

第 1 章 Visual C++ 6.0 集成开发环境

Visual C++是微软公司的重要产品 Visual Studio 工具集的重要组成部分,它提供了一个集源程序编辑、代码编译和调试于一体的可视化开发环境,即所谓的集成开发环境(Integrated Development Environment, IDE),它包含文本编辑器、资源编辑器、工程编译工具、源代码浏览器、集成调试工具,以及一套联机文档。开发环境是程序员同 Visual C++的交互界面,通过它,程序员可以完成创建、调试、修改应用程序等的各种操作,对于集成开发环境的熟悉程度直接影响到程序设计的效率。集成开发环境采用标准的多窗口 Windows 用户界面,并增加了一些新特性,使得开发环境更易于使用,用户将很容易地学会它的使用方法。

Visual C++ 6.0 是微软公司推出的 Visual C++开发工具的 6.0 版本,它不仅支持 Internet 特性,还增加了 ADO(ActiveX Data)数据绑定、ATL(Active Template Library,活动模板库)复合控件、编辑和继续特性等,开发者可以充分利用 Visual C++ 6.0 的这些特性,构建自己的 Windows 应用程序。

由于 Visual C++是一个可视化的开发工具,在介绍 Visual C++ 6.0 的各个组成部分之前,首先了解一下可视化编程的概念。可视化技术是当前发展迅速并引人注目的技术之一,它的特点是把原来抽象的数字、表格、功能逻辑等用直观的图形、图像形式表现出来。可视化编程,亦即可视化程序设计,以“所见即所得”的编程思想为原则,力图实现编程工作的可视化,即随时可以看到结果,程序与结果的调整同步。可视化编程是与传统的编程方式相比而言的,这里的“可视”,指的是无须编程,仅通过直观的操作方式即可完成界面的设计工作,是目前最好的 Windows 应用程序开发工具。这种可视化的编程方法易学易用,而且大大提高了工作效率。在 Visual C++ 6.0 中提供了大量的用于可视化编程的资源编辑工具。

1.1 Visual C++ 6.0 集成开发环境的启动

在计算机上安装完 Visual C++ 6.0 应用程序后,可以选择以下两种方式启动:

(1) 单击 Windows 的“开始”菜单,选择“程序”组下 Microsoft Visual Studio 6.0 子菜单下的 Microsoft Visual C++ 6.0,启动 Visual C++ 6.0。

(2) 单击 Windows 的“开始”菜单,选择“运行”,输入 msdev,即可启动 Visual C++ 6.0。

Visual C++ 6.0 启动后的界面如图 1.1 所示。

Visual C++ 6.0 是一个 Windows 应用程序,启动后显示一个标准的窗口,由标题栏、菜单栏、工具栏、工作区窗口、源代码编辑窗口、输出窗口和状态栏组成。

主窗口的左侧是项目工作区窗口,右侧是程序编辑窗口,下面是调试信息窗口。工作区窗口显示所设定的工作区的信息,程序编辑窗口用来输入和编辑源程序,调试信息窗口用来显示程序出错信息和结果有无错误(errors)或警告(warnings)。

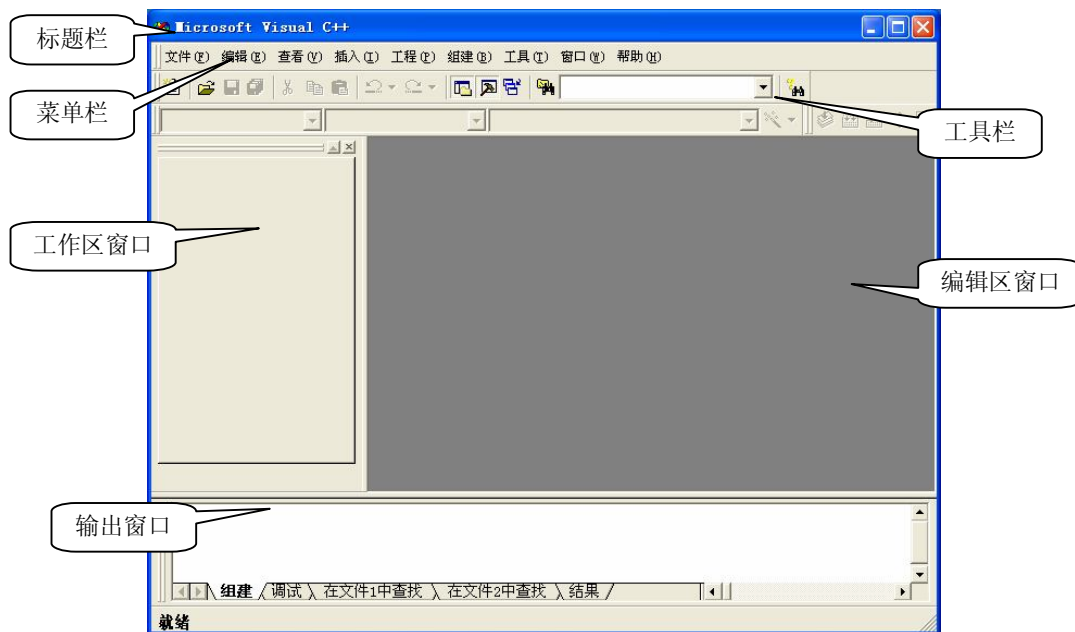


图 1.1 Visual C++ 6.0 启动界面

1.2 Visual C++ 6.0 集成开发环境的使用

1.2.1 菜单栏

在 Visual C++ 6.0 集成开发环境中，标题栏下方是菜单栏，由 9 个菜单项组成。单击菜单项弹出下拉式菜单，可使用这些菜单项实现集成环境的各种功能。菜单选项及其功能介绍如下。

(1) 文件 (File)。“文件”菜单包括对文件、项目、工作区及文档进行文件操作的相关命令或子菜单。

(2) 编辑 (Compile)。“编辑”菜单不仅包括常用的剪切、复制、粘贴命令，还包括为调试程序设置的“断点”命令，实现设置、删除、查看断点。此外，还有为方便程序员输入源代码的 List Members、Type Info 等命令。

(3) 查看 (View)。“查看”菜单中的命令主要用于改变窗口和工具栏的显示方式、检查源代码、激活调试时所用的各个窗口等。

(4) 插入 (Insert)。“插入”菜单包括创建新类、新表单、新资源以及新的 ATL 对象等命令。

(5) 工程 (Project)。使用“工程”菜单可以创建、修改和存储正在编辑的工程文件。

(6) 组建 (Build)。“组建”菜单用来编译、链接程序，创建及执行应用程序。

(7) 工具 (Tools)。“工具”菜单允许用户简单快速地访问多个不同的开发工具，如定制工具栏与菜单、激活常用的工具 (Spy++等) 或者更改选项等。

(8) 窗口 (Window)。该菜单的命令用来进行有关窗口的操作。

(9) 帮助 (Help)。与标准 Help 菜单相似，该菜单用来获得大量的帮助信息。

集成开发环境还为一些常用的命令分配了默认的快捷键，记住这些常用的快捷键，用户就可以更高效地使用集成开发环境。其中常用的快捷键如表 1-1 所示。

表 1-1 常用快捷键

快捷键	功能	快捷键	功能
Ctrl+O	打开文件 (File→Open)	F4	下一条错误
Ctrl+S	保存文件 (File→Save)	Shift+F4	上一条错误
Ctrl+C	编辑复制 (Edit→Copy)	Ctrl+F6	下一个窗口
Ctrl+T	编辑剪切 (Edit→Cut)	Alt+Enter	对象属性
Ctrl+V	编辑粘贴 (Edit→Paste)	F7	建立可执行文件
Ctrl+F	编辑查找 (Edit→Find)	Ctrl+F5	运行 (Execute)
Ctrl+Z	编辑取消 (Edit→Undo)		

1.2.2 工具栏

菜单栏下方是工具栏，它由若干个功能按钮组成，单击按钮可实现某种操作功能。工具栏也是 IDE 的重要组成部分，主要列出在代码开发过程中经常用到的一些功能，具有直观和快捷的特点，熟练使用这些工具按钮将大大提高工作效率。在 Visual C++ 6.0 开发环境中提供了丰富的工具栏，这些工具栏并不都显示在开发环境中，可以在工具栏上任意位置右击，在弹出的快捷菜单中选择要显示的工具栏，如图 1.2 所示。

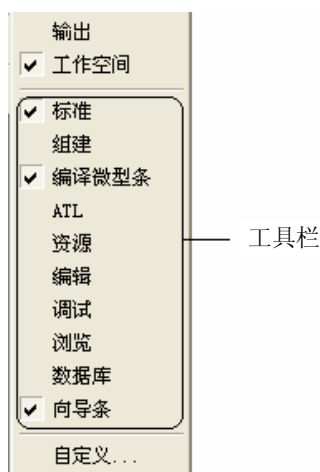


图 1.2 工具栏快捷菜单

下面介绍三个常用的工具栏。

1. 标准 (Standard) 工具栏

标准工具栏用于维护和编辑工作区的文本和文件，如图 1.3 所示。



图 1.3 标准工具栏

标准工具栏中各命令按钮的功能说明如表 1-2 所示。

表 1-2 标准工具栏中各命令按钮的功能说明

命令按钮	功能说明
	创建一个新的文件、项目和工作区
	打开一个已存在的文件、项目和工作区
	保存当前打开的文件
	保存所有打开的文件
	将所选择的内容剪切掉，移到剪贴板中
	将所选内容复制到剪贴板中
	在当前位置粘贴剪贴板中最新的内容
	取消上一次的操作
	恢复被取消的操作
	激活工作区窗口，用来管理工程中的文件和资源
	激活输出窗口，用来显示编译、调试和查找的信息
	管理当前打开的窗口
	在所有窗口中查找指定字符串
<input type="text" value="fg"/>	在当前窗口中查找指定字符串
	调用 MSDN

2. 编译微型条 (Build MiniBar) 工具栏







编译微型条工具栏用于运行程序和调试程序，如图 1.4 所示。



图 1.4 编译微型条工具栏

编译微型条工具栏中各命令按钮的功能说明如表 1-3 所示。

表 1-3 编译微型条工具栏中各命令按钮的功能说明

命令按钮	功能说明
	用于编译当前在源代码编辑窗口中的源文件
	用于编译、连接当前工程中的文件，生成一个可执行文件
	终止编译或连接的程序
	运行程序
	开始或继续调试程序
	编辑程序中的断点

3. 调试 (Debug) 工具栏

调试工具栏用于调试程序，如图 1.5 所示。

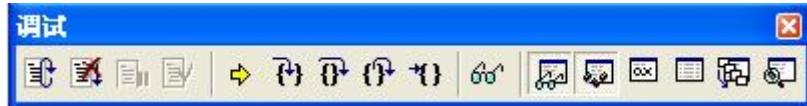


图 1.5 调试工具栏

调试工具栏中各命令按钮的功能说明如表 1-4 所示。

表 1-4 调试工具栏中各命令按钮的功能说明

命令按钮	功能说明
	放弃当前的调试，重新开始调试
	终止调试，返回到编辑状态
	暂停调试
	改编代码后调试
	显示将要运行的代码行
	单步执行程序，进入到函数内部
	单步执行程序，不进入到函数内部
	跳出当前函数
	运行到光标
	弹出 Quick Watch 窗口
	显示/隐藏 Watch 窗口
	显示/隐藏 Variables 窗口
	显示/隐藏 Registers 窗口
	显示/隐藏 Memory 窗口
	显示/隐藏 Call Stack 窗口
	显示/隐藏 Disassembly 窗口

1.2.3 项目和项目工作区

Visual C++应用程序的核心是项目 (Project, 也称“工程”)，它通常位于项目工作区 (Workspace) 中。Visual C++的项目工作区可以容纳多个项目。

1. 项目

在 Visual C++集成开发环境中，可以通过选择“文件”→“新建”菜单命令，打开“新建”对话框来创建一个新的项目。这个项目可以是一个游戏程序，或者是其他的应用程序。项目名是项目中其他文件命名的基础，它的后缀名为.dsp (Developer Studio Project)。

“新建”对话框的“工程”选项卡的内容如图 1.6 所示。



图 1.6 “新建”对话框的“工程”选项卡

可供选择的工程类型名称及含义如表 1-5 所示。

表 1-5 工程类型名称及含义

类型	含义
ATL COM AppWizard	ATL 应用程序创建向导
Cluster Resource Type Wizard	簇资源类型创建向导
Custom AppWizard	自定义的应用程序创建向导
Database Project	数据库工程
DevStudio Add-in Wizard	插件创建向导
Extended Stored Proc Wizard	扩展存储编程创建向导
ISAPI Extension Wizard	扩展创建向导
Makefile	C/C++生成文件
MFC ActiveX ContorWizard	MFC ActiveX 控制程序创建向导
MFC AppWizard(exe)	MFC 可执行程序创建向导
MFC AppWizard(dll)	MFC 动态链接库创建向导
New Database Wizard	新数据库创建向导
Utility Project	单元工程
Win32 Application	Win32 应用程序
Win32 Console Application	Win32 控制台应用程序
Win32 Dynamic-Link Library	Win32 动态链接库
Win32 Static Library	Win32 静态库

需要注意的是，要建立一个扩展名是.c 的程序文件，在“工程”选项卡中要选择 Win32 Console Application。Win32 Console Application 的入口函数是 main 函数。

2. 项目工作区

在创建一个项目的同时，也创建了一个项目工作区。项目工作区文件的后缀名为 .dsw (Developer Studio Workspace)。通过项目工作区窗口可以查看和访问项目的各种组件，它用于保存工作区的设置。项目工作区文件含有工作区的定义和项目中所包含文件的全部信息。

在创建一个项目的同时，Visual C++ 为这个项目创建了一个默认的项目工作区。

1.3 C 程序的运行步骤

1.3.1 编辑

启动 Visual C++ 6.0，如图 1.1 所示。

选择“文件”→“新建”菜单命令，出现如图 1.7 所示的“新建”对话框。选择“文件”选项卡，选择 C++ Source File 文件类型，输入程序名（程序的扩展名是.c，否则系统默认为 C++ 源程序的扩展名.cpp）及保存程序的目录。

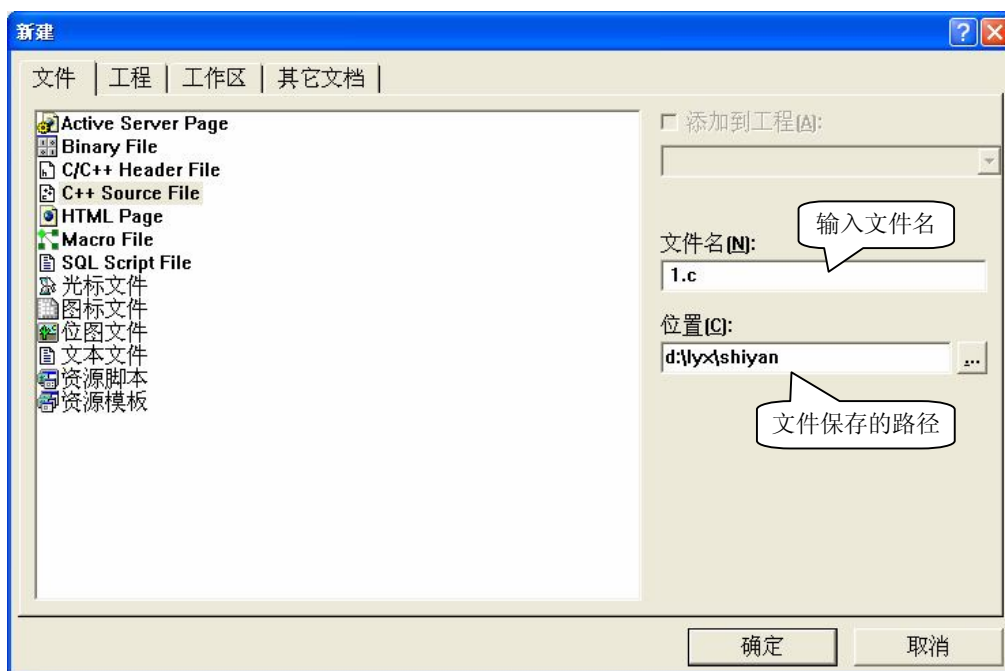


图 1.7 “新建”对话框

注意：输入文件名时要指定文件名的扩展名为.c，如果输入的文件名为*.cpp，则表示要建立的是 C++ 源程序。如果不写后缀，系统会默认指定为 C++ 源程序文件，自动加上扩展名.cpp，因此，编写 C 语言程序不能省略扩展名.c。

“文件”选项卡中显示出可创建的文件类型名称及含义，如表 1-6 所示。

表 1-6 文件类型名称及含义

类型	含义
Active Server Page	服务器页文件
Binary File	二进制文件
Bitmap File	位图文件
C/C++ Header File	C/C++头文件
C++Source File	C++源程序文件
Cursor File	光标文件
HTML Page	HTML 页文件
Icon File	图标文件
Macro File	宏文件
Resource Script	资源脚本文件
Resource Template	资源模板文件
SQL Script File	SQL 脚本文件
Text File	文本文件

单击“确定”按钮，出现如图 1.8 所示的 VC++ 6.0 编辑窗口，在此窗口中输入 C 程序，如图 1.8 所示。

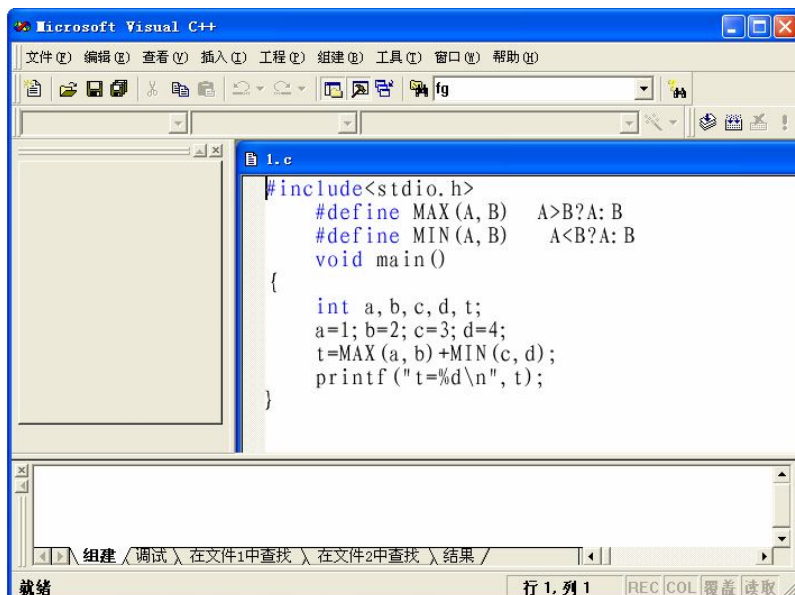


图 1.8 VC++ 6.0 编辑窗口

1.3.2 编译

在 VC++6.0 环境中，选择“组建”菜单下的“编译”菜单项，如图 1.9 所示。对 C 语言

的源程序进行编译，并生成扩展名为.OBJ 的目标文件。

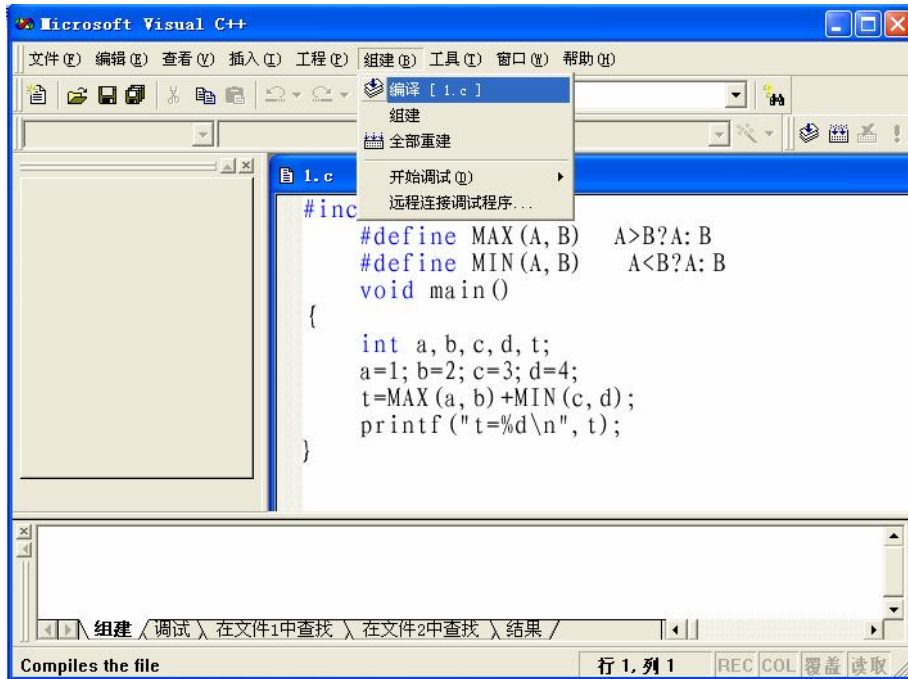


图 1.9 “编译”命令

编译时，在屏幕下方的调试信息窗口中将显示出编译的结果，指出有多少错误（error(s)）和警告性的错误（warning(s)）。可以根据显示的错误信息所在的行号尽快地找到错误语句，修改程序，然后再编译。

对程序进行“编译”还可以使用系统提供的“编译微型条”工具栏。在系统提供的“编译微型条”工具栏上，包括经常使用的“编译”、“连接”、“运行”等功能按钮，如图 1.10 所示。

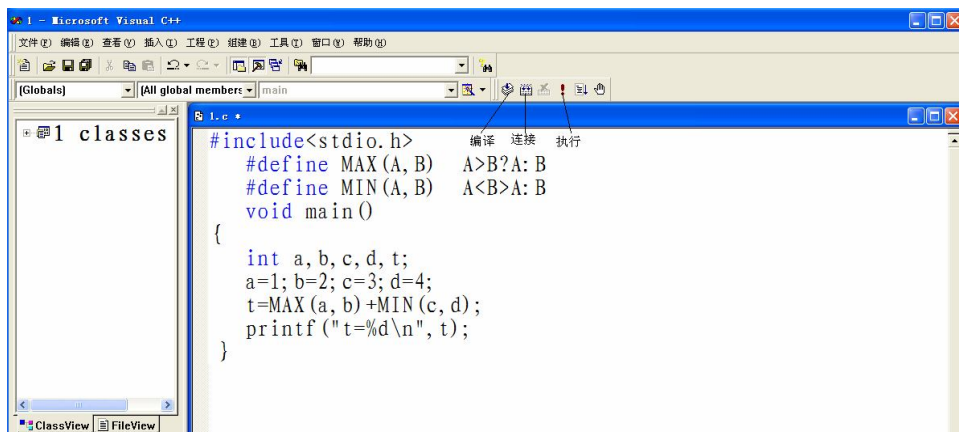


图 1.10 “编译微型条”工具栏

程序第一次编译会弹出一个窗口，要求创建一个默认的工作区，如图 1.11 所示。

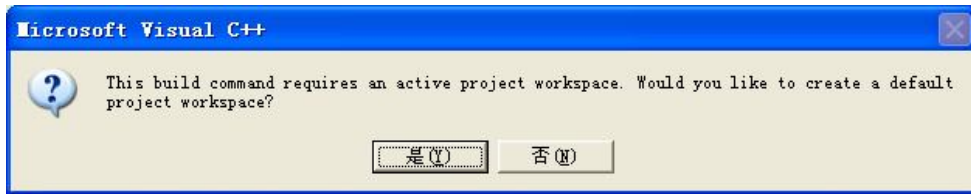


图 1.11 编译时弹出的窗口

提示：如果创建程序之前已经建立了项目工作区，则不会出现这个对话框。

单击“是”按钮，表示同意由系统建立默认的项目工作区。

对图 1.8 窗口中的程序进行编译后，在屏幕下方的调试信息窗口中将产生编译结果，如图 1.12 所示。

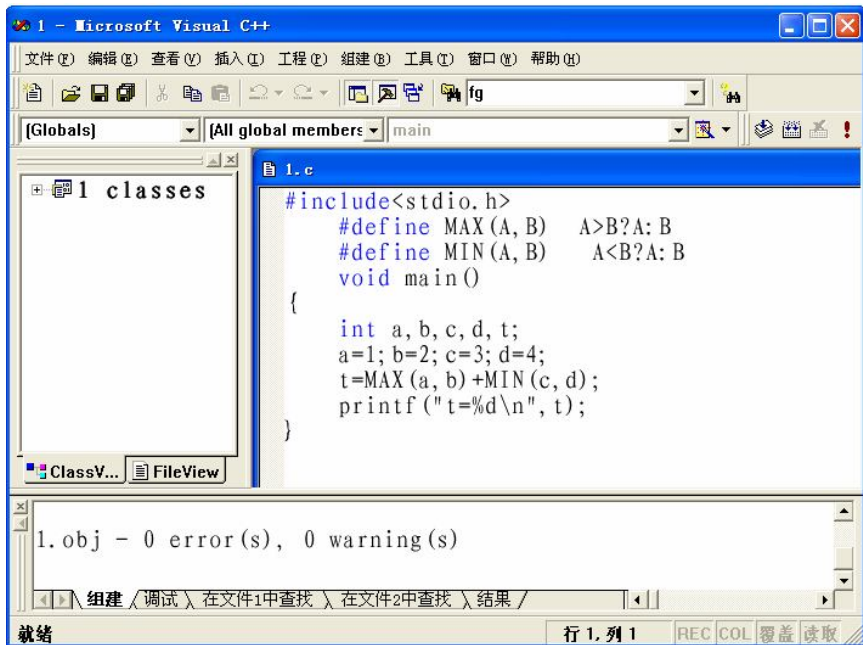



图 1.12 编译后的“调试”窗口


屏幕下面的调试信息窗口指出源程序有无错误，在此开始程序的调试，发现和改正程序中的错误，编译系统能检查程序中的语法错误。语法错误分为两类：一类是致命错误，以 **error** 表示，如果程序有这类错误，就通不过编译，无法形成目标程序，更谈不上运行了；另一类是轻微错误，以 **warning** 表示，这类错误不影响生成目标程序和可执行程序，但有可能影响运行的结果，也应当改正，使程序既无 **error(s)**，又无 **warning(s)**。

本结果说明源程序没有任何错误，已经通过了编译，可以进行“连接”了。

1.3.3 连接

在 VC++ 6.0 环境中，选择“组建 (Build)”菜单下的“组建 (Build)”菜单项，如图 1.9 所示，或使用“编译微型条”工具栏上的“连接”按钮，对 C 语言的目标文件进行连接，并生成扩展名为 .EXE 的可执行文件。

1.3.4 运行

在 VC++ 6.0 环境中，选择“组建 (Build)”菜单下的“执行 (Execute)”菜单项，如图 1.9 所示，或使用“编译微型条”工具栏上的“运行”按钮，运行程序。

1.3.5 关闭工作区

如果要继续编写下一个程序，必须关闭工作区。关闭工作区的方法为：选择“文件”菜单的“关闭工作区”菜单项，如图 1.13 所示。然后执行“文件”菜单下的“新建”菜单项，新建源程序。



图 1.13 “文件”菜单的“关闭工作区”命令