

第 1 章 C++程序的集成开发环境

运行 C++语言源程序需要相应编译系统的支持。C++的编译系统有许多，常用的系统都是集成开发环境，即源程序的输入、编辑修改、调试及运行都可以在同一环境下完成。这种集成开发环境功能齐全，操作方便。本章介绍 Visual C++ 6.0 和 Turbo C++ 3.0 两种流行的 C++语言程序集成开发环境，以方便读者掌握编辑、编译、连接和运行一个 C++程序。

1.1 Visual C++ 6.0 集成开发环境

Visual C++ 6.0（简称 VC 6.0）是 Microsoft 公司研制开发的基于 Windows 的可视化 C++ 语言版本，它是一个集 C++程序编辑、编译、连接、运行和调试等功能以及可视化软件开发为一体的开发工具。相比其他的编程工具而言，Visual C++在提供可视化的编程方法的同时，也适应编写直接对系统进行底层操作的程序。Visual C++提供的 Microsoft 基础类库（Microsoft Foundation Class Library, MFC）对 Windows 9X/NT 所用的 Win32 应用程序接口（Win32 Application Programming Interface）进行了十分彻底的封装，这使得 Windows 9X/NT 应用程序的开发可以使用完全的面向对象的方法进行，从而能够大大地缩短应用程序的开发周期，降低开发成本，也使得 Windows 程序员能从大量的、繁杂的劳动中解脱出来。

VC 6.0 是 Microsoft Visual Studio 套件的一个有机组成部分。Visual C++软件包含有许多单独的组件，例如编辑器、编译器、链接器、生成实用程序和调试器等，以及各种各样为开发 Windows 下的 C/C++程序而设计的工具。Visual Studio 把所有的 Visual C++工具结合在一起，集成为一个整体，通过一个由窗口、对话框、菜单、工具栏、快捷键等组成的完整系统，可以观察和控制整个开发过程。

1.1.1 Visual C++ 6.0 的安装与启动

在使用 VC 6.0 之前，首先要安装 VC 6.0。VC 6.0 的安装方法和其他 Windows 应用程序的安装方法类似。将 VC 6.0 系统安装盘放入光驱，一般情况下系统能自动运行安装程序；否则运行安装盘中的 setup.exe 文件。启动安装程序后，根据屏幕提示依次回答有关内容，便可完成系统安装。

启动 VC 6.0 的过程十分简单。常用的方法是，在 Windows 桌面选择“开始 | 程序 | Microsoft Visual Studio 6.0 | Microsoft Visual C++ 6.0”命令，即可启动 VC 6.0，屏幕上将显示如图 1.1 所示的 VC 6.0 主窗口。

本章开始使用中文版 Visual C++ 6.0 系统，这与主教材中使用的英文版系统一样，均为 6.0 系统，只是窗口界面提示使用中文提示，这样做是方便读者对照中、英文提示信息。

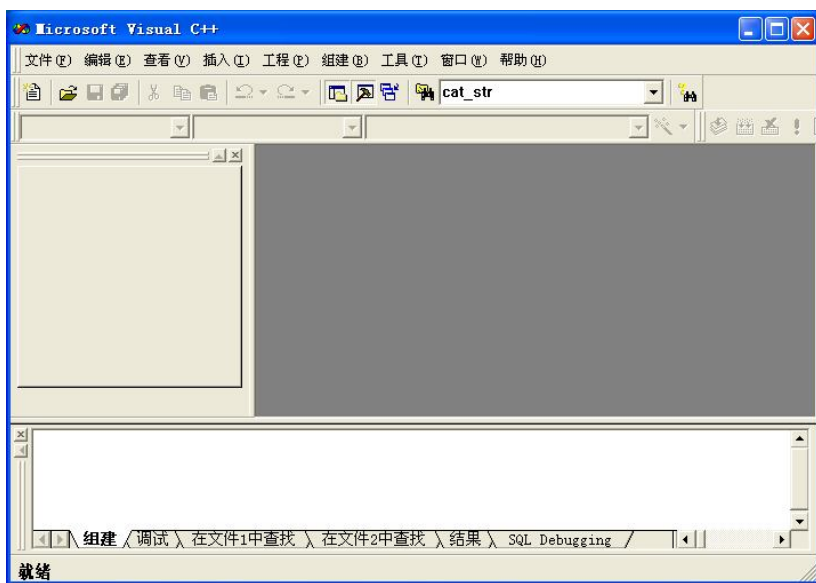


图 1.1 VC 6.0 主窗口

1.1.2 Visual C++ 6.0 主窗口的组成

和其他 Windows 应用程序一样, VC 6.0 主窗口也具有标题栏、菜单栏和工具栏。标题栏的内容是 Microsoft Visual C++。菜单栏提供了编辑、运行和调试 C/C++ 程序所需要的菜单命令。工具栏是一些菜单命令的快捷按钮, 单击工具栏上的按钮, 即可执行该按钮所代表的操作。

在 VC 6.0 主窗口的左侧是项目工作区 (Workspace) 窗口, 右侧是程序编辑窗口, 下方是输出 (Output) 窗口。项目工作区窗口用于显示所设定的工作区的信息, 程序编辑窗口用于输入和修改源程序, 输出窗口用于显示程序编译、运行和调试过程中出现的状态信息。

1. 菜单栏

VC 6.0 的菜单栏共有 9 个菜单项: 文件、编辑、查看、插入、工程、组建、工具、窗口和帮助, 每个菜单项都有下拉菜单, 用鼠标单击菜单项可弹出其下拉菜单, 下拉菜单中的每个菜单命令执行不同的功能。下面对各菜单项进行详细介绍。

(1) “文件”菜单。“文件”菜单中包含用于对文件进行各种操作的命令选项, 其快捷键及功能如表 1.1 所示。

表 1.1 “文件”菜单命令

菜单命令	英文版命令	快捷键	功能说明
新建	New	Ctrl+N	创建一个新的文件、项目或工作区
打开	Open	Ctrl+O	打开一个已存在的文件
关闭	Close		关闭当前被打开的文件
打开工作空间	Open Workspace		打开一个已存在的工作区
保存工作空间	Save Workspace		保存当前被打开的工作区

续表

菜单命令	英文版命令	快捷键	功能说明
关闭工作空间	Close Workspace		关闭当前被打开的工作区
保存	Save	Ctrl+S	保存当前打开文件
另存为	Save As		将当前文件另存为一新的文件名
保存全部	Save All		保存所有打开的文件
页面设置	Page Setup		对页面的布局进行设置
打印	Print	Ctrl+P	打印当前被打开的文件
最近文件	Recent Files		最近使用的文件列表
最近工作空间	Recent Workspaces		最近使用的工作区
退出	Exit		退出集成开发环境

(2) “编辑”菜单。“编辑”菜单包含所有与文件编辑操作有关的命令选项，其快捷键及功能如表 1.2 所示。

表 1.2 “编辑”菜单命令

菜单命令	英文版命令	快捷键	功能说明
取消	Undo	Ctrl+Z	撤销上一次的操作
重做	Redo	Ctrl+Y	恢复被取消的操作
剪切	Cut	Ctrl+X	将所选中的内容剪切掉并送至剪贴板中
复制	Copy	Ctrl+C	将所选内容复制至剪贴板中
粘贴	Paste	Ctrl+V	将当前剪贴板中的内容粘贴到当前插入点
删除	Delete	Del	删去选中的内容
选择全部	Select All	Ctrl+A	选定当前窗口中的全部内容
查找	Find	Ctrl+F	查找指定的字符串
在文件中查找	Find in Files		在多个文件中查找指定字符串
替换	Replace	Ctrl+H	替换指定字符串
转到	Go To	Ctrl+G	光标自动转移到指定位置
书签	Bookmarks	Alt+F2	设置书签或书签导航
高级	增加搜索	Ctrl+I	开始向前搜索
	格式选择内容	Alt+F8	对选中对象进行快速缩排
	制表符替换空格		在选中对象中用跳格代替空格
	空格替换制表符		在选中对象中用空格代替跳格
	大写选择内容	Ctrl+Shift+U	把选中部分改成大写
	小写选择内容	Ctrl+U	把选中部分改成小写
	a-b 查看制表符	Ctrl+Shift+8	显示或隐藏空格点
断点	Breakpoints	Alt+F9	设置程序中的断点

续表

菜单命令	英文版命令	快捷键	功能说明
列出成员	List Member	Ctrl+Alt+T	显示出全部关键字
类型信息	Type Info	Ctrl+T	显示变量、函数或方法的语法
参数信息	Parameter Info	Ctrl+Shift+Space	显示函数的参数
完成字词	Complete Word	Ctrl+Space	给出相关关键字的全称

(3) “查看”菜单。“查看”菜单中包含用于检查源代码和调试信息的各种命令选项，其快捷键及功能如表 1.3 所示。

表 1.3 “查看”菜单命令

菜单命令	英文版命令	快捷键	功能说明
建立类向导	Class Wizard	Ctrl+W	编辑应用程序的类
资源符号	Resource Symbols		浏览和编辑资源文件中的资源标识符 (ID 号)
资源包含	Resource Includes		编辑修改资源文件名及预处理命令
全屏显示	Full Screen		切换到全屏幕显示方式
工作空间	Workspace	Alt+0	激活项目工作区 (Workspace) 窗口
输出	Output	Alt+2	激活输出 (Output) 窗口
调试窗口	Watch	Alt+3	激活监视 (Watch) 窗口
	Call Stack	Alt+7	激活调用栈 (Call Stack) 窗口
	Memory	Alt+6	激活内存 (Memory) 窗口
	Variables	Alt+4	激活变量 (Variables) 窗口
	Registers	Alt+5	激活寄存器 (Registers) 窗口
	Disassembly	Alt+8	激活反汇编 (Disassembly) 窗口
更新	Refresh		更新选中区域
属性	Properties	Alt+Enter	打开源文件属性窗口

(4) “插入”菜单。使用“插入”菜单，用户可以向当前项目中插入新类、新资源等，其快捷键及功能如表 1.4 所示。

表 1.4 “插入”菜单命令

菜单命令	英文版命令	快捷键	功能说明
类	New Class		在项目中添加一个新类
窗体	New Form		在项目中添加一个新表单
资源	Resource	Ctrl+R	创建各种新资源
资源副本	Resource Copy		对选定的资源进行复制
作为文本文件	File As Text		可以将一个已存在的文件插入到当前焦点中
ATL 对象	New ATL Object		在项目中添加一个新的 ATL 对象

(5) “工程” 菜单。“工程” 菜单包含用于管理项目和工作区的一系列菜单项，其快捷键及功能如表 1.5 所示。

表 1.5 “工程” 菜单命令

菜单命令		英文版命令	快捷键	功能说明
设置活动工程		Set Active Project		选择指定项目为当前工作区中的活动项目
增加到工程	新建	New		在项目中增加新文件
	新建文件夹	New Folder		在项目中增加新文件夹
	文件	Files		在项目中插入已存在的文件
	数据连接	Data Connection		在当前项目中增加数据连接
		Components and Controls		在当前项目中插入一个组件或 ActiveX 控件
从属性		Dependencies		编辑项目组件
设置		Settings	Alt+F7	编译及调试的设置
导出制作文件		Export Makefile		以制作文件（.mak）形式输出可编译项目
插入工程到工作空间		Insert Project into Workspace		将项目插入到项目工作区窗口中

(6) “组建” 菜单。“组建” 菜单中包含的命令选项用于编译、创建、调试及执行应用程序，其快捷键及功能如表 1.6 所示。

表 1.6 组建菜单命令

菜单命令		英文版命令	快捷键	功能说明
编译		Compile	Ctrl+F7	编译当前编辑窗口中打开的文件
组建		Build	F7	生成一个可执行文件，即编译一个项目
全部重建		ReBuild All		编译和连接多个项目文件
批组建		Batch Build		一次编译和连接多个项目文件
清除		Clean		删除当前项目中所有中间文件及输出文件
开始调试 (Start Debug)	Go	Go	F5	开始或继续调试程序
	Step Into	Step Into	F11	单步运行调试
	Run to Cursor	Run to Cursor	Ctrl+F10	运行程序到光标所在处
	Attach to Process	Attach to Process		连接正在运行的进程
远程连接调试程序		Debugger Remote Connection		用于编辑远程调试连接设置
执行		Excute	Ctrl+F5	运行可执行文件
移除工程配置		Set Active Configuration		选择激活的项目及配置
配置		Configurations		编辑项目配置
配置文件		Profile		选中该菜单项，用户可以检查代码的执行情况

(7) “工具” 菜单。“工具” 菜单中包含 VC 6.0 中提供的各种工具，用户可以直接从菜

单中调用它们，其快捷键及功能如表 1.7 所示。

表 1.7 “工具”菜单命令

菜单命令	英文版命令	快捷键	功能说明
源浏览器	Source Browser	Alt+F12	浏览对指定对象的查询及相关信息
关闭源浏览器文件	Close Source Browser File		关闭信息浏览文件
	Visual Component Manager		激活组件管理器
	Register Control		激活注册控件
	Error Lookup		激活错误查找器
	ActiveX Control Text Container		激活 ActiveX 控件测试器
	OLE/COM Object Viewer		激活 OLE/COM 对象查看器
	Spy++		激活 Spy++ 工具包
	MFC Tracer		激活 MFC 跟踪器
定制	Customize		定制 Tool 菜单和工具栏
选项	Options		改变集成开发环境的各项设置
宏	Macro		创建和编辑宏
记录宏操作	Record Quick Macro	Ctrl+Shift+R	记录宏
播放宏操作	Play Quick Macro	Ctrl+Shift+P	运行宏

(8) “窗口”菜单。“窗口”菜单用于设置 VC 6.0 开发环境中窗口的属性，其快捷键及功能如表 1.8 所示。

表 1.8 “窗口”菜单命令

菜单命令	英文版命令	快捷键	功能说明
新建窗口	New Window		为当前文档打开一个新窗口
分割	Split		将窗口拆分为多个窗口
组合	Docking View	Alt+F6	启动或关闭 Docking View 模式
关闭	Close		关闭当前窗口
全部关闭	Close All		关闭所有打开的窗口
下一个	Next		激活下一个未停放的窗口
上一个	Previous		激活上一个未停放的窗口
层叠	Cascade		将所有打开的窗口重叠地排列
水平平铺	Tile Horizontally		将工作区中所有打开窗口按照横向平铺
垂直平铺	Tile Vertically		将工作区中所有打开窗口按照纵向平铺
窗口	Windows		管理当前打开的窗口

(9) “帮助”菜单。“帮助”菜单提供了详细的帮助信息，其快捷键及功能如表 1.9 所示。

表 1.9 “帮助” 菜单命令

菜单命令	英文版命令	快捷键	功能说明
内容	Contents		显示所有帮助信息的内容列表
搜索	Search		利用在线查询获得帮助信息
索引	Index		显示在线文件的索引
使用扩展帮助	Use Extension Help		开启或关闭 Extension Help
键盘设置	Keyboard Map		显示所有键盘命令
每日提示	Tip of the Day		显示 Tip of the Day
技术支持	Technical Support		显示 Visual Studio 的支持信息
网上微软	Microsoft on the Web		有关 Microsoft 的网站或网页
关于 Visual C++	About Visual C++		显示版本的有关信息

2. 工具栏

默认情况下，VC 6.0 提供了 11 个工具栏，但只显示 3 个工具栏。用户可以通过鼠标右击工具栏，在弹出的快捷菜单中选择需要显示的工具栏，如图 1.2 所示。在图 1.2 中（左边是中文版，右边是英文版），显示了系统提供的工具栏，其中带有复选标记的菜单项，表示在开发环境中显示的工具栏。用户可以通过单击菜单项来控制该工具栏是否显示。



图 1.2 工具栏弹出式菜单

实际上，工具栏中的命令按钮多数是与菜单栏中的菜单项对应的。这里只介绍在开发环境中显示的工具栏。对于其他工具栏，可以参考菜单操作。

(1) 标准工具栏 (Standard)。标准工具栏主要用来帮助用户维护和编辑在工作区中的文本和文件，如图 1.3 所示。



图 1.3 标准工具栏

标准工具栏对应的菜单项如表 1.10 所示。

表 1.10 标准工具栏中各功能按钮所对应的菜单项

名称	相应菜单项	名称	相应菜单项
新建 	文件 新建	重做 	编辑 重做
打开 	文件 打开	工作空间 	查看 工作空间
保存 	文件 保存	输出 	查看 输出
全部保存 	文件 全部保存	窗口列表 	查看 窗口列表
剪切 	编辑 剪切	在文件中查找 	编辑 在文件中查找
复制 	编辑 复制	查找 	编辑 查找
粘贴 	编辑 粘贴	cat_str 	
撤消 	编辑 撤消	搜索 	帮助 搜索

(2) 编译微型条工具栏 (Build MiniBar)。编译微型条工具栏是 Build 工具栏的子集。编译微型条工具栏如图 1.4 所示。



图 1.4 编译微型条工具栏

编译微型条工具栏按钮对应的菜单项如表 1.11 所示。

表 1.11 编译微型条工具栏中各功能按钮对应的菜单项

名称	相应菜单项	名称	相应菜单项
Compile 	组建 Compile	Build Execute 	组建 Execute
Bulid 	组建 组建	Go 	组建 Start Debug Go
Build Stop 	组建 Stop Bulid	Insert/Remove Breakpoint 	编辑 断点

(3) 向导条工具栏 (WizardBar)。向导条工具栏主要用于向类中添加成员变量、方法，以及查看类中某个方法的声明和定义。向导条工具栏如图 1.5 所示。

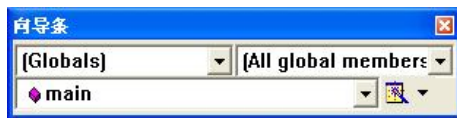


图 1.5 向导条工具栏

向导条工具栏由 3 个下拉式列表框和 1 个 WizardBar Actions 控制按钮组成。第 1 个下拉式列表框中列出了当前项目中定义的类。如果第 1 个下拉式列表框中当前显示的类关联一个对话框，则第 2 个下拉式列表框中还包含该对话框中控件的 ID 号。第 3 个下拉式列表框中的内容会根据第 2 个下拉式列表框的内容变化。如果在第 2 个下拉式列表框中选择 All class members 选项，则第 3 个下拉式列表框中会显示当前类的所有成员及成员函数，当在第 3 个

下拉式列表框中选择某个成员函数时，代码编辑窗口会直接定位到该函数的定义处。如果在第 2 个下拉式列表框中选择一个控件 ID，则会在第 3 个下拉式列表框中会显示与该控件 ID 有关的方法。如果某个方法没有定义，则会弹出一个对话框要求用户添加该方法；否则，代码编辑窗口会直接定位到方法的定义处。

此外，在向导工具栏中还包含了一个 Actions 按钮，单击该按钮，会弹出一个快捷菜单，从中可以选择要执行的菜单命令。

(4) 工具栏的定制。VC 6.0 为用户提供了 11 个工具栏，用户还可以根据需要对工具栏进行一些自定义的设置。常用的方法是：

①鼠标右键单击工具栏，在弹出的快捷菜单中选择“自定义”菜单项，打开“定制”对话框，选择“工具栏”选项卡，如图 1.6 所示。



图 1.6 “定制”对话框

②若要显示某工具栏，用鼠标单击该栏名称前面的复选框，使之出现一个选中标志即可。同样的操作再做一次，即可使该工具栏从主窗口中隐藏。

③单击“新建”按钮，弹出 New Toolbar 对话框，在 Toolbar name 编辑框中输入工具栏名称，单击 OK 按钮，将新建一个工具栏。单击“删除”按钮可以删除建立的工具栏。

3. 项目工作区窗口

项目工作区窗口通常包括 3 个标签页面，即 ClassView、ResourceView 和 FileView，分别显示项目中的类信息、资源信息和文件信息。在窗口底端单击相应图标标签可在 3 个标签页面之间切换。

(1) ClassView 标签页。ClassView 标签页用来显示当前工作区中所有的类、结构体、全局的变量和函数。ClassView 标签页提供了项目中所有类、结构体和全局变量的层次列表，通过单击列表左侧小的加号(+)或减号(-)可以扩展或折叠列表。双击列表开头靠近文件夹或书本形状图标的文字，也可以扩展或折叠列表。

在层次列表的每个项目前面都有一个特殊的图标。如公有成员函数的图标是一个紫色菱形框，公有成员变量是一个青绿色图标。如果为私有的，则在相应的图标前加一把锁。当用户双击某成员函数名时，在程序编辑窗口将打开定义该函数的源代码。

用户在某一个列表项目名上单击鼠标右键时，将弹出一个菜单，从中可以选择要执行的命令。单击的项目名不同，菜单中的命令也就不同。

如图 1.7 所示是工程项目 ex 工作区的 ClassView 标签页，表明该项目只有一个 main() 函数，它是全局的。



图 1.7 项目工作区的 ClassView 标签页

(2) ResourceView 标签页。ResourceView 标签页在层次列表中列出了项目中用到的所有资源。任何图像、字符串值以及程序所需要的其他编程部件都可以作为资源使用。

Visual C++ 中可以创建的资源在资源视图标签页中都有自己的文件夹。例如，Icon 文件夹中包含的都是图标文件，Dialog 文件夹中包含所有的对话框资源。如果用户想要创建某个资源或者从磁盘导入某个资源，可以在资源视图标签页中用鼠标右击某一资源类型节点，在弹出的快捷菜单中选择 Insert 或者 Import 菜单项就可以了。

如果是 Win32 Console Application 程序，则没有 ResourceView 标签页。

(3) FileView 标签页。FileView 标签页用于管理项目中使用的文件。它根据文件的不同类型，将其放置在不同的节点下。例如，源文件 (.cpp) 被放置在 Source Files 节点下，头文件 (.h) 被放置在 Header Files 节点下。

在 FileView 标签页中，用户不仅可以把文件从一个文件夹移动到另一个文件夹中，也可以创建保存特定类型文件（根据其扩展名）的新文件夹。创建新文件夹的方法是：右击要添加新文件夹的文件夹或项目，然后在快捷菜单中选择“新建文件夹”菜单项，系统将显示“新建文件夹”对话框，输入文件夹的名称以及相应的文件扩展名，然后单击“确定”按钮完成创建过程。双击显示在 FileView 标签页中的文件名，即可以编辑该文件。

如图 1.8 所示是工程项目 ex 工作区的 FileView 标签页，从中可以看到该项目只有一个 ex.cpp 文件，这是 C++ 源程序文件。



图 1.8 项目工作区的 FileView 标签页

4. 程序编辑窗口

VC 6.0 提供的程序编辑窗口是一个功能齐全的文本编辑器，可用于编辑 C/C++ 头文件、C/C++ 程序文件、Text 文本文件和 HTML 文件等。当打开或建立上述类型的文件时，该编辑器将自动打开。VC 6.0 编译器除了具有复制、查找、替换等一般文本编辑器的功能外，还具有很多特色功能，如根据 C++ 语法将不同元素按照不同颜色显示、根据合适长度自动缩进等。

程序编辑器还具备自动提示的功能。当用户输入程序代码时，代码编辑器会显示对应的成员函数和变量，用户可以在成员列表中选择需要的成员，减少了输入工作量，也避免了手动输入错误。

5. 输出窗口

输出窗口主要用于显示编译结果、调试结果以及文件的查找信息等，它共有 6 个标签，如表 1.12 所示。

表 1.12 输出窗口中标签的功能

标签名称	功能说明
组建	显示编译和连接结果
调试	显示调试信息
在文件 1 中查找	显示“编辑 在文件中查找”命令的查找结果。默认情况下，查找结果显示在“在文件 1 中查找”页面中，但“在文件中查找”对话框中有一个复选框，允许把结果显示在“在文件 2 中查找”页面
在文件 2 中查找	显示“编辑 在文件中查找”命令的查找结果
结果	显示 Profile 工具执行的结果。Profile 是一个辅助工具，能够显示编译程序的时间、线程等信息，这些信息显示在“结果”标签页中
SQL Debugging	显示 SQL 调试信息

1.1.3 Visual C++ 6.0 环境下 C++ 程序的运行

用 C++ 语言编写好一个程序后，要上机调试运行。第 1 步是上机编辑源程序文件，其扩展名为 .cpp；第 2 步是编译源程序文件生成目标文件，目标文件的文件名与源程序文件名相同，其扩展名为 .obj；第 3 步是运行连接程序，将目标文件与 C++ 的库文件相连接，生成可执行文件，可执行文件的文件名与源程序文件名相同，而扩展名为 .exe；第 4 步是运行可执行文件，实现程序所具有的功能。

第一次使用 VC 6.0 时，在进入开发环境之前，最好建立一个专用子目录。该子目录专门用于保存 C++ 源程序文件、目标文件、可执行文件以及系统自动生成的其他文件。

在计算机上安装好 VC 6.0 集成开发环境并启动该系统，然后就能应用该系统运行 C++ 程序了。

1. 创建一个工程

在 VC 6.0 主窗口的菜单栏中选择“文件 | 新建”命令，这时屏幕出现一个“新建”对话框，如图 1.9 所示。单击该对话框的“工程”标签，在“工程”选项卡中选择 Win32 Console Application，表示建立 Win32 控制台应用程序。在对话框右半部分的“位置”文本

框中指定文件存储路径（如 d:\mfc），在“工程名称”文本框中输入工程名称（如 ex）。同时选中“创建新的工作空间”单选按钮，表示在建立工程的同时建立项目工作区，单击“确定”按钮。



图 1.9 新建工程项目对话框

在弹出的“Win32 Console Application – 步骤 1 共 1 步”对话框中选择“一个空工程”选项，表示建立一个空的工程，然后单击“完成”按钮，出现“新建工程信息”对话框，在确认项目建立的信息后，单击“确定”按钮，从而完成了一个工程的创建工作。

2. 新建或打开 C++源程序文件

新建的工程是空的，其中没有任何具体内容，下面要在新工程中建立一个 C++源程序文件。在 VC 6.0 主窗口的菜单栏中选择“文件 | 新建”命令，在出现的“新建”对话框中单击“文件”标签，选中 C++ Source File 项，表示要建立新的源程序，如图 1.10 所示。在对话框右半部分的“文件名”文本框中输入源程序文件的名称，如 ex1，其扩展名默认为.cpp。单击“确定”按钮，回到 VC 6.0 主窗口，可以在编辑窗口中输入或修改源程序。由于完全是 Windows 界面，可以使用鼠标操作，输入和修改都十分方便。



图 1.10 新建源文件对话框

在程序编辑窗口中输入源程序，输入修改完成后，选择“文件 | 保存”命令，保存已经输入的源程序文件。

如果源程序文件已经存在，可选择“文件 | 打开”命令，并在查找范围中找到正确的文件路径，打开指定的程序文件。

3. 源程序的编译

在 VC 6.0 主窗口菜单栏选择“组建 | 编译”命令，对源程序进行编译。在编译过程中，编译系统检查源程序中是否有语法错误，然后在输出窗口显示编译信息。如果程序没有语法错误，则生成目标文件 ex1.obj，并在输出窗口中显示：

ex1.obj - 0 error(s), 0 warning(s)

表示没有任何错误（见图 1.11）。

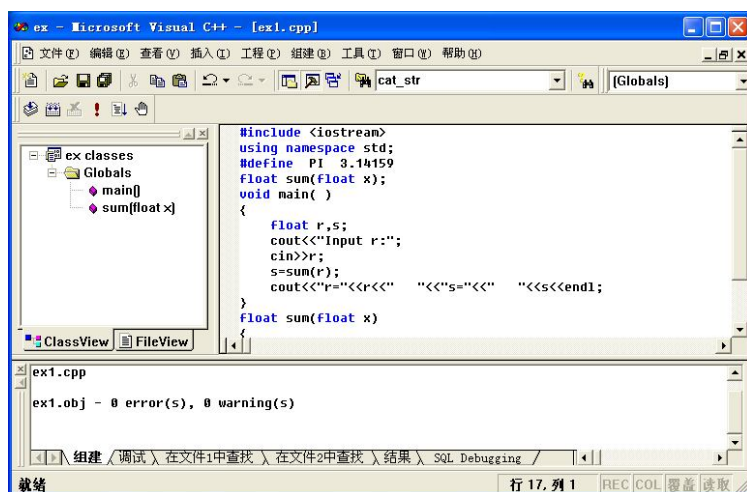


图 1.11 编译后程序无语法错误

编译一个源程序的过程是：首先处理所有预处理命令行，将预处理包含命令替换为头文件名所指定的头文件中的全部内容。在使用#include 命令时也可以包含扩展名为.cpp 的源程序文件。其次删除掉所有注释内容，用一个空格取代之。然后按行从上到下进行语法分析。最后生成相应的目标文件。

编译时检查出的错误分为两类：一类是严重的错误（error），又称为致命的错误，必须进行修复，否则不能进一步向下处理程序。另一类是警告性错误（warning），它不影响进入下一步处理过程。但最好修改程序，纠正警告性错误，使程序在编译后没有任何错误。

假如有致命性错误（error），则系统会指出错误的位置和信息，双击某行出错信息，程序窗口中会指示对应的出错位置，根据信息窗口的提示分别予以修改。为了“造”一个错误，将上面 ex1.cpp 中第 10 行“s=sum(r);”语句后面的分号去掉，重新编译，编译出错提示信息如图 1.12 所示。

要根据错误信息分析错误原因并找到错误位置，对源程序进行修改。修改后再编译，直至编译后状态输出窗口显示“0 error(s), 0 warning(s)”信息，表示没有任何错误。

当编辑输入多个程序文件和头文件时，它们的文件名将出现在菜单栏上的“窗口”菜单项的下拉式菜单的底部。选择其中任何一个文件名使它成为当前的编辑文件，即可对它进行

编辑和编译。

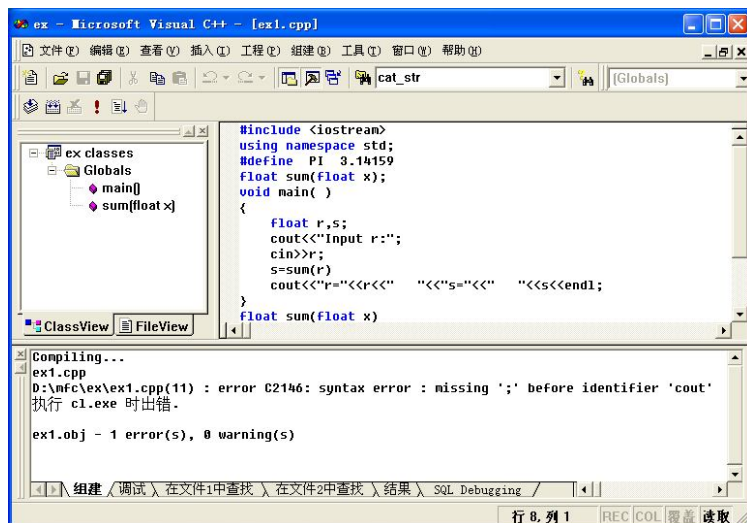


图 1.12 编译出错提示信息

4. 目标程序的连接

在生成目标程序后，还要把程序和系统提供的资源（如库函数、头文件等）连接起来，生成可执行文件后才能运行。此时在主窗口菜单栏选择“**组建** | **组建**”命令，表示要求连接并生成一个可执行文件 `ex.exe`。同样，在输出窗口会显示连接信息，如果有错，则要返回去修改源程序。修改后再重新进行编译和连接，直至没有连接错误为止。

以上分别介绍的是进行程序的编译和连接，也可以在主窗口菜单栏选择“**组建** | **组建**”命令（或按 **F7** 键）一次完成编译与连接。

5. 可执行文件的运行

在生成可执行文件后，就可以运行该文件了。在 **VC 6.0** 主窗口菜单栏选择“**组建** | **执行**”命令（或按 **Ctrl+F5** 键）运行可执行文件。程序运行时将自动打开一个输出窗口，并使显示光标处于该窗口的左上角位置。每次执行 `cout` 语句时，要输出的内容将在当前的光标位置上显示出来，然后显示光标自动后移；每次执行 `cin` 语句时，从键盘输入的内容也在当前的光标位置上显示出来，然后显示光标自动后移。程序运行结束，将在该窗口显示“**Press any key to continue**”提示信息，此时按下任意键后将关闭窗口，重新回到 **VC 6.0** 主窗口。

6. 关闭程序工作区

在完成一个 **C++** 程序（项目）后，若想执行第 2 个程序，必须选择“**文件** | **关闭工作空间**”命令关闭前一个程序的工作区，然后通过建立新的工程，产生第 2 个程序的工作区，否则运行的将一直是前一个程序。

1.2 Turbo C++ 3.0 集成开发环境

Turbo C 3.0（简称 **TC 3.0**）是美国 **Borland** 公司推出的一个集程序编辑、编译、连接、调试为一体的 **C++** 程序开发环境。它继承发展 **Turbo C 2.0** 的集成开发环境，并包含了面向

对象的基本思想和设计方法，具有方便、直观和易用的界面，支持鼠标操作。

1.2.1 Turbo C 3.0 的安装与启动

1. 系统安装

TC 3.0 的安装非常简单。在 Windows 中,启动 MS-DOS 方式,运行安装盘上的安装程序 install,然后按屏幕提示操作,即可完成安装过程。TC 3.0 的安装只是一个解压和复制的过程,而不修改系统参数,用户从一台装有 TC 3.0 的机器上将整个 TC 子目录复制到自己的机器上,也能达到在自己的机器上安装 TC 3.0 的目的。

2. 系统启动

TC 3.0 是基于 DOS 操作系统的应用程序，因此只能在 DOS 下运行。目前主流的操作系统已经从 DOS 发展到 Windows，DOS 系统已经很少使用。但是由于 Windows 系列操作系统充分向后兼容了 DOS 平台的应用软件，因此大部分的 DOS 系统下的应用软件可以在 Windows 下运行。TC 3.0 程序也不例外。下面介绍在 Windows 系列操作系统下 TC 3.0 的启动方法。

在 Windows 系列操作系统下，首先在 Windows 桌面选择“程序 | 附件 | 命令提示符”选项，进入 MS-DOS 命令提示符。在命令提示符下，通过 CD 命令进入 TC 3.0 的安装目录，输入 TC 命令启动 TC 3.0 集成环境。假定 TC 3.0 安装在 C 驱动器的 TC 目录下，需要输入命令：

C:\>CD \TC\BIN↵

C:\TC\BIN>TC ✓

进入 TC 3.0 集成开发环境中后，屏幕上显示如图 1.13 所示的 TC 3.0 主界面。

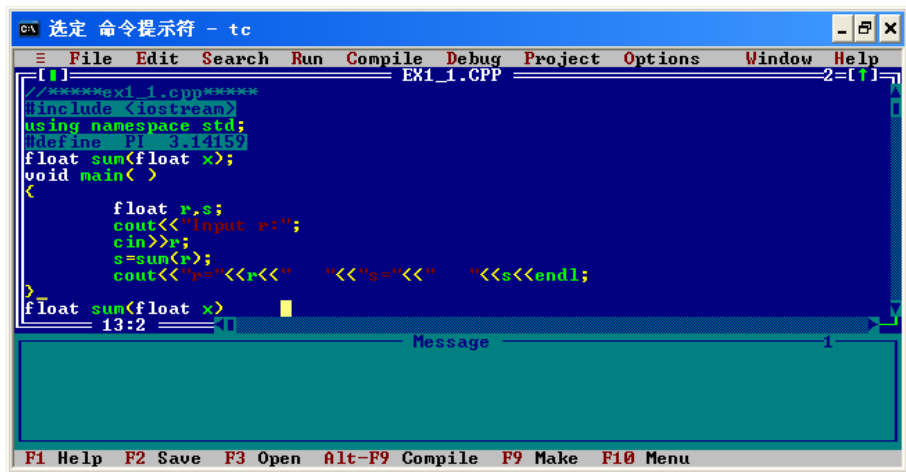


图 1.13 TC 3.0 主界面

3. 系统退出

退出 TC 3.0 的方法是选择 File | Quit 命令，或使用快捷键 Alt+X。在退出 TC 3.0 时，如果当前编辑的文件没有存盘，系统将提示如下信息：

ex1 1.cpp not Saved. Save?

输入 Yes 后，系统将提示输入文件名和所在的路径。

1.2.2 Turbo C 3.0 主界面的组成

TC 3.0 主界面的最上一行为 TC 3.0 系统菜单，中间窗口为编辑窗口，接下来是信息窗口，最下面一行为状态行。这 4 个部分构成了 TC 3.0 的主界面，以后的编程、编译、调试和运行都将在这个主界面中进行。

1. 系统菜单

TC 3.0 以系统菜单的形式提供一组与编辑、编译、调试以及运行相关的命令。TC 3.0 提供 10 个主菜单，分别为 File、Edit、Search、Run、Compile、Debug、Project、Options、Window 和 Help。每一主菜单均由一组菜单项或子菜单构成，每个菜单项都代表一个命令。

2. 编辑窗口

编辑窗口为 TC 3.0 的主窗口，在此窗口可以输入、编辑和调试 C 语言程序文件。编辑窗口由下列几部分组成：

(1) 标题条。标题条位于编辑窗口顶部，标题条上有窗口名、窗口号、窗口关闭图标和窗口缩放图标。窗口名显示所编辑的文件名，在该区域双击可以对该窗口进行缩放。在窗口关闭图标上单击可以关闭该窗口（也可以使用菜单命令 Window | Close）。在窗口缩放图标上单击可以放大或缩小该窗口（也可以使用菜单命令 Window | Zoom）。窗口号表示该窗口是第几号窗口，按 Alt+n 键可以将第 n 号窗口置为当前活动窗口。

(2) 滚动条。滚动条分为水平滚动条和垂直滚动条，在滚动条上单击可以使编辑窗口中的内容横向或纵向滚动。

(3) 行列号显示区。行列号显示区位于编辑窗口的左下角，用于显示被编辑文件的当前行列号。如果对被编辑文件作了修改，行列号的左边会出现一个星号（*）。

(4) 窗口角。编辑窗口的 4 个角都是窗口角，通过用鼠标拖曳窗口四角中的任何一个均可以任意改变编辑窗口的大小（也可以使用菜单命令 Window | Size/Move）。

在编辑状态，闪烁的“_”代表“插入点”，表示当前输入文字将出现的位置。另外，通过改变选中文本的颜色表示选中的文本。

3. 信息窗口

此窗口为在编译和调试程序时的信息输出窗口。在编译或连接时输出编译或连接结果，在调试程序的情况下，此窗口又称为跟踪窗口，用于显示跟踪表达式或变量的当前值。编辑窗口与信息窗口切换通过功能键 F6 来实现。

4. 状态行

状态行为用户提供了最基本的帮助信息，以及当前的键盘状态。例如，提示按功能键 F1 可以进入联机帮助，按 F10 键可以进入系统菜单。状态行右端的 NUM 代表键盘上 Numlock 键的状态，显示 NUM 表示数字小键盘已经打开，否则表示没有打开。状态行右端的 CAPS 代表键盘上 Capslock 键的状态，显示 CAPS 表示默认为大写字母输入，否则表示默认为小写字母输入。

1.2.3 Turbo C 3.0 环境下 C++程序的运行

下面以具体的实例说明在 TC 3.0 集成环境中运行一个 C 语言程序的操作过程。

1. 建立或打开 C++源程序文件

(1) 建立 C++源程序文件。在每次启动 TC 3.0 后, TC 3.0 系统自动创建文件名为 NONAME00.CPP 的新文件。在编辑窗口内输入程序代码, 并保存到磁盘, 就可以建立一个 C++语言源程序文件。

另外, 用户可以在任何时候选择 File | New 命令创建一个新文件。按 F10 键, 选择主菜单中的 Edit 选项, 则光标进入编辑区, 在编辑区中输入程序。

输入修改完成后, 需要将程序存盘。TC 3.0 为新建立的文件指定一个名称为 NONAME00.CPP 的通用文件名, 首次保存文件时需要输入新的文件名, 并确定存放文件的目录。可以通过选择 File | Save 命令实现, 这时系统显示输入文件名的窗口。

如果希望将当前文件换一个新的文件名或将当前的文件保存到其他目录中, 需要使用 File | Save as 命令, 然后按系统提示输入新文件的目录和文件名称。

(2) 打开 C++源程序文件。选择 File | Open 命令, 系统将弹出输入文件名的对话框, 直接输入程序文件名, 则可以将此文件打开, 并显示在当前的编辑窗口中。另外程序文件名可以是包含文件所在路径的全名或简单的文件名, 如果是简单的文件名, 系统只是在当前的目录中查找并打开。如果输入的文件名是包含目录的全名, 则在指定的目录中查找并打开文件。

如果忘记了具体的文件名, 则可以通过输入通配符 “*” 或 “?” 进行查找, 其原理与 DOS 系统中文件查找类似。例如, 如果在文件名称处输入 C:\TC*.CPP, 系统显示 TC 目录下所有扩展名为 .cpp 的文件, 通过光标键选择需要的文件, 回车确认即可。

2. 源程序的编译

选择 Compile | Compile 命令或按 Alt+F9 键, 对程序进行编译, 以生成目标文件。如果有错, 在屏幕上显示该程序在编译过程中出现的错误。

如图 1.14 所示是对一个 C++程序进行编译后的界面。有 5 个语法错误 (Errors) 和一个警告错误 (Warnings), 需要回到编辑状态进行修改。

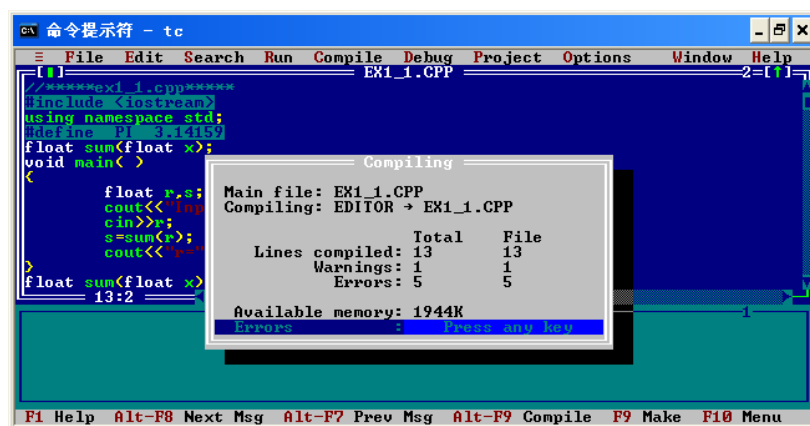


图 1.14 编译结果窗口

按回车键, 则回到编辑器中, 出现如图 1.15 所示的编辑窗口, 有一亮条出现在第 2 行, 信息窗第 2 行错误提示: 第 2 行在 TC 3.0 中应写为 “#include <iostream.h>”, 第 3 行内

容改为注释，即改为“//using namespace std;”。修改后再重新编译，直到不出错为止。

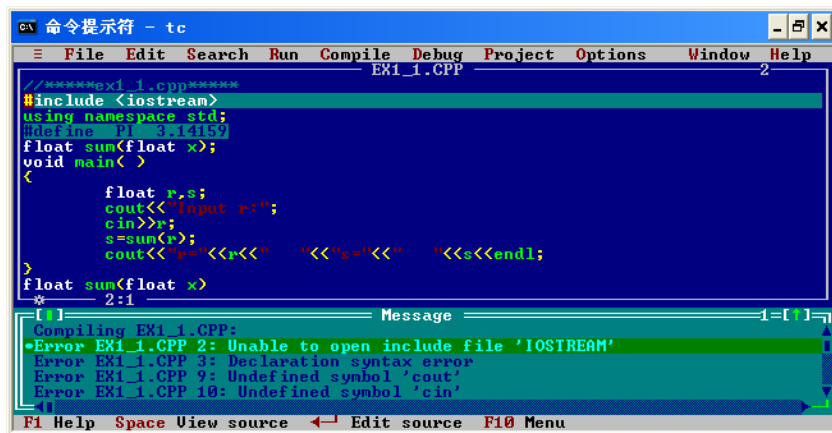


图 1.15 编辑窗口

注意：信息窗的提示为修改错误提供线索，但对错误的定性和定位不绝对准确。

3. 目标程序的连接

选择 Compile | Link 命令，将生成的.obj 文件和库文件进行连接，以生成可执行文件。在把目标文件与库文件连接时，若发现程序中有错误存在，需要返回编辑器进行修改，修改后再重新编译、连接，直到没错为止。

也可以选择 Compile | Make 命令一次完成编译和连接过程，如中途遇到错误也按上述方法处理。

4. 可执行程序的运行

最后，选择 Run | Run 命令，运行可执行文件。也可直接执行 Run | Run 命令或使用组合键 Ctrl+F9 一次完成程序的运行。如中途遇到错误也按上述方法处理。

选择 Window | User screen 命令或按 Alt+F5 键，在用户屏幕观察输出结果。如果结果不对，还要返回去修改。

注意：当程序进入死循环时，可通过按 Ctrl+Break 组合键退出程序运行状态。

1.2.4 系统设置

1. 当前工作目录设置

工作目录指 TC 3.0 用来存放输出结果文件的目录。编辑的源程序文件、OBJ 文件和 EXE 文件等均存于此目录中。

当用户进入 TC 3.0 集成开发环境后，当前目录可能不是用户所需的目录，这时需用户自己设置当前目录。其方法是：

按 F10 键进入主菜单，然后选择 File | Change dir 命令，屏幕上显示 Change Directory 对话框，其中显示的是默认的当前目录。用户可在此对话框中输入所需要的目录，则该新目录将成为当前目录。

注意：新目录应该是首先已建好的目录，如果输入的目录不存在，则屏幕会显示出错误信息：

Invalid drive or directory

2. 环境工作设置

若用户对环境的默认设置不满意或默认设置不符合要求时，则在编辑、编译及调试 C/C++ 语言程序之前，用户可对 TC 3.0 的工作环境进行设置。设置操作均在 Options 主菜单中进行。这些设置将影响到程序的编译、连接、库、包含目录等操作。

设置参数保存在配置文件 TCCONFIG.TC 中。配置文件是包含 TC 3.0 有关信息的文件，主要包括编译、连接的选择和路径等信息。

如果有需要，可以使用菜单命令 Options | Directories... 进入目录对话框，如图 1.16 所示，配置系统工作环境。



图 1.16 目录对话框

Include Directories 和 Library Directories 分别表示系统的头文件和标准库函数的目录位置，一般不要改变。Output Directories 和 Source Directories 分别表示系统的目标文件和源程序文件的存放目录，输入相应目录后单击 OK 按钮完成设置。