

普通高校系列教材·信息技术

Visual C++与Windows编程

普通高校系列教材（信息技术）编委会组编

汪小龙 编

学习参考



南京大学出版社

普通高校系列教材·信息技术

《Visual C++ 与 Windows 编程》学习参考

汪小龙 编

南京大学出版社

内 容 简 介

本书是依据《Visual C++ 与 Windows 编程》教材编写的学习参考书。该书适合于在学完“Visual C++ 与 Windows 编程”课程各章后复习时使用,其中包括各章的基本内容、重点难点、章后习题解答,以及针对各章内容而挑选出的典型例题。另外在本书的后面还针对性地给出模拟测试卷,使读者在学完整个课程后进行自我测试和检查。为了培养读者的实际应用能力,本书的最后安排了十个具有代表性的实验题目。通过这些实验题目,使读者能够举一反三,提高设计应用程序的能力。本书可作为“Visual C++ 与 Windows 编程”课程的学习参考教材,也可作为 Visual C++ 程序员学习 Windows 编程的辅导材料。

图书在版编目(CIP)数据

《Visual C++ 与 Windows 编程》学习参考/汪小龙编.—南京:南京大学出版社,2003.5
ISBN 7-305-04073-8

I.V... II.汪... III.C语言-程序设计-高等学校-教学参考资料 IV.TP312
中国版本图书馆 CIP 数据核字(2003)第 037593 号

书 名 《Visual C++ 与 Windows 编程》学习参考
编 者 汪小龙
出版发行 南京大学出版社
社 址 南京市汉口路 22 号 邮编 210093
电 话 025-3596923 025-3592317 传真 025-3303347
网 址 www.njupress.com
电子函件 nupress1@public1.ptt.js.cn
经 销 全国新华书店
印 刷 合肥学苑印刷厂印刷
开 本 787×1092 1/16 印张:12.625 字数:303 千
版 次 2003 年 6 月第 1 版 2003 年 6 月第 1 次印刷
书 号 ISBN 7-305-04073-8/TP·261
定 价 18.00 元

* 版权所有,侵权必究

* 凡购买南大版图书,如有印装质量问题,请与所购
图书销售部门联系调换

普通高校系列教材(信息技术)编委会

主任:孙钟秀 中国科学院院士

副主任:张福炎 南京大学教授

陈国良 中国科学技术大学教授

钱洲胜 中国计算机函授学院教授

委员(按姓氏笔画排序):

于学锋	中国计算机函授学院	王佩珠	西安交通大学
王文兰	桂林电子工业学院	王蔚韬	重庆建筑大学
伍良富	成都电子科技大学	成松林	东南大学
刘存书	郑州信息工程大学	朱大奇	安徽工业大学
朱宝长	西安电子科技大学	孙德文	上海交通大学
杜象元	上海交通大学	李茂青	厦门大学
李学干	西安电子科技大学	杨来利	兰州大学
何淑兰	北京科技大学	张凤祥	华中科技大学
张绍林	河北行政学院	张维勇	合肥工业大学
张民坤	云南工业大学	张景书	哈尔滨工程大学
赵良全	新疆大学	皇甫正贤	东南大学
洪志全	成都理工学院	姚君遗	合肥工业大学
高平	浙江大学	陶世群	山西大学
曹翊旺	湖南省计算机高等专科学校	梁文康	山东大学
韩国强	华南理工大学	舒洪	南昌大学
葛燕	中国科学技术大学	解世耀	辽宁大学
谭耀铭	南京大学	黎庆国	合肥工业大学

出版前言

近些年来我国的高等教育事业有了长足的发展,高校招生人数年年递增,越来越多的年轻人有机会接受正规的高等教育。这一举措无疑对我国的社会进步和经济发展有着重要的意义。但是人们也深刻的认识到,高等教育质量的好坏是一个不容忽视的关键性问题,而保证教育质量的一个重要环节就是抓好教材建设。但是教材内容陈旧,教学手段落后的现象一直存在着。尤其像计算机技术这样的新兴领域发展迅猛,知识更新日新月异,教学内容落后于新技术新知识的矛盾显得尤为突出。基于上述两方面考虑,在南京大学出版社的鼎力相助下,一个以组编高校信息、电子类专业教材为主要任务的教材编委会成立了。

针对我国高等教育的现状和信息、电子技术的发展趋势,编委会组织部分高校的专家教授进行了深入的专题研讨。大家一致认为,在当前情况下组编一套紧跟新技术发展、符合高校教学需要、满足大学生求知欲望的系列教材势在必行,这将有助于规范教学体系、更新教学内容、把握教学质量,培养合格人才。专家们还对教材的结构、内容、体例及配套服务等提出了具体要求。为了能使这套教材逐步完善,并促进全国各地高校教学质量的提高,编委会决定在教材之外认真做好三件事:第一,为每本教材配备一本供学生使用的学习参考书,其主体内容为学习方法指导、习题分析与解答、典型题解或课程设计、模拟测试卷及解答、实验指导书;第二,定期对教材内容进行修订,及时补充新技术新知识,并根据具体情况组编新的教材;第三,有计划的组织各地高校教师进行教学交流与研讨,通过这种途径来提高偏远地区的师资水平。我们相信,通过各方面的大力支持和大家的不懈努力,这套教材会逐步被广大师生所接受,并在使用过程中得到完善、充实。

大家都知道,组编这样一套系列教材是个牵涉面很广的大工程。这个工程不仅在起步阶段需要得到各级教育主管部门、各高等院校、出版社的大力支持和协助,而且在使用过程中也离不开各位专家、教授、学生的热心呵护和指导。因此,殷切期待所有的能人志士关心我们,帮助我们,向我们提出好的建议或意见,为我们指出教材中的不足之处。

最后,感谢所有为本套系列教材出版付出辛勤劳动的同志们。

普通高校系列教材(信息技术)编委会

2003年3月

编者的话

该书是普通高校系列教材《Visual C++ 与 Windows 编程》的配套辅导教材,适合于在学完“Visual C++ 与 Windows 编程”课程每章后复习时使用。本书的主要内容有:第一部分是学习方法指导,包括各章节基本内容提要、重点难点讲解、学习思路和方法等,使读者既掌握教材的重点和难点,又能把握学习思路和方法;第二部分是课后习题解答,给出了教材各章后的习题答案,通过解答书后的习题,读者可进一步巩固所学的知识,增强编程能力;第三部分是典型题解,整理出各章的典型题目,并给予分析和解答,以培养读者分析问题、解决问题的能力;第四部分是模拟测试,总结性地给出五套模拟测试卷,供读者学完整个课程后进行自我测试和检查;第五部分是实验指导,安排了十个具有代表性的实验题目,以此帮助读者举一反三,培养实际应用能力,提高应用程序的设计水平。

本书作为“Visual C++ 与 Windows 编程”课程的学习参考教材,也可作为 Visual C++ 程序员学习 Windows 编程的辅导材料。

由于编者水平有限,难免有不妥或疏漏之处,请批评指正和提出宝贵意见。

编者

2003年3月

目 录

第一部分 学习方法指导	(1)
一、课程的性质与要求	(1)
二、如何学好本课程	(1)
三、课程的主要内容	(2)
四、课程的基本要求	(2)
五、重点难点解析	(3)
第二部分 习题解答	(19)
第一章	(19)
第二章	(20)
第三章	(21)
第四章	(23)
第五章	(24)
第六章	(30)
第七章	(32)
第八章	(33)
第九章	(34)
第十章	(48)
第三部分 典型题解	(76)
第四部分 模拟测试卷	(113)

模拟测试卷(一)·····	(113)
模拟测试卷(一)参考答案·····	(115)
模拟测试卷(二)·····	(120)
模拟测试卷(二)参考答案·····	(121)
模拟测试卷(三)·····	(127)
模拟测试卷(三)参考答案·····	(128)
模拟测试卷(四)·····	(135)
模拟测试卷(四)参考答案·····	(137)
模拟测试卷(五)·····	(143)
模拟测试卷(五)参考答案·····	(145)
第五部分 实验指导 ·····	(155)
实验一·····	(155)
实验二·····	(157)
实验三·····	(158)
实验四·····	(160)
实验五·····	(162)
实验六·····	(164)
实验七·····	(165)
实验八·····	(167)
实验九·····	(168)
实验十·····	(170)
附录 程序调试与优化 ·····	(173)

第一部分 学习方法指导

一、课程的性质与要求

“Visual C++ 与 Windows 编程”是一门理论性、实践性和前沿性都很强的课程。对于计算机等电子信息类专业的高年级学生来说,掌握 Visual C++ 这样一个功能最强大的编程工具是必不可少的。本课程的学习是以微机原理、操作系统、编译原理、C 语言和数据结构等专业基础课为基础,同时也是对以上专业基础课的应用与升华。

但是,由于 Visual C++ 对 Windows 系统 API(Application Programming Interface 应用编程接口)的完美封装,使得使用者可以跳过操作系统、编译原理和数据结构的学习。因此,对于广大的非计算机专业电子类高校学生来说,初步掌握 Visual C++ 和 Windows 编程没有额外的课程负担。考虑到国内仍有些高校没有开设 C 语言课程或者将其作为选修课,因此,教材用较少的篇幅介绍了 C 语言的基本语法和知识点。对于有 PASCAL、BASIC 或者 FORTRAN 等高级语言基础的初学者来说,也可以跳过 C 语言的学习而直接进入“Visual C++ 与 Windows 编程”课程的学习。

与此同时,由于 Visual C++ 在很大程度上是用来开发系统软件和设备驱动程序,因此,随着程序员的编程能力提升和工作范围拓宽,前面介绍的专业基础课知识将会相继用到。读者可以在以后的学习和工作过程中,不断充实自己关于操作系统、编译原理和数据结构的知识。

二、如何学好本课程

很多关于 Visual C++ 的书籍都提供了类似说明书的步骤指导,因此,读者完全可以像学习 VB 和 Delphi 那样写一些简单的程序。从这一点来说,初学者能够很快进入 Visual C++ 的编程状态。但是,Visual C++ 的魅力不在于简单,而在于其强大的功能和广博的应用领域。

实际上,Visual C++ 与 Windows 编程的学习过程是一个知识面不断拓宽的过程。下面的学习方法和步骤是作者和一些业界人士的共同心得,希望能够帮助读者更好地学习本课程,减少不必要的弯路。

第一步,通过学习本学习参考和其配套教材《Visual C++ 与 Windows 编程》,获得 Visual C++ 和 Windows 编程的基本知识和技能,为进一步提高打下坚实的基础。

第二步,通过阅读 MSDN 中关于 Visual C++ 和 Windows 的内容,加深对系统运行机制的理解和领悟,提高自己作为程序员的素质。

第三步,通过调试、运行和阅读 MSDN 提供的大量示例程序,获得处理实际编程问题的基本思路、方法和常用技巧,提高自己作为程序员的能力。

第四步,通过编写、调试和优化自己的程序代码,完成特定的功能、速度和实时性要求,培养实际工作能力和技巧。

读者应该能够理解,以上四个步骤并不是相互独立的过程。在总体规划上,具有时间的先后性与过程的延续性。而在处理具体编程问题时,往往是相互穿插进行的。

三、课程的主要内容

《Visual C++ 与 Windows 编程》全书可以分为三个部分:入门部分、MFC 部分、提高部分。

入门部分包括第一章、第二章和第三章。该部分介绍 Visual C++ 6.0 的强大功能,将读者引入 Windows 编程的奇妙世界,让读者迈出作为程序员最坚实的第一步。

MFC 部分以入门部分内容为基础,深入探讨基于 C++ 的微软基础类(MFC)。该部分由五章组成,涵盖的内容有 MFC 的世界观、理解 MFC 程序、理解 AppWizard 生成的代码、使用 Class Wizard 和丰富多彩的窗口。

提高部分介绍在实际编程中应用最为广泛的三个知识点,即文档视图结构、ActiveX 控件和动态链接库。

全书各部分的内容安排如图 1-1 所示。

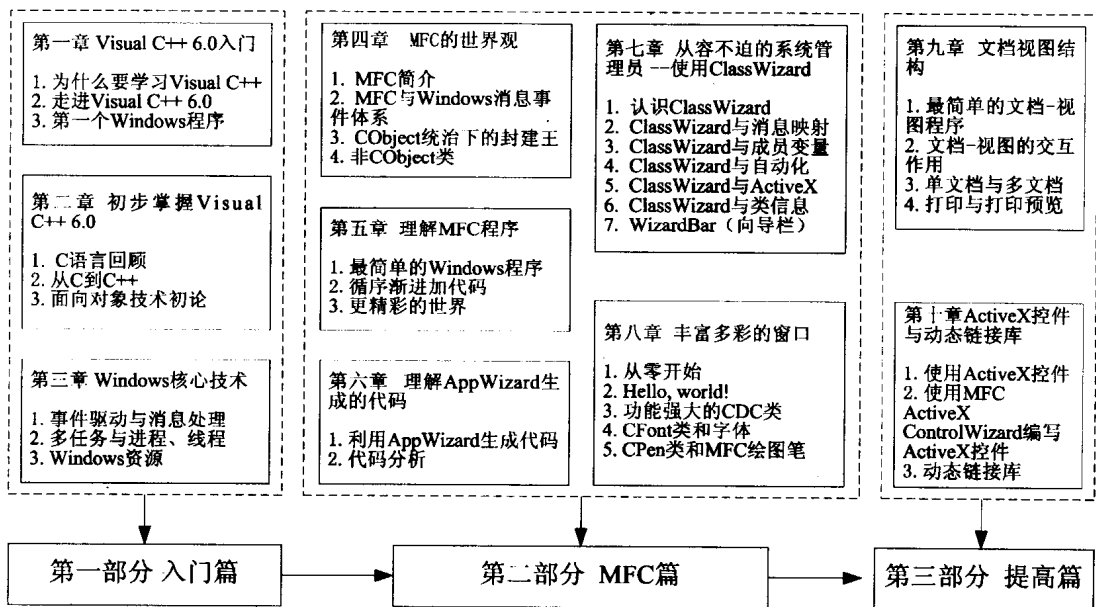


图 1-1

四、课程的基本要求

《Visual C++ 与 Windows 编程》是《普通高校系列教材·信息技术》系列教材中的一本,介绍了 Visual C++ 与 Windows 编程的基本知识和技能。课程的基本要求如下:

- ① 熟悉 Visual C++ 的集成环境。
- ② 初步了解 Windows 程序的来龙去脉。

- ③ 熟悉 C/C++ 语言基本语法和知识要点。
- ④ 理解面向对象程序设计中的最基本概念,包括:对象、类、消息、方法、类与封装性、派生类与继承性、虚函数与多态性。
- ⑤ 理解 Windows 程序运行的机制,即“事件发出消息,消息激活程序(进程)”。熟悉消息的数据结构和消息的产生机理,重点加深理解消息处理循环的机制和作用。
- ⑥ 理解 Windows 环境多任务机制与进程、线程的关系,重点掌握多线程的实现过程。
- ⑦ 了解 Windows 系统可执行程序独特的资源配置结构:允许将资源附加在应用程序可执行代码的后面,而不必嵌入到程序可执行部分的内部。熟悉 Visual C++ 提供的资源和资源编辑器的使用,Visual C++ 提供的十种资源包括加速键、位图、光标、对话框、超文本、图标、菜单、串表、工具条和版本信息。
- ⑧ 了解 MFC 及其与 Windows 消息事件体系的关系,掌握 MFC 应用程序中消息处理的方法,熟悉消息映射中使用的宏、消息声明宏及其函数原型。
- ⑨ 熟悉 CObject 及其派生类。
- ⑩ 熟悉非 CObject 类。
- ⑪ 了解 MFC 程序的项目建立、代码和资源的添加方法,理解简单 MFC 程序的格局和运行机理,初步掌握程序中 bug 的修正方法。
- ⑫ 掌握利用 AppWizard 生成代码的方法,学会分析 AppWizard 生成的复杂代码。
- ⑬ 学会使用 ClassWizard 进行系统管理,理解 ClassWizard 中消息映射、成员变量、自动化、ActiveX 事件以及类信息的关系,并学会使用 ClassWizard 的精简版本 WizardBar(向导栏)。
- ⑭ 学会如何为 AppWizard 生成的程序添加功能代码,实现丰富多彩的窗口。
- ⑮ 理解文档视图结构的机理,学会如何实现文档一视图结构的编辑和串行操作功能;重点掌握 CEditView 类与 CArchive 类的应用。
- ⑯ 理解单文档与多文档的实现机理,了解它们的异同点。
- ⑰ 掌握打印与打印预览的实现方法。
- ⑱ 了解 ActiveX 控件的相关背景知识,初步掌握 COM、OLE、客户和服务器的基本概念。
- ⑲ 学会如何在对话框中使用 ActiveX 控件。
- ⑳ 学会如何在 Web 页中使用 ActiveX 控件。
- ㉑ 学会如何使用 MFC AppWizard 创建 DLL 项目,以及如何添加功能代码,学会如何调用 MFC DLL。

五、重点难点解析

对应课程的基本要求,本书的重点难点解析如下:

1. Visual C++ 6.0集成环境

本部分内容位于教材 1.2 节。由于该部分覆盖了全书内容,因此,大部分菜单命令将在相关章节中进一步解释。在这里,大部分菜单命令只需要熟练使用,而不需要深入探讨。本部分的重点有:

(1)“File”菜单中的“New (Ctrl + N)”命令

该命令用于创建新文件、工程(Projects)、工程工作区(Workspaces)和其他文档。该命令运行时将打开一个对话框, Visual C++ 将根据用户选择的文件类型和指定文件及路径自动创建文件,并且打开相应的编辑器。

(2)“Edit”菜单中的“Find (Ctrl + F)”命令

执行该命令激活“Find”对话框,在对话框中可输入在当前窗口中要查找的字符串,同时可以选择查找方向和匹配模式。

该对话框中各选项含义如下:

- ① “Match whole word only”:只匹配整个单词,不包括以子字符串形式出现的部分。
- ② “Match case”:将大小写不同的单词作为不同的单词对待。
- ③ “Regular expression”:只搜索常用表达式,不包括注释。
- ④ “Search all open documents”:搜索“Window”菜单栏中列出的所有文档,无论该文档是否属于本次编程范围。
- ⑤ “Up”:从光标处向前搜索。若搜索所有文档,则按“Window”菜单栏中的排列顺序进行,下同。
- ⑥ “Down”:从光标处向后搜索。
- ⑦ “Mark All”:将所有搜索到的字符串反白显示。
- ⑧ “Find Next(F3)”:搜索下一个。
- ⑨ “Cancel”:取消本次操作。

(3)“View”菜单中的“Debug Windows”子菜单

该菜单中的命令主要用于控制调试工具 Debug 窗口在工作台上的显示。菜单中的命令简要介绍如下:

- ① “Watch(Alt + 3)”命令:用于在 Debug 过程中激活观察窗口,增加被观察的项。
- ② “Call Stack (Alt + 7)”命令:按调用顺序显示程序调用的函数窗口。
- ③ “Memory (Alt + 6)”命令:显示观察程序加载内存值的窗口。
- ④ “Variables (Alt + 4)”命令:该命令用于在 Debug 过程中显示观察程序中各变量值的窗口。
- ⑤ “Register (Alt + 5)”命令:该命令用于在 Debug 过程中显示当前寄存器的值。
- ⑥ “Disassembly(Alt + 8)”命令:该命令用于在 Debug 过程中显示程序的汇编代码。

2. Windows 程序来龙去脉

所有 Windows 程序都必须从 WinMain 函数开始。WinMain 函数被系统调用,用于初始化应用程序、显示窗口、进入消息循环。在 WinMain 的函数体中,程序除了进行一些必要的资源配置以外,主要进行以下工作:

- ① 注册窗口类。
- ② 应用程序初始化。

③ 开始消息循环。

WinMain 函数以外的其他函数都是由 WinMain 函数直接、间接调用的;除了这些函数外的 # include 语句、# define 宏定义、全局变量、函数预定义,都是为这些函数服务的。

而除了“消息循环”以外,WinMain 函数的其他部分都是只执行一次。因此,在程序正常运行时,真正活动的只是“消息循环”和被“消息循环”触发的那些“动作”。通常所说的“消息循环是 Windows 程序运行的脉搏”,就是这个意思。

3. C/C++ 基本语法和知识要点

(1) Microsoft C 与 ANSI C 标准

C 语言的语法来自 ANSI X3.159 - 1989(美国国家信息系统——程序设计语言)——C 标准,简称为 ANSI C 标准。Microsoft C 符合并且扩充了 ANSI C 标准,这些扩充限制了 Windows 系统与非 Windows 系统之间程序的移植性。为了禁止这些扩充必须指定 /Za 编译器选项,这样,非 ANSI 代码会产生编译错误或警告。

(2) 标识符

① 标识符在拼写和大小写上必须不同于关键词。

② 标识符名称的第一个字符必须是非数字。

③ C 编译器把大写和小写字母当作不同的字符。

④ 标识符名称不能以“两个下划线”或“一个下划线加上一个大写字母”开头,文件层范围的标识符也不要以一个下划线加上一个小写字母开头来命名。

(3) 逗号运算符

逗号运算符“,”是一种特别的运算符。用逗号将两个表达式连接起来,即可形成逗号表达式(如 $3 + 15, 4 * 6$)。

逗号表达式先计算前一个表达式 $3 + 15$ 得到值 18,再计算后一个表达式 $4 * 6$ 得到值 24,整个逗号表达式的值为后一个表达式的值,即 24。

(4) 条件运算符

条件运算符是三目运算符,是用“?”和“:”把三个运算对象连接在一起的表达式。

“?”前的关系表达式用于表示条件;中间和后面的两个表达式分别代表条件表达式可取的两个值。当条件运算结果为真时,取前一个值,否则取后一个值。

例如: $x > y ? 3 : 9$

当 $x > y$ 成立时,表达式的值为 3,否则为 9。

(5) 运算符的优先级和结合性

为了确定各种运算的先后顺序,C 语言规定了各种运算符的优先级和结合性,见表 1-1。

表 1-1 运算符的优先级和结合性

运算符	名称	优先级	结合性	运算符	名称	优先级	结合性
()	括号,函数调用	1	自左至右	> =	大于或等于	7	自左至右
[]	下标			< =	小于或等于		
- >	间接成员选择符			= =	等于		
.	直接成员选择符			! =	不等于		
!	逻辑非	2	自右至左	&	按位与	8	自左至右
~	按位取反			^	按位异或	9	自左至右
+	增 1+				按位或	10	自左至右
-	减 1-			&&	逻辑与	11	自左至右
&	取地址				逻辑或	12	自左至右
*	取内容			?	条件运算符	13	自左至右
-	单目减			=	等于	14	自右至左
(类型)	类型强制转换			+ =	复合运算赋值符		
sizeof	计算类型长度			- =			
*	乘			* =			
/	除	/ =					
%	取模	% =					
+	加	& =					
-	减	^ =					
<<	按位左移	=					
>>	按位右移	>> =					
>	大于	<< =	15	自左至右			
<	小于	,			逗号运算符		

表中运算符的优先级以 1 级为最高,数值越大优先级越低,优先级高的运算符先运算。结合性“自左至右”是把运算对象先与左边的运算符结合,“自右至左”则反之。

(6) 生存期、范围、可见性和连接

“生存期”是程序执行期间一个变量或函数存在的时间。

标识符的范围是指可以使用该标识符的程序部分,有时称之为“词法范围”。标识符仅在它的范围内可见。

同一个标识符在不同的范围可以指不同的标识符。一个标识符在不同范围内说明,或者在同一范围内说明多次,可以通过称为“连接”的过程使之指向相同的标识符或函数。

表 1-2 是大多数标识符的生存期和可见性特性的总结。开头三列给出定义生存期和可见性的属性,具有开头三列属性的标识符的生存期和可见性在第 4 列和第 5 列中给出。该表不包括所有可能的情况。

表 1-2 生存期和可见性总结

属 性			结 果	
层	项	存储类指示符	生存期	可见性
文件范围	变量定义	static	全局的	在出现它的源文件中的剩余部分
	变量说明	extern	全局的	在出现它的源文件中的剩余部分
	函数原型或定义	static	全局的	单个源文件
	函数原型	extern	全局的	源文件的剩余部分
块范围	变量说明	extern	全局的	块
	变量定义	static	全局的	块
	变量定义	auto 或 register	局部的	块

(7) 结构说明与位域

一个“结构说明”命名一个类型,并指定具有不同类型变量值(称为结构的“成员”或域)的序列。一个称为“标志”的任选标识符给出结构类型的名称,该名称可以用在后面对结构类型的引用中。一个结构类型的变量保存该类型定义的全部序列。

除了说明结构的成员外,结构说明符还可以指定位的个数,称为位域,其长度由说明符的域名后的冒号来设置。一个位域解释为一个整型。

(8) 联合说明

联合是 C 语言中又一种构造类型,它的形式与结构比较相似,它是一个特殊形式的结构。在结构中各成员有各自的存储区域,而联合则不同,它把不同类型数据放在存储器相同的位置中。也就是说,联合中的各成员占用的存储区域是相互重叠的,通过联合中不同的成员可以用不同的方式访问同一块存储区域。在任何给定的时刻,仅有一个成员驻留在联合中,这和结构也是不同的。

(9) 复合语句

复合语句也称“块”,一般以另一语句的语句体形式出现。每个在复合语句开头说明的标识符,其范围是从说明点开始延伸到该块结束。

(10) switch 语句

switch 语句帮助控制复杂条件和分支操作。switch 语句可以包含任何个数的 case 实例,但在同一个 switch 语句中不能有两个 switch 常量具有相同的值。switch 语句把控制转向该语句体中的一个语句。可以使用 break 语句终止 switch 语句中一个特殊情况的处理,并跳转到该 switch 语句的末尾。若没有 break 语句,则继续执行下一个 case,直到碰到一个 break 语句或到达该语句末尾。switch 表达式和 case 常量表达式的类型必须是整数。每个 case 常量

表达式的值在语句体中必须是唯一的。

(11) 函数属性

`fastcall` 指定函数的参数,在可能的情况下将通过寄存器传输。

`inline` 关键字告诉编译器在一个函数调用的每个实例中用该函数的定义替换该代码。

联编汇编器 `asm` 允许 C 源程序中直接嵌入汇编语言指令,而不要额外的汇编和链接步骤。

`dllimport` 和 `dllexport` 存储类修饰符显式定义了 DLL 的客户界面,DLL 界面指的是系统中某个程序中输出的所有已知项(函数和数据)。`dllimport` 和 `dllexport` 必须与扩充的属性语法关键字 `declspec` 一起使用。

对于用 `naked` 存储类属性说明的函数,编译器生成没有序言和结尾部分的代码。

(12) `new`、`delete` 操作符

`new`、`delete` 可以帮助用户方便地申请内存和回收内存。

`new` 操作符可以指定数据类型,则 C++ 编译器将按类型分配一块自由内存,返回分配块地址的指针。`new` 除了可以分配 C++ 内部类型对象的内存,还可以按照用户自定义的结构分配内存。如果分配内存失败则返回为 0。

`delete` 操作符用以回收 `new` 操作分配的内存。`new` 和 `delete` 操作符通常用于精确控制对象寿命。特别是在编译时对象的大小和数量未知的时候,动态地生成内存对象,对控制程序使用内存及加快程序运行速度都大有益处。

4. 面向对象程序设计基本概念

面向对象的程序是围绕被操作的数据而设计的,而不是操作本身。

(1) 对象、方法、消息、类的相互关系

对象是基本的运行时的实体,它既包括数据(反映对象的状态),也包括作用于数据的操作(对象的行为)。对象将数据与操作封装成一个整体。从程序设计者来看,对象是一个程序模块;从用户来看,对象是对客观事物的一种抽象,它包括了此事物的状态及所具有的行为。

在对象内的操作通常称为方法。

对象之间进行通信的一种构造叫作消息(Message)。

将具有相同结构、操作并遵守相同约束规则的对象聚合在一组,这组对象集合就称为类(Class)。

类是对象的抽象,对象是类的具体化,是类的实例(instance)。

(2) 虚函数与多态性

虚函数是一种在基类中说明为 `virtual`,并由派生类重新定义的成员函数。虚函数实现了面向对象编程的多态性这一重要特征。

多态性(Polymorphism)是允许把同一消息发送给父类及其子类的所有对象的能力。多

态性的重要性在于允许父类对象各自以不同的方法响应同一消息,即所谓“同一接口,不同方法”。多态性有两种:编译时的多态性和运行时的多态性,前者由重载函数和重载运算符来实现,后者由继承和虚函数来实现。

虚函数在不同的类中执行不同的操作。由派生类重新定义的虚函数的原型,必须与基类中说明的虚函数原型完全匹配,即函数返回类型、参数类型与个数都必须一致。虚函数可以确保在一个对象中调用正确的函数,而不管用于调用函数的表达式。

虚函数必须是基类的成员函数,而不能是友员函数。构造函数不能是虚拟的,而析构函数可以是虚拟的。

5. Windows 程序的运行机制

Windows 程序运行的机制是:事件发出消息,消息激活程序(进程)。

事件的含义很广泛,任何一个拥有设备驱动程序的系统输入设备(包括端口),都可以产生各种事件,发送自己的消息。除了键盘和鼠标等系统输入设备可以发出消息外,Windows 定时程序或 Windows 本身也可以产生消息。此外,事件也可以由用户来定义。程序员可以在程序的任何地方利用 SendMessage 等函数发送自己的消息。

Windows 系统在将事件用 MSG 结构打包后,将其放在消息队列中以供应用程序取用。

接受并处理消息的主角是窗口。当应用程序收到一个事件消息(除 WMQUIT)时,就传递消息值。首先,该消息到达 TranslateMessage 函数,进行具体的按键翻译。然后被传递到 DispatchMessage 处理器上,消息被传递到适当的消息处理过程上。最后,当消息处理循环结束时,最终读取到的消息的 wParam 参数又返回到调用应用程序(Return msg, wParam)。

事件驱动程序设计是围绕着消息的产生与处理而展开的。作为一个 Windows 程序员,应用程序所做的最主要的工作,就是对各种消息进行处理。程序运行后,就在那里静静地等候各种事件发出消息,然后进行相应的处理。

消息驱动,也称事件驱动,是一种进程,各种子进程和应用程序通过它进行通信。Windows 系统用该进程来管理多任务系统,通过它给应用程序、应用程序实例和应用程序内部的进程分发信息,从而共享键盘、鼠标和其他资源。

6. 多任务与进程、线程

进程是加载到内存中的程序,以及分配给这个程序的所有资源。每个进程都拥有一个主线程,就是在 WinMain 开始执行的那个线程,该线程运行时可以创建其他线程。一个进程中的所有线程都共享可执行文件的一个映像,每个新进程都要求系统把可执行文件映射到一个新的地址空间。

线程与进程密切相关。进程是静态的,本身不完成任何事情。线程执行程序命令,遵循通往代码的一个路径。每个进程都拥有一个主线程,主线程可以有选择地创建其他线程。属于一个进程的所有线程都共享这个进程的财富,它们都遵循来自相同代码映像中的指令,引用相同的全局变量,写入到相同的私有地址空间,以及具有对相同对象的访问权。

系统保存所有线程的一个列表,并循环通过它们在处理器上给每个线程都分配一段时间。当一段时间结束时,系统在线程对象中记录当前的 CPU 寄存器值,包括线程将要执行的指令指针。然后系统选择另一个线程,恢复其 CPU 寄存器值,从上次离开线程的位置继续