

C++ 17

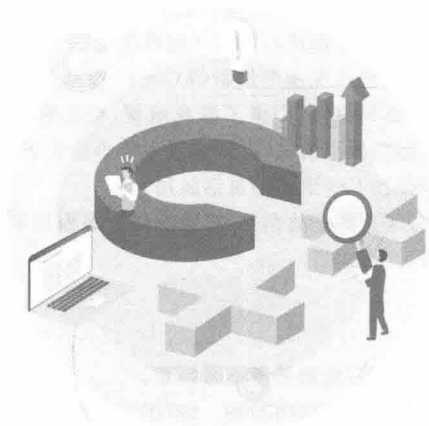
从入门到精通

◎ 董洪伟 编著



清华大学出版社





C++ 17从入门到精通

◎ 董洪伟 编著



清华大学出版社
北京

内 容 简 介

本书以简明扼要的语言、配合丰富的实例,针对初学者从最基础的变量、表达式、数组、指针、引用和函数等,到面向对象的类和对象、继承与派生、虚函数与多态,从泛型编程的函数模板和类模板到移动语义、头等函数(函数指针、函数对象、Lambda 表达式),从 C++ 标准库的输入输出流库、容器、迭代器、算法、智能指针等工具到异常处理和 RAII 等,由浅入深地对最新的 C++17 标准语法进行了系统的讲解。对一些关键的语法概念如函数、类与对象、派生类等内容,提供了游戏编程、信息管理、数据结构、机器学习、人工智能等学科领域的一些经典的、实际问题的实战演练,以加强读者将语法知识用于解决各种实际问题 and 进行实际编程能力的训练,让读者领悟和体会 C++ 语言的灵活运用。

本书描述精炼、简单易懂,并有丰富的实战案例,既适合作为编程初学者的学习用书,也适合有编程基础的开发人员迅速学习和掌握现代 C++ 语言。

本书封面贴有清华大学出版社防伪标签,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++17 从入门到精通/董洪伟编著. —北京:清华大学出版社,2019
ISBN 978-7-302-52743-5

I. ①C… II. ①董… 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 067169 号

责任编辑:闫红梅 张爱华

封面设计:刘 键

责任校对:徐俊伟

责任印制:宋 林

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

课件下载: <http://www.tup.com.cn>, 010-62795954

印 装 者:清华大学印刷厂

经 销:全国新华书店

开 本:185mm×260mm 印 张:30.5 字 数:765 千字

版 次:2019 年 8 月第 1 版 印 次:2019 年 8 月第 1 次印刷

印 数:1~1500

定 价:79.00 元

产品编号:081355-01



前言

C++ 编程语言以其具有“可操纵底层硬件”“程序效率高”和“面向对象”的优势被广泛应用于系统软件和应用软件的开发,不但是企业界开发重量级软件或平台的首选语言,也是国内外高校广泛采用的计算机编程教学语言,更是衡量一个程序员功力的标尺。

然而,尽管企业界早已使用 C++11/14/17 标准,但国内高校仍然沿用的是传统的、过时的 C++98 标准,已经和业界普遍使用的现代 C++ 语法标准有很大的脱节。

目前市场上还未见到国内作者编写的现代 C++ 语言教材,虽然有少量国外作者编写的现代 C++ 语言教材,但国外作者的思维模式和语言文化差异使得这些书难以被国内读者,特别是初学者阅读理解,这些书往往都是大部头的著作,令人望而生畏。由于 C++ 语言本身语法的复杂,琐碎的语法讲解使初学者感到枯燥乏味;缺少实践性的例子,初学者很难理解这些语法知识的价值和适用场景,不知如何将语法知识应用于实际编程中。这就造成了许多学过 C++ 语言的计算机专业的本科毕业生实际并没有掌握最基本的 C++ 编程知识,缺乏实际应用编程的能力。

本书作者在 C++ 课程的教学中心深感缺少一本适合中国读者的、没有冗余语句、注重实战的现代 C++ 教材。于是参考了各种英文教材和网上资料,编写了这本面向初学者的、遵循 C++17 标准的语法与实践结合的入门教材,并且书中的实战案例对于有经验的程序员也很有参考价值。

本书的编写遵循下面几个目标。

(1) 针对没有任何编程基础的学生,直接讲解最新的 C++17 标准,避免传统的从 C 到 C++ 的教学模式和国内高校采用的过时的 C++98 标准语法,使读者可以直接学会使用最新的现代 C++ 语法特征,如 auto、range for、Lambda、移动语义、变长模板等,避免了 C 和 C++ 比较式教学带来的混乱,也不需要浪费时间在 C++ 旧标准语法上,可使无编程基础的初学者在较短时间内快速掌握现代 C++ 编程语言的核心内容。

(2) 突出重点,讲解主要的常用语法,而不是一本面面俱到的语法手册,由浅入深、由易到难,尽量用浅显易懂的例子说明语法概念,力求简明扼要,避免空洞的概念和冗长的描述。

(3) 从入门到实战,只有通过具体、长期的实战训练,才能逐步熟练精通一门编程语言。

语法知识可能短时间就能理解,但只有通过大量的实战训练才能真正熟练使用一门编程语言。本书准备了从游戏编程、信息管理、数据结构、机器学习、人工智能等不同领域的一些经典的实战案例,希望这些案例能够帮助读者消化语法知识、提高学习兴趣,逐步将 C++ 用于解决各种实际问题,避免出现“只会考试而不会编程”的普遍问题。

由于实战案例涉及一些其他学科专业知识,初学者和教师可以根据自己的需要选读实战部分,甚至完全跳过实战案例也不影响 C++17 语言的教学。本书的源代码可以登录清华大学出版社网站(www.tup.com.cn)下载。

由于作者知识和水平所限,错误之处在所难免,欢迎读者批评指正。

作者

2019年1月



目录

第 1 章 C++ 介绍	1
1.1 程序与编程语言	1
1.1.1 计算机是什么	1
1.1.2 计算机编程	3
1.1.3 编译器、解释器和 C++ 语言	4
1.1.4 C++ 语言介绍	4
1.1.5 C++ 程序开发步骤	5
1.2 C++ 程序结构	6
1.2.1 最简单的 C++ 程序	6
1.2.2 函数	6
1.2.3 语句	7
1.2.4 程序注释	7
1.2.5 hello world 程序	8
1.2.6 标准输入输出库和 cout	8
1.2.7 名字空间	9
1.2.8 字符串和字符	10
1.2.9 运算符和运算数	11
1.2.10 宏定义 #define	12
1.2.11 变量	13
1.2.12 标准输入流对象 cin	14
1.2.13 用户定义类型	14
1.3 数和字符的表示	15
1.3.1 数的表示	15
1.3.2 字符的表示	17

1.4	编译、执行 C++ 程序	19
1.5	习题	20
第 2 章	变量和类型	23
2.1	变量	23
2.1.1	变量的定义及初始化	23
2.1.2	auto	24
2.1.3	typeid 运算符	24
2.1.4	decltype	25
2.1.5	赋值运算符 =	25
2.1.6	const	25
2.1.7	标识符、关键字、文字量	26
2.2	数据类型	26
2.2.1	基本类型	27
2.2.2	sizeof 运算符	29
2.2.3	文字量	30
2.2.4	格式化输出	33
2.2.5	类型转换	34
2.2.6	类型别名	36
2.2.7	枚举	37
2.3	局部变量与全局变量、变量的作用域与生命期	37
2.3.1	程序块、局部变量和全局变量	37
2.3.2	作用域和生命期	38
2.4	习题	39
第 3 章	运算符与表达式	42
3.1	运算符	42
3.1.1	运算符的分类	42
3.1.2	优先级和结合性	43
3.2	表达式	44
3.3	算术运算符	44
3.3.1	算术运算符需要注意的几个问题	45
3.3.2	自增++和自减--	46
3.3.3	数学计算函数库 cmath	47
3.4	位运算	49
3.5	赋值运算符	51
3.6	关系运算符	52
3.7	逻辑运算符	54
3.8	特殊运算符	54

3.8.1	条件运算符	54
3.8.2	逗号运算符	55
3.9	习题	55
第4章	语句	58
4.1	简单语句、复合语句和控制语句	58
4.1.1	简单语句	58
4.1.2	复合语句	58
4.1.3	控制语句	59
4.2	条件语句	59
4.2.1	if 语句	59
4.2.2	switch 语句	62
4.2.3	if/switch 语句中的初始化语句	65
4.3	循环语句	66
4.3.1	while 语句	66
4.3.2	for 语句	68
4.4	跳转语句	69
4.5	实战：控制台游戏——Pong 游戏	70
4.5.1	Pong 游戏	70
4.5.2	初始化	71
4.5.3	绘制场景	71
4.5.4	让球动起来	73
4.5.5	事件处理：用挡板击打球	77
4.6	习题	78
第5章	复合类型：数组、指针和引用	82
5.1	引用	82
5.2	指针	83
5.2.1	指针类型	83
5.2.2	指针的其他运算	85
5.2.3	void * 无类型指针	85
5.2.4	指针的指针	86
5.2.5	指针的引用	87
5.2.6	引用和指针的比较	87
5.3	数组	87
5.3.1	数组和下标运算符	87
5.3.2	复杂的数组声明	89
5.3.3	C 风格字符串	90
5.3.4	指针访问数组	91

5.3.5	range for	94
5.3.6	多维数组	95
5.4	动态内存	98
5.4.1	程序堆栈区	98
5.4.2	new 和 delete 运算符	99
5.4.3	动态内存表示多维数组	101
5.5	const 修饰符	102
5.5.1	const 和指针	102
5.5.2	const 对象的引用	104
5.6	实战:查找、排序、最短路径	105
5.6.1	二分查找	105
5.6.2	排序:冒泡、选择	108
5.6.3	Floyd 最短路径算法	109
5.7	习题	113
第 6 章	函数	119
6.1	函数是命名的程序块	119
6.1.1	最大公约数	119
6.1.2	函数的定义	122
6.2	静态变量	124
6.3	函数的形参	125
6.3.1	参数传递	125
6.3.2	默认参数	126
6.3.3	数组作为形参	127
6.3.4	const 与形参	129
6.3.5	可变数目的形参	129
6.4	递归函数:调用自身的函数	131
6.4.1	递归和递归函数	131
6.4.2	实战:二分查找的递归实现	133
6.4.3	实战:汉诺塔问题	133
6.4.4	实战:快速排序算法	135
6.4.5	实战:迷宫问题	137
6.5	函数重载与重载解析	139
6.5.1	函数重载	139
6.5.2	重载解析	140
6.5.3	const 对象的引用或指针	142
6.6	inline 函数	142
6.7	constexpr	143
6.8	实战:二维字符图形库 ChGL	145

6.8.1	如何在字符终端上绘图	145
6.8.2	字符图形库 ChGL	146
6.8.3	曲线绘制 API 函数 plot()	149
6.9	实战：基于 ChGL 的控制台游戏	151
6.9.1	游戏程序的框架	151
6.9.2	用 ChGL 和函数重写 Pong 游戏	151
6.10	实战：机器学习-线性回归	156
6.10.1	机器学习	156
6.10.2	假设函数、回归和分类	157
6.10.3	线性回归	157
6.10.4	多变量函数的最小值、正规方程	158
6.10.5	梯度下降法	159
6.10.6	梯度下降法求解线性回归问题：模拟数据	160
6.10.7	批梯度下降法	165
6.10.8	房屋价格预测	166
6.10.9	样本特征的规范化	167
6.10.10	预测房屋价格	170
6.11	习题	170
第 7 章	类和对象	174
7.1	面向对象编程	174
7.2	类	177
7.2.1	定义一个类	177
7.2.2	定义类的对象(变量)	178
7.2.3	成员函数	180
7.2.4	this 指针	180
7.2.5	类对象的大小	183
7.3	构造函数	183
7.3.1	创建类对象的构造函数	183
7.3.2	初始化成员列表	187
7.3.3	拷贝构造函数	187
7.3.4	赋值运算符：operator=	189
7.3.5	隐式类型转换、explicit	190
7.3.6	委托构造函数	192
7.3.7	delete	193
7.3.8	类对象数组	193
7.3.9	类体外定义成员函数和构造函数	194
7.4	访问控制和接口	195
7.5	const 对象、const 成员函数、mutable 成员变量	196

7.5.1	const 对象和 const 成员函数	196
7.5.2	重载 const	198
7.5.3	mutable 成员变量	200
7.6	析构函数	200
7.7	静态成员	202
7.7.1	非静态成员变量和静态成员变量	202
7.7.2	静态常量	204
7.7.3	静态成员函数	205
7.7.4	类自身类型的静态成员变量	205
7.8	友元	207
7.9	内联成员函数	207
7.10	重新定义拷贝构造函数和赋值运算符函数	208
7.11	实战：线性表及应用	209
7.11.1	线性表	209
7.11.2	线性表的顺序实现：顺序表	211
7.11.3	线性表的链式实现：链表	215
7.11.4	实现一个图书管理的程序	222
7.12	实战：面向对象游戏——基于链表的贪吃蛇游戏	224
7.12.1	面向对象游戏引擎	224
7.12.2	贪吃蛇游戏	226
7.13	习题	237
第 8 章	运算符重载	242
8.1	运算符重载的 2 种方式	242
8.2	赋值运算符 =	246
8.3	下标运算符 []	246
8.4	输入输出运算符	247
8.5	比较运算符	248
8.6	函数调用运算符 ()	250
8.7	类型转换运算符	250
8.8	自增和自减运算符	252
8.9	可以重载的运算符	253
8.10	实战：矩阵	253
8.11	习题	257
第 9 章	派生类	259
9.1	继承与派生	259
9.1.1	继承关系和派生类	259
9.1.2	is a 和 belong to	260

9.1.3	派生类的定义	260
9.1.4	成员的隐藏	261
9.1.5	继承方式	263
9.1.6	基类指针和派生类指针	264
9.2	派生类的构造函数和析构函数	266
9.3	多继承和虚基类	272
9.3.1	多继承	272
9.3.2	虚基类	274
9.4	多态	276
9.4.1	对象的切割和类型转换	276
9.4.2	基类指针(引用)和向下类型转换	277
9.4.3	虚函数和多态	280
9.4.4	虚函数的一些语法规则	283
9.4.5	基类指针数组	285
9.4.6	虚析构函数	286
9.4.7	纯虚函数和抽象类	286
9.5	实战:仿“雷电战机”游戏	288
9.5.1	精灵	288
9.5.2	游戏引擎 GameEngine	291
9.5.3	碰撞检测和精灵的销毁	295
9.5.4	让敌方战机运动和发射子弹	297
9.6	习题	300
第 10 章	模板	305
10.1	函数模板	305
10.1.1	函数模板的定义与实例化	306
10.1.2	模板参数推断	307
10.1.3	模板专门化	308
10.1.4	函数模板和重载	309
10.1.5	模板的返回类型推断	310
10.1.6	非类型模板参数	311
10.1.7	模板模板参数	313
10.1.8	模板参数的默认值	313
10.1.9	可变模板参数	314
10.1.10	constexpr if	317
10.2	类模板	317
10.2.1	标准库类模板 vector	317
10.2.2	类模板 Vector	320
10.2.3	定义类模板的成员函数	321

10.2.4	类模板的模板参数推断	327
10.2.5	类模板的专门化	328
10.2.6	类模板的友元	329
10.2.7	类模板 <code>std::initializer_list<></code>	330
10.3	实战：强化学习 Q-Learning 求解最佳路径	332
10.3.1	强化学习	332
10.3.2	Q-Learning	334
10.3.3	Q-Learning 的 C++ 实现	336
10.4	习题	343
第 11 章	移动语义	347
11.1	左值和右值	347
11.1.1	左值和右值概述	347
11.1.2	左值和右值的转换	349
11.1.3	左值引用和右值引用	349
11.2	移动	350
11.2.1	复制和移动	350
11.2.2	移动构造函数	353
11.2.3	移动赋值运算符函数	353
11.2.4	<code>std::move</code>	354
11.2.5	右值引用	355
11.2.6	<code>push_back()</code>	355
11.3	习题	357
第 12 章	函数指针、函数对象、Lambda 表达式	359
12.1	函数指针	359
12.1.1	函数类型和函数指针类型	359
12.1.2	给函数指针类型起别名	361
12.1.3	函数指针作为其他函数的参数	361
12.2	函数对象	363
12.3	Lambda 表达式	366
12.3.1	定义和使用 Lambda 表达式	366
12.3.2	捕获子句	368
12.3.3	返回类型	370
12.3.4	Lambda 表达式的实质	371
12.4	<code>std::function</code>	371
12.5	<code>std::bind</code>	374
12.6	习题	376

第 13 章 C++标准库介绍	378
13.1 输入输出流库	379
13.1.1 C++的 I/O 流库	379
13.1.2 格式化输入输出	382
13.1.3 非格式化输入输出	386
13.1.4 文件位置	392
13.1.5 流状态	393
13.1.6 管理输出缓冲区	395
13.1.7 文件输入输出	395
13.1.8 字符串流	397
13.2 容器	399
13.2.1 标准容器	399
13.2.2 序列容器	401
13.2.3 容器适配器	406
13.2.4 关联容器	408
13.3 迭代器	411
13.3.1 迭代器及其分类	411
13.3.2 迭代器适配器	416
13.3.3 数组、字符串和迭代器	423
13.4 算法	423
13.4.1 自定义通用算法	424
13.4.2 策略参数	425
13.4.3 标准库的常用算法	426
13.5 智能指针	442
13.5.1 raw 指针和智能指针	442
13.5.2 unique_ptr	443
13.5.3 shared_ptr	447
13.5.4 weak_ptr	448
13.6 字符串	449
13.6.1 字符: < ctype >、< cwstring >	449
13.6.2 C 风格字符串	449
13.6.3 C++的字符串	450
13.7 习题	455
第 14 章 异常处理	459
14.1 错误和异常处理	459
14.1.1 错误的分类	459
14.1.2 传统的错误处理方法	459

14.1.3	C++的异常处理	460
14.2	throw、try、catch	461
14.2.1	throw	461
14.2.2	try、catch	461
14.2.3	异常类型的匹配	463
14.3	堆栈展开和 RAII	464
14.3.1	堆栈展开	464
14.3.2	资源获取即初始化	466
14.4	习题	470
参考文献		473

C++介绍

1.1 程序与编程语言

1.1.1 计算机是什么

计算机是一种根据指令对数据处理的通用计算设备。每台计算机都有一个称为中央处理器(CPU)的微处理器芯片执行对数据处理的指令,不同计算机的指令集是不一样的。

1. 计算机指令

计算机接受一系列指令作为输入,逐个处理它们,并且通常显示某种输出表示它已完成的操作。这类似于人们日常生活中通过一系列操作步骤完成一个任务的方式,如一个人通过如下一系列步骤完成“做饭”的任务。

- (1) 从容器(米桶)中取出米,放入洗米盆中。
- (2) 用自来水对洗米盆中的米进行冲洗。
- (3) 如果电饭锅没洗净,洗净电饭锅。
- (4) 打开电饭锅盖,将米和水放入电饭锅中。
- (5) 插上电源,按下开关。
- (6) 饭后,拔下电源(任务结束)。

虽然人们可以理解自然语言(如英语)中的复杂指令,但计算机只能理解为用计算机语言表达的非常简单的机器指令集中的指令。不管多么复杂的计算,在计算机内都是被分解成许多简单的、逐条执行的机器指令。告诉计算机如何执行复杂任务的指令序列称为程序。

以下是一些简单的计算机指令示例。

- (1) 算术: 加、减、乘、除法。这些通常被称为**算术操作**。
- (2) 比较: 比较两个数字,看哪个较大或者它们是否相等。这些通常被称为**逻辑操作**。
- (3) 分支: 跳转到程序中的另一条指令,并从那里继续。这些通常被称为**控制语句**。

2. 计算机的组成部分

1) 计算机的主要类型的组件

计算机包含 4 种主要类型的组件。

(1) 输入：允许计算机从用户接收信息的任何设备，包括键盘、鼠标、扫描仪和话筒等。

(2) 处理：处理信息的计算机组件。计算机的主要处理部件是中央处理器(CPU)，但在现代计算机中也可能有其他处理器。例如，许多图形卡都带有图形处理器(GPU)，GPU 以前只用于处理图形，现在也可用于通用程序。

(3) 存储：存储信息的组件，包括主存储器(也称为内存)和二级存储器(如硬盘驱动器、CD 或闪存盘等外部存储器)。存储器是存储运行程序的指令和数据的地方。

(4) 输出：用于向用户显示信息的任何设备，包括显示器、扬声器和打印机。

2) 举例理解计算机

可以通过自动售票机来理解计算机(尽管自动售票机严格地说并不是计算机)。

(1) 输入：投币口和选择按钮是自动售票机的输入设备。

(2) 处理：当进行选择后，自动售票机会执行以下几个步骤——验证是否有满足条件的票、验证身份信息、检查和验证是否收到足够的资金、修改数据库、计算差额。执行所有这些步骤的机器都可以被认为是处理器。

(3) 存储：自动售票机需要在某个地方保存信息，如票的库存、价格等。

(4) 输出：自动售票机显示结果、打印票。

3. 中央处理器

中央处理器(CPU)是计算机中最重要的部分，是计算机的大脑，负责计算、处理数据、控制其他设备等。它有几个重要的子组件，具体如下所示。

(1) 算术逻辑单元(ALU)：执行算术和比较操作。

(2) 控制单元：确定下一个要执行的指令。

(3) 寄存器：形成一个高速存储区以保存临时结果。

不同种类的 CPU 可以理解不同的指令集。例如，Intel IA-32、x86-64、IBM PowerPC 和 ARM。

4. 存储器

计算机将信息(程序、数据)存储在存储器(memory)中，存储器有两种类型：主存储器(也称为内存)和辅助存储器(也称为外存)。

主存储器直接连接 CPU(或其他处理单元)，通常称为 RAM(随机存取存储器)。计算机关闭时，大多数主存储器都会丢失其内容，即具有易失性。

可以将主存储器想象成一个排成一列的存储器单元，每个单元都可以通过其存储器地址寻址。对于第一个单元，地址从零开始，并且每个后续单元的地址比它之前的地址多一个，正如一个班级中学生的学号从 1 开始依次递增。每个单元只能保存长度固定的用二进制表示的数值，但 CPU 可以随时用新的数值替换原有内容。

辅助存储器比主存储器便宜，但可以存储更多的内容。虽然它慢得多，但它是非易失性的，也就是说，即使在计算机关闭后其内容也会保留，如硬盘和闪存盘。

计算机的操作系统提供操作辅助存储器的高级接口。这些接口允许信息以文件的形式