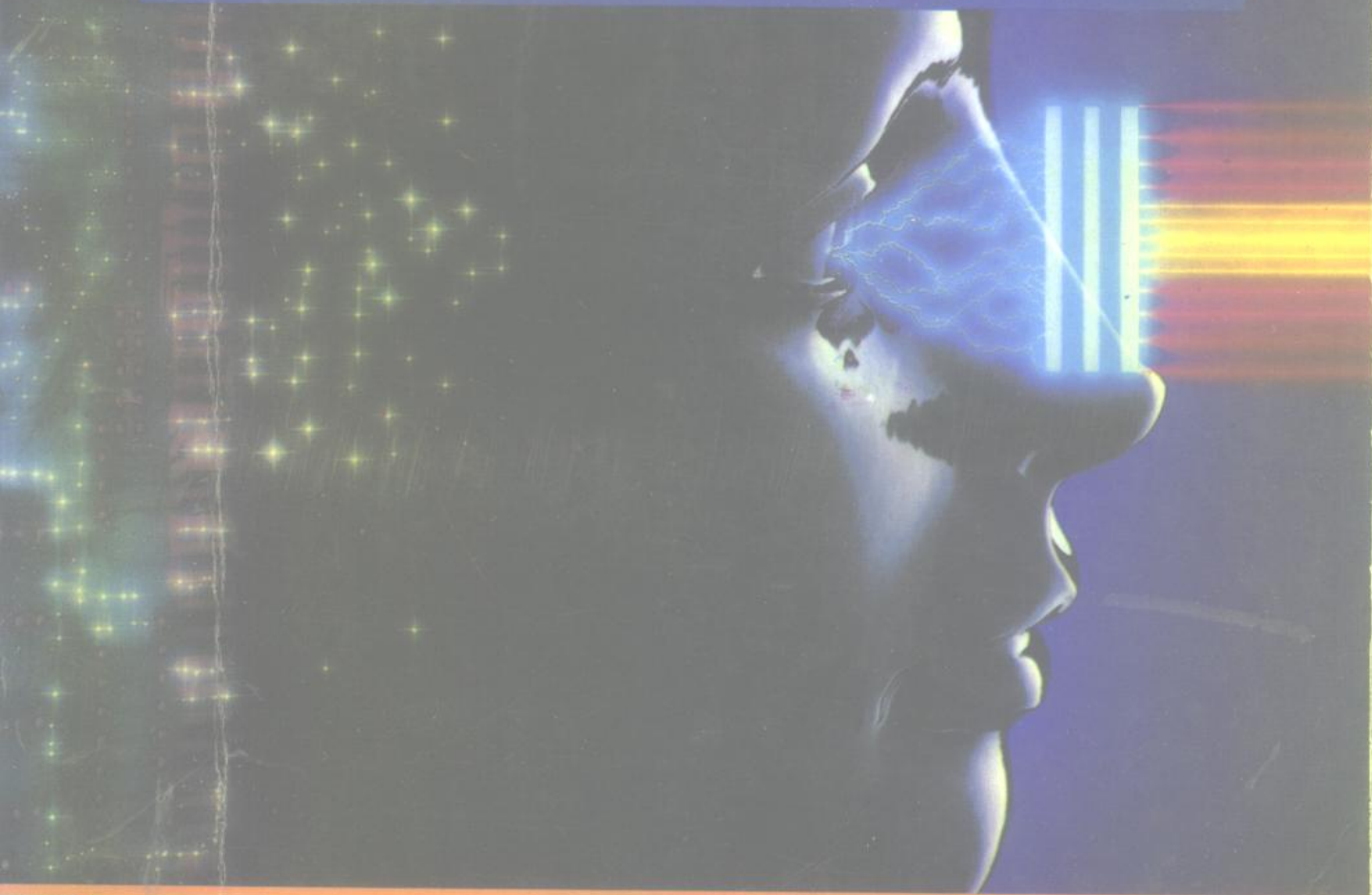


VISUAL 1.5 指南

OOP WITH C++



成都科技大学出版社



205594

Visual C⁺⁺ 1.5 程序设计指南

张天庆 谭胜勇 黄武 编写

成都科技大学出版社

(川)新登字015号

责任编辑:张成生

封面设计:王松

JS205/30 57

Visual C++1.5 程序设计指南

张青天 等编写

成都科技大学出版社出版发行

峨眉电影制片厂印刷厂 印刷

787×1092毫米 1/16 印张:125

1995年3月第一版 1995年3月第1次印刷

印数0001 3000 字数:2000千字

ISBN 7 5616 3010 7/TP·123

全套定价:128.00元

前 言

本书是理解 Microsoft Visual C++ 及学习编程的最完整的工具书。C++ 程序设计语言已经成为软件开发商的标准开发语言。许多人都看到 C++ 正成为程序员的标准编程语言,使用它能在个人计算机上开发出效率高、功能强的软件。

大多数讲述 C++ 编程的书一开始都只讲 C 语言,然后把 C++ 当作 C 的扩展来讲述。事实上,C++ 是一种独立的语言。要成为 C++ 专业程序员并不需要学习 C 语言。本书从一开始就讲述如何用 C++ 编程,并介绍面向对象程序设计的概念。这种尝试,只讲 C++ 而不讲述 C 应该更有助于掌握 C++。

本书从一开始就着重于对象和 C++ 的特征,而不管 C 语言的特征。

读者将会发现本书是最容易理解的一本 C++ 教材,通过它可以迅速掌握 C++。

本书尽管不是初学者的工具书,但初学者也能受益匪浅。通过本书,初学者可以迅速编写出 Visual C++ 应用程序。

本书的目的是为了满足想学习 C++ 程序设计的读者。读者只要以前有任何语言编程经验,例如 BASIC、FORTRAN、Visua、Basic,最好是 C,将有助于本书的学习,但并不是必不可少。本书读者应能使用 Microsoft Windows,不需要专业水平,只要会用鼠标点取对话框和菜单即可。

大多数讲述 C++ 的书开始总是讲 C 语言,然后把 C++ 讲述为加在 C 之上的不同特征。这种办法对于大多数学习 C++ 的程序员已熟悉 C 编程的情况是较为有利的。但目前,新程序员从一开始就学习 C++,完全没有必要浪费时间去学 C,尤其是 C 的输入输出例程及库功能等。这些功能在 C++ 中用面向对象的方法实现了,同 C 中所用的方法有着本质的不同。

例如,假定你曾用 C 编过程序,你一定会对本书从未用 printf() 和 scanf() 进行输入输出感到奇怪。取而代之的是 cout 和 cin 流操作,相对于传统 C 的库函数而言,它们显得更为简单明白。

Microsoft 提供了两种版本的 Visual C++:标准版和专业版。学习本书可用任一版本。两者都提供了丰富的环境,不仅有编程语言,还有 Visual Workbench,即 VWB。Visual WorkBench 提供了多种功能,包括输入和编辑 C++ 源程序的程序编辑器,内部编译和链接程序,还有一个特殊的 debugger 帮助用户查错并改正。

Visual C++ 是 Microsoft Windows 下的编程环境。尽管 Windows 已成为世界上最流行的操作系统,但基于它学习 C++ 编程并不容易。好在读者不必同时学习 C、C++ 和 Windows 程序设计。本书所写程序既可在 MS-DOS 下运行,也可以在 Windows 下运行。

第一章“C++ 程序设计基础”介绍怎样用 Visual Workbench 输入和编辑程序。读者能编写出自己的第一个 C++ 程序并学到若干 C++ 编程语句及变量的基本概念。读者还能学到怎样向屏幕输出信息和从键盘输入信息。读者学完第一章后就会编写简单的 C++ 程序以及在 Visual Workbench 下输入、编译和执行程序。

计算机程序也被称作包含数据的程序指令(或语句)集。第二章“C++ 数据类型和表达

式”讲述基本的用于存储和操作的不同数据类型,如数字型、字符型和文本型等。同时,也讲述了在 C++ 语言中要用到的各种数学表达式(等式)。

第三章“C++ 高级数据类型”通过数组、结构和联合讨论了 C++ 高级数据存储技术。本章还引入了对象,它是面向对象程序设计的核心。

从第四章“选择和循环语句”起,开始深入讨论 C++ 程序设计语句。探讨在程序中用 if 语句实现条件分支,同时还要讨论 switch 语句用于多种可能流向的选择控制。本章也讲述了 C++ 的三种循环语句,用于在程序中重复执行部分代码。

第五章“指针”讨论 C++ 的指针类型。指针,正如它的名字指出,是内存中存储的指向另一地址单元的指针。指针的概念比较难掌握,所以整个第五章都讲述这一数据类型。使用指针是提高 C++ 编程能力的必要手段,因此理解指针的精髓并灵活用于程序设计是非常重要的。

用 C++ 编写的程序,尤其是那些不是太大的程序,都被分为较小的称为函数的多个部分。大多数 C++ 程序都包含多个函数,而并非一个大的 main() 函数。把代码分在不同的函数中可以在其它程序中重用它们。

第六章“函数和宏”讨论如何设计和实现函数。随着程序的变大,需要把一个较大的程序分解成多个源程序文件,以便于编辑和编译。C++ 提供了特殊的用于管理工程文件的工具,工程文件包含多个源文件和目标模块。

第七章“工程”,讲述如何把程序分成多个文件,以及如何用 Visual Workbench 工程功能管理编译和链接过程,还讲述了如何用 Microsoft Visual C++ (MSVC) 提供的外部工具创建独立于 Workbench 的工程。

第八章“类”扩展了对象的概念,讲述如何用类说明定义对象,将讨论构造和析构函数、友元函数,以及把对象作为参数传递给函数等。

第九章“继承的能力”继续讨论类以及面向对象程序设计,如怎样从已定义类中派生出新类,这样就可以继承(inherit)或者说是重用代码,而不必重写代码。

第十章“虚函数和多态性”扩充了继承的概念。本章讲述面向对象程序设计最显著的特征:多态性(Polymorphism),讨论怎样用继承和虚函数实现多态性。

本书一开始就提到通过 C++ 流来处理 C++ 输入和输出操作。

第十一章“C++ 输入输出”讨论各种格式化输出技术,如屏幕字符定位和选择色彩等,还讲述了如何用磁盘永久存储数据。

第十二章“调试 Visual C++ 程序”介绍初学者可能碰到的各种错误,讲述如何用调试程序查找程序错误。本章还介绍怎样准备程序调试,怎样用 debugger 跟踪及改错等方面内容。

第十三章“Quickwin 图形程序设计”介绍 Quickwin 的作图功能以及怎样使用和联接图形库。Visual C++ 为基于 Quickwin 的应用程序提供了一种易用的扩展二维计算机图形支持库,它包含基本的作图例程(直线、矩形、圆、椭圆、多边形等等),多种文本字体,文本输出例程,以及用于创建各种图形的特殊功能例程。

如能在别人的基础上工作并继承预先设计的类功能,面向对象程序设计就能显示出强大威力了。Visual 提供了各种通用类,包含在一个类库集中,包括处理存取数据的多个例程。考虑到所有程序依赖一种或另一种数据,类库中提供的类可以减少创建应用程序时所需的时间。

第十四章“MFC 通用类和集合类”介绍了类库集和许多类集。

第十五章“Visual C++ 用于 Windows 程序设计”介绍怎样真正用 Microsoft 基本库类编制 Windows 程序。MFC 简化了开发 Windows 应用程序时其它方法的复杂程序接口。

学习程序设计的唯一途径就是编写程序。要用好本书,读者应利用每一章后面的复习题,确信自己已弄清了本章内容。每一章后面的练习有助于读者提高 C++ 编程技巧。读者应该做这些练习,或补充自己的习题,尽可能多的练习编程,尽可能多在 Visual C++ 下实习。只要有信心并掌握一定技巧,就能够编写出能解决很多实际问题的软件。读者所写的每一个新程序都有助于提高自己的编程技巧。

本书常常出现竖写的省略号,表示程序中的若干行。例如:

```
Void main(void)
{
    .
    .
    .
}
```

省略号表示这里有不指定的行或列若干行程序。省略号本身并非 C++ 的语句。

还可能看到如 Ctrl+F3 之类的组合键。它表示按下并保持 Ctrl 键,然后按 F3 键。本书第二章讲述的几种 Visual C++ 命令就使用了这种组合键。可能遇到如 Alt+F, X 之类的情况,此时,按 Alt+F 再按 X。再按第二个字母时不必按住 Alt 键(本例中是 X)。

本书中的示例程序是精确的。作者鼓励读者使用本书中提供的示例程序来构造自己的应用程序。但是,在程序中使用这些例程时,读者必须测试它们(需要单测和合测)。因为作者提供的这些源代码不一定能在应用程序中执行得很好。

本书中提供的示例程序在左边印有行号,便于讲解时查找指定行号。但是当读者输入这些程序时,不要输入这些行号。

本书由于时间紧迫,任务重,难免有错误及不妥之处,敬请读者谅解并批评指正。

编 者

目 录

第一章 C++程序设计基础

第一节 使用 Microsoft Windows	(1)
§ 1.1.1 定义通用窗口	(1)
§ 1.1.2 定义对话框	(1)
第二节 安装 Visual C++	(4)
第三节 使用 Visual C++	(5)
§ 1.3.1 使用 Visual Workbench	(5)
§ 1.3.2 Visual Workbench 工具条	(6)
§ 1.3.3 Visual Workbench 帮助系统	(8)
§ 1.3.4 退出 Visual Workbench	(8)
第四节 使用 Visual Workbench 编辑器	(9)
§ 1.4.1 编辑器操作	(9)
§ 1.4.2 选择文本及执行文本块操作	(11)
§ 1.4.3 打开已存文件	(12)
第五节 学写第一个 C++ 程序	(12)
§ 1.5.1 创建工程文件	(14)
§ 1.5.2 编译用户程序	(15)
§ 1.5.3 运行用户程序	(16)
§ 1.5.4 QuickWin 库	(16)
第六节 一个 C++ 示例程序	(16)
§ 1.6.1 cout 和 cin 流	(18)
§ 1.6.2 编译运行用户程序	(18)
§ 1.6.3 选择内存模式	(21)
§ 1.6.4 编译程序、链接程序和语法错误	(21)
第七节 C++ 程序的构成	(21)
§ 1.7.1 C++ 的输入输出	(22)
§ 1.7.2 C++ 函数	(22)
§ 1.7.3 变量	(22)
§ 1.7.4 符号常量	(23)
§ 1.7.5 表达式	(25)
§ 1.7.6 语句	(25)
§ 1.7.7 函数	(29)

第八节 小结	(30)
第二章 C++数据类型和表达式	
第一节 标识符名	(32)
第二节 基本数据类型	(33)
§ 2.2.1 整数(Integer)	(33)
§ 2.2.2 长整数(Long Integer)	(34)
§ 2.2.3 字符(character)	(35)
§ 2.2.4 把 char 类型用于算术运算	(36)
§ 2.2.5 浮点类型	(37)
第三节 变量的生存期	(37)
第四节 定义符号常量	(41)
§ 2.4.1 使用 const	(41)
§ 2.4.2 使用 #define	(42)
第五节 表达式和 C++ 运算符	(44)
§ 2.5.1 运算符	(44)
§ 2.5.2 赋值运算符	(45)
§ 2.5.3 关系运算符	(47)
§ 2.5.4 逻辑运算符	(49)
§ 2.5.5 位操作运算符	(49)
§ 2.5.6 其它赋值运算符	(53)
第六节 运算符优先级	(54)
第七节 数据类型转换	(55)
第八节 枚举类型及常量	(56)
§ 2.8.1 用 enum 产生枚举类型	(57)
§ 2.8.2 把枚举类型用作标志变量	(57)
第九节 小结	(58)
第三章 C++高级数据类型	
第一节 C++组合数据类型	(62)
第二节 数组	(62)
§ 3.2.1 说明数组	(62)
§ 3.2.2 初始化数组	(64)
§ 3.2.3 字符数组和文本串	(64)
§ 3.2.4 使用数组	(66)
§ 3.2.5 使用 char 型数组的示例	(67)
§ 3.2.6 多维数组	(68)
第三节 结构	(72)

§ 3.3.1	结构赋值	(74)
§ 3.3.2	初始化结构说明	(74)
§ 3.3.3	结构数组	(74)
§ 3.3.4	结构内的结构	(77)
§ 3.3.5	位域	(78)
第四节	typedef 的用法	(80)
第五节	联合	(80)
第六节	对象的引入	(82)
第七节	小结	(87)
第四章	选择和循环语句	
第一节	判断和选择语句	(89)
§ 4.1.1	if 语句	(89)
§ 4.1.2	if-else 语句	(91)
§ 4.1.3	if-else-if 语句	(91)
§ 4.1.4	switch 语句	(92)
§ 4.1.5	break 语句	(94)
第二节	循环语句	(94)
§ 4.2.1	for 循环	(94)
§ 4.2.2	break 语句作为循环出口	(99)
§ 4.2.3	continue 语句	(100)
§ 4.2.4	while 循环	(100)
§ 4.2.5	do-while 循环	(102)
第三节	优化循环	(103)
第四节	使用 goto 和语句标号	(104)
第五节	小结	(105)
第五章	指针	
第一节	理解指针	(109)
第二节	使用指针	(110)
§ 5.2.1	使用 NULL 指针	(110)
§ 5.2.2	使用指向数组的指针	(111)
§ 5.2.3	对指针执行算术运算	(111)
§ 5.2.4	简洁用法	(112)
§ 5.2.5	用指针加速数组取存	(113)
§ 5.2.6	指针算术运算	(113)
§ 5.2.7	设置一个指针等于另一个指针	(114)
第三节	动态内存分配	(115)

§ 5.3.1	堆	(115)
§ 5.3.2	new 操作符	(115)
§ 5.3.3	delete 操作符	(116)
§ 5.3.4	使用 new 和 delete 的示例	(117)
§ 5.3.5	new 用于多维数组	(118)
§ 5.3.6	内存不足处理	(118)
§ 5.3.7	初始化动态分配	119(C)
§ 5.3.8	使用指针可能出现的问题	(119)
§ 5.3.9	使用指针访问数组	(120)
§ 5.3.10	使用指向数组的指针示例程序	(121)
§ 5.3.11	指针数组	(123)
第四节	指针和结构	(126)
§ 5.4.1	把指针放到结构定义中	(126)
§ 5.4.2	自引用结构	(127)
§ 5.4.3	实现链表	(131)
第五节	Near、Far 和 Huge 指针	(134)
第六节	小节	(136)
第六章 函数和宏		
第一节	使用库函数	(139)
§ 6.1.1	使用指定库函数	(140)
§ 6.1.2	使用串函数	(143)
第二节	自定义函数	(147)
§ 6.2.1	函数结果类型	(148)
§ 6.2.2	函数参数	(149)
第三节	在函数中定义变量	(161)
§ 6.3.1	自动变量	(161)
§ 6.3.2	静态变量	(161)
§ 6.3.3	寄存器变量	(162)
第四节	把函数定义和实现分开	(162)
第五节	为什么要使用函数原型	(163)
第六节	组合一个示例程序	(165)
第七节	函数重载	(168)
第八节	函数模板	(169)
第九节	类函数宏	(170)
§ 6.9.1	# undef 指示符	(173)
§ 6.9.2	使用宏的问题	(173)

第十节 小结	(174)
第七章 工程	
第一节 把程序分成多个源文件	(177)
§ 7.1.1 分解文件容易实现	(177)
§ 7.1.2 创建源程序模块	(178)
§ 7.1.3 创建头文件	(179)
§ 7.1.4 创建主文件	(180)
第二节 使用 Visual Workbench 创建工程	(182)
§ 7.2.1 向工程中加入文件	(183)
§ 7.2.2 使用 Rebuild All 菜单选项	(185)
§ 7.2.3 使用 Build 菜单选项	(185)
§ 7.2.4 使用 Compile 菜单选项	(185)
§ 7.2.5 使用工具条按钮	(185)
§ 7.2.6 处理依赖性问题	(185)
§ 7.2.7 使用 DOS 命令行	(186)
第三节 条件编译指示符	(186)
§ 7.3.1 定义标志符号	(187)
§ 7.3.2 使用 #ifdef 和 #ifndef	(187)
§ 7.3.3 使用条件编译以避免重复定义符号	(188)
§ 7.3.4 使用 #if, #elif, #else 和 #endif	(188)
§ 7.3.5 使用 defined() 操作符	(189)
第四节 变量的作用域	(189)
§ 7.4.1 函数作用域	(190)
§ 7.4.2 块作用域	(190)
§ 7.4.3 文件作用域	(190)
§ 7.4.4 函数原型作用域	(190)
第五节 访问其它源文件中的变量和函数	(190)
§ 7.5.1 内部链接	(191)
§ 7.5.2 外部链接	(191)
§ 7.5.3 使用 extern 关键字	(191)
第六节 内存模式	(192)
§ 7.6.1 80x86 内存结构	(193)
§ 7.6.2 80x86CPU 寄存器	(193)
§ 7.6.3 选择内存模式	(196)
第七节 小结	(198)
第八章 类	

第一节	用 struct 来创建对象	(200)
第二节	C++ 关键字	(203)
第三节	外联和内联函数	(204)
第四节	构造函数	(205)
§ 8.4.1	使用构造函数	(206)
§ 8.4.2	具有多构造函数的类	(206)
§ 8.4.3	拷贝构造函数	(207)
§ 8.4.4	缺省构造函数	(207)
第五节	析构函数	(207)
第六节	指向对象的指针	(208)
§ 8.6.1	使用指针的另一例	(210)
§ 8.6.2	释放对象指针	(210)
第七节	对象数组	(210)
第八节	将对象用作函数参数	(211)
第九节	类的静态数据成员	(211)
第十节	实现 List 类	(214)
第十一节	* this 指针	(215)
第十二节	友元函数	(217)
第十三节	重载操作符函数	(219)
§ 8.13.1	重载单目和二目操作符	(224)
§ 8.13.2	将友元函数用于重载操作符	(225)
§ 8.13.3	用重载的操作符实现定点数运算	(225)
第十四节	小结	(228)
第九章 继承的功能		
第一节	继承如何工作	(230)
§ 9.1.1	关键字 protected	(233)
§ 9.1.2	派生类和访问描述符	(234)
§ 9.1.3	调用祖先类的构造函数	(235)
§ 9.1.4	sprintf() 函数	(235)
§ 9.1.5	继承的数据成员	(235)
§ 9.1.6	类与类型相容	(239)
第二节	多继承如何工作	(239)
第三节	小结	(242)
第十章 虚函数和多态性		
第一节	重载函数的问题	(244)
第二节	为什么需要虚函数	(246)

第三节	虚函数如何工作	(249)
第四节	static 与 Virtual:选择哪一个?	(250)
第五节	多态性	(251)
§ 10.5.1	设计一个支持多态性的类	(251)
§ 10.5.2	用虚函数实现多态性	(254)
§ 10.5.3	在一个图形应用程序中使用多态函数	(259)
§ 10.5.4	用多态函数实现一个列表框	(259)
§ 10.5.5	创建纯虚函数	(261)
10.5.6	从抽象类中派生出一个子类	(262)
第六节	对象库	(263)
第七节	设计面向对象的程序	(263)
第八节	小结	(264)
第十一章 C++ 输入和输出		
第一节	屏幕输出	(266)
第二节	键盘输入	(267)
§ 11.2.1	文本和字符输入	(268)
第三节	格式化输出	(269)
§ 11.3.1	操作程序(/操作符)的使用	(269)
§ 11.3.2	设置格式化流标志	(270)
§ 11.3.3	格式化输出函数	(271)
§ 11.3.4	流错误处理	(272)
§ 11.3.5	检查流错误	(272)
§ 11.3.6	检查文件结束状态	(272)
§ 11.3.7	检查特殊错误状态	(273)
§ 11.3.8	清错误状态	(273)
§ 11.3.9	用户定义类输入和输出	(273)
§ 11.3.10	串流格式化	(275)
§ 11.3.11	ostrstream 的使用	(276)
§ 11.3.12	istrstream 的使用	(277)
第四节	磁盘文件输入和输出	(278)
§ 11.4.1	对文件写文本输出	(278)
§ 11.4.2	打开文件的一个可选方法	(279)
§ 11.4.3	从文件读入文本	(280)
§ 11.4.4	文件拷贝	(282)
§ 11.4.5	二进制数据文件的使用	(282)
§ 11.4.6	随机访问数据文件	(284)

§ 11.4.7 打印.....	(286)
第五节 小结.....	(286)
第十二章 调试 Visual C++ 程序	
第一节 理解上面提到过的三种类型的程序错误.....	(288)
§ 12.1.1 语法错误.....	(288)
§ 12.1.2 动态错误.....	(289)
§ 12.1.3 测试函数的返回值.....	(289)
§ 12.1.4 异常处理.....	(290)
§ 12.1.5 逻辑错误.....	(291)
第二节 在你的用户使用之前发现错误.....	(291)
§ 12.2.1 使用桌面检查和代码预视技术.....	(292)
§ 12.2.2 使用代码回顾(即代码重查)的方法.....	(293)
§ 12.2.3 防止以后发生的问题.....	(294)
第三节 准备你的程序去调试.....	(295)
§ 12.3.1 使用调试型创建(build)与自由型创建方式创建程序	(295)
§ 12.3.2 建立一个调试型文件.....	(295)
第四节 使用集成调试器.....	(296)
§ 12.4.1 控制程序执行.....	(297)
§ 12.4.2 设置断点.....	(297)
§ 12.4.3 查看和修改变量.....	(298)
§ 12.4.4 用 QuickWatch 对话框检查变量	(298)
§ 12.4.5 修改变量.....	(299)
§ 12.4.6 各种各样的调试器功能.....	(299)
§ 12.4.7 使用 Call stack 对话框	(299)
§ 12.4.8 设置调试器参数.....	(299)
第五节 使用调试窗口.....	(301)
§ 12.5.1 Watch 窗口	(301)
§ 12.5.2 监视结构、类和表达式	(302)
§ 12.5.3 使用 Watch 窗口修改变量值	(302)
§ 12.5.4 Locals 窗口	(302)
§ 12.5.5 Registers 窗口.....	(302)
§ 12.5.6 Output 窗口	(303)
第六节 小结.....	(303)
第十三章 QuickWin 图形程序设计	
第一节 使用 Visual C++ 图形	(305)
第二节 使用 QuickWin 图形库	(307)

§ 13.2.1 QuickWin 库窗口	(308)
§ 13.2.2 QuickWin 视频硬件配置	(308)
§ 13.2.3 QuickWin 用户界面	(309)
第三节 理解图形坐标系	(310)
第四节 画圆	(310)
第五节 显示文本	(311)
第六节 选择字体和字符尺寸	(311)
第七节 查找图形错误	(312)
§ 13.7.1 错误报告函数 <code>_grstatus()</code>	(312)
§ 13.7.2 QuickWin 错误报告	(312)
第八节 使用视窗	(313)
第九节 使用当前指针	(314)
第十节 选择颜色	(314)
§ 13.10.1 从彩色调色板中选择颜色	(315)
§ 13.10.2 为对象选择内部颜色和模式	(316)
§ 13.10.3 使用并填充多边形	(319)
第十一节 在 QuickWin 应用程序中作图表	(320)
第十二节 小结	(328)
第十四章 MFC 通用类和集合类	
第一节 MFC 简介	(329)
第二节 准备 QuickWin 应用程序使用 MFC 库	(329)
§ 14.2.1 在 QuickWin 应用程序中包含头文件 <code>AFX.H</code>	(329)
§ 14.2.2 为 QuickWin 应用程序使用 MFC 库的工程选项	(330)
第三节 CString 类	(331)
第四节 CTime 类	(332)
第五节 CFile 类	(333)
§ 14.5.1 文本文件与二进制文件	(335)
§ 14.5.2 使用 <code>CStdioFile</code> 类	(335)
第六节 集合类	(336)
§ 14.6.1 理解集合类	(337)
§ 14.6.2 了解集合类	(337)
第七节 使用集合类	(338)
第八节 数组集合类	(338)
§ 14.8.1 向数组集合中插入元素	(339)
§ 14.8.2 访问数组项	(339)
第九节 链表集合类	(339)

§ 14.9.1	向链表集合中插入数据项	(339)
§ 14.9.2	访问链表节点	(341)
§ 14.9.3	删除链表	(342)
§ 14.9.4	搜索链表	(342)
第十节	映射集合类	(342)
§ 14.10.1	联系关键字与映射集合的值	(343)
§ 14.10.2	查找映射定义	(343)
第十一节	小结	(344)
第十五章 Visual C++ 用于 Windows 程序设计		
第一节	使用 AppWizard 编写第一个 Windows 程序	(345)
§ 15.1.1	AppWizard	(345)
§ 15.1.2	AppWizard 定义类	(348)
第二节	生成应用程序	(348)
第三节	运行应用程序	(348)
第四节	一个通用 MFC 应用程序	(349)
§ 15.4.1	WinMain() 函数	(349)
§ 15.4.2	应用程序窗口	(350)
第五节	FIRSTWIN 读者的第一个应用程序	(350)
第六节	初始化步骤	(352)
第七节	MFC 文档/视图体系结构	(352)
§ 15.7.1	创建主框架窗口	(353)
§ 15.7.2	显示主框架窗口	(353)
第八节	消息循环	(354)
第九节	处理命令消息	(355)
第十节	使用 ClassWizard	(355)
§ 15.10.1	用 ClassWizard 增加消息映射入口	(356)
§ 15.10.2	用 ClassWizard 编辑消息映射函数	(356)
第十一节	资源	(357)
第十二节	MFC 类的组织	(360)
§ 15.12.1	再谈文档	(361)
第十三节	小结	(365)

第一章 C++ 程序设计基础

在开始学写 C++ 程序前,先得熟悉 Visual C++ 程序设计环境。可以用 Visual C++ 编辑、编译和调试程序。在编辑阶段,要用 Visual C++ 输入或修改 C++ 源程序文件。在编译阶段,把源程序转移成目标文件,然后链接各个带启动代码的目标文件和库代码,创建一个可执行的 .EXE 应用程序。Visual C++ 还有几种工具,包括一个内部调试工具,以尽可能地提高编程效率。

本章将介绍 Visual C++ Visual Wordbench,这是 Microsoft 公司命名的 Visual C++ 下编辑及程序开发工具集。然后介绍第一个 C++ 程序和 C++ 程序常用的结构,包括若干 C++ 语句。学完本章,即可编辑和编译简单的 C++ 程序。读者应着重理解变量的基本概念,执行输入和输出功能的 cin 和 cout 的用法,以及用 if 语句和 for 循环语句控制程序的执行等几个方面内容。

第一节 使用 Microsoft Windows

因为 Visual C++ 是一种宿主为 Microsoft Windows 的编程环境,所以需要熟悉 Windows 用户界面的操作,下面几段将简单介绍怎样使用 Windows。如果你会使用 Windows,可以直接跳到下一部分。

Microsoft Windows 是一种图形用户界面(graphical wuer interface)或称 GUI,它使得计算机的使用较容易。与输入难以记忆的命令的交互式计算机系统不同,GUI 使用户能通过选择按钮和图形(叫作图标)与计算机进行图形交互。图 1.1 示例了 Windows 的屏幕。

注意:本书的术语 Windows(带有大写 W)指 Microsoft 的产品 Microsoft Windows。而术语窗口(window)指通用窗口。

可以用键盘和一个叫作鼠标的工具与 Windows 交互。鼠标用于在全屏幕中移动光标以选择和执行命令,鼠标光标通常是一箭头形状的指示符。要选某项时,需要把鼠标光标移到此项上,然后单击一次鼠标左键。要击活一命令时,通常需要双击此项。双击(double-click)表示在很短时间内按下并释放鼠标键两次。一般只需要几分钟就能在 Windows 下熟悉这些操作技术。

§ 1.1.1 定义通用窗口

窗口是屏幕上的一块区域,用于与用户界面的不同部分交互。这种用户界面最开始由 Xerox 开发,而由 Apple MacIntosh 推向流行。图 1.1 显示了一个带多种构件的典型窗口。

§ 1.1.2 定义对话框

对话框是 Windows 中用于控制选项的窗口。图 1.2 表示了一个示例对话框。