

MS C/C++7.0程序设计

辜健飞 辛达雅 等编译

同济大学出版社

MS C/C++ 7.0 程序设计

辜健飞 辛达雅 等编译

同济大学出版社

(沪)新登字 206 号

内 容 简 介

Microsoft C/C++ 7.0 编译程序软件包为开发 MS-DOS 及 Windows 应用程序提供了最全面的、最新的、产品级的开发环境。Microsoft 的这一最新版本吸收了许多新的、经过升级的特性,有了多方面的增强,如支持 AT & T C++ 2.1、预编译的头文件、自动嵌入扩展、覆盖管理器、P 代码,等等。

本书旨在为初学者或具有不同程序设计背景的人介绍 MS C/C++ 的基本内容和最新特色,使之可以熟练地进行 C 和 C++ 程序设计。书中介绍了所有重要的 C 和 C++ 程序设计概念,包括面向过程的传统程序设计概念和新的面向对象程序设计概念,并给出了大量的程序例子。

本书适合用作高等学校 C 和 C++ 程序设计课程的教材和教学参考书,亦可供广大计算机用户作为学习和使用 MS C/C++ 7.0 的手册使用。

责任编辑 王建中

封面设计 王肖生

MSC/C++ 7.0 程序设计

辜健飞 辛迅雅 等编译

同济大学出版社出版

(上海四平路 1239 号)

新华书店上海发行所发行

江苏大丰印刷二厂印刷

开本:787×1092 1/16 印张:31.5 字数:800 千字

1995 年 4 月第 1 版 1995 年 4 月第 1 次印刷

印数:1—5000 定价:30.00 元

ISBN 7-5608-1508-1/TP·158

前 言

Microsoft C/C++ 7.0 编译程序软件包为开发 MS-DOS 及 Windows 应用程序提供了最全面的、最新的、产品级的开发环境。Microsoft 的这一最新版本吸收了许多新的、经过升级的特性,有了多方面重要的增强,如支持 AT & T C++ 2.1、预编译的头文件、自动嵌入扩展、覆盖管理器、P 代码,等等。

Microsoft C/C++ 编译程序软件包还提供构造 Windows 程序的工具,包括生成一个真正可靠实用的 Windows 应用程序所需的所有头文件、库、对话框及资源编辑器。它既可以用来进行传统的结构化程序设计,又可以用来进行较新颖的面向对象程序设计,是强有力的开发环境与 C 语言、C++ 语言和库的有机结合。

本书旨在为初学者或具有不同程序设计背景的人介绍 MS C/C++ 的基本内容和最新特色,使之可以熟练地进行 C 和 C++ 程序设计。书中介绍了 C 和 C++ 中所有重要的概念,解释了如何调试程序,如何编写无语法和逻辑编程错误的程序。读者还将了解过程式程序设计与面向对象程序设计的区别,以及如何开发简单的 OOP 程序。对于初学者来说,通过学习本书将为编写更复杂的程序打下坚实的基础,对于熟练的编程人员来说,可将本书用作一本参考书,同时也可了解新颖的 C++ 概念和程序设计思想。

本书给出了大量的 C 和 C++ 程序的例子,试图通过例子来具体而形象地介绍 MS C/C++ 的重要概要和特性。通过对例子的分析学习后,读者对书中的内容就更容易理解,并可以仿照例子很快地学会编程方法。

本书共分三篇,第一篇包括第一章到第十章,第二篇包括第十一章到第十四章,第三篇包括第十五章到第十七章。第一章到第十章介绍了基本的 C 和 C++ 程序设计概念,主要是面向过程的传统程序设计概念。

第十一章到第十四章完整地介绍了如何用 C++ 进行面向对象程序设计,给出了有关的术语、定义和完整的程序例子。

第十五章到第十七章介绍了如何开发 DOS 图形应用程序,如何利用各种重要的库函数。第十五章透彻地讨论了 C、C++ 和汇编语言的接口问题,包括如何综合 C、C++ 和汇编语言例程、传递参数等。

参加本书编写工作的主要人员有辜健飞、辛达雅、高青、周颖、张汝平、胡志东、左龙顺、薛仲清、张浩、周爽、刘东、袁兵、宁文英、叶晓东、李强、陈昕、王林、王长友等共 20 人。并由辜健飞、辛达雅负责全书的统编。由于水平、时间所限,不妥之处在所难免,欢迎广大读者批评指正。

需要本书中示例程序盘和希望知道原版 MS C/C++ 7.0 系统盘信息的读者,可与出版社软件部或本书的编者联系。

编 者

1994 年 5 月

目 录

第一篇 C 和 C++ 程序设计基础

第一章 C 和 C++ 基础	(3)
1.1 C 的历史	(3)
1.1.1 与其他语言的关系	(4)
1.1.2 C 的特点	(5)
1.1.3 C 的不足	(7)
1.1.4 程序设计训练	(8)
1.2 ANSI C 标准	(8)
1.3 C++ 和面向对象程序设计的演化	(9)
1.4 C++ 的历史	(10)
1.4.1 利用 C++ 对象来简化代码设计	(10)
1.4.2 对 C 的一些小的改进	(11)
1.4.3 对 C 的主要增强	(13)
1.5 C 程序的基本成分	(14)
1.5.1 程序的五种基本成分	(15)
1.5.2 第一个 C 程序	(15)
1.5.3 第一个 C++ 程序	(16)
1.5.4 第二个 C 程序	(16)
1.5.5 第二个 C++ 程序	(20)
1.5.6 文件	(21)
第二章 数据	(25)
2.1 标识符	(25)
2.2 关键字	(26)
2.3 标准 C 和 C++ 数据类型	(27)
2.3.1 字符	(28)
2.3.2 三种整数	(29)
2.3.3 unsigned 修饰符	(29)
2.3.4 浮点类型	(31)
2.3.5 枚举类型	(32)
2.4 存取修饰符	(33)
2.4.1 const 修饰符	(33)

2.4.2	#define 常量	(34)
2.4.3	volatile 修饰符	(35)
2.4.4	同时使用 const 和 volatile	(35)
2.5	pascal, cdecl, near, far, huge 修饰符	(35)
2.5.1	pascal	(36)
2.5.2	cdecl	(37)
2.5.3	near, far, huge	(37)
2.6	数据类型转换	(38)
2.7	存储类	(39)
2.7.1	位于外层的变量声明	(40)
2.7.2	位于内层的变量声明	(41)
2.7.3	变量作用域	(43)
2.7.4	外层的函数声明	(43)
2.8	操作符	(43)
2.8.1	位操作符	(43)
2.8.2	左移和右移	(45)
2.8.3	增一和减一	(45)
2.8.4	算术操作符	(46)
2.8.5	赋值操作符	(47)
2.8.6	复合赋值操作符	(47)
2.8.7	关系操作符和逻辑操作符	(49)
2.8.8	条件操作符	(52)
2.8.9	逗号操作符	(52)
2.9	操作符优先级的理解	(52)
2.10	标准的 C 和 C++ 库	(53)

第三章 控制

3.1	条件语句	(56)
3.1.1	if 语句	(56)
3.1.2	if-else 语句	(57)
3.1.3	嵌套的 if-else 语句	(59)
3.1.4	if-else-if 语句	(60)
3.1.5	? 条件语句	(61)
3.1.6	switch 语句	(62)
3.1.7	if-else-if 和 switch 语句的组合使用	(69)
3.2	循环语句	(71)
3.2.1	for 循环	(71)
3.2.2	while 循环	(75)
3.2.3	do-while 循环	(78)
3.2.4	break 语句	(80)
3.2.5	continue 语句	(81)

3.2.6	break 和 continue 组合使用	(82)
3.2.7	exit()语句	(84)
3.2.8	atexit()语句	(87)
第四章	函数	(90)
4.1	函数原型和风格	(90)
4.1.1	原型	(90)
4.1.2	按值调用和按引用调用	(93)
4.1.3	存储类	(94)
4.1.4	作用域	(95)
4.1.5	递归	(95)
4.2	函数参数	(96)
4.2.1	形式参数和实在参数	(96)
4.2.2	void 参数	(97)
4.2.3	字符参数	(97)
4.2.4	整数参数	(98)
4.2.5	浮点数参数	(99)
4.2.6	双精度浮点数参数	(101)
4.2.7	数组参数	(102)
4.3	函数类型	(104)
4.3.1	void 函数类型	(104)
4.3.2	char 函数类型	(105)
4.3.3	int 函数类型	(106)
4.3.4	long 函数类型	(107)
4.3.5	float 函数类型	(108)
4.3.6	double 函数类型	(109)
4.4	main()函数的参数	(110)
4.4.1	串	(110)
4.4.2	整数	(111)
4.4.3	浮点数	(113)
4.5	一些重要的 C++ 特性	(114)
4.5.1	inline	(114)
4.5.2	重载	(115)
4.5.3	省略号(...)	(116)
4.6	有关作用域的问题	(118)
4.6.1	在一个 C 程序中未定义的符号	(118)
4.6.2	具有文件作用域的变量的使用	(119)
4.6.3	局部变量覆盖全局变量	(120)
4.6.4	C++ 中的一个作用域问题	(121)
4.6.5	C++ 域分辨符	(122)

第五章 数组	(124)
5.1 什么是数组?	(124)
5.2 数组和 C	(124)
5.3 数组声明	(124)
5.4 数组的初始化	(125)
5.4.1 缺省的初始化	(126)
5.4.2 显式的初始化	(126)
5.4.3 数目未定的初始化值	(127)
5.5 数组元素的访问	(127)
5.6 数组维数的计算(sizeof())	(129)
5.7 越界的数组下标	(132)
5.8 串的输入和输出	(132)
5.9 多维数组	(134)
5.10 作为函数参数的数组	(138)
5.10.1 将数组传给 C 函数	(138)
5.10.2 将数组传递给 C++ 函数	(139)
5.11 串函数和字符数组	(145)
5.11.1 gets(), puts(), fgets(), fputs(), sprintf()	(145)
5.11.2 strcpy(), strcat(), strcmp(), strlen()	(147)
第六章 指针	(151)
6.1 指针变量的定义	(151)
6.1.1 指针变量声明	(152)
6.1.2 使用指针变量的简单语句	(153)
6.1.3 指针变量的初始化	(156)
6.1.4 取地址操作符的不当使用	(157)
6.1.5 数组指针	(158)
6.1.6 指向指针的指针	(158)
6.1.7 字符串指针	(160)
6.1.8 指针运算	(161)
6.1.9 指针运算和数组	(163)
6.1.10 操作符++和--使用时的的问题	(165)
6.1.11 指针的比较	(165)
6.1.12 指针的可移植性	(165)
6.1.13 将 sizeof 作用于指针	(166)
6.2 函数指针	(167)
6.3 动态存储	(170)
6.4 指针与数组	(175)

6.4.1	串(char 数组)	(175)
6.4.2	指针数组	(176)
6.4.3	有关指向指针的指针的进一步讨论	(178)
6.4.4	串指针数组	(183)
6.5	C++ 引用类型	(185)
6.5.1	返回地址的函数	(186)
6.5.2	使用 CodeView	(187)
6.5.3	何时使用引用类型?	(187)
第七章	C 中的 I/O	(188)
7.1	流函数	(190)
7.1.1	流的打开	(191)
7.1.2	输入和输出重定向	(191)
7.1.3	改变流缓冲区	(192)
7.1.4	流的关闭	(194)
7.2	C 中的低级输入和输出	(194)
7.3	字符的输入和输出	(194)
7.3.1	getc(),putc(),fgetc()和 fputc()的使用	(195)
7.3.2	getchar(), putchar(), fgetchar()和 fputchar()的使用	(195)
7.3.3	getch()和 putch()的使用	(195)
7.4	串输入和输出	(196)
7.5	整数的输入和输出	(197)
7.6	格式化输出	(200)
7.7	fseek(), ftell(), rewind()的使用	(203)
7.8	格式化输入	(207)
第八章	C++ 中 I/O 简介	(210)
8.1	用 C++ 简化 I/O	(210)
8.1.1	cin,cout 和 cerr	(210)
8.1.2	>>和<<操作符	(210)
8.2	从 stream.h 到 iostream.h	(217)
第九章	结构、联合及其他	(230)
9.1	C 和 C++ 中的结构	(230)
9.1.1	C 和 C++ 结构:语法和规则	(230)
9.1.2	C++ 结构:语法和规则扩充	(232)
9.1.3	结构成员的存取	(232)
9.1.4	构造一个简单的结构	(233)
9.1.5	将结构传递给函数	(234)

9.1.6	构造结构数组	(236)
9.1.7	结构指针的使用	(239)
9.1.8	将结构数组传给函数	(242)
9.1.9	结构在 C++ 中的使用	(244)
9.1.10	关于结构的其他操作	(248)
9.2	联合	(249)
9.2.1	联合:语法和规则	(249)
9.2.2	构造一个简单的联合	(250)
9.3	其他特性	(252)
9.3.1	typedef 的使用	(252)
9.3.2	enum 的使用	(253)

第十章 C 和 C++ 高级程序设计课题

10.1	类型相容性	(256)
10.1.1	类型相容性的 ANSI C 定义	(256)
10.1.2	什么是相同的类型?	(256)
10.1.3	枚举类型	(257)
10.1.4	数组类型	(257)
10.1.5	函数类型	(257)
10.1.6	结构和联合类型	(258)
10.1.7	指针类型	(258)
10.1.8	多源文件相容性	(258)
10.2	宏	(258)
10.2.1	宏的定义	(259)
10.2.2	宏与参数	(259)
10.2.3	宏扩展中可能出现的问题	(260)
10.2.4	宏的创建和使用	(261)
10.2.5	编译程序中提供的宏	(262)
10.3	高级预处理器指令	(262)
10.3.1	#ifdef 和 #endif 指令	(263)
10.3.2	#undef 指令	(263)
10.3.3	#ifndef 指令	(263)
10.3.4	#if 指令	(264)
10.3.5	#else 指令	(264)
10.3.6	#elif 指令	(264)
10.3.7	#line 指令	(265)
10.3.8	#error 指令	(265)
10.3.9	#pragma 指令	(265)
10.4	条件编译	(266)

10.5	高级预处理器操作符	(267)
10.5.1	#串长度操作符	(267)
10.5.2	##并置操作符	(267)
10.5.3	#@字符化操作符	(268)
10.6	头文件的正确使用	(268)
10.7	使头文件更加有效	(269)
10.8	预编译的头文件	(270)
10.8.1	创建预编译的头文件	(270)
10.8.2	预编译的头文件与PWB一起使用	(270)
10.9	limits.h和float.h	(270)
10.10	错误处理:perror()	(271)
10.11	内存模式	(272)
10.11.1	微型模式	(272)
10.11.2	小模式	(272)
10.11.3	中型模式	(272)
10.11.4	紧凑模式	(273)
10.11.5	大型模式	(273)
10.11.6	巨型模式	(273)
10.12	动态内存分配:链表	(273)
10.12.1	使用链表时的一些考虑	(274)
10.12.2	一个简单的链表	(275)

第二篇 C++面向对象程序设计基础

第十一章	面向对象程序设计引论	(281)
11.1	传统的结构程序设计	(281)
11.2	面向对象程序设计	(281)
11.3	C++与面向对象程序设计	(282)
11.4	面向对象程序设计的术语	(282)
11.4.1	封装性	(282)
11.4.2	类层次	(283)
11.4.3	继承性	(283)
11.4.4	多态性	(283)
11.4.5	虚函数	(284)
11.5	C++类初探	(284)
11.5.1	作为原始类的结构	(284)
11.5.2	C++类的语法和规则	(289)
11.5.3	一个简单的C++类	(290)

第十二章 C++类	(293)
12.1 类的另一些特性	(293)
12.1.1 一个简单的类	(293)
12.1.2 嵌套类	(293)
12.1.3 构造函数和析构函数	(297)
12.1.4 类成员函数的重载	(304)
12.1.5 利用友函数以访问私有的类变量	(308)
12.1.6 this 指针的使用	(311)
12.2 操作符重载	(312)
12.2.1 操作符重载和函数调用	(312)
12.2.2 重载的语法	(312)
12.3 派生类	(315)
12.3.1 派生类的语法	(315)
12.3.2 派生类的创建	(315)
第十三章 C++中的输入输出	(320)
13.1 enum 类型	(320)
13.2 引用变量	(321)
13.3 缺省参数	(323)
13.4 memset()	(324)
13.5 格式化输出	(325)
13.6 C/C++ I/O 选项	(329)
13.7 iostream 类表	(329)
13.7.1 输入流类	(333)
13.7.2 输出流类	(334)
13.7.3 带缓冲的流类	(335)
13.7.4 串流类	(337)
13.8 二进制文件	(339)
13.9 用 extern "C" 组合 C 和 C++ 代码	(341)
13.10 编写自己的操纵符	(343)
13.10.1 不带参数的操纵符	(343)
13.10.2 带一个参数的操纵符	(344)
13.10.3 带多参数的操纵符	(345)
第十四章 面向对象的环境	(348)
14.1 一个以 C++ 编写的面向对象栈	(348)
14.2 以 C++ 编写的一个面向对象的链表	(351)
14.2.1 创建一个父类	(351)

14.2.2	一个派生子类	(352)
14.2.3	友元类的使用	(353)
14.2.4	分析完整的程序	(356)
14.2.5	链表输出	(366)

第三篇 C++高级程序设计技术

第十五章	DOS 下的图形程序设计	(371)
-------------	---------------------------	--------------

15.1	正文和图形模式	(371)
15.2	使用简单的图形函数	(374)
15.2.1	矩形的绘制	(375)
15.2.2	确定屏幕的分辨率	(376)
15.2.3	有关图形环境的信息	(378)
15.2.4	图形细节	(379)
15.2.5	图形原语的使用	(386)
15.3	图形模式下字型的使用	(391)
15.3.1	基本的字型属性	(393)
15.3.2	以不同的大小显示不同的字型	(393)
15.3.3	字型旋转	(397)
15.4	包含图形原语的科学和商业应用程序	(401)
15.4.1	正弦曲线的绘制	(401)
15.4.2	一个富里叶级数	(405)
15.4.3	用图形原语绘制一个饼图	(410)
15.5	展示图形	(416)
15.5.1	一个交互的饼图	(420)
15.5.2	一个交互的直方图	(424)
15.5.3	一个交互的直线图	(427)
15.5.4	一个分散图	(430)
15.6	特殊的图形效果	(433)
15.6.1	使用四个视区	(433)
15.6.2	调整视区的大小	(435)
15.6.3	简单的动画技术	(438)

第十六章	重要的 C 和 C++ 库	(443)
-------------	----------------------------	--------------

16.1	Microsoft C 和 C++ 头文件	(443)
16.2	标准库函数(stdlib.h)	(444)
16.2.1	数据转换	(445)
16.2.2	查找和排序	(447)
16.2.3	其他操作	(449)

16.3	字符函数 ctype.h)	(451)
16.3.1	检查是否是字母数字、字母和 ASCII 值	(452)
16.3.2	检查是否是控制、白空和标点符号字符	(453)
16.3.3	转换为 ASCII,小写和大写字符	(455)
16.4	串函数(string.h)	(456)
16.4.1	内存函数	(457)
16.4.2	串函数	(459)
16.5	数学函数(math.h)	(462)
16.6	时间函数(time.h)	(465)
16.7	与系统有关的函数	(470)
16.7.1	bios.h 头文件	(470)
16.7.2	dos.h 头文件	(472)

第十七章 Microsoft C/C++与汇编语言的混合编程 (478)

17.1	嵌入式汇编语代码	(478)
17.2	创建 C/C++和汇编语言模块	(480)
17.2.1	函数参数的传递	(480)
17.2.2	不同数据类型参数的传递	(481)
17.2.3	将数组从 C 中传递给汇编语言	(486)

第一篇

C 和 C++ 程序设计基础

第一章 C 和 C++ 基础

从本章开始，我们要介绍 C 和 C++ 语言的起源、语法和用法。了解一下 C 语言的历史是有益的，它可以揭示 C 的成功的设计哲学，有助于理解为什么在未来的数年内，C 和 C++ 会是人们所看好的语言。

1.1 C 的历史

要介绍 C 的历史，就要先来讨论一下 UNIX 操作系统，因为该系统和在它上面运行的大多数程序都是用 C 写的。然而，这并不表示 C 只局限于用在 UNIX 或其他操作系统、其他机器上。UNIX/C 构成了一种程序开发环境，人们将 C 看作是一种系统程序设计语言 (system programming language)，因为它非常适合于书写编译程序和操作系统等系统程序。此外，C 还可以用来书写应用于各种领域中的主要程序。

UNIX 最初是 1969 年在贝尔实验中开发出来的。该实验室位于美国新泽西州的 Murray Hill。UNIX 基于的硬件系统是 DEC PDP-7，按现在的眼光来看，它应该算是“小”机器了。当时，UNIX 完全是用 PDP-7 汇编语言写的。根据设计目标，这个操作系统将是“程序员友好的” (programmer friendly)，可提供实用的开发工具，简洁的命令，以及一个相当开放的环境。在开发出 UNIX 后不久，Ken Thompson 就实现了一种新语言的编译程序，这种新语言就是 B。

讲到这儿，我们可以来回顾一下 B 语言的起源和历史，该语言是 C 语言的直接前驱。具体发展过程如下：

Algol60 1960 由一个国际委员会设计。

CPL (Combined Programming Language, 组合程序设计语言) 1963 年由剑桥大学和伦敦大学开发。

BCPL (Basic Combined Programming Language, 基本组合程序设计语言) 1967 年在剑桥大学由 Martin Richards 开发。

B 1970 年由 Ken Thompson 在贝尔实验室开发。

C 1972 年由 Dennis Ritchie 在贝尔实验室开发。

此后，在 1983 年，为了设计 ANSI C (一种标准化的 C 语言)，形成了美国国家标准协会 (the American National Standards Institute, 简称 ANSI)。

Algol60 语言在 FORTRAN 之后的几年内出现的。这一新的语言要更复杂些，对它以后的程序设计语言的设计产生了很大的影响。它的作者将注意力主要放在了高级结构化语言的语法、模块结构及其他特性的规范性上。不幸的是，Algol60 始终没有能流行起来。人们认为，这主要是因为它太抽象、太通用了。

CPL 的发明者意在把 Algol60 的宏伟的目标变得更现实一些，即与具体的计算机结合起来。然而，与 Algol60 一样，CPL 也难以学习和难以实现。这就导致了它最终的衰落。BCPL