

2 CD-ROMs



SAMS

计算机技术

译林

精选系列

21天学通

C++

(第三版)

〔美〕 Jesse Liberty 著
康博创作室 译

人民邮电出版社
www.pptph.com.cn



77512
LBD/1

计算机技术译林精选系列

21 天学通 C++ (第三版)

[美] 丁利伯特^著 Jesse Liberty

康博创作室 译

人民邮电出版社

计算机技术译林精选系列
21 天学通 C++ (第三版)

◆ 著 [美] Jesse Liberty
译 康博创作室
责任编辑 李 际

JS266/12

◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号
邮编 100061 电子函件 315@pptph.com.cn
网址 <http://www.pptph.com.cn>
北京汉魂图文设计有限公司制作
北京顺义振华印刷厂印刷
新华书店总店北京发行所经销

◆ 开本:787×1092 1/16
印张:52.5

字数:1 302 千字

2000 年 8 月第 1 版

印数:1 - 5 000 册

2000 年 8 月北京第 1 次印刷

著作权合同登记 图字:01 - 1999 - 0482 号

ISBN 7-115-08569-2/TP·1664

定价:105.00 元

内容提要

本书全面系统地介绍了 C++ 编程的基本知识,包括 C++ 的基本数据类型、基本语法和面向对象编程的基础知识,同时介绍了 C++ 新增的内容。全书共分 21 章,读者大约用一天时间可学完一章的内容。

本书既可作为 C++ 新手开始学习 Visual C++ 6.0 编程的入门书籍,也可作为有经验的 C++ 编程人员的编程参考书。书中还提供了大量实例代码和练习及测验题,以加深广大读者对所学知识的理解。

版权声明

Jesse Liberty: Teach Yourself C++ in 21 Days (Third Edition)。

Authorized translation from English language edition published by SAMS Publishing。

Copyright © 1999 by SAMS Publishing。

All rights reserved. For sale in Mainland China only。

本书中文简体字版由美图 SAMS 出版公司授权人民邮电出版社出版, 未经出版者书面许可, 对本书的任何部分不得以任何方式复制或抄袭。

版权所有, 侵权必究。

译者序

近年来, Visual C++ 受到越来越多的应用程序开发人员的欢迎, 正在成为 Windows 应用程序开发的事实上的标准。有关用 Visual C++ 的各个版本进行编程的书籍也越来越多。然而这些书籍基本上都是对使用 C++ 各个版本开发应用程序的介绍, 而有关 C++ 基本知识的书籍却不多见。这给广大 C++ 初学者带来了许多不便。因此有必要推出一本有关 C++ 编程基础的书籍。

本书作者曾编写过多本有关 C++ 及面向对象分析和设计的书籍, 在用 C++ 编程、面向对象分析和设计方面具有丰富经验。本书的目的是帮助读者自学如何用 C++ 编程。在总共 21 天(每天一章)的学习中, 读者将学到有关 C++ 的基本数据类型、输入输出、循环和数组、面向对象编程、模板、创建 C++ 应用程序等方面的基本知识。本书的所有内容都经过了精心编排, 并采用循序渐进的方式进行组织。

为了帮助读者加深理解所学内容, 每章结束时都提供了一组常见问题及答案、练习题和测验。每周课程结束时都对本周内容进行了总结和复习。通过对照检查附录所给出的测验和练习题的答案, 读者可评估自己的学习情况。

用本书学习 C++ 不要求读者以前有任何编程经验。本书从入门开始, 既讲授 C++ 语言, 也介绍 C++ 编程中涉及到的一些概念。本书提供了大量语法实例及对代码的详细分析, 它们是引导读者快速进入 C++ 的优秀向导。无论你是刚开始编程还是已有一些编程经验, 都会发现本书的精心安排使得学习 C++ 变得快捷又轻松。

参加本书翻译的人员有阙开良、郑秋秋、李锋、张瑞、张开良、宋勇、曹峰、许学军、陈安全、孙江宏、赵文、李建明、杨霞、王为林、回学明等。

在翻译过程中, 译者本着严谨求实的态度, 参考了各种相关书籍和资料。对书中出现的新术语, 均进行了仔细的推敲和斟酌; 对于疑难和分歧进行了广泛的讨论, 并最终达成了一致的意见。但是由于时间仓促, 加之本书内容较新, 疏漏和争议在所难免。我们热诚期望广大读者多提宝贵意见。

康博创作室

作者简介

Jesse Liberty 是编著了 6 本有关 C++ 及面向对象分析和设计的书籍的作者,此外他还是 C++ Report 月刊的定期专栏作者。他是 Liberty Associates 公司的总裁,该公司提供有关面向对象软件开发的网上培训、指导、咨询和交互编程,该公司的网址是 <http://www.libertyassociates.com>。

Jesse 曾是一名杰出的 AT&T 软件工程师,花旗银行开发部的副总裁。他和他的妻子 Stacey 及两个女儿 Robin 和 Rachel 共同生活在 Massachusetts 州 Cambridge 郊区。你可以通过 jliberty@libertyassociates.com 与他联系。另外,在其站点 <http://www.libertyassociates.com> 上单击 Books and Resources 就可得到 Jesse 对他的著作的支持。

第三版中新增内容

本书对前一版本全部内容进行了修改和更新。下面列出本书的部分新特点：

- 对全书内容进行了更新,以符合新发布的 ANSI/ISO C++ 标准。
- 在最新的 Microsoft 和 Borland 编译器上测试了全部源代码,这样可以确保更好地符合 ANSI 标准。
- 提供了有关 C++ 设计模式的补充说明。
- 扩充了标准模板库(STL)和 C++ 标准库。
- 提供了有关名字空间的全部内容。
- 全面修改了有关面向对象链表的讨论。
- 全部重写了面向对象的分析和设计,并介绍了统一建模语言(UML)。
- 对本书的前一版本中的错误进行了更正。
- 根据大量的读者反馈意见新增加了常见问题(FAQ)。

除了上述新增和修改的内容外,读者还可以链接到作者的 Web 站点或 Sams 出版社的站点,以获取最新的源代码和有关本书的最新补充资料。

本书作者的站点:

www.libertyassociates.com

Sams 出版公司的站点:

www.sampublishing.com/codecenter.html

前 言

编写本书的目的是帮助读者自学如何用 C++ 编程。在 21 章里,读者将学会如何管理输入输出、循环和数组、面向对象编程、模板、创建 C++ 应用程序等方面的基本知识。所有内容都是经过精心编排并采用循序渐进的方式组织的。书中给出了示例清单,每个清单都包含完整的示例输出结果和代码分析,以此来说明每章的主题。为了便于读者参考,每个示例都做了清晰的标记。

为了帮助读者加深理解所学内容,每章结束时都提供了一组常见问题及答案、练习题和测验。通过对照检查附录 D 所给出的测验和练习题的答案,读者可评估自己的学习进展情况。

本书读者对象

通过本书学习 C++ 时,并不要求读者以前有任何编程经验。本书从入门开始,既讲授 C++ 语言,也介绍 C++ 编程中涉及到的一些概念。本书提供了大量语法实例及代码的详细分析,它们是引导读者快速进入 C++ 的优秀向导。无论读者是刚开始编程还是已经有了一些编程经验,书中精心安排的内容都会使你的 C++ 学习变得既快又轻松。

本书约定

注意:

这种字体样式用来强调一些信息,它们可使 C++ 编程更有效、更实用。

常见问题:

常见问题有何作用?

答:常见问题提供了对 C++ 语言用法的深入剖析,并澄清一些容易混淆的问题。

警告：

警告用来提醒读者注意在特殊情况下可能出现的问题或副作用。

建议做的事

通过这部分内容,读者可找出某一章中基本原理的简洁摘要。

不该做的事

不要小看这部分所提供的有用信息,它很重要。

在本书的程序清单中,每行代码都编了号。如果在程序清单中看到某行未编号,表示该行是前一行的续行(有的代码行太长,只能分行列出)。在这种情况下,在输入时必须将其按一行输入,而不能分开。

第一篇.....	(1)
第一章 开始	(3)
1.1 引言.....	(3)
1.2 C++ 历史简介	(3)
1.3 程序.....	(4)
1.4 本书要解决的问题.....	(4)
1.4.1 过程编程、结构化编程和面向对象编程	(5)
1.4.2 C++ 和面向对象编程	(6)
1.5 C++ 的发展历程	(7)
1.6 学习 C++ 应该先学 C 吗	(7)
1.7 C++ 和 Java	(7)
1.8 ANSI 标准	(8)
1.9 准备编程.....	(8)
1.10 开发环境	(9)
1.11 编译源代码	(9)
1.12 用链接器生成可执行文件	(10)
1.13 程序开发周期	(10)
1.14 HELLO.cpp——你的第一个 C++ 程序	(10)
1.15 开始学习 Visual C++ 6.0	(13)
1.16 编译错误	(13)
1.17 小结	(14)
1.18 问与答	(14)
1.19 习题	(15)
1.19.1 测验	(15)
1.19.2 练习	(15)
第二章 C++ 程序的组成部分	(17)
2.1 一个简单程序.....	(17)
2.2 cout 简介	(19)
2.3 注释.....	(20)
2.3.1 注释的类型	(21)

2.3.2	使用注释	(21)
2.3.3	对注释的忠告	(22)
2.4	函数	(22)
2.4.1	函数的使用	(23)
2.5	小结	(25)
2.6	问与答	(25)
2.7	习题	(25)
2.7.1	测验	(26)
2.7.2	练习	(26)
第三章	变量与常量	(27)
3.1	什么是变量	(27)
3.1.1	留出内存	(27)
3.1.2	整型量的大小	(28)
3.1.3	有符号和无符号	(29)
3.1.4	基本变量类型	(29)
3.2	定义一个变量	(30)
3.2.1	大小写敏感性	(31)
3.2.2	关键字	(32)
3.3	一次创建多个变量	(32)
3.4	给变量赋值	(33)
3.5	typedef	(34)
3.6	short 与 long 的使用	(35)
3.6.1	无符号整型量的溢出	(36)
3.6.2	有符号整型量的溢出	(36)
3.7	字符型变量	(37)
3.7.1	字符和数字	(38)
3.7.2	特殊打印字符	(38)
3.8	常量	(39)
3.8.1	字面常量	(39)
3.8.2	符号常量	(39)
3.9	枚举型常量	(41)
3.10	小结	(43)
3.11	问与答	(43)
3.12	习题	(44)
3.12.1	测验	(44)
3.12.2	练习	(45)

第四章 表达式和语句	(47)
4.1 语句	(47)
4.1.1 空白	(47)
4.1.2 块和复合语句	(48)
4.2 表达式	(48)
4.3 运算符	(49)
4.3.1 赋值运算符	(49)
4.3.2 数学运算符	(50)
4.3.3 整型量除法和取模	(51)
4.4 赋值运算符与数学运算符的组合	(52)
4.5 自加与自减	(53)
4.6 优先级	(55)
4.7 括号的嵌套	(56)
4.8 真值的本质	(56)
4.8.1 关系运算符	(57)
4.9 if 语句	(58)
4.9.1 缩进样式	(61)
4.9.2 else	(61)
4.9.3 高级 if 语句	(63)
4.10 在嵌套的 if 语句中使用大括号	(65)
4.11 逻辑运算符	(67)
4.11.1 逻辑“与”	(68)
4.11.2 逻辑“或”	(68)
4.11.3 逻辑“非”	(68)
4.12 简化求值	(68)
4.13 关系运算符的优先级	(69)
4.14 有关真假关系的进一步讨论	(69)
4.15 条件运算符(三目运算符)	(70)
4.16 小结	(71)
4.17 问与答	(72)
4.18 习题	(72)
4.18.1 测验	(72)
4.18.2 练习	(73)
第五章 函数	(75)
5.1 什么是函数	(75)
5.2 函数的返回值、参量和变元	(75)
5.3 声明和定义函数	(76)

5.3.1	函数的声明	(76)
5.3.2	函数原型	(77)
5.3.3	定义函数	(79)
5.4	函数的执行	(80)
5.5	局部变量	(80)
5.6	全局变量	(83)
5.7	使用全局变量的注意事项	(84)
5.8	有关局部变量的更多知识	(84)
5.9	函数语句	(86)
5.10	有关函数变元的更多知识	(86)
5.10.1	使用函数作为函数的参量	(86)
5.11	参量作为局部变量	(87)
5.12	有关返回值的更多知识	(88)
5.13	默认参量	(90)
5.14	函数重载	(92)
5.15	有关函数的一些专门话题	(95)
5.15.1	内嵌函数	(96)
5.15.2	递归	(97)
5.16	函数是如何工作的	(101)
5.16.1	抽象层次	(101)
5.16.2	划分 RAM	(102)
5.16.3	堆栈与函数	(104)
5.17	小结	(105)
5.18	问与答	(105)
5.19	习题	(106)
5.19.1	测验	(106)
5.19.2	练习	(106)
第六章	基本类	(109)
6.1	创建新类型	(109)
6.1.1	为什么要创建新类型	(110)
6.2	类和成员	(110)
6.2.1	声明一个类	(110)
6.2.2	有关命名习惯的一点说明	(111)
6.2.3	定义一个对象	(111)
6.2.4	类与对象	(112)
6.3	访问类成员	(112)
6.3.1	给对象赋值而不是给类赋值	(112)

6.3.2	如果你不声明它,你的类就没有它	(113)
6.4	私有和公有	(113)
6.4.1	使成员数据私有	(115)
6.4.2	私有性与安全性	(116)
6.5	实现类的方法	(118)
6.6	构造函数和析构函数	(120)
6.6.1	默认的构造函数和析构函数	(121)
6.6.2	使用默认构造函数	(121)
6.7	成员函数 const	(124)
6.8	界面与实现	(125)
6.9	在何处放置类声明和方法定义	(128)
6.10	内嵌实现	(129)
6.11	用其他类作为成员数据的类	(131)
6.12	结构体	(135)
6.12.1	为什么两个关键字做同一件事	(135)
6.13	小结	(136)
6.14	问与答	(136)
6.15	习题	(137)
6.15.1	测验	(137)
6.15.2	练习	(137)
第七章	程序流程	(139)
7.1	循环	(139)
7.1.1	循环语句的起源——goto 语句	(139)
7.1.2	为什么现在不再使用 goto 语句	(140)
7.2	while 循环	(141)
7.2.1	复杂的 while 语句	(142)
7.2.2	continue 和 break	(144)
7.2.3	while(true)循环	(146)
7.3	do...while 循环	(147)
7.4	Do...while	(148)
7.5	for 循环	(150)
7.5.1	高级 for 循环	(152)
7.5.2	多重初始化和自增运算	(152)
7.5.3	在 for 循环中使用空语句	(153)
7.5.4	空 for 循环	(155)
7.5.5	嵌套循环	(156)
7.5.6	for 循环的作用域	(157)

7.6	循环总结	(158)
7.7	switch 语句	(160)
7.7.1	使用带有菜单的 switch 语句	(163)
7.8	小结	(167)
7.9	问和答	(167)
7.10	习题	(168)
7.10.1	测验	(168)
7.10.2	练习	(168)
第一篇内容回顾		(171)
第二篇		(177)
第八章 指针		(179)
8.1	什么是指针	(179)
8.1.1	在指针中存储地址	(181)
8.1.2	指针名	(183)
8.1.3	间接引用运算符	(183)
8.1.4	指针、地址和变量	(184)
8.1.5	使用指针操作数据	(184)
8.1.6	检查地址	(185)
8.2	为什么要使用指针	(187)
8.3	栈和自由存储区	(188)
8.3.1	new	(189)
8.3.2	delete	(189)
8.4	内存泄漏	(191)
8.5	在自由存储区内创建对象	(192)
8.6	删除对象	(192)
8.7	访问数据成员	(193)
8.8	自由存储区中的成员数据	(195)
8.9	this 指针	(197)
8.10	迷途指针、失控指针或悬浮指针	(199)
8.11	const 指针	(202)
8.11.1	const 指针和 const 成员函数	(202)
8.11.2	const this 指针	(204)
8.12	指针算法	(204)
8.13	小结	(207)
8.14	问与答	(207)
8.15	习题	(208)
8.15.1	测验	(208)

8.15.2 练习	(208)
第九章 引用	(211)
9.1 什么是引用	(211)
9.2 对引用使用取址运算符	(212)
9.2.1 引用不能被重新赋值	(214)
9.3 什么可以被引用	(215)
9.4 空指针和空引用	(217)
9.5 通过引用传递函数参数	(217)
9.5.1 用指针使 swap()工作	(219)
9.5.2 使用引用实现 swap()	(220)
9.6 理解函数头文件和原型	(222)
9.7 返回多个值	(222)
9.8 使用引用传递来提高效率	(225)
9.8.1 传递 const 指针	(228)
9.8.2 使用引用代替指针	(231)
9.9 何时使用引用以及何时使用指针	(233)
9.10 混合使用引用和指针	(234)
9.11 不要返回对不在作用域中的对象的引用	(235)
9.12 返回对一个在堆中的对象的引用	(237)
9.13 指针的传递	(239)
9.14 小结	(240)
9.15 问与答	(240)
9.16 习题	(240)
9.16.1 测验	(241)
9.16.2 练习	(241)
第十章 高级函数	(243)
10.1 重载成员函数	(243)
10.2 使用默认值	(245)
10.3 默认值和重载函数的选择	(248)
10.4 默认构造函数	(248)
10.5 重载构造函数	(249)
10.6 初始化对象	(250)
10.7 复制构造函数	(251)
10.8 运算符重载	(255)
10.8.1 写一个自加函数	(257)
10.8.2 重载前置运算符	(258)