类模板

B.6　异常

第 14 章讨论了异常和异常的支持。有两个头文件提供了大多数所需的定义，但其他一些领域的异常定义在相关领域的头文件中，如表 B-7 所示。

表 B-7　异常定义

头文件名	内容
<exception>	定义 exception 和 bad_exception 类，以及 set_terminate()和 uncaught_exception()函数
<stdexcept>	没有定义在<exception>中的非领域相关的异常

B.7　I/O 流

表 B-8 列出了 C++中所有和 I/O 流相关的头文件。不过通常情况下应用程序只需要包含<fstream>、<iomanip>、<iostream>、<istream>、<ostream>和<sstream>。详情请参阅第 13 章。

表 B-8　和 I/O 流相关的头文件

头文件名	内容
<fstream>	定义了 basic_filebuf、basic_ifstream、basic_ofstream 和 basic_fstream 类，声明了 filebuf、wfilebuf、ifstream、wifstream、ofstream、wofstream、fstream 和 wfstream 类型别名
<iomanip>	声明了其他地方没有声明的 I/O 运算符(大部分都声明在<ios>中)
<ios>	定义了 ios_base 和 basic_ios 类，声明了大部分流运算符。几乎不需要直接包含这个头文件
<iosfwd>	其他 I/O 流头文件中出现的模板和类型别名的前置声明。几乎不需要直接包含这个头文件
<iostream>	声明了 cin、cout、cerr 和 clog 以及对应的宽字符版本。包括了<istream>、<ostream>、<streambuf>和<ios>。注意这不仅是<istream>和<ostream>的组合
<istream>	定义了 basic_istream 和 basic_iostream 类，声明了 istream、wistream、iostream 和 wiostream 类型别名
<ostream>	定义了 basic_ostream 类。声明了 ostream 和 wostream 类型别名
<sstream>	定义了 basic_stringbuf、basic_istringstream、basic_ostringstream 和 basic_stringstream 类，声明了 stringbuf、wstringbuf、istringstream、wistringstream、ostringstream、wostringstream、stringstream 和 wstringstream 类型别名
<streambuf>	声明了 basic_streambuf 类以及 streambuf 和 wstreambuf 类型别名。几乎不需要直接包含这个头文件
<strstream>	已不赞成使用
<syncstream>	定义了与同步输出流相关的类，例如 osyncstream 和 wosyncstream。详见第 27 章“C++多线程编程”

B.8 线程库

C++包含一个线程库，允许编写与平台无关的多线程应用程序。详情请参阅第 27 章。这个线程库由表 B-9 中的头文件组成。

表 B-9 线程库

头文件名	内容
<atomic>	定义了原子类型、atomic<T>以及原子操作
C++20 < barrier >	定义了 barrier 类
<condition_variable>	定义了 condition_variable 和 condition_variable_any 类
C++20 <coroutine>	定义了编写协程所需的所有功能
<future>	定义了 future、promise、packaged_task 和 async()
C++20 <latch>	定义了 latch 类
<mutex>	定义了不同的非共享互斥体、锁类以及 call_once()
C++20 <semaphore>	定义了 counting_semaphore 和 binary_semaphore 类
<shared_mutex>	定义了 shared_mutex、shared_timed_mutex 和 shared_lock 类
C++20 <stop_token>	定义了 stop_token、stop_source 和 stop_callback 类
<thread>	定义了 thread 和 jthread 类，以及 yield()、get_id()、sleep_for()和 sleep_until()函数

附录 C
UML 简介

UML(Unified Modeling Language，统一建模语言)是用于显示类层次结构、子系统交互和序列图等的行业图形标准。本书使用 UML 来表示类图。要完整解释 UML 标准，需要一整本书，因此本附录仅简要介绍本书正文中涉及的 UML 方面。UML 标准有多个版本。本书使用 UML 2.0。

C.1 图形类型

UML 定义了以下图形的集合：

- 结构 UML 图
  - 类图
  - 对象图
  - 包图
  - 组合结构图
  - 构件图
  - 部署图
  - Profile 图
- 行为 UML 图
  - 用例图
  - 活动图
  - 状态机图
  - 交互图
    - 顺序图
    - 通信图
    - 定时图
    - 交互概览图

由于本书仅使用类图和顺序图，因此本附录仅讨论这种 UML 图。

C.2 类图

类图用于可视化单个类，可以包括数据成员和成员函数。它们经常被用来展示不同类之间的关系，

C.2.1 类的表示

在 UML 中，类表示为方框，最多有 3 个隔间，包含的内容如下：

- 类名
- 类的数据成员
- 类的方法

图 C-1 显示了一个示例。

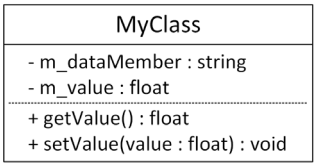

图 C-1 示例

MyClass 类有两个数据成员，一个数据成员的类型是 string，另一个数据成员的类型是 float；还有两个方法成员。每个方法成员前面的加号和减号用于指定可视性。表 C-1 列出了最常用的可见性。

表 C-1 最常见的可视性

可见性	含义
+	public 成员
–	private 成员
#	protected 成员

根据类图的目的，有时会忽略成员的细节。此时，类表示为如图 C-2 所示。如果只想显示不同类之间的关系，而不想显示各个类的成员详情，则可使用这种形式。

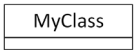

图 C-2 类

C.2.2 关系的表示

UML 2.0 支持类之间的 6 种不同关系：继承、实现、聚合、组合、关联和依赖。接下来将讨论这些关系。

1. 继承

使用从派生类到基类的线来表示继承(Inheritance)关系。线的末端是一个空心三角箭头，位于基类一侧，描述“是一个”关系。图 C-3 显示了一个示例。

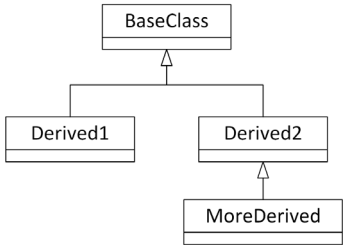

图 C-3　“是一个”关系的示例

2. 实施/实现

实施/实现(Realization/Implementation)某个接口的类基本是从那个接口继承(“是一个”关系)。但是，要区分一般的继承和接口实现，后者看上去像是继承，但用的是虚线而非实线，如图 C-4 所示。ListBox 类从 UIElement 继承，并实现了 Clickable 和 Scrollable 接口。

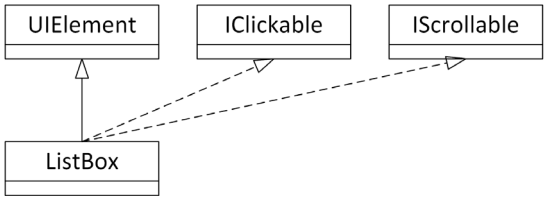

图 C-4　虚线表示

3. 聚合

聚合(Aggregation)表示“有一个”关系，用的是一条线。如果一个类包含另一个类的实例，那么前者的一侧是一个空心菱形。在聚合关系中，也可以可选地指定关系中每个参与者的多重性。多重性的位置刚开始时令人有些困惑，如图 C-5 所示。在本例中，一个 Class 可以包含/聚合一个或多个 Student，每个 Student 可属于 0 个或多个 Class。聚合关系意味着，即使聚合者被销毁，被聚合对象也可继续存在。例如，如果 Class 被销毁，Student 不会被销毁。

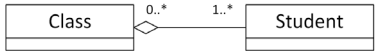

图 C-5　“有一个”关系的示例

表 C-2 列出了几个可能的多重性示例。

表 C-2　多重性示例

多重性	含义
N	正好 N 个实例
0..1	0 个或 1 个实例
0..*	0 个或多个实例
N..*	N 个或多个实例

4. 组合

组合(Composition)非常类似于聚合，表示形式几乎完全相同，只是用实心菱形替代了空心菱形。与聚合的不同之处在于，使用组合时，如果包含另一个类的实例的类被销毁，被包含的实例也会被销毁。

图 C-6 显示了一个示例。一个 Window 可包含 0 个或多个 Button，每个 Button 必须正好被 1 个 Window 包含。如果销毁 Window，Window 中包含的所有 Button 也将被销毁。

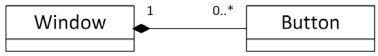

图 C-6 组合的示例

5. 关联

关联(Association)是聚合的更通用形式，它表示类之间的双向链接，而聚合是单向链接。图 C-7 显示了一个示例。每个 Book 都知道 Author 是谁，每个 Author 都知道自己所写的每个 Book。

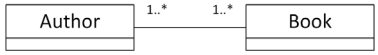

图 C-7 关联的示例

6. 依赖

依赖(Dependency)表示一个类依赖于另一个类。表示形式是一条虚线，箭头指向被依赖的类。通常情况下，虚线上的一些文本描述依赖关系。再来分析第 33 章的汽车工厂示例，CarFactory 依赖于 Car，因为工厂生产汽车，如图 C-8 所示。

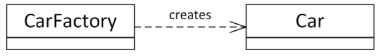

图 C-8 依赖的示例

C.3 交互图

UML2 支持 4 种类型的交互图：顺序图、通信图、定时图和交互概览图。本书仅使用顺序图，将在下一节中简要讨论。

顺序图

顺序图以图形方式表示在不同对象之间发送哪些消息以及这些消息的发送顺序。顺序图由以下组件组成。

- 对象：交互中涉及的对象实例。
- 生命线：图形化表示的对象生命周期。
- 消息：消息由一个对象发送给另一个对象。
- 回复：当对象收到另一个对象的消息时，会发送一个回复。
- 自我消息：对象发送给自己的消息。
- 选择：表示选择流，类似 if-then-else 语句的分支。

图 C-9 显示了顺序图的示例。它是从第 4 章“设计专业的 C++程序”的图表的简化版本，但这次是用标签表示图表重要部分的含义。

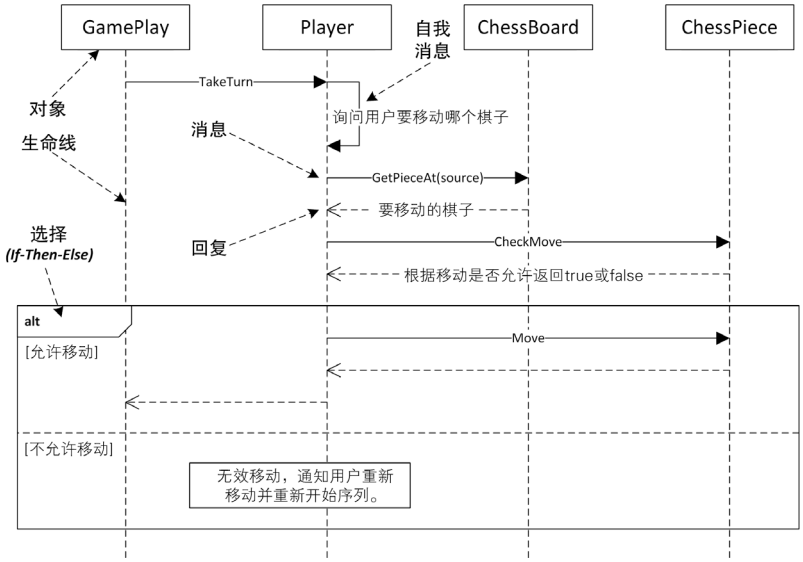
GamePlay
Player
ChessBoard
ChessPiece
自我
消息
对象
生命线
TakeTurn
询问用户要移动哪个棋子
消息
GetPieceAt(source)
要移动的棋子
选择
(If-Then-Else)
回复
CheckMove
根据移动是否允许返回true或false
alt
[允许移动]
Move
[不允许移动]
无效移动，通知用户重新
移动并重新开始序列。

图 C-9　顺序图的示例

附录 D
带注解的参考文献

附录 D 包含本书在撰写过程中参阅的与各种不同 C++主题相关的书籍和在线资源，还包含一些用于深入阅读或背景阅读的建议。

C++

C++入门(不要求具有编程经验)

- Ivor Horton and Peter Van Weert. *Beginning C++*20, 6th ed. Apress, 2020. ISBN: 978-1-4842-5883-5。
  这本书从基础开始介绍，通过例子逐步提高，使你成为一名熟练的 C++程序员。这个版本包含了 C++20 的新特性。该书不要求读者具有编程经验。
- Bjarne Stroustrup. *Programming: Principles and Practice Using* C++, 2nd ed. Addison-Wesley Professional, 2014. ISBN: 0-321-99278-4。
  由 C++语言的发明者简要介绍 C++编程。本书不要求读者具有编程经验，即便如此，它也仍然是经验丰富的编程人员的有益读物。
- Steve Oualline. *Practical C++ Programming, 2*nd ed. O'Reilly Media, 2003. ISBN: 0-596-00419-2。
  C++基础读物，本书不要求读者具有编程经验。
- Walter Savitch. *Problem Solving with C++*, 9th ed. Pearson, 2014. ISBN: 0-133-59174-3。
  该书不要求读者具有编程经验。通常作为入门编程课程的教材。

C++入门(要求具有编程经验)

- Bjarne Stroustrup. *A Tour of C++, 2*nd ed. Addison-Wesley Professional, 2018. ISBN: 0-134-99783-2。
  共约 256 页，简述整个 C++语言和标准库，面向已经了解 C++或一定经验的中高级读者，本书包含 C++17 特性。
- Stanley B. Lippman, Josée Lajoie, and Barbara E. Moo, *C++ Primer,* 5th ed. Addison-Wesley Professional, 2012, ISBN: 0-321-71411-3。
  该书非常详细地介绍 C++，以非常易用的格式详细讨论该语言的所有方面。

- Andrew Koenig, Barbara E. Moo, *Accelerated C++: Practical Programming by Example,* Addison-Wesley Professional, 2000, ISBN: 0-201-70353-X。
  内容与 *C++ Primer* 相同，但篇幅较短，因为它假定读者具有其他编程语言的经验。
- Bruce Eckel, *Thinking in C++*, *Volume 1: Introduction to Standard C++* , 2nd ed. Prentice Hall, 2000, ISBN: 0-139-79809-9。
  一本介绍 C++编程知识的卓越入门读物，读者需要预先掌握 C 语言知识。

综合 C++

- The C++ Programming Language，网址为 isocpp.org。
  标准 C++在网络上的主页，包含 C++标准在所有编译器和平台上的新闻、状态和讨论。
- C++ Super-FAQ，网址为 isocpp.org/faq。
  有关 C++的大量问答。
- Marius Bancila. *Modern C++ Programming Cookbook,* 2nd ed. Packt, 2020. ISBN: 9781800208988。

该书是以实用的方法组织的，涵盖了 C++开发者所面临的各种各样的问题。第 2 版提供了 30 个新的或更新的 C++20 方法。

- Paul Deitel, Harvey Deitel. *C++20 for Programmers,* 3rd ed. O'Reilly, 2020. ISBN: 9780136905776。
  该书是一本中级的基于教程的编程演示，介绍了最新版本的 C++程序设计语言 C++ 20。
- Scott Meyers. *Effective Modern C++:* 42 *Specific Ways to Improve Your Use of C++*11 *and C++*14. O'Reilly, 2014. ISBN: 1-491-90399-6。
- Scott Meyers. *Effective C++:* 55 *Specific Ways to Improve Your Programs and Designs*, 3rd ed. Addison-Wesley Professional, 2005. ISBN: 0-321-33487-6。
- Scott Meyers. *More Effective C++:* 35 *New Ways to Improve Your Programs and Designs*. Addison-Wesley Professional, 1996. ISBN: 0-201-63371-X。
  以上 3 本书对 C++中经常误用和误解的特性提供极好的技巧和诀窍。
- Bjarne Stroustrup. *The C++ Programming Language*, 4th ed. Addison-Wesley Professional, 2013. ISBN: 0-321-56384-0。
  C++"圣经"，由 C++设计者本人编写。每一位 C++程序员都应该有该书，但 C++初学者可能会感到晦涩难懂。
- Herb Sutter. *Exceptional C++:* 47 *Engineering Puzzles, Programming Problems, and Solutions*. Addison-Wesley Professional, 1999. ISBN: 0-201-61562-2。
  呈现为一组谜题。亮点是透彻讨论通过 RAII 实现适当的资源管理和异常安全。该书还深入介绍各种主题，如 pimpl idiom、名称查找、良好的类设计和 C++内存模型。
- Herb Sutter. *More Exceptional C++:* 40 *New Engineering Puzzles, Programming Problems, and Solutions.* Addison-Wesley Professional, 2001. ISBN: 0-201-70434-X。
  介绍 *Exceptional C++:* 47 *Engineering Puzzles, Programming Problems, and Solutions* 中未涵盖的其他异常安全主题。本书还讨论有效的面向对象编程以及正确使用标准库的某些方面。
- Herb Sutter. *Exceptional C++ Style:* 40 *New Engineering Puzzles, Programming Problems*, *and Solutions*. Addison-Wesley Professional, 2004. ISBN: 0-201-76042-8。
  讨论一般性编程、优化和资源管理。该书还很好地呈现如何使用非成员函数和单职责原理来编写模块化代码。

- Stephen C. Dewhurst. *C++ Gotchas: Avoiding Common Problems in Coding and Design*. Addison-Wesley Professional, 2002. ISBN: 0-321-12518-5。
  提供 99 个和 C++编程相关的技巧。
- Bruce Eckel and Chuck Allison. *Thinking in C++, Volume 2: Practical Programming*. Prentice Hall, 2003. ISBN: 0-130-35313-2。
  Eckel 所写书籍的第二卷，介绍更高级的 C++主题。
- Ray Lischner. *C++ in a Nutshell*. O'Reilly, 2009. ISBN: 0-596-00298-X。
  一本 C++参考书，覆盖从基础知识到高级主题的所有内容。
- Stephen Prata, *C++ Primer Plus*, 6th ed. Addison-Wesley Professional, 2011. ISBN: 0-321-77640-2。
  最全面的 C++书籍之一。
- *The C++ Reference*，可从 www.cppreference.com 访问。
  C++98、C++03、C++11、C++14、C++17 和 C++20 的优秀参考资源。
- *C++ Resources Network*，网址为 www.cplusplus.com。
  这个网站包含很多与 C++相关的信息，包括 C++语言的完整参考，也包含 C++20 的内容。

I/O 流和字符串

- Cameron Hughes and Tracey Hughes, *Stream Manipulators and Iterators in C++*, www.informit.com/ articles/article.aspx?p=171014。
  这篇好文章解释了如何自定义 C++中的流运算符。
- Philip Romanik and Amy Muntz, *Applied C++: Practical Techniques for Building Better Software*, Addison-Wesley Professional, 2003, ISBN: 0-321-10894-9。
  除了独特的软件开发建议和 C++相关知识，该书还是关于 C++ locale 和 Unicode 支持的最好解释。
- Joel Spolsky, *The Absolute Minimum Every Software Developer Absolutely, Positively Must Know About Unicode and Character Sets (No Excuses!)*，www.joelonsoftware.com/articles/Unicode.html。
  读完 Joel 的这篇阐述本地化重要性的文章后，你肯定想阅读 Joel on Software 网站的其他文章。
- The Unicode Consortium, *The Unicode Standard 5.0*, 5th ed. Addison-Wesley Professional, 2006, ISBN: 0-321-48091-0。
  这是有关 Unicode 的权威书籍，所有使用 Unicode 的开发人员都应该有这本书。
- Unicode, Inc., *Where is my Character?* www.unicode.org/standard/where。
  查找 Unicode 字符和图表的最佳资源。
- Wikipedia.*Universal Character Set*, en.wikipedia.org/wiki/Universal_Character_Set。
  解释 Universal Character Set(UCS)的文章，其中包含 Unicode 标准。

C++标准库

- Peter Van Weert and Marc Gregoire. *C++17 Standard Library Quick Reference*. Apress, 2019. ISBN: 978-1-4842-4922-2。
  简要描述 C++17 标准库提供的所有数据结构、算法和函数。

- Rainer Grimm. *The C++ Standard Library,* 3rd ed. Leanpub, 2020。
  第 3 版包括了 C++20。该书的目标是在大约 300 页的篇幅内提供 C++20 标准库的简明参考。该书假设你对 C++很熟悉。
- Nicolai M. Josuttis. *The C++ Standard Library: A Tutorial and Reference,* 2nd ed. Addison-Wesley Professional, 2012. ISBN: 0-321-62321-5。
  该书覆盖整个标准库，包括 I/O 流和字符串，还包含容器和算法。这是一本很好的参考书。
- Scott Meyers, *Effective STL:* 50 *Specific Ways to Improve Your Use of the Standard Template Library*, Addison-Wesley Professional, 2001, ISBN: 0-201-74962-9。
  Meyers 编写该书时采取与 *Effective C++*系列书籍同样的思路。该书以使用标准库为目标提供了很多技巧，但不是一本参考书或教程。
- Stephan T. Lavavej. *Standard Template Library (STL)*. channel9.msdn.com/Shows/Going+Deep/C9-Lectures-Introduction-to-STL-with-Stephan-T-Lavavej。
  关于 C++标准库的一系列有趣的视频演讲。
- David R. Musser, Gillmer J. Derge, and Atul Saini. *STL Tutorial and Reference Guide: Programming with the Standard Template Library*, 2nd ed. Addison-Wesley Professional, 2001. ISBN: 0-321-70212-3。
  该书类似于 Josuttis 的教材，但是只覆盖标准库中的容器和算法等部分。

C++模板

- Herb Sutter, “Sutter’s Mill: Befriending Templates.” *C/C++ User’s Journal*, www.drdobbs.com/befriending-templates/184403853。
  有关编写类的友元函数模板的最佳解释文章。
- David Vandevoorde, Nicolai M. Josuttis, and Douglas Gregor. *C++ Templates: The Complete Guide*, 2nd ed. Addison-Wesley Professional, 2017. ISBN: 0-321-71412-1。
  有关 C++模板的一切。该书要求具备很好的 C++综合背景。
- David Abrahams and Aleksey Gurtovoy, *C++ Template Metaprogramming: Concepts, Tools, and Techniques from Boost and Beyond,* Addison-Wesley Professional, 2004, ISBN: 0-321-22725-5。
  该书为程序员的日常工作提供实用的元编程工具和技术。

C++11/C++14/C++17/C++20

- *C++ Standards Committee Papers*, www.open-std.org/jtc1/sc22/wg21/docs/papers。
  由 C++标准委员会编写的大量白皮书。
- Nicolai M. Josuttis. *C++17 - The Complete Guide.* NicoJosuttis, 2019. ISBN: 3-967-30017-X
  该书介绍了 C++17 的所有特性，重点介绍了这些特性如何影响日常编程，组合特性会产生什么影响，以及在实践中如何从中受益。
- Wikipedia. *C++11*. en.wikipedia.org/wiki/C%2B%2B11。
- Wikipedia. *C++14*. en.wikipedia.org/wiki/C%2B%2B14。
- Wikipedia. *C++17*. en.wikipedia.org/wiki/C%2B%2B17。
- Wikipedia. *C++20*. en.wikipedia.org/wiki/C%2B%2B20。
  这 4 篇 Wikipedia 文章描述 C++11、C++14、C++17 和 C++20 的新功能。

- Scott Meyers, *Presentation Materials: Overview of the New C++(C++11/14)*, Artima, 2013. www.artima.com/shop/overview_of_the_new_cpp。
  包含 Scott Meyers 有关新 C++标准的培训课程的展示材料，这是获得所有 C++11 新特性列表和选择 C++14 特性的极佳参考。
- *ECMAScript* 2017 *Language Specification*. www.ecma-international.org/publications/files/ECMA-ST/ECMA-262.pdf。
  C++使用的正则表达式语法和 ECMAScript 语言使用的正则表达式语法一样，这个规范文档描述了 ECMAScript 语言。

统一建模语言(UML)

- Russ Miles and Kim Hamilton, *Learning UML* 2.0: *A Pragmatic Introduction to UML*. O'Reilly Media, 2006, ISBN:0-596-00982-8。
  关于 UML 2.0 的可读性非常好的一本书，它的示例使用了 Java，但可不太费力地转换为 C++。

算法和数据结构

- Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein, *Introduction to Algorithms*, 3rd ed. The MIT Press, 2009, ISBN: 0-262-03384-4。
  最流行的算法书籍之一，涵盖所有常见的数据结构和算法。
- Donald E. Knuth. *The Art of Computer Programming Volume* 1*: Fundamental Algorithms, 3*rd ed. Addison-Wesley Professional, 1997. ISBN: 0-201-89683-1。
- Donald E. Knuth. *The Art of Computer Programming Volume* 2*: Seminumerical Algorithms, 3*rd ed. Addison-Wesley Professional, 1997. ISBN: 0-201-89684-2。
- Donald E. Knuth. *The Art of Computer Programming Volume* 3*: Sorting and Searching, 2*nd ed. Addison-Wesley Professional. 1998. ISBN: 0-201-89685-0。
- Donald E. Knuth. *The Art of Computer Programming Volume* 4*A: Combinatorial Algorithms, Part* 1. Addison-Wesley Professional, 2011. ISBN: 0-201-03804-8。
  如果喜欢数学的严谨，那么 Knuth 的这 4 部巨著是最好的算法和数据结构教材。但是，如果没有数学知识和理论以及计算机科学相关的知识，可能不好领悟这些书。
- Kyle Loudon. *Mastering Algorithms with C: Useful Techniques from Sorting to Encryption*. O'Reilly Media, 1999. ISBN: 1-565-92453-3。
  一本通俗易懂的数据结构和算法参考书。

随机数

- Eric Bach and Jeffrey Shallit. *Algorithmic Number Theory, Efficient Algorithms*. The MIT Press, 1996. ISBN: 0-262-02405-5。
- Oded Goldreich. *Modern Cryptography, Probabilistic Proofs and Pseudorandomness*. Springer, 2010. ISBN: 3-642-08432-X。
  以上两本书讲解计算伪随机性的理论。

- Wikipedia. *Mersenne Twister,* en.wikipedia.org/wiki/Mersenne_twister。
  通过梅森旋转算法生成伪随机数的数学理论。

开源软件

- The Open Source Initiative：opensource.org。
- The GNU Operating System — Free Software Foundation：gnu.org。
  这是推动开源运动的两个主要网页，解释了它们的哲学，并提供了有关获得开源软件以及为开源软件开发做贡献的信息。
- Boost C++ Libraries：boost.org。
  提供大量同行评议的、可移植的免费 C++源代码库，肯定值得一看。
- GitHub(www.github.com)以及 SourceForge(www.sourceforge.net)。

这两个网站为很多开源项目提供服务。这是查找有用开源软件的极佳资源。

- www.codeguru.com 和 www.codeproject.com。
  查找免费库以及可在自己的项目中重用的代码的极佳资源。

软件工程方法论

- Robert C. Martin. *Agile Software Development, Principles, Patterns, and Practices*. Pearson, 2013. ISBN: 978-1292025940。
  面向“战壕里”的软件工程师。该书主要介绍技术(原理、模式和过程)，帮助软件工程师有效地管理日趋复杂的操作系统和应用程序。
- Mike Cohn. *Succeeding with Agile: Software Development Using Scrum*. Addison-Wesley Professional, 2009. ISBN: 0-321-57936-4。
  开始使用 Scrum 方法的优秀指南。
- David Thomas and Andrew Hunt, *The Pragmatic Programmer, Your Journey To Mastery, 2*nd ed. Addison-Wesley Professional, 2019. ISBN: 978-0135957059。
  一本经典书籍的最新版，是每位软件工程师的必读书籍。在第 1 版出版后将近 20 年的时间里，所提的建议历久弥新。它分析核心过程，指出了要使创建的软件能供用户方便地使用，团队和成员该做什么。
- Barry W. Boehm, TRW Defense Systems Group, “A Spiral Model of Software Development and Enhancement” *IEEE Computer*, 21(5): 61–72, 1988。
  这篇重要论文描述了当时软件开发的状态，并提出了 Spiral 模型。
- Kent Beck and Cynthia Andres, *Extreme Programming Explained: Embrace Change*, 2nd, ed. Addison-Wesley Professional, 2004, ISBN: 0-321-27865-8。
  推崇极限编程为新型软件开发方法的一系列书籍中的一本。
- Robert T. Futrell, Donald F. Shafer, and Linda Isabell Shafer, *Quality Software Project Management*, Prentice Hall, 2002, ISBN: 0-130-91297-2。
  负责管理软件开发过程的人员的指南。

- Robert L. Glass, *Facts and Fallacies of Software Engineering*, Addison-Wesley Professional, 2002, ISBN: 0-321-11742-5。
  该书讨论软件开发中几个不同的方面，展示一些隐藏的道理。
- Philippe Kruchten, *The Rational Unified Process: An Introduction*, 3rd ed. Addison-Wesley Professional, 2003, ISBN: 0-321-19770-4。
  概述 RUP，包括其使命和过程。
- Edward Yourdon, *Death March* ,2nd ed. Prentice Hall, 2003, ISBN: 0-131-43635-X。
  介绍软件开发中的策略和现实，是一本极佳的启蒙读物。
- Wikipedia. *Scrum*, en.wikipedia.org/wiki/Scrum_(software_development)。
  对 Scrum 方法做了详细讨论。
- *Manifesto for Agile Software Development*, agilemanifesto.org/。
  完整的敏捷开发宣言。
- Wikipedia. *Version control*. en.wikipedia.org/wiki/Version_control。
  解释版本控制系统的概念，包括一些可用的解决方案。

编程风格

- Bjarne Stroustrup and Herb Sutter. *C++ Core Guidelines*. github.com/isocpp/CppCoreGuidelines/blob/master/CppCoreGuidelines.md。
  讲述如何正确使用 C++，旨在帮助人员高效地使用现代 C++。
- Martin Fowler. *Refactoring: Improving the Design of Existing Code, 2*nd ed. Addison-Wesley Professional, 2018, ISBN: 0-134-75759-9。
  一本讲解识别和改进糟糕代码实践的经典书籍的新版。
- Herb Sutter and Andrei Alexandrescu, *C++ Coding Standards:* 101 *Rules, Guidelines, and Best Practices*, Addison-Wesley Professional, 2004, ISBN: 0-321-11358-0。
  关于 C++设计和编码风格的必备书籍。这里的“编码标准”并非意味着“应当为代码留多少缩进量”。本书包含 101 条最佳实践、习惯语法和常见陷阱，可帮助编写正确、易懂、高效的 C++代码。
- Diomidis Spinellis, *Code Reading: The Open Source Perspective*, Addison-Wesley Professional, 2003, ISBN: 0-201-79940-5。
  这本独特的书籍用反向思维讲解编程风格的问题，教会读者正确地阅读代码，成为更优秀的程序员。
- Dimitri van Heesch, *Doxygen*. www. doxygen.nl/index.html。
  一个高度可定制的程序，从源代码和注释生成文档。
- John Aycock, *Reading and Modifying Code,* John Aycock, 2008, ISBN 0-980-95550-5。
  内容简练，给出代码最常见操作的相关建议，包括阅读、修改、测试、调试和编写代码。
- Wikipedia. *Code Refactoring*. en.wikipedia.org/wiki/Refactoring。
  讨论代码重构的意义，包括一些重构的技术。
- Google. *Google C++ Style Guide.* google.github.io/styleguide/cppguide.html。
  讨论 Google 中使用的 C++风格指南。

计算机体系结构

- David A. Patterson and John L. Hennessy, *Computer Organization and Design: The Hardware/Software Interface*, 4th ed. Morgan Kaufmann, 2011, ISBN: 0-123-74493-8。
- John L. Hennessy and David A. Patterson, *Computer Architecture: A Quantitative Approach,* 5th ed. Morgan Kaufmann, 2011. ISBN: 0-123-83872-X。
  这两本书提供大部分软件工程师需要知道的所有关于计算机体系结构的知识。

效率

- Dov Bulka and David Mayhew, *Efficient C++: Performance Programming Techniques*, Addison-Wesley Professional, 1999, ISBN: 0-201-37950-3。
  少有的专门讨论高效C++编程的书籍之一，既包含语言层次的效率，又包含设计层次的效率。
- GNU gprof. sourceware.org/binutils/docs/gprof/。
  提供关于 gprof 性能剖析工具的信息。

测试

- Elfriede Dustin, *Effective Software Testing:* 50 *Specific Ways to Improve Your Testing,* Addison-Wesley, 2002, ISBN: 0-201-79429-2。
  尽管这本书面向 QA 专业人士，但任何软件工程师都可通过其中介绍的软件测试过程获益。

调试

- Diomidis Spinellis. *Effective Debugging:* 66 *Specific Ways to Debug Software and Systems.* Addison-Wesley Professional, 2016. ISBN: 978-0134394794.
  该书通过系统地对最有用的调试方法、策略、技术和工具进行分类、解释和说明，帮助经验丰富的程序员加速他们的掌握之旅。
- Microsoft Visual Studio Community Edition, microsoft.com/vs。
  Microsoft Visual Studio 的 Community Edition 版本可供以下人员免费使用：学生、开源开发人员以及创建免费和收费应用程序的开发人员。5 人以内的组织也可免费使用。这个版本带有卓越的图形符号调试器。
- The GNU Debugger(GDB)：www.gnu.org/software/gdb/gdb.html。
  GDB 是极佳的命令行符号调试器。
- Valgrind，valgrind.org/。
  Linux 下的开源内存调试工具。
- Microsoft Application Verifier. Part of the Windows 10 SDK, developer.microsoft.com/windows/downloads/windows-10-sdk。
  用于 C++代码的运行时验证工具，能帮助找到微妙的编程错误和安全性问题，用一般的应用程序测试技术很难找到这些问题。

设计模式

- Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides, *Design Patterns: Elements of Reusable Object-Oriented Software*, Addison-Wesley Professional, 1994, ISBN: 0-201-63361-2。
  该书是设计模式的开山之作。
- Andrei Alexandrescu, *Modern C++ Design: Generic Programming and Design Patterns Applied,* Addison-Wesley Professional, 2001, ISBN: 0-201-70431-5。
  为 C++编程的高度可重用的代码和模式提供方法。
- John Vlissides, Pattern Hatching: *Design Patterns Applied,* Addison-Wesley Professional, 1998, ISBN: 0-201- 43293-5。
  该书解释如何实际应用模式。
- Eric Freeman, Bert Bates, Kathy Sierra, and Elisabeth Robson, *Head First Design Patterns*, O'Reilly Media, 2004, ISBN: 0-596-00712-4。
  该书不仅讲解设计模式，还更进一步给出使用设计模式的好例子和坏例子，并说明每种设计模式的有力推论。
- Wikipedia. *Software design pattern.*en.wikipedia.org/wiki/Design_pattern_(computer_science)。
  描述计算机编程中使用的大量设计模式。

操作系统

- Abraham Silberschatz, Peter B. Galvin, and Greg Gagne. *Operating System Concepts*, 10th ed. Wiley, 2018. ISBN: 1-119-45633-9。
  讨论操作系统的优秀著作，包括多线程的问题，例如死锁和争用条件。

多线程编程

- Anthony Williams, *C++ Concurrency in Action*，2nd ed. 2019, ISBN: 1-617-29469-1。
  有关多线程编程实践的一本优秀书籍，包含最新的 C++线程库。
- Cameron Hughes and Tracey Hughes, *Professional Multicore Programming: Design and Implementation for C++ Developers*, Wrox, 2008, ISBN: 0-470-28962-7。
  该书适用于想要转移到多核编程的不同层次的开发人员。
- Maurice Herlihy and NirShavit, *The Art of Multiprocessor Programming*, Morgan Kaufmann, 2012, ISBN: 0-123-70591-6。
  一本关于多处理器和多核系统编程的优秀著作。