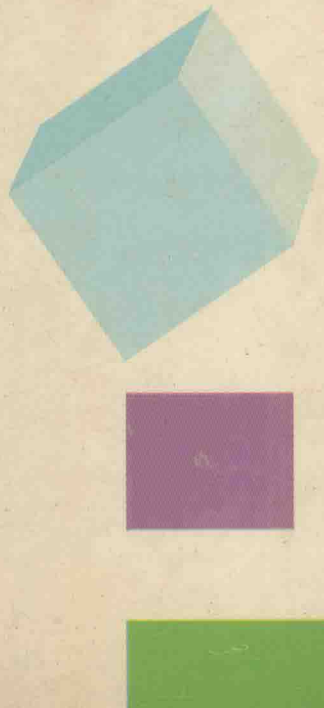
Visual C++ Windows 程序设计

周升锋 编著



西安交通大学出版社

Visual C++
Windows 程序设计

周升锋　编著

西安交通大学出版社

内容提要

在 Windows 下编程已是大势所趋，而 Visaul C++则是在 Windows 下编程最好的选择。如何快速进入 Windows/Visual C++编程，一直是令 DOS 程序员感到困难的事情。本书就是为了解决这个难题而编写的，目的是使程序员在短时间内学会 Windows 编程。本书主要内容有：Windows 编程基础，数据输出设计，鼠标和键盘的应用，滚动条、定时器、控制元件、资源设计以及基本图形设计。书中有大量实例，并用汉字注释，便于读者充分地理解和掌握。

本书适合具有 C 语言基础的大中专学生和程序设计人员阅读。

(陕)新登字 007 号

Visual C++Windows 程序设计

周升锋　编著

责任编辑　何向荣

组稿编辑　叶　涛

*

西安交通大学出版社出版发行

(西安市咸宁西路 28 号　邮政编码:710049)

西安电子科技大学印刷厂印装

各地新华书店经销

*

开本 787×1092　1/16　印张:18.75　字数:454 千字

1996 年 6 月第 1 版　1996 年 6 月第 1 次印刷

印数:1——5000

ISBN7-5605-0847-2/TP·133　定价:18.00 元

前　言

Windows 是一个以图形用户界面为基础、功能强大的计算机软件环境。它以丰富多彩的图形取代了 MS-DOS 以字符为基础的单一形态，提供了漂亮、统一、友好的用户界面。特别是 90 年代初 Windows 3.X 的推出，在计算机领域产生了巨大而深远的影响。它以 80386CPU 为硬件平台，充分发挥了 80386 处理器的能力，使其在图形功能、运行速度、内存管理、多任务环境、资源共享以及支持多媒体等方面都有了很大的发展。

Windows 支持面向对象的程序设计方法，提供了多窗口处理、多任务环境以及消息驱动的程序结构。Windows 强大的内存管理能力，使得在 Windows 平台上开发的计算机软件可以不考虑内存的管理就可以获得良好的运行品质。Windows 支持资源共享的能力以及丰富的资源，形成了良好的软件开发环境。Visual C++面向对象开发方式和可视化开发工具，极大地提高了软件开发效率。这对于缩短软件开发周期、开发出高质量、用户界面好的应用软件，都具有十分重要的意义。

Windows 3.X 推出的时间虽然不长，但销售量已达数千万套，用户更是逾以亿计，其程序设计思想和方法正被越来越多的软件开发人员所接受。几乎市场上所有畅销的 PC 软件都有 Windows 版本。

如何紧紧跟上这个迅猛发展的潮流，掌握 Windows 环境下编程的技术，已成为国内软件开发人员十分关切的问题。但目前可以得到的参考资料和书籍，大多以详尽、深入地介绍 Windows 的方方面面功能的使用为目的，这就使得初学者很难在短时间内掌握 Windows 的程序设计要领，从而对开发 Windows 应用程序望而生畏。本书就是为了解决这些问题而写的。本书的内容包括：Windows 编程基础，Visual C++编辑环境介绍，数据输出设计，鼠标和键盘应用、滚动条、定时器、控制元件及资源设计。每一部分内容都有应用程序实例，并用汉字加了注释，便于读者充分地理解和掌握。

本书是 Windows 程序设计的入门教材，读者只要具备 Windows 的基本知识，学习过 C 语言，即可顺利阅读本书，学会用 Visaul C++进行 Windows 程序设计。

本书在写作过程中，得到了马明建副教授、周长城副教授、崔胜民教授、张宇副教授、王好臣副教授的帮助，作者表示衷心的感谢！

对本书存在的错误，敬请读者批评指正。

作　者

1996 年 2 月

目　录

第1章

Windows/Visual C++编程概述

1.1 Windows程序特点

Windows是Microsoft公司开发的一个图形窗口环境软件，1985年11月推出了第一个公开版本。此后，Windows经过了不断修改与更新，目前正在流行Windows 3.1版。Windows的推出，特别是1990年5月推出3.0版以后，立即引起了用户的强烈反响，使得操作计算机的方法和软件开发过程发生了根本性的变化。

Windows运行于MS-DOS之上，是MS-DOS的扩展。它与MS-DOS共同管理计算机的硬件资源。MS-DOS继续管理文件系统，而Windows管理其它一切，包括显示器、键盘、鼠标器、打印机和串行口，并负责内存管理和程序的执行、调度。

Windows丰富多彩的界面改变了存储管理能力，它能同时运行多个任务；可以通过多种方式进行应用程序间的数据交换，实现信息共享；能支持网络系统；在多媒体技术方面更是大显身手。

与DOS环境相比，Windows不管是对用户还是对应用程序开发人员，都提供了更加强大的功能，特别是其先进的程序设计思想和方法，更是程序设计人员必须熟悉和掌握的。

1. 漂亮、统一的用户界面

用户界面友好是衡量应用程序质量的一个重要方面。在DOS环境下，程序员采用C语言，为了设计出友好的用户界面，往往要花费很多的精力和时间，而设计出的界面还是因人而异，没有统一的风格。这给用户使用和掌握应用程序带来了困难。

Windows为设计漂亮、统一和友好的用户界面提供了一种全新的方法。Windows是一个图形用户界面(GUI)，各种操作对象都是以图形形式显示在屏幕上，通过鼠标器或键盘，用户就可以在屏幕上直接操纵这些对象。

所有Windows应用程序的基本外观相差不大，每个程序占据一个窗口。窗口是屏幕上的一个矩形区域，它是由标题条来标识的。程序中绝大多数功能均可通过菜单来执行。某些菜单会触发出对话框，通过对话框，用户可以输入较多的附加信息。图1-1是一个典型的Windows程序界面。

利用Windows提供的Help工具，可以为应用程序创建统一的联机求助系统，用户可以很方便地搜索、查询有关信息。

此外，Windows还提供了一种多文档界面技术。利用这种技术，在一个应用程序中可以创建多个窗口，分别用来显示和处理不同的文档。

由于用户界面的一致性，用户一旦学会了如何使用Windows应用程序，就会很快掌握其它Windows应用程序的使用方法。在Windows下设计统一的用户界面是件很容易的事情。

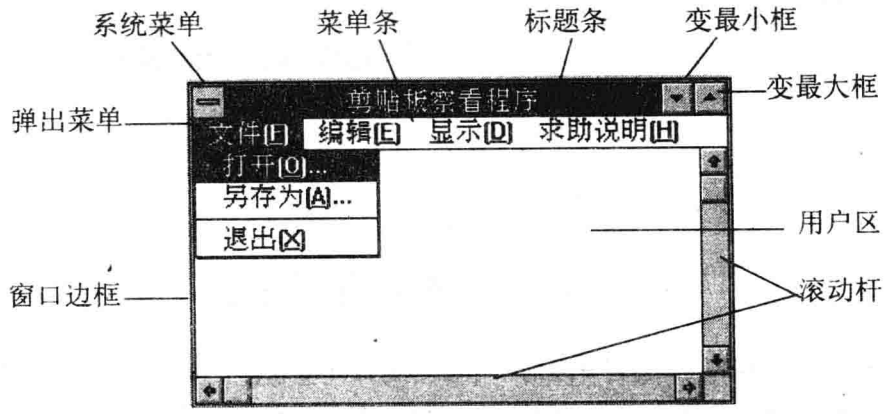


图 1-1 典型 Windows 程序界面

2. 面向对象的程序设计

Windows 程序设计方法实际上是面向对象的程序设计，这是目前最先进的程序设计方法之一。Windows 编程所涉及的对象很多，主要有窗口、菜单、对话框等。

如前所述，窗口是屏幕上的一个矩形区域，窗口通过键盘或鼠标接受用户的输入，并将图形输出到显示屏幕上。

菜单是 Windows 应用程序接收用户输入的主要手段。一个菜单是一组可供用户选择和查看的命令列表，用户只要选择菜单项就能执行相应的功能。

对话框是一种特殊的窗口，它为用户显示更多的有关命令的信息，并能接收用户的输入信息。一个对话框可以包含一个或多个控制窗口。控制窗口也是一种特殊的窗口，又称子窗口控制。控制窗口的形式包括各种按钮、文本编辑框、列表框、组合框和滚动杆，可以供用户选择和输入。

用户在屏幕上看到这些对象，可直接与它们进行交互。交互的方法是按一下按钮、滚动一个滚动杆或选中一个菜单项等。其实，用户所执行的这些操作是通过发送消息来告诉程序的，消息是应用程序执行命令的信号。

在面向对象的程序设计中，对象表示代码和数据的集合。以窗口为例，窗口是一个对象，其代码是窗口函数(该函数处理窗口的各种消息)，而数据就是与窗口相关的消息。

3. 消息驱动的程序结构

Windows 消息提供了应用程序与应用程序之间和应用程序与 Windows 系统之间进行通信的手段。几乎每个 Windows 程序所做的事情都是为了响应发送到窗口函数的某条消息。在大多数应用程序中，很大一部分代码内容是用来处理这些消息的。

Windows 的这些消息由三部分组成：消息号、字参数和长参数。消息号由事先定义的消息名标识，字参数和长参数包含与消息号相关的值。例如，当光标在窗口用户区时，按下鼠标器的左按钮，Windows 就向该窗口发送一条 WM_LBUTTONDOWN 消息(宏定义是在 windows.h 中定义的)，其长参数含有光标的 X 坐标和 Y 坐标(分别在其低位字和高位字中)，而其字参数则包含指示各种虚拟键盘是否被按下的值。窗口函数接收到此消息后，就可以作出相应的

处理。Windows 应用程序创建的每一个窗口都有一个窗口函数，用以处理发送到该窗口的消息。窗口函数由 Windows 采用消息驱动的形式直接调用，而不是由应用程序显式调用的。窗口函数处理完消息后，又将控制返回给 Windows。

Windows 中有一个系统消息队列，简称系统队列。当开始执行一个 Windows 应用程序时，Windows 为该程序建立一个“消息队列”，称为应用程序队列。这个队列存放该程序可能创建的各种窗口的所有消息。程序中含有一段“消息循环”代码，用来从消息队列中检索这些消息，并把它们分发到相应的窗口函数中。有些消息不必放到消息队列中，而直接送给窗口函数。系统队列、应用程序队列、消息循环、窗口函数之间的关系如图 1-2 所示。

Windows 应用程序接收用户输入的方式也不同于 DOS 环境下的应用程序。在 DOS 环境下，程序通过显式地调用一个函数(如 getch())来读取键盘的输入，该函数一般要等待用户按下一个键后，再把相应的字符返回给程序。在 Windows 环境下，Windows 接收所有来自键盘、鼠标器等外设的输入，并把它们放在相应的应用程序的消息队列中。在应用程序需要获取输入时，只不过是简单地从消息队列中读取下一个输入的消息。

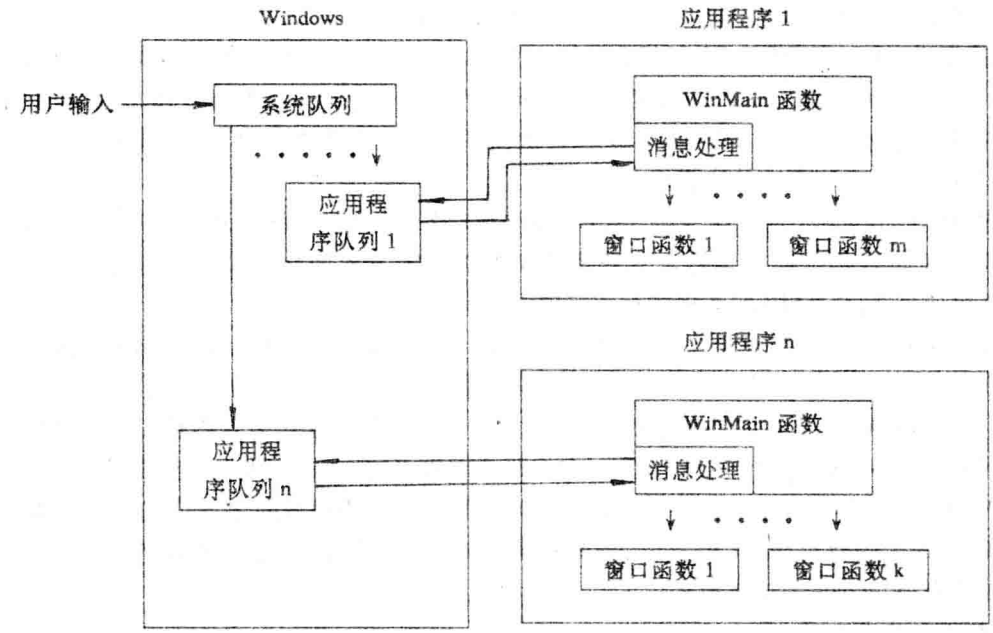


图 1-2　Windows 程序中的消息处理

4. 多任务

Windows 是一种多任务系统，可以同时显示和运行多个应用程序，每个应用程序有各自的窗口。用户可以在屏幕上移动这些窗口，改变它们的尺寸，在不同的程序间进行切换，并可以在应用程序之间进行数据交换。

标准 MS-DOS 环境下编写的应用程序一般都假定它独占计算机的所有资源。这些资源包括输入输出设备、内存、系统显示器、CPU 等。而在 Windows 环境下，应用程序必须与所有当前正在运行的其它应用程序共享这些资源。在程序设计时，要考虑多任务的特点，协调并共享资源，以避免耗尽资源。

Windows 的多任务是通过在程序之间进行切换来实现的，只有当正在运行的应用程序消息队列中没有消息(或只有 WM_PAINT 和 WM_TIMER 消息)时，才能转去执行其它等待消息的程序。因此，Windows 是一种非抢先的多任务。

5. 高效的内存管理

Windows 为保证多任务的运行，采取了一系列措施来提高内存的利用率，从而保证能以较少的内存资源运行较大的程序。Windows 内存管理的特点是：

(1) 当 Windows 运行同一应用程序的多个拷贝(每个拷贝称为一个实例)时，每个实例使用相同的代码段和相同的资源；

(2) 在 Windows 环境中分配的内存块，多数是可移动的，从而便于 Windows 对内存块的管理和提高内存资源利用率；

(3) 代码段和程序资源通常都是视需要而被装入内存的，而且在多数情况下分配为可丢弃的内存块。在内存资源紧张时，这种内存块可以被丢弃而释放所占内存空间。当再次需要时，Windows 能自动地从磁盘上复制相应的内容到内存中。

Windows 突破了 MS-DOS 对内存 640KB 的限制，提供了实模式、标准模式和 386 增强模式等运行模式。标准模式和 386 增强模式统称为保护模式。

在基于 Intel 8086CPU(或 80286，80386 以及以上的 CPU，内存少于 1MB)的机器上，Windows 3.0 将以实模式运行。在这种模式下，Windows 系统以及 Windows 应用程序占用 640KB 常规内存的上部，而 MS-DOS、设备驱动程序以及内存驻留程序占用 640KB 常规内存的下部。Windows 3.1 不支持实模式。

在基于 80286 CPU、内存至少为 1MB，或 80386 及以上 CPU、内存为 1～2MB 的机器上，Windows 3.X 将以标准模式运行。这种模式又称为 286 兼容的保护模式。在这种模式下，Windows 可使用多达 16MB 的常规内存和扩展内存。

在基于 80386 CPU、内存至少为 2MB 的机器上，Windows 3.X 将以 386 增强模式运行。这种模式的主要特点是使用 CPU 的页寄存器实现虚拟内存管理。CPU 的内存页面长度规定为 4KB，Windows 可将内存页面交换到磁盘上，需要时再从磁盘上重新装入内存。页交换和重新装入的过程均由 Windows 系统自动进行，在编写应用程序时，不必考虑页交换的过程。

当 Windows 系统启动时，它根据机器的硬件配置自动选择适当的运行模式运行。用户也可以从 DOS 命令中使用/R(实模式)，/2(标准模式)，/3(386 增强模式)参数来启动 Windows 系统，以改变 Windows 的缺省模式。

6. 数据交换与共享

Windows 提供了多种手段来实现应用程序之间的数据交换与共享，包括剪贴板(Clipboard)、动态数据交换(DDE)和动态连接库(DLL)。

剪贴板是共享内存块的管理器，应用程序既可以向其中写入数据，也可以从中读出数据。利用剪贴板，能实现应用程序内部或应用程序之间的数据交换。

动态数据交换是一种更高级的数据交换手段，也是一种数据交换的消息协议。通过建立应用程序之间的数据连接，使得应用程序之间可以进行自动的数据传送，而不需要用户干预。这种功能对于开发需要连接实时数据的应用程序具有十分重要的意义。

动态连接库是应用程序之间实现代码和资源共享的一种手段。在 Windows 环境下，由于可以同时执行多个任务，使用 DLL 比使用静态连接库有更多的优点。如果两个应用程序同时

运行，且它们使用了某一静态库中的同一函数，那么系统中就会出现该函数的两个副本，而DLL能允许若干个应用程序共享某个函数的单个副本，从而节省了内存空间。此外，DLL还可以用来实现其它资源的共享，如数据和硬件资源的共享。

Windows所有的运行库均是DLL，如GDI.EXE，USER.EXE等，它们是Windows的主要构成部分之一。标准的Windows设备驱动程序也是用DLL实现的。程序员也可以根据需要，开发出自己的DLL。

7. 与设备无关的图形接口

Windows提供了丰富的、与设备无关的图形处理功能。应用程序能很方便地画出各种图形，而不需要直接与具体的输出设备打交道。由于Windows提供了设备无关性，因此，在应用程序中可以使用同一函数显示或在打印机上输出同一图形。

Windows应用程序并不直接访问图形显示设备，而是通过其图形程序设计语言(图形设备接口，即GDI)来实现图形输出的。Windows应用程序的输出可以适用于任何显示或打印设备，只要在Windows环境下安装相应的设备驱动程序即可，由设备驱动程序将GDI图形输出请求转换为显示器、打印机等输出设备上的图形输出。

1.2 Windows/Visual C++编程环境

1.2.1 开发环境

从开发应用程序的角度看，我们建议最好使用如下的硬件环境：

1. 80386或以上的处理器；
2. 8MB以上的内存空间，200MB以上的硬盘；
3. VGA以上的显示卡和相应的彩色监视器；
4. 一个Microsoft或兼容鼠标器。

软件环境建议用：

1. MS-DOS 5.0以上；
2. Microsoft Visual C++ 1.0以上；
3. Windows 3.1。

1.2.2 Visual C++ Workbench编程环境

Visual C++编译器所支持的最重要的工具就是程序员工作平台。Microsoft习惯上把它称作Visual Workbench。Visual Workbench是一个集成开发环境，共有10个选项菜单，每个菜单包含一系列选择，这些选择执行某项任务或设置一个选项。

在此环境下运行应用程序非常方便。用户首先进入Windows环境，如图1-3所示。

用鼠标双击Visual C++图标(或光标在Visual C++上时按ENTER键)，将看到图1-4所示的Visual C++的Workbench集成环境。

1. File项

当弹出Visual C++ Workbench的File菜单时，将看到如图1-5所示的选项。File项基本可以用来装载和保存文件、打印及退出Workbench。本项包括以下几个选择：

图 1-3 包含 Visual C++程序组的 Windows 界面

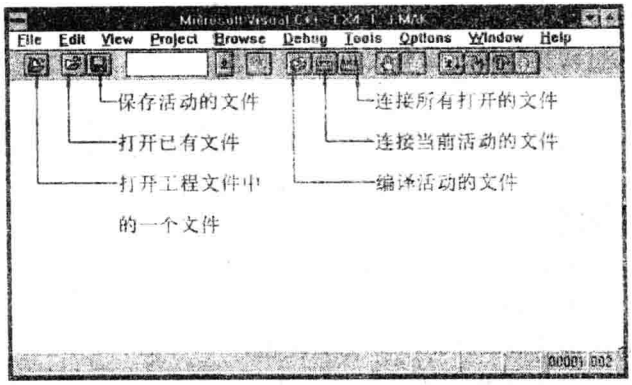


图 1-4 Visual Workbench 编辑环境

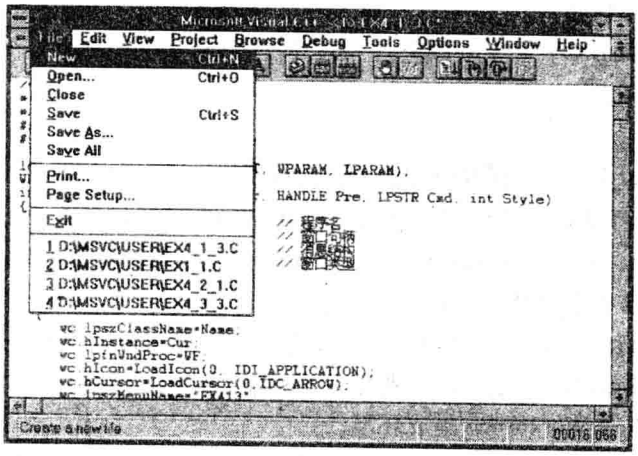


图 1-5 Visual Workbench File 菜单命令

(1) New 选择

New 选择用于打开一个新的编辑窗口。一般是在这一点上开始设计应用程序。

(2) Open 选择

Open 选择打开一个已经存在的文件。这两项选择文件都要指定目录。

(3) Close 选择

Close 选择用于关闭一个已打开的文件。

(4) Save 选择

Save 选择把当前被选的内容或活动窗口存储到指定的文件中去。

(5) Save As 选择

Save As 选择用于用于保存一个新的窗口或将当前编辑的内容写到一个不同的文件中去。

(6) Save All 选择

Save All 选择快速保存 Workbench 中的所有窗口。

(7) Print 选择

Print 选择用来打印全部或部分当前被编辑的窗口。

(8) Page Setup 选择

Page Setup 选择激活一个对话框，用来指定页的头尾和文件四周的页边空。

(9) Exit 选择

Exit 选择退出 Visual C++ Workbench。

(10) Most Recent Files 选择

Visual C++ Workbench 利用本选择保存操作过的最新的文件，Workbench 在 Exit 选择下边显示这些文件的名字。

2. Edit 项

Visual Workbench Edit 列出了如图 1-6 所示的常用命令。Edit 项中包含用于进行四类操作的选择：不做更改；删除和插入操作；寻找/替换操作；触发只读模式。

图 1-6 Visual Workbench Edit 菜单命令

(1) Undo 选择

Undo 选择允许你不做当前正在编辑的任务。放弃的编辑内容数量取决于 Undo 缓冲区的大小。缓冲区的大小范围是 0～64KB，缺省为 4K，可使用 Options 选择项中的 Editor 选择改变

Undo 缓冲区的大小。

(2) Redo 选择

Redo 选择不做由 Undo 选择所做的工作，如果你过多地使用了 Undo 选择，Redo 选择也只能支持你恢复部分被编辑的文件。

(3) Cut 选择

Cut 选择删除被选择的文本，并把这部分文本复制到剪贴板，被删除的文本可以使用 Paste 选择重新插入。

(4) Copy 选择

Copy 选择将被选择的文本复制到剪贴板，覆盖先前剪贴板的内容，随后可以使用 Paste 选择重新插入。

(5) Paste 选择

Paste 选择把剪贴板中的文本插入在当前光标所在的位置。使用 Copy 和 Paste 选择可以在一个或几个窗口中复制文本。

(6) Delete 选择

Delete 选择删除被选择的文本，被删除的文本不被拷贝到剪贴板上去。

(7) Find 选择

Find 选择可以在当前编辑的窗口中搜索文本。该选择项激活对话框，允许你指定如下任选项：

①搜索文本；

②搜索一个整字；

③区分大小写；

④使用规则表达式，搜索具有更高级模式的文本；

⑤搜索方向。

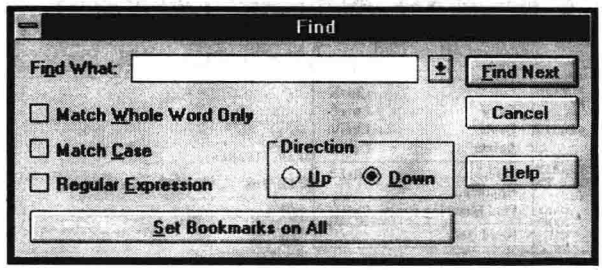


图 1-7 Find 对话框

(8) Replace 选择

Replace 选择替换在当前编辑窗口中的文本。该选择项激活对话框，允许你指定如下任选项：

①搜索文本；

②替换文本；

③搜索一个整字；

④区分大小写；

⑤使用规则表达式，搜索具有更高级模式的文本。

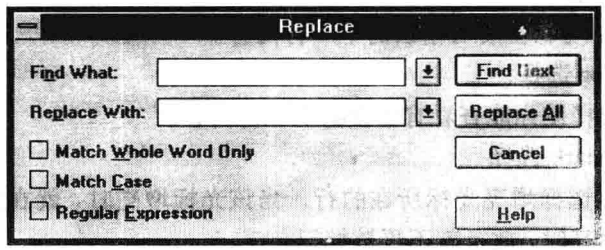


图 1-8　Replace 对话框

Replace 对话框将所有搜索到的文本用替换文本替代。对话框中还有一个 Find Next 选择，用于继续搜索文本。

(9) Find Match Brace 选择

Find Match Brace 选择能够寻找{和}的搭配。

(10) Read Only 选择

Read Only 选择将一个窗口中的文本设置成只读模式。当设置成只读模式时，在这一选择的左边，Workbench 设置一小方块标志。

菜单命令后跟三个点，表示屏幕上的对话框需要用户附加输入。

3. View 项

View 项使你能快速阅读各类消息，本选择项包含一些选择能显示一指定的行、错误信息、书标、工具棒和状态棒。图 1-9 是 Visual Workbench View 菜单命令。

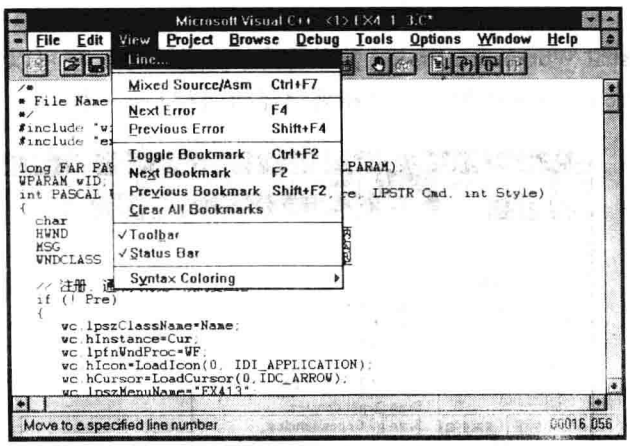


图 1-9　Visual Workbench View 菜单命令

(1) Line 选择

Line 选择激活一对话框，提示和接受你要查看的行号，选择可以准确地跳到被指定的那些行。

(2) Mixed Source/Asm 选择

Mixed Source/Asm 选择显示混有 C++语句的汇编代码。该选择是为高水平的程序员设