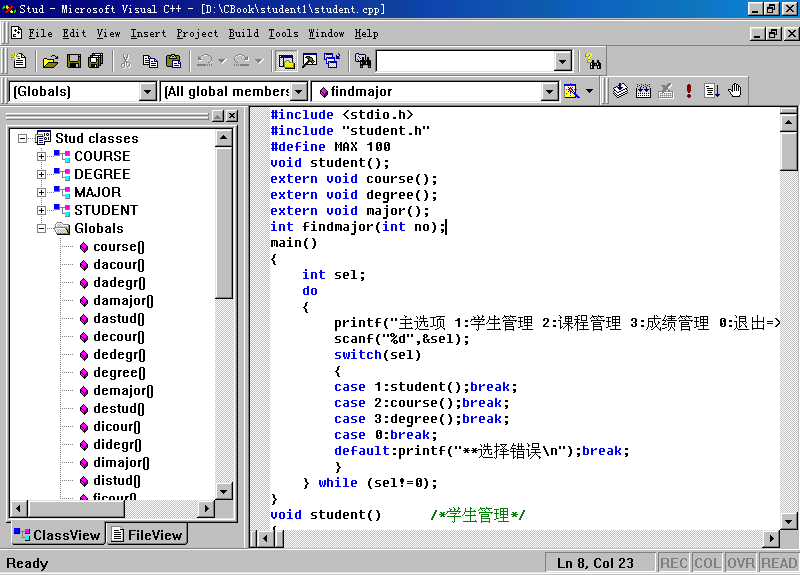
# VC++6.0系统开发环境

在Windows 98/2000/XP操作系统中正确安装了Visual C++ 6.0后，可以启动Visual C++6.0系统。第一次运行时，将显示出“Tip of the Day”对话框，单击“Next”命令按钮，可看到有关各种操作的提示；如果不选中“Show tips at startup”复选框，那么以后运行Visual C++ 6.0时将不再出现此对话框。单击“Close”命令按钮关闭此对话框，进入 Visual C++6.0系统开发环境。

## 1 开发环境的菜单功能

Visual C++6.0开发环境界面由标题栏、菜单栏、工具栏、项目工作区窗口、文档窗口、输出窗口以及状态栏等组成。在开发环境界面中，可以看到在它的上方排列着一系列菜单，如图 1所示，而每一个菜单下都有各自的菜单命令。在进一步与开发环境打交道之前，先了解各个菜单命令的基本功能是很有必要的，因为大部分的操作都是通过菜单来完成的。



文件 编辑 查看 插入 工程 组建 工具 窗口 帮助

图 1 VC++主菜单栏

#### 1. File莱单

File菜单中的命令主要用来对文件和项目进行操作，如“新建”、“打开”、“保存”、“打印”等。其中各项命令的功能描述如表 1所示。

表 1 File菜单命令的快捷键及功能描述

|  |  |  |
| --- | --- | --- |
| 菜单命令 | 快捷键 | 功能描述 |
| New新建 | Ctrl+N | 创建一个新项目或文件 |
| Open打开 | Ctrl+O | 打开已有的文件 |
| Close关闭 |  | 关闭当前被打开的文件 |
| Open Workspace打开工作空间 |  | 打开一个已有的项目 |
| Save Workspace保存工作空间 |  | 保存当前项目 |
| Close Workspace关闭工作空间 |  | 关闭当前项目 |
| Save 保存 | Ctrl+S | 保存当前文件 |
| Save As另存为 |  | 将当前文件用新文件名保存 |
| Save All保存全部 |  | 保存所有打开的文件 |
| Page Setup页面设置 |  | 文件打印的页设置 |

（续表）

|  |  |  |
| --- | --- | --- |
| 菜单命令 | 快捷键 | 功能描述 |
| Print打印 | Ctrl+P | 打印当前文件内容或选定的当前内容 |
| Recent Files最近文件 |  | 选择打开最近的文件 |
| Recent Workspace最近工作空间 |  | 选择打开最近的项目 |
| Exit 退出 |  | 退出 Visual C++6.0开发环境 |

#### 2. Edit菜单

Edit菜单中的命令用来使用户方便快捷地编辑文件内容，如进行删除、复制等操作，其中大多数命令功能与Windows中标准字处理程序的编辑命令一致，各项命令的快捷键及它们的功能描述如表 2所示。

表 2 Edit菜单命令的快捷键及功能描述

|  |  |  |
| --- | --- | --- |
| 菜单命令 | 快捷键 | 功能描述 |
| Undo取消 | Ctrl+Z | 撤销上一次操作 |
| Redo重做 | Ctrl+Y | 恢复被撤销的操作 |
| Cut剪切 | Ctrl+X | 将当前选定的内容剪切掉，并移至剪贴板中 |
| Copy复制 | Ctrl+C | 将当前选定的内容复制到剪贴板中 |
| Paste粘贴 | Ctrl+V | 将剪贴板中的内容粘贴到光标当前位置处 |
| Delete删除 | Del | 删除当前选定的对象或光标位置处的字符 |
| Select All选择全部 | Ctrl+A | 选定当前活动窗口中的全部内容 |
| Find查找 | Ctrl+F | 查找指定的字符串 |
| Find in Files在文件中查找 |  | 在指定的多个文件（夹）中查找字符串 |
| Replace替换 | Ctrl+H | 替换指定的字符串 |
| Go to转到 | Ctrl+G | 将光标移到指定位置处 |
| Bookmark书签 | Alt+F2 | 在光标当前位置处定义一个书签 |
| Advanced高级 |  | 其他一些编辑操作，如将指定内容进行大小写转换 |
| Breakpoints断点 | Alt+F9 | 在程序中设置断点 |
| List Members列出成员 | Ctrl+Alt+T | 显示“词语敏感器”的“成员列表”选项 |
| Type Info类型信息 | Ctrl+T | 显示“词语敏感器”的“类型信息”选项 |
| Parameter Info参数信息 | Ctrl+Shift+Space | 显示“词语敏感器”的“参数信息”选项 |
| Complete Word完成字词 | Ctrl+Space | 显示“词语敏感器”的“词语自动完成”选项 |

#### 3. View菜单

View菜单中的命令主要用来改变窗口和工具栏的显示方式，激活调试时所用的各个窗口等，其中各项命令的功能描述如表 3所示。

表 3 View菜单命令的快捷键及功能描述

|  |  |  |
| --- | --- | --- |
| 菜单命令 | 快捷键 | 功能描述 |
| Class Wizard建立类向导 | Ctrl+W | 弹出类编辑对话框 |
| Resource symbols资源符号 |  | 显示和编辑资源文件中的资源标识符（ID号） |
| Resource Includes资源包含 |  | 修改资源包含文件 |
| Full Screen全屏显示 |  | 切换到全屏显示方式 |
| Workspace工作空间 | Alt+0 | 显示并激活项目工作区窗口 |
| Output输出 | Alt+2 | 显示并激活输出窗口 |
| Debug Windows调试窗口 |  | 操作调试窗口 |
| Refresh更新 |  | 刷新当前选定对象的内容 |
| Properties属性 | Alt+Enter | 编辑当前选定对象的属性 |

#### 4. Insert菜单

Insert菜单中的命令主要用于创建和添加项目及资源，表 4列出了Insert菜单各项命令的快捷键及它们的功能。

表 4 Insert菜单命令的快捷键及功能描述

|  |  |  |
| --- | --- | --- |
| 菜单命令 | 快捷键 | 功能描述 |
| New Class类 |  | 插入一个新类 |
| New Form窗体 |  | 插入一个新的表单类 |
| Resource资源 | Ctrl+R | 插入指定类型的新资源 |
| Resource Copy资源副本 |  | 创建一个不同语言的资源副本 |
| File As Text作为文本文件 |  | 在当前光标位置处插入文本文件内容 |
| New ATL Object ATL对象 |  | 插入一个新的ATL对象 |

#### 5. Project菜单

Project菜单中的命令主要用于项目的一些操作，如向项目中添加源文件等。表 5列出了Project菜单各项命令的快捷键及它们的功能。

表 5 Project菜单命令的快捷键及功能描述

|  |  |  |
| --- | --- | --- |
| 菜单命令 | 快捷键 | 功能描述 |
| Set Active Project设置活动工程 |  | 激活指定的项目 |
| Add To Project增加到工程 |  | 将组件或外部的源文件添加在当前项目中 |
| Dependencies从属性 |  | 编辑当前项目的依赖关系 |
| Settings设置 | Alt+F7 | 修改当前编译和调试项目的一些设置 |
| Export Makefile导出制作文件 |  | 生成当前可编译项目的（.MAK）文件 |
| Insert Project into Workspace插入工程到工作空间 |  | 将项目加入到项目工作区中 |

#### 6. Build菜单

Build菜单中的命令主要用来编译、连接、调试和运行应用程序。表 6列出了Build菜单各项命令的快捷键及它们的功能。

表 6 Build菜单命令的快捷键及功能描述

|  |  |  |
| --- | --- | --- |
| 菜单命令 | 快捷键 | 功能描述 |
| Compile XXXX①编译 | Ctrl+F7 | 编译C或C++源代码文件 |
| Build XXXX.exe组建 | F7 | 生成应用程序的.EXE文件（编译、连接又称编连②） |
| Rebuild All全部组建 |  | 重新编连整个项目文件 |
| Batch Build批组建 |  | 成批编连多个项目文件 |
| Clean清除 |  | 清除所有编连过程中产生的文件 |
| Start Debug开始调试 |  | 给出调试的一些操作 |
| Debugger Remote Connection远程连接调试程序 | | 做远程调试连接的各项环境设置 |
| Execute XXXX.exe执行 | Ctrl+F5 | 执行应用程序 |
| Set Active Configuration移除工程配置 |  | 设置当前项目的配置 |
| Configuration配置 |  | 设置、修改项目的配置 |
| Profile配置文件 |  | 为当前应用程序设定各选项 |

注：① XXXX为当前编连的C++源代码主文件名。

② 生成应用程序的EXE文件过程是一个编译、连接过程，故把Build称为“编连”。

#### 7. Tools菜单

Tools菜单命令主要用于选择或制定开发环境中的一些实用工具，其中除了 Visual C++ 6.0的组件（如Spy++等）外，其余各项命令的快捷键及功能描述如表 7所示。

表 7 Tools菜单命令的快捷键及功能描述

|  |  |  |
| --- | --- | --- |
| 菜单命令 | 快捷键 | 功能描述 |
| Source Browser源浏览器 | Alt+F12 | 浏览对指定对象的查询及其相关信息 |
| Close Source Browser File关闭源浏览器文件 |  | 关闭浏览信息文件 |
| Customize定制 |  | 定制菜单及工具栏 |
| Options选项 |  | 改变开发环境的各种设置 |
| Macro宏 |  | 进行宏操作 |
| Record Quick Macro记录宏操作 | Ctrl+Shift+R | 录制新宏 |
| Play Quick Macro播放宏操作 | Ctrl+Shift+P | 运行新录制的宏 |

#### 8. Window菜单

Window菜单中的命令主要用于文档窗口的操作，如排列文档窗口、打开或关闭一个文档窗口、重组或切分文档窗口等。其中各项命令的快捷键及它们的功能描述如表 8所示。

表 8 Window菜单命令的快捷键及功能描述

|  |  |  |
| --- | --- | --- |
| 菜单命令 | 快捷键 | 功能描述 |
| New Window新建窗口 |  | 再打开一个文档窗口显示当前窗口内容 |
| Split分割 |  | 文档窗口切分 |
| Docking View组合 | Alt+F6 | 浮动显示项目工作区窗口 |
| Close关闭 |  | 关闭当前文档窗口 |
| Close All全部关闭 |  | 关闭所有打开过的文档窗口 |
| Next下一个 |  | 激活并显示下一个文档窗口 |
| Previous下一个 |  | 激活并显示上一个文档窗口 |
| Cascade层叠 |  | 层铺所有的文档窗口 |
| Tile Horizontally水平平铺 |  | 多个文档窗口上下依次排列 |
| Tile Vertically垂直平铺 |  | 多个文档窗口左右依次排列 |
| Window窗口 |  | 文档窗口操作 |

注：在打开多个窗口时，在Window菜单的底部列有所有打开的文档名称。

#### 9. Help菜单

Visual C++6.0同大多数其他的Windows应用软件一样，提供了大量详细的帮助信息，而Help菜单便是得到这些帮助信息最有效和最主要的途径。其中的各项功能如表 9所示。

表 9 Help菜单命令的功能描述

|  |  |
| --- | --- |
| 菜单命令 | 功能描述 |
| Contents内容 | 按“文件夹”方式显示帮助信息 |
| Search搜索 | 用查询方式获得帮助信息 |
| Index索引 | 按“索引”方式显示帮助信息 |
| Use Extension Help使用扩展帮助 | 选中此命令，按F1或其他帮助命令将显示外部的帮助信息；若此命令没有选中，则启动MSDN |
| Keyboard Map键盘设置 | 显示所有的键盘命令 |
| Tip of the Day每日提示 | 显示“每天一贴”对话框 |
| Technical Support技术支持 | 用微软技术支持的方式获得帮助 |
| Microsoft on the Web网上微软 | 微软网站 |
| About Visual C++关于Visual C++ | Visual C++的版本、注册等信息 |

注：像Contents、Search和Index等菜单项，只有在安装MSDN后才能使用。MSDN是微软提供的有关技术支持文档。

## 2 项目和项目工作区

一个Windows应用程序通常有许多源代码文件以及菜单、工具栏、对话框、图标等资源文件，这些文件都将纳入应用程序的项目中。通过对项目工作区的操作，可以显示、修改、添加、删除这些文件。项目工作区可以管理多个项目。

### 2.1 项目基本概念

在Windows环境下，大多数应用程序除了许多源代码文件外，还包含菜单、工具栏、对话框、图标等，Visual C++称它们为资源，这些资源通常用资源文件保存起来。另外，还要包含应用程序代码源文件编连时所需要的库文件、系统DLL文件等。有效组织这些文件并维护各源文件之间的依赖关系是应用程序最先要达到的目的，Visual C++中的项目就起这样的作用。实际上，项目作为工作区中的主要内容已加入集成开发环境中，不再需要自己来组织这些文件，只需要在开发环境中进行设置、编译、连接等操作，就可创建可执行的应用程序文件或DLL文件。

在Visual C++中，项目中所有的源文件都是采用文件夹方式进行管理的，它将项目名作为文件夹名，在此文件夹下包含源程序代码文件（.cpp，.h）、项目文件（.dsp）、项目工作区文件（.dsw）以及项目工作区配置文件（.opt），还有相应的Debug（调试）或Release（发行）、Res（资源）等子文件夹。

在开发环境中，Visual C++是通过左边的项目工作区窗口对项目进行各种管理。项目工作区窗口包含3页，它们分别是ClassView页、ResourceView页和FileView页。

注意

### 2.2 ClassView页

项目工作区窗口的ClassView页用以显示项目中的所有各类信息。假设打开的项目名为Stud，单击项目区窗口底部的ClassView，则显示出一个标题“Stud classes”的树状条目，在它的前面是一个图标和一个套在方框中的符号“+”，单击符号“+”或双击图标，Stud中的所有类名（包括结构体类型名）将被显示。例如，图 2所示是第13章上机实验题的对应项目，它有4个结构体类型名（如学生基本记录类型STUDENT等）和若干个函数（如学生基本记录输出函数distud()等）。

在ClassView页中，每个类名前也有一个图标和一个套在方框中的符号“+”，双击图标，则直接打开并显示类定义的头文件（如student.h）；单击符号“+”，则会显示该类中的成员函数和成员变量；双击成员函数前的图标，则在文档窗口中直接打开源文件并显示相应函数体代码。

这里要注意一些图标所表示的含义。例如，在成员函数的图标中，使用紫色方块表示公共成员函数（包括普通函数），使用紫色方块和一把钥匙表示私有成员函数，使用紫色方块和一把锁表示保护型成员函数；又如，用蓝绿色图标表示成员变量等。

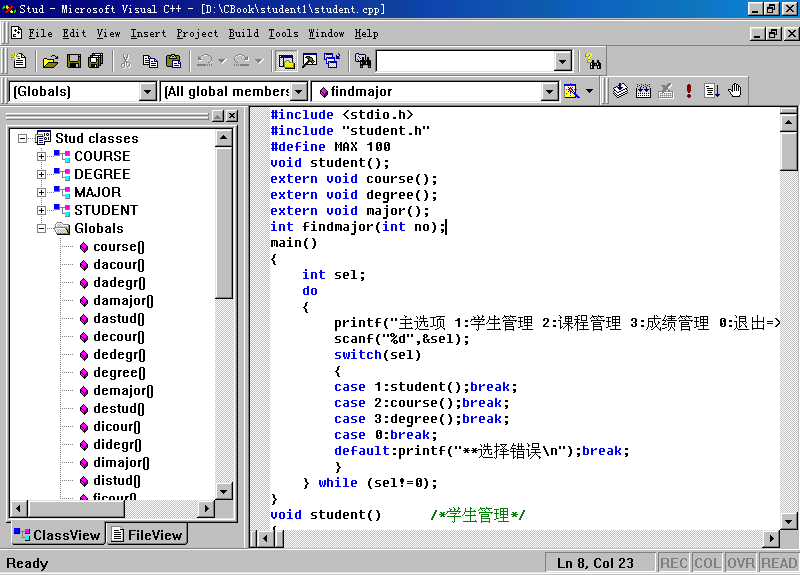


图 2 ClassView页

### 2.3 FileView页

FileView可将项目中的所有文件（C++源文件、头文件、资源文件、Help文件等）分类显示，如图 3所示。

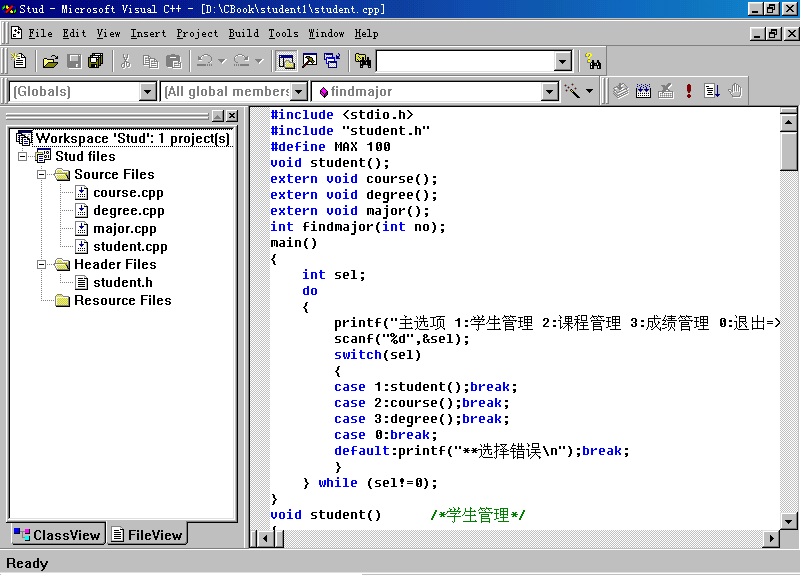


图 3 FileView页

每一类文件在FileView页中都有自己的目录项，例如所有的C++源文件都在 Source File目录项中。你不仅可以在目录项中移动文件，而且还可以创建新的目录项以及将一些特殊类型的文件放在该目录项中。

若创建一个新目录项，可在添加目录项的地方右击，弹出一个快捷菜单，从中选择“New Folder”，将出现如图 4所示的对话框，只要输入目录项名称和相关文件的扩展名，单击“OK”命令按钮即可。



图 4 “New Folder”对话框

在大型项目开发中，除了ClassView和FileView页外，还包括资源管理页，即ResourceView页，后者已超出C程序设计的内容，在这里不作介绍。

附录Ⅰ：**常见错误对照表**

|  |
| --- |
| fatal error C1003: error count exceeds number; stopping compilation  中文对照：（编译错误）错误太多，停止编译  分析：修改之前的错误，再次编译 |
| fatal error C1004: unexpected end of file found  中文对照：（编译错误）文件未结束  分析：一个函数或者一个结构定义缺少“}”、或者在一个函数调用或表达式中括号没有配对出现、或者注释符“/\*…\*/”不完整等 |
| fatal error C1083: Cannot open include file: 'xxx': No such file or directory  中文对照：（编译错误）无法打开头文件xxx：没有这个文件或路径  分析：头文件不存在、或者头文件拼写错误、或者文件为只读 |
| fatal error C1903: unable to recover from previous error(s); stopping compilation  中文对照：（编译错误）无法从之前的错误中恢复，停止编译  分析：引起错误的原因很多，建议先修改之前的错误 |
| error C2001: newline in constant  中文对照：（编译错误）常量中创建新行  分析：字符串常量多行书写 |
| error C2006: #include expected a filename, found 'identifier'  中文对照：（编译错误）#include命令中需要文件名  分析：一般是头文件未用一对双引号或尖括号括起来，例如“#include stdio.h” |
| error C2007: #define syntax  中文对照：（编译错误）#define语法错误  分析：例如“#define”后缺少宏名，例如“#define” |
| error C2008: 'xxx' : unexpected in macro definition  中文对照：（编译错误）宏定义时出现了意外的xxx  分析：宏定义时宏名与替换串之间应有空格，例如“#define TRUE"1"” |
| error C2009: reuse of macro formal 'identifier'  中文对照：（编译错误）带参宏的形式参数重复使用  分析：宏定义如有参数不能重名，例如“#define s(a,a) (a\*a)”中参数a重复 |
| error C2010: 'character' : unexpected in macro formal parameter list  中文对照：（编译错误）带参宏的形式参数表中出现未知字符  分析：例如“#define s(r|) r\*r”中参数多了一个字符‘|’ |
| error C2014: preprocessor command must start as first nonwhite space  中文对照：（编译错误）预处理命令前面只允许空格  分析：每一条预处理命令都应独占一行，不应出现其他非空格字符 |
| error C2015: too many characters in constant  中文对照：（编译错误）常量中包含多个字符  分析：字符型常量的单引号中只能有一个字符，或是以“\”开始的一个转义字符，例如“char error = 'error';” |
| error C2017: illegal escape sequence  中文对照：（编译错误）转义字符非法  分析：一般是转义字符位于 ' ' 或 " " 之外，例如“char error = ' '\n;” |
| error C2018: unknown character '0xhh'  中文对照：（编译错误）未知的字符0xhh  分析：一般是输入了中文标点符号，例如“char error = 'E'；”中“；”为中文标点符号 |
| error C2019: expected preprocessor directive, found 'character'  中文对照：（编译错误）期待预处理命令，但有无效字符  分析：一般是预处理命令的#号后误输入其他无效字符，例如“#!define TRUE 1” |
| error C2021: expected exponent value, not 'character'  中文对照：（编译错误）期待指数值，不能是字符  分析：一般是浮点数的指数表示形式有误，例如123.456E |
| error C2039: 'identifier1' : is not a member of 'identifier2'  中文对照：（编译错误）标识符1不是标识符2的成员  分析：程序错误地调用或引用结构体、共用体、类的成员 |
| error C2041: illegal digit 'x' for base 'n'  中文对照：（编译错误）对于n进制来说数字x非法  分析：一般是八进制或十六进制数表示错误，例如“int i = 081;”语句中数字‘8’不是八进制的基数 |
| error C2048: more than one default  中文对照：（编译错误）default语句多于一个  分析：switch语句中只能有一个default，删去多余的default |
| error C2050: switch expression not integral  中文对照：（编译错误）switch表达式不是整型的  分析：switch表达式必须是整型（或字符型），例如“switch ("a")”中表达式为字符串，这是非法的 |
| error C2051: case expression not constant  中文对照：（编译错误）case表达式不是常量  分析：case表达式应为常量表达式，例如“case "a"”中“"a"”为字符串，这是非法的 |
| error C2052: 'type' : illegal type for case expression  中文对照：（编译错误）case表达式类型非法  分析：case表达式必须是一个整型常量（包括字符型） |
| error C2057: expected constant expression  中文对照：（编译错误）期待常量表达式  分析：一般是定义数组时数组长度为变量，例如“int n=10; int a[n];”中n为变量，这是非法的 |
| error C2058: constant expression is not integral  中文对照：（编译错误）常量表达式不是整数  分析：一般是定义数组时数组长度不是整型常量 |
| error C2059: syntax error : 'xxx'  中文对照：（编译错误）‘xxx’语法错误  分析：引起错误的原因很多，可能多加或少加了符号xxx |
| error C2064: term does not evaluate to a function  中文对照：（编译错误）无法识别函数语言  分析：1、函数参数有误，表达式可能不正确，例如“sqrt(s(s-a)(s-b)(s-c));”中表达式不正确  2、变量与函数重名或该标识符不是函数，例如“int i,j; j=i();”中i不是函数 |
| error C2065: 'xxx' : undeclared identifier  中文对照：（编译错误）未定义的标识符xxx  分析：1、如果xxx为cout、cin、scanf、printf、sqrt等，则程序中包含头文件有误  2、未定义变量、数组、函数原型等，注意拼写错误或区分大小写。 |
| error C2078: too many initializers  中文对照：（编译错误）初始值过多  分析：一般是数组初始化时初始值的个数大于数组长度，例如“int b[2]={1,2,3};” |
| error C2082: redefinition of formal parameter 'xxx'  中文对照：（编译错误）重复定义形式参数xxx  分析：函数首部中的形式参数不能在函数体中再次被定义 |
| error C2084: function 'xxx' already has a body  中文对照：（编译错误）已定义函数xxx  分析：在VC++早期版本中函数不能重名，6.0版本中支持函数的重载，函数名可以相同但参数不一样 |
| error C2086: 'xxx' : redefinition  中文对照：（编译错误）标识符xxx重定义  分析：变量名、数组名重名 |
| error C2087: '<Unknown>' : missing subscript  中文对照：（编译错误）下标未知  分析：一般是定义二维数组时未指定第二维的长度，例如“int a[3][];” |
| error C2100: illegal indirection  中文对照：（编译错误）非法的间接访问运算符“\*”  分析：对非指针变量使用“\*”运算 |
| error C2105: 'operator' needs l-value  中文对照：（编译错误）操作符需要左值  分析：例如“(a+b)++;”语句，“++”运算符无效 |
| error C2106: 'operator': left operand must be l-value  中文对照：（编译错误）操作符的左操作数必须是左值  分析：例如“a+b=1;”语句，“=”运算符左值必须为变量，不能是表达式 |
| error C2110: cannot add two pointers  中文对照：（编译错误）两个指针量不能相加  分析：例如“int \*pa,\*pb,\*a; a = pa + pb;”中两个指针变量不能进行“+”运算 |
| error C2117: 'xxx' : array bounds overflow  中文对照：（编译错误）数组xxx边界溢出  分析：一般是字符数组初始化时字符串长度大于字符数组长度，例如“char str[4] = "abcd";” |
| error C2118: negative subscript or subscript is too large  中文对照：（编译错误）下标为负或下标太大  分析：一般是定义数组或引用数组元素时下标不正确 |
| error C2124: divide or mod by zero  中文对照：（编译错误）被零除或对0求余  分析：例如“int i = 1 / 0;”除数为0 |
| error C2133: 'xxx' : unknown size  中文对照：（编译错误）数组xxx长度未知  分析：一般是定义数组时未初始化也未指定数组长度，例如“int a[];” |
| error C2137: empty character constant。  中文对照：（编译错误）字符型常量为空  分析：一对单引号“''”中不能没有任何字符 |
| error C2143: syntax error : missing 'token1' before 'token2'  error C2146: syntax error : missing 'token1' before identifier 'identifier'  中文对照：（编译错误）在标识符或语言符号2前漏写语言符号1  分析：可能缺少“{”、“)”或“；”等语言符号 |
| error C2144: syntax error : missing ')' before type 'xxx'  中文对照：（编译错误）在xxx类型前缺少‘）’  分析：一般是函数调用时定义了实参的类型 |
| error C2181: illegal else without matching if  中文对照：（编译错误）非法的没有与if相匹配的else  分析：可能多加了“；”或复合语句没有使用“{}” |
| error C2196: case value '0' already used  中文对照：（编译错误）case值0已使用  分析：case后常量表达式的值不能重复出现 |
| error C2296: '%' : illegal, left operand has type 'float'  error C2297: '%' : illegal, right operand has type 'float'  中文对照：（编译错误）%运算的左(右)操作数类型为float，这是非法的  分析：求余运算的对象必须均为int类型，应正确定义变量类型或使用强制类型转换 |
| error C2371: 'xxx' : redefinition; different basic types  中文对照：（编译错误）标识符xxx重定义；基类型不同  分析：定义变量、数组等时重名 |
| error C2440: '=' : cannot convert from 'char [2]' to 'char'  中文对照：（编译错误）赋值运算，无法从字符数组转换为字符  分析：不能用字符串或字符数组对字符型数据赋值，更一般的情况，类型无法转换 |
| error C2447: missing function header (old-style formal list?)  error C2448: '<Unknown>' : function-style initializer appears to be a function definition  中文对照：（编译错误）缺少函数标题(是否是老式的形式表？)  分析：函数定义不正确，函数首部的“( )”后多了分号或者采用了老式的C语言的形参表 |
| error C2450: switch expression of type 'xxx' is illegal  中文对照：（编译错误）switch表达式为非法的xxx类型  分析：switch表达式类型应为int或char |
| error C2466: cannot allocate an array of constant size 0  中文对照：（编译错误）不能分配长度为0的数组  分析：一般是定义数组时数组长度为0 |
| error C2601: 'xxx' : local function definitions are illegal  中文对照：（编译错误）函数xxx定义非法  分析：一般是在一个函数的函数体中定义另一个函数 |
| error C2632: 'type1' followed by 'type2' is illegal  中文对照：（编译错误）类型1后紧接着类型2，这是非法的  分析：例如“int float i;”语句 |
| error C2660: 'xxx' : function does not take n parameters  中文对照：（编译错误）函数xxx不能带n个参数  分析：调用函数时实参个数不对，例如“sin(x,y);” |
| error C2664: 'xxx' : cannot convert parameter n from 'type1' to 'type2'  中文对照：（编译错误）函数xxx不能将第n个参数从类型1转换为类型2  分析：一般是函数调用时实参与形参类型不一致 |
| error C2676: binary '<<' : 'class istream\_withassign' does not define this operator or a conversion to a type acceptable to the predefined operator  error C2676: binary '>>' : 'class ostream\_withassign' does not define this operator or a conversion to a type acceptable to the predefined operator  分析：“>>”、“<<”运算符使用错误，例如“cin<<x; cout>>y;” |
| error C4716: 'xxx' : must return a value  中文对照：（编译错误）函数xxx必须返回一个值  分析：仅当函数类型为void时，才能使用没有返回值的返回命令。 |
| fatal error LNK1104: cannot open file "Debug/Cpp1.exe"  中文对照：（链接错误）无法打开文件Debug/Cpp1.exe  分析：重新编译链接 |
| fatal error LNK1168: cannot open Debug/Cpp1.exe for writing  中文对照：（链接错误）不能打开Debug/Cpp1.exe文件，以改写内容。  分析：一般是Cpp1.exe还在运行，未关闭 |
| fatal error LNK1169: one or more multiply defined symbols found  中文对照：（链接错误）出现一个或更多的多重定义符号。  分析：一般与error LNK2005一同出现 |
| error LNK2001: unresolved external symbol \_main  中文对照：（链接错误）未处理的外部标识main  分析：一般是main拼写错误，例如“void main()” |
| error LNK2005: \_main already defined in Cpp1.obj  中文对照：（链接错误）main函数已经在Cpp1.obj文件中定义  分析：未关闭上一程序的工作空间，导致出现多个main函数 |
| warning C4003: not enough actual parameters for macro 'xxx'  中文对照：（编译警告）宏xxx没有足够的实参  分析：一般是带参宏展开时未传入参数 |
| warning C4067: unexpected tokens following preprocessor directive - expected a newline  中文对照：（编译警告）预处理命令后出现意外的符号 - 期待新行  分析：“#include<iostream.h>;”命令后的“；”为多余的字符 |
| warning C4091: '' : ignored on left of 'type' when no variable is declared  中文对照：（编译警告）当没有声明变量时忽略类型说明  分析：语句“int ;”未定义任何变量，不影响程序执行 |
| warning C4101: 'xxx' : unreferenced local variable  中文对照：（编译警告）变量xxx定义了但未使用  分析：可去掉该变量的定义，不影响程序执行 |
| warning C4244: '=' : conversion from 'type1' to 'type2', possible loss of data  中文对照：（编译警告）赋值运算，从数据类型1转换为数据类型2，可能丢失数据  分析：需正确定义变量类型，数据类型1为float或double、数据类型2为int时，结果有可能不正确，数据类型1为double、数据类型2为float时，不影响程序结果，可忽略该警告 |
| warning C4305: 'initializing' : truncation from 'const double' to 'float'  中文对照：（编译警告）初始化，截取双精度常量为float类型  分析：出现在对float类型变量赋值时，一般不影响最终结果 |
| warning C4390: ';' : empty controlled statement found; is this the intent?  中文对照：（编译警告）‘；’控制语句为空语句，是程序的意图吗？  分析：if语句的分支或循环控制语句的循环体为空语句，一般是多加了“；” |
| warning C4508: 'xxx' : function should return a value; 'void' return type assumed  中文对照：（编译警告）函数xxx应有返回值，假定返回类型为void  分析：一般是未定义main函数的类型为void，不影响程序执行 |
| warning C4552: 'operator' : operator has no effect; expected operator with side-effect  中文对照：（编译警告）运算符无效果；期待副作用的操作符  分析：例如“i+j;”语句，“+”运算无意义 |
| warning C4553: '==' : operator has no effect; did you intend '='?  中文对照：（编译警告）“==”运算符无效；是否为“=”？  分析：例如 “i==j;” 语句，“==”运算无意义 |
| warning C4700: local variable 'xxx' used without having been initialized  中文对照：（编译警告）变量xxx在使用前未初始化  分析：变量未赋值，结果有可能不正确，如果变量通过scanf函数赋值，则有可能漏写“&”运算符，或变量通过cin赋值，语句有误 |
| warning C4715: 'xxx' : not all control paths return a value  中文对照：（编译警告）函数xxx不是所有的控制路径都有返回值  分析：一般是在函数的if语句中包含return语句，当if语句的条件不成立时没有返回值 |
| warning C4723: potential divide by 0  中文对照：（编译警告）有可能被0除  分析：表达式值为0时不能作为除数 |
| warning C4804: '<' : unsafe use of type 'bool' in operation  中文对照：（编译警告）‘<’：不安全的布尔类型的使用  分析：例如关系表达式“0<=x<10”有可能引起逻辑错误 |

**附录Ⅱ：常见单词**

Pointers（指针） references（引用） casts（类型转换）

arrays（数组） constructors（构造）

abstraction抽象 action行动 action-oriented面向行动

analysis分析 ANSI/ISO standard Ｃ++标准Ｃ＋＋

arithmetic and logic unit**(ALU)** 算术和逻辑单元

arithmetic operators 算术操作符 assenbly language汇编语言

association关联 associativity of operators地址操作符

assignment operator赋值操作符 attribute属性

attributes of an object对象的属性 behavior行为

binary operator二元操作符 C++ standard library C++标准库

compile error 编译错误 compiler编译器 component组件

date member 数据成员 distributed computing 分布式计算

editor编辑器 encapsulation封装 execution-time error执行期错误

fatal error致命错误 flow of control控制流程

function函数 identifier标识符 information hiding信息隐藏

inheritance继承 instantiate实例化

interface接口 interpreter解释器 linking连接

logic error逻辑错误 modeling建模 inheritance多重继承

multiprogramming多路程序 Object Management Group（OMG）对象管理组

object-oriented analysis and design（OOAD）面向对象分析和设计

operator associativity操作符的结合性 precedence优先级

preprocessor预处理器 prompt提示 pseudocode伪代码 satement语句

structured programming结构化编程 syntax error语法错误

Unified Modeling Language（UML）统一建模语言

user-defined type 用户自定义类型 variable变量名

algorithm算法 block代码块 case label标签

infinite loop无限循环 delay loop延迟循环

parameterized stream manipulator参数化流操纵元

syntax error语法错误 composition合成

Object Constraint Language（OCL）对象限制语言

argument in a function call函数调用中的参数

automatic storage class自动存储类 call-by-reference按引用调用

coercion of arguments强制类型转换

dangling reference悬挂引用 enumeration枚举

access function访问函数 class scope类作用域

constructor构造函数 destructor析构函数

global object全局对象 header file头文件

interface to a class类的接口 proxy class代理类

rapid applications development（RAD）快速应用程序开发

source-code file源代码文件 handle句柄

abstract data type（ADT）抽象数据类型

first-in-first-out（FIFO）先进先出 iterator迭代器

member access specifiers成员访问说明符

pop（stack operation）弹出（堆栈操作）

forward declaration 提前声明

1、int: integer整数 2、const: constant常量

3、Variable:变量 4、IDE=Integrated Development Environment:集成开发环境

5、Visual C: Microsoft公司开发的C语言集成开发环境软件

6、Turbo C: Borland公司开发的C编译器

7、GCC: Linux下的C编译器

8、C Builder: Borland公司开发的C语言IDE

9、Compile:编译 10、compiler:编译器 11、float:浮点数,实数12、double:双精度浮点数，实数 13、debug:调试

14、Dennis Ritchie: C语言的创始人

15、《The C Programming Language》:布莱恩·柯林汉（Brian Kernighan）和丹尼斯·里奇（Dennis Ritchie）编写的C语言（K&R）

16、ANSI C: C语言国际准则，亦称ISO C，ANSI= American National Standards Institute美国国家标准学会

17、AT&T=American Telephone & Telegraph美国电话电报公司

18、Bell Labs:贝尔实验室（C语言的创造地，隶属于AT&T）

19、MSDN=Microsoft Developer Network:微软开发者网络

20、MSDN Library: 微软开发者技术库

21、MFC= Microsoft Foundation Classes:微软基础类

22、Visual Studio:微软开发的编程IDE，包括VC、V、VC#等组件

23、Bite:字节,存储容量单位 24、KB:千字节 25、MB:兆字节

26、file:文件 27、input/output:输入输出 28、class:类 29、Array:数组

30、object:对象 31、loop:循环体 32、operator:运算符 33、function:函数

34、macro:宏 35、define:定义 36、include:包含 37、math:数学

38、Microsoft:美国微软公司

39、Windows:微软开发的系统,用C语言编写

**附录Ⅲ 文件扩展名**

.APS：资源辅助文件。存放二进制资源的中间文件，VC把当前资源文件转换成二进制格式，并存放在APS文件中，以加快资源装载速度。

.BMP：位图资源文件。

.BSC：浏览信息文件。由浏览信息维护工具（BSCMAKE）从原始浏览信息文件（.SBR）中生成，BSC文件可以用来在源代码编辑窗口中进行快速定位。若用source browser就必须有这个文件。可以在project options里去掉Generate Browse Info File以加快编译进度。

.C：用C语言编写的源代码文件。

.CLW：ClassWizard生成的用来存放类信息的文件。classwizard信息文件，ini文件的格式。

.CNT：用来定义帮助文件中“Contents”的结构。

.CPP或.CXX：用C++语言编写的源代码文件。

.CUR：光标资源文件。

.DEF：模块定义文件，供生成动态链接库时使用。

.DLG：定义对话框资源的独立文件。对于VC工程来说并非必需，因为VC一般把对话框资源放在.RC资源定义文件中。

.DSP：项目文件。VC开发环境生成的工程文件，VC4及以前版本使用MAK文件来定义工程，文本格式。

.DSW：工作区文件。VC开发环境生成的WorkSpace文件，用来把多个工程组织到一个WorkSpace中，与.dsp差不多。

.EXP：由LIB工具从DEF文件生成的输出文件，其中包含了函数和数据项目的输出信息，LINK工具将使用EXP文件来创建动态链接库。只有在编译DLL时才会生成，记录了DLL文件中的一些信息。

.H/.HPP或.HXX：用C/C++语言编写的头文件，通常用来定义数据类型，声明变量/函数/结构/类。

.HLP：Windows帮助文件。

.HM：在Help工程中，该文件定义了帮助文件与对话框、菜单或其它资源之间ID值的对应关系。

.HPJ：由Help Workshop生成的Help工程文件，用来控制Help文件的生成过程。

.HPG，生成帮助的文件的工程。

.ICO：图标资源文件。

.ILK：连接过程中生成的一种中间文件，只供LINK工具使用。

.INI：配置文件。

.LIB：库文件，LINK工具将使用它来连接各种输入库，以便最终生成EXE文件。

.LIC：用户许可证书文件，使用某些ActiveX控件时需要该文件。

.MAK：即MAKE文件，VC4及以前版本使用的工程文件，用来指定如何建立一个工程，VC6把MAK文件转换成DSP文件来处理。

.MAP：执行文件的映像信息记录文件。由LINK工具生成的一种文本文件，其中包含有被连接的程序的某些信息，例如程序中的组信息和公共符号信息等。

.MDP：旧版本的项目文件，相当于.dsp

.NCB：NCB是“No Compile Browser”的缩写，其中存放了供ClassView.WizardBar和Component Gallery使用的信息，由VC开发环境自动生成。无编译浏览文件。当自动完成功能出问题时可以删除此文件，编译工程后会自动生成。

.OBJ：由编译器或汇编工具生成的目标文件，是模块的二进制中间文件。

.ODL：用对象描述语言编写的源代码文件，VC用它来生成TLB文件。

.OLB：带有类型库资源的一种特殊的动态链接库，也叫对象库文件。

.OPT：VC开发环境自动生成的用来存放WorkSpace中各种选项的文件。工程关于开发环境的参数文件。如工具条位置信息等。

.PBI/.PBO/.PBT：由VC的性能分析工具PROFILE生成并使用的三种文件。

.PCH：预编译头文件。比较大，由编译器在建立工程时自动生成，其中存放有工程中已经编译的部分代码，在以后建立工程时不再重新编译这些代码，以便加快整个编译过程的速度。

.PDB：程序数据库文件。在建立工程时自动生成，其中存放程序的各种信息，用来加快调试过程的速度。记录了程序有关的一些数据和调试信息。

.PLG：编译信息文件，编译时的error和warning信息文件。

.RC：资源定义文件。

.RC2：资源定义文件，供一些特殊情况下使用。

.REG：注册表信息文件。

.RES：二进制资源文件，资源编译器编译资源定义文件后即生成RES文件。

.RTF：Rich Text Format（丰富文本格式）文档，可由Word或写字板来创建，常被用来生成Help文件。

.SBR：VC编译器为每个OBJ文件生成的原始浏览信息文件，浏览信息维护工具（BSCMAKE）将利用SBR文件来生成BSC文件。

.TLB：OLE库文件，其中存放了OLE自动化对象的数据类型.模块和接口定义，自动化服务器通过TLB文件就能了解自动化对象的使用方法。

.WAV：声音资源文件。