

Visual C++ 6.0

编程



学习捷径

胡海生 李升亮 编著



清华大学出版社

Visual C++ 6.0 编程学习捷径

胡海生 李升亮 编著

清华大学出版社

北 京

内 容 简 介

Visual C++ 6.0 是目前最流行、使用最广泛的软件开发工具之一。它在编程的深入性、运行的快速性等方面都具有很强的功能。

本书以 MFC 编程方式为主、Windows API 编程方式为辅，循序渐进地把 Visual C++ 6.0 程序设计的主要特色展示给读者；全书结合实例讲解，从易到难地将 Visual C++ 6.0 的编程技术和设计难点娓娓道来，为编程入门人员提供了一条学习 Visual C++ 6.0 的捷径。全书共分为 11 章，内容包括：VC++ 的基本操作、对话框编程、常用控件编程、绘图及打印编程、位图编程、文件操作、动态链接库、进程和线程、串口通信、多媒体编程、Windows 套接字。

本书主要针对初学者和有一定编程基础的读者，同时也可作为自学用书或相关专业的培训教材，其中的实例对程序设计人员也有一定的参考价值。

版权所有，翻印必究。

本书封面贴有清华大学出版社激光防伪标签，无标签者不得销售。

图书在版编目 (CIP) 数据

Visual C++ 6.0 编程学习捷径/胡海生，李升亮编著.

北京：清华大学出版社，2003.9

ISBN 7-302-07446-1

I. V… II. ①胡… ②李… III. C 语言-程序设计 IV. TP312

中国版本图书馆 CIP 数据核字 (2003) 第 094381 号

出 版 者：清华大学出版社

<http://www.tup.com.cn>

社 总 机：010-62770175

地 址：北京清华大学学研大厦

邮 编：100084

客户服务：010-62776969

组稿编辑：科海

文稿编辑：陈轶

封面设计：杨月静

版式设计：科海

印 刷 者：北京科普瑞印刷有限责任公司

发 行 者：新华书店总店北京发行所

开 本：787×1092 1/16 印张：27 字数：657 千字

版 次：2003 年 10 月第 1 版 2003 年 12 月第 2 次印刷

书 号：ISBN 7-302-07446-1/TP·5502

印 数：5001~7000

定 价：39.00 元

前 言

Visual C++ 6.0 是 Microsoft 公司推出的基于 C/C++ 的集成开发工具，它不仅是 C/C++ 语言的集成开发环境，而且与 Win32 紧密相连，功能相当强大，代码效率很高，可以实现从底层软件直到上层直接面向用户的软件。Visual C++ 6.0 也是目前最为流行、使用最为广泛的软件开发工具之一。

Visual C++ 6.0 向用户提供了一个面向对象、可视化的编程环境，它的 MFC (Microsoft Foundation Class, 微软基础类库) 封装了大部分 Windows API (Application Programming Interface, 应用程序编程接口) 函数，大大简化了用户的编程工作，提高了代码的可重用性，使得用户可以快速地进行软件开发。

在 Visual C++ 6.0 集成开发环境下，用户可以使用两种方式开发 Windows 应用程序，一种是基于 Windows API 的 C 编程方式，这种方式的开发难度与开发的工作量较大，但代码的运行效率较高；另一种是基于 MFC 的 C++ 编程方式，相对第一种方式，用这种方式编写的代码运行效率较低，但开发难度小、开发工作量小、源代码效率较高。本书将以 MFC 编程方式为主并结合 Windows API 编程方式向读者介绍 Visual C++ 6.0 的编程技术。

本书的内容安排如下：

- 第 1 章 Visual C++ 6.0 编程基础，主要介绍 Visual C++ 6.0 集成开发环境和 Windows 编程的一些基础知识，为读者学习后面各章提供基础。
- 第 2 章 对话框编程，主要介绍对话框、一些通用对话框和消息框的基本编程方法。
- 第 3 章 常用控件编程，主要介绍一些常用控件（包括按钮、编辑框、列表框等）的基本使用方法。
- 第 4 章 绘图及打印编程，主要介绍如何利用 GDI (Graphic Device Interface, 图形设备接口) 实现绘图操作和如何进行打印输出。
- 第 5 章 位图编程，主要介绍 Windows 位图编程技术，并介绍如何将 Windows 位图编程应用到 GIS (Geography Information System, 地理信息系统) 中去。
- 第 6 章 文件操作，主要介绍如何利用 MFC 的文件类进行文件存取操作等。
- 第 7 章 动态链接库，主要介绍如何编写动态链接库 (Dynamic-Link Library, DLL)。
- 第 8 章 进程和线程，主要介绍进程和线程的基本知识，以及如何实现进程和线程的同步。
- 第 9 章 串口通信，主要介绍如何利用 Windows 通信 API 编写串口通信程序。
- 第 10 章 多媒体编程，主要介绍媒体控制接口 (MCI) 编程技术。
- 第 11 章 Windows 套接字，主要介绍 Windows 套接字 (WinSocket) 方面的基本知识，以及如何利用 WinSocket 编写 Windows 网络通信程序。

本书由胡海生和李升亮编写，其中第 3 章由李升亮编写，其余章节由胡海生编写，最后由胡海生修改定稿。

由于编者水平有限，缺点和错误在所难免，恳请读者朋友批评指正。最后祝愿每一位读者在学习程序设计的过程中获得真正的乐趣。

读者联系方式: huhs2000@sina.com

编者

2003 年 8 月

目 录

第1章 Visual C++ 6.0编程基础	1
1.1 Visual C++ 6.0集成开发环境概述	1
1.1.1 标题栏	2
1.1.2 菜单栏	2
1.1.3 工具栏	2
1.1.4 项目工作区	2
1.1.5 文档窗口	4
1.1.6 信息输出窗口	4
1.1.7 状态栏	4
1.1.8 应用程序向导AppWizard	4
1.1.9 类向导ClassWizard	5
1.2 Visual C++项目及其文件构成	5
1.2.1 Visual C++项目	5
1.2.2 文件构成	5
1.3 Windows编程基础	6
1.3.1 匈牙利符号表示法	6
1.3.2 Windows消息和命令	8
1.3.3 句柄	10
1.4 一个简单的应用程序	10
1.4.1 使用AppWizard生成应用程序基本框架	11
1.4.2 应用程序框架剖析	15
1.4.3 添加代码	18
1.4.4 编译执行程序	19
第2章 对话框编程	20
2.1 模式对话框	20
2.1.1 创建项目	20
2.1.2 创建对话框模板	20
2.1.3 创建对话框类	21
2.1.4 添加消息处理函数	22
2.1.5 定义对话框对象	23
2.1.6 编译并运行程序	24
2.2 非模式对话框	24
2.2.1 创建对话框	24

2.2.2 创建对话框类	25
2.2.3 添加代码	26
2.3 通用对话框和消息对话框	34
2.3.1 通用对话框	34
2.3.2 消息对话框	43
第3章 常用控件编程	45
3.1 控件的创建和使用	45
3.1.1 控件的创建	45
3.1.2 Windows常用控件介绍	47
3.2 Windows常用控件	49
3.2.1 静态控件	49
3.2.2 按钮	49
3.2.3 编辑框	52
3.2.4 列表框	55
3.2.5 列表控制	59
3.2.6 组合框	68
3.2.7 滚动条	71
3.2.8 滑动条	73
3.2.9 旋转按钮	74
第4章 绘图及打印编程	81
4.1 图形设备接口	81
4.1.1 设备描述表及其获取方法	81
4.1.2 映像模式	83
4.1.3 坐标系统	84
4.1.4 GDI对象	86
4.1.5 常用的绘图函数	91
4.2 实现绘图程序	94
4.2.1 建立Draw程序框架	94
4.2.2 实现Draw图形类	96
4.2.3 实现文档类	109
4.2.4 添加绘图菜单和工具条	113
4.2.5 实现视图类	124
4.2.6 编译并执行程序	135
4.3 打印和打印预览	136
4.3.1 打印	136
4.3.2 打印预览	139
4.3.3 给对话框实现打印功能	139

第5章 位图编程	155
5.1 设备相关位图和设备无关位图	155
5.1.1 设备相关位图 (DDB)	155
5.1.2 设备无关位图 (DIB)	155
5.2 使用DDB位图	163
5.3 DIB编程示例	167
5.3.1 BMP地图文件	168
5.3.2 CDib类	168
5.3.3 CImageMap类	188
5.3.4 图像配准	211
5.3.5 添加菜单、工具条和光标	214
5.3.6 实现CImageView类	215
5.3.7 程序运行结果	226
第6章 文件操作	228
6.1 MFC文件类简介	228
6.2 利用CFile类进行文件操作	228
6.2.1 CFile类成员函数	228
6.2.2 打开文件	229
6.2.3 文件的读写	230
6.2.4 文件的随机访问	231
6.2.5 文件的关闭	232
6.3 使用CStdioFile类进行文件读写	233
6.4 CMemFile类	234
6.5 CShareFile类	235
6.6 CArchive类	236
第7章 动态链接库	238
7.1 基本动态链接库理论	238
7.1.1 动态链接库简介	238
7.1.2 导出和导入函数的匹配	239
7.1.3 调用约定	239
7.1.4 修饰约定	240
7.1.5 DllMain函数	242
7.1.6 模块句柄	242
7.2 使用动态链接库	243
7.2.1 隐式链接	243
7.2.2 显式链接	244
7.3 动态链接库的创建	249
7.3.1 Non-MFC DLL	249

7.3.2	Regular DLL.....	253
7.3.3	Extension DLL	255
第8章	进程和线程	258
8.1	进程	258
8.1.1	创建一个进程	259
8.1.2	终止一个进程	260
8.2	线程	260
8.2.1	线程的类型	260
8.2.2	工作者线程	261
8.2.3	用户界面线程	264
8.2.4	线程调度	266
8.2.5	线程局部存储	267
8.3	线程同步	270
8.3.1	临界区	270
8.3.2	互斥量	273
8.3.3	信号量	278
8.3.4	事件	279
8.3.5	互锁函数	284
8.3.6	等待函数	286
8.4	进程间同步	290
第9章	串口通信	291
9.1	Win32 API串口通信编程概述	291
9.2	串口通信函数	292
9.2.1	打开和关闭串口	292
9.2.2	配置串口	293
9.2.3	配置超时值	296
9.2.4	缓冲区操作	298
9.2.5	串口读写操作	299
9.2.6	获取通信状态和通信错误	303
9.2.7	使用通信事件	305
9.2.8	设备控制	308
9.3	串口通信实例程序	309
9.3.1	功能目标	309
9.3.2	主要技术和算法	310
9.3.3	程序实现	311
第10章	多媒体编程	334
10.1	MCI简介	334

10.2	MCI函数.....	335
10.3	MCI命令.....	339
10.3.1	MCI命令集	339
10.3.2	标志 (FLAG)	341
10.4	MCI基本操作.....	342
10.4.1	启动MCI装置.....	342
10.4.2	设定MCI装置的时间表示法.....	344
10.4.3	操作MCI装置.....	344
10.4.4	关闭MCI装置.....	345
10.4.5	获取时间信息	345
10.5	一个MCI实例程序.....	346
10.5.1	功能目标	346
10.5.2	程序实现	347
第11章	Windows套接字.....	377
11.1	WinSock概述	377
11.1.1	套接字 (Sockets)	377
11.1.2	基本概念	378
11.2	WinSock编程原理.....	379
11.2.1	WinSock的启动和终止	380
11.2.2	创建套接字	381
11.2.3	使用数据报套接字	382
11.2.4	使用流套接字	388
11.2.5	等待事件	391
11.2.6	套接字选项和I/O控制命令	394
11.3	一个网络通信程序示例.....	396
11.3.1	功能目标	396
11.3.2	程序实现	397
11.4	WinSock 2的扩展特性.....	419
11.4.1	WinSock 2的新函数	419
11.4.2	使用多种协议	419
11.4.3	重叠I/O.....	420
11.4.4	服务质量 (QOS)	420
11.4.5	多点套接字	420
11.4.6	共享的套接字	421
11.4.7	扩展的连接操作	421

第 1 章 Visual C++ 6.0 编程基础

Visual C++ 是 Microsoft 公司推出的 Windows 平台下最主要的可视化编程环境之一，它不仅是 C++ 语言的集成开发环境，而且与 Win32 紧密相连，功能相当强大，可以完成从底层软件直到上层直接面向用户的软件。此外，Visual C++ 还提供了强大的调试功能。Visual C++ 6.0 是目前最为流行的版本，本章主要介绍 Visual C++ 6.0 集成开发环境和 Windows 编程的一些基础知识，为读者学习后面各章提供基础。

1.1 Visual C++ 6.0 集成开发环境概述

Visual C++ 6.0 集成开发环境提供了各种工具和向导以支持可视化编程，它们包括标题栏、菜单栏、工具栏、项目工作区、文档窗口、信息输出窗口和状态栏等工具，以及应用程序向导 AppWizard、类向导 ClassWizard 等向导，如图 1.1 所示。

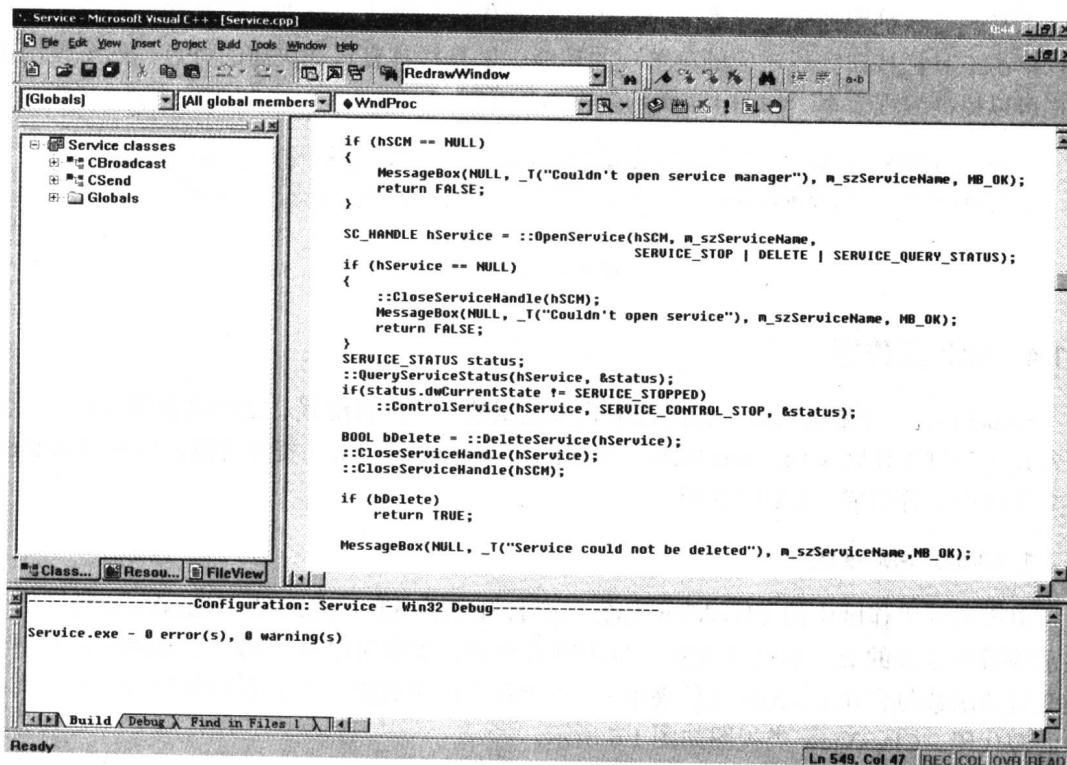


图 1.1 Visual C++ 6.0 开发环境

1.1.1 标题栏

标题栏包括最小化、最大化或还原、关闭按钮，如图 1.2 所示，标题栏上还可显示当前打开的文件名。



图 1.2 标题栏 1

1.1.2 菜单栏

菜单栏包括 File、Edit、View、Insert、Project、Build、Tools、Window、Help 等菜单项，它们包含了开发环境几乎所有的操作命令，为用户提供工程的创建、设置、编译、调试以及文件操作、窗口操作等功能。如图 1.3 所示，我们将在后面的各章节中介绍各个菜单的功能。

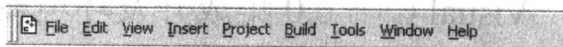


图 1.3 菜单栏

1.1.3 工具栏

工具栏是一种图形化的操作界面，由某些操作按钮组成，分别对应某些菜单项，用户可以单击工具栏按钮完成对应的菜单项的功能。工具栏具有直观快捷的特点，熟练使用工具栏可以提高编程效率。如图 1.4 所示。

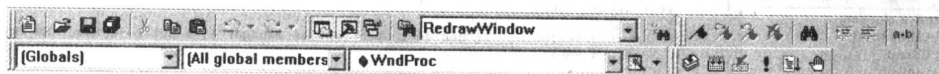


图 1.4 工具栏

1.1.4 项目工作区

Visual C++ 6.0 是以项目工作区的形式来组织项目、项目配置、文件以及项目信息的显示方式。项目工作区是由 ClassView（类视图）、ResourceView（资源视图）以及 FileView（文件视图）等组成。如图 1.5 所示。

1. ClassView 视图

单击项目工作区中的 ClassView 标签就会打开 ClassView 类视图，ClassView 视图用于显示项目中定义的类，通过该视图可以浏览各个类、类成员以及全局变量和函数，其中全局变量和函数包含在 Globals 文件夹中。双击类和类成员就能在文档窗口分别看到类和类成员的源代码。ClassView 类视图如图 1.6 所示。

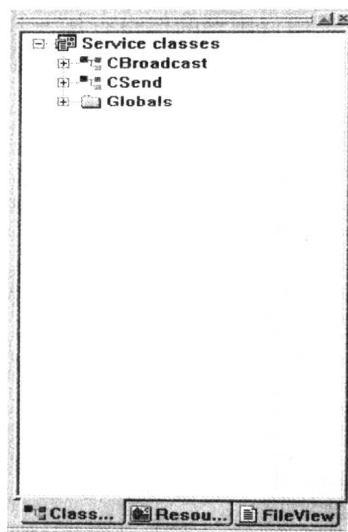


图 1.5 项目工作区

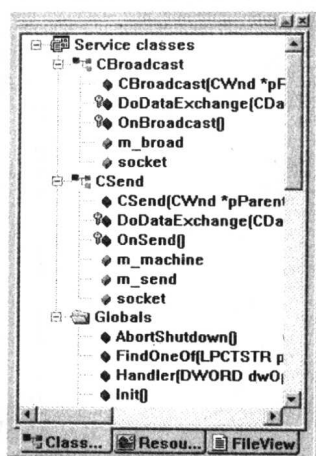


图 1.6 ClassView 类视图

2. ResourceView 视图

单击项目工作区中的 ResourceView 标签就会打开 ResourceView 资源视图，ResourceView 视图用于管理项目中所有的资源文件，展开最顶层文件夹就会看到项目中所有的资源类型，如图 1.7 所示。如加速键资源 (Accelerator)、位图资源 (Bitmap)、对话框资源 (Dialog)、图标资源 (Icon)、菜单资源 (Menu) 以及字符串资源 (String Table) 等。

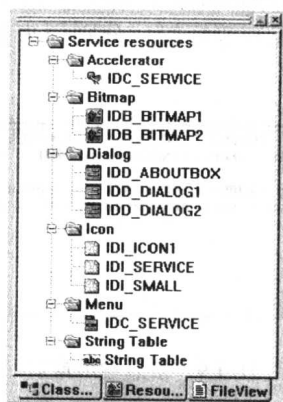


图 1.7 ResourceView 资源视图

3. FileView 视图

单击项目工作区中的 FileView 标签就会打开 FileView 文件视图，FileView 视图用于所创建的项目和项目中包含的文件，如图 1.8 所示。Source Files 文件夹管理源程序文件 (.cpp)，Header Files 文件夹管理头文件 (.h)，Resource Files 文件夹管理各个资源文件 (包括位图、

图标等), 还有 External Dependencies 文件夹用于管理项目用到的但没有加入到项目中的外部文件。双击各个文件就会打开各个文件对应的编辑器。

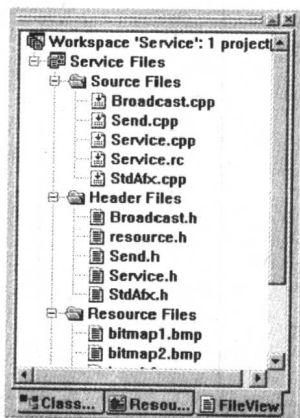


图 1.8 FileView 文件视图

1.1.5 文档窗口

如图 1.1 所示, 文档窗口一般位于开发环境的右边, 用于显示和编辑各种源文件、资源文件以及其他文件的程序代码。

1.1.6 信息输出窗口

信息输出窗口一般位于开发环境的底部, 如图 1.9 所示, 它包括 Build (编译和连接)、Debug (调试)、Find in Files (查找) 等标签, 分别用来显示编译和连接信息、调试信息和查找信息等。

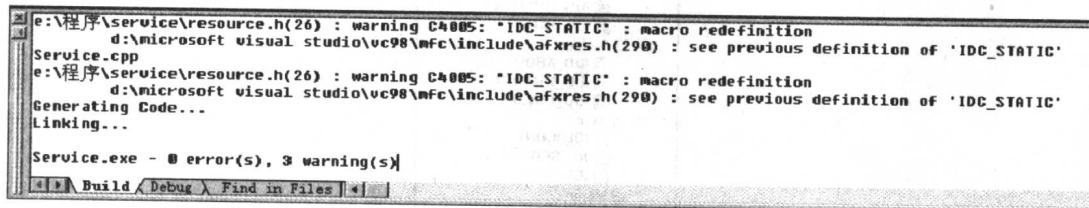


图 1.9 信息输出窗口

1.1.7 状态栏

状态栏位于开发环境的最低部, 用来显示当前操作状态、光标位置以及键盘上某些键的状态等信息。

1.1.8 应用程序向导 AppWizard

AppWizard 作为 Visual C++ 6.0 的一个高级编程工具, 它的作用是帮助用户一步步生成

一个新的应用程序框架，并自动生成应用程序框架代码，这样可以为用户节省开发应用程序的时间和精力，使用户可以用较少的时间完成这些重复性的工作，而把主要时间和精力放在设计程序的功能上。使用 AppWizard 时，会有一系列对话框提示用户输入创建程序框架所需的一些信息，AppWizard 会根据这些信息自动生成一些文件和文件代码。在后面的各章节中我们会向读者展示 AppWizard 强大的功能。

1.1.9 类向导 ClassWizard

Visual C++ 6.0 提供的另一个很重要的高级编程工具是 ClassWizard，它主要用来管理程序中的类、对象和消息。ClassWizard 是一个交互式工具，可以用来建立程序所需要的新类、添加、修改各种消息和消息映射，如图 1.10 所示。我们将在后面介绍它的用法。

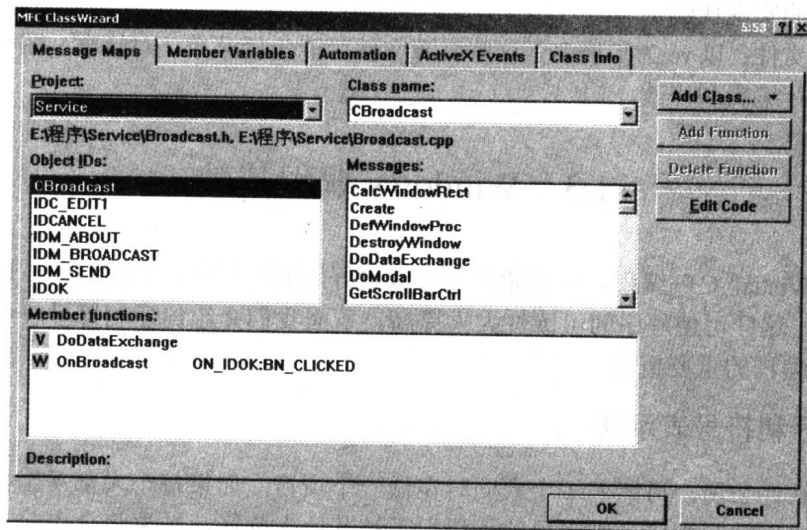


图 1.10 类向导 ClassWizard

1.2 Visual C++项目及其文件构成

1.2.1 Visual C++项目

Visual C++项目也称为工程，Visual C++应用程序是以项目的形式存在的。Visual C++应用程序除了源代码文件外，还包含菜单、工具栏、图标、对话框等资源文件，另外还包括一些应用程序所需的库文件和系统 DLL 文件等。在 Visual C++中项目的作用就是有效组织这些文件并维护这些文件相互间的依赖关系。

1.2.2 文件构成

在 Visual C++项目中包含有不同类型的文件，这些不同类型的文件各自有不同的作用，

其中，主要包括以下类型的文件：

- 项目文件：以 `dsp` 为扩展名，每个项目对应一个 `dsp` 文件。
- 项目工作区文件：以 `dsw` 为扩展名，在一个 `dsw` 文件中可以包含一个以上项目，由项目工作区文件对它们进行统一的协调和管理。
- 项目工作区配置文件：以 `opt` 为扩展名，它与 `dsw` 文件相配合，这个文件中包含 `dsw` 文件中要用到的本地计算机的有关配置信息。
- 类和资源配置文件：以 `clw` 为扩展名，用来存放有关类和资源的信息，ClassWizard 工具使用这些信息来管理类和资源。
- 头文件：以 `h` 为扩展名，一般包含类、一些常量和变量的定义。
- 实现文件：以 `cpp` 为扩展名，一般和以 `h` 为扩展名的文件相对应，主要包含类成员函数的实现代码。
- 资源文件：以 `rc` 为扩展名，这些资源文件可能包括菜单、对话框、图标等资源，可以进行编辑和修改。

1.3 Windows 编程基础

学习用 Visual C++ 编程，一开始就要养成良好的书写程序代码的习惯，遵守编写代码的标准规范，这样可使程序的可读性大大提高。下面我们介绍匈牙利符号表示法和其他一些 Windows 编程的重要概念。

1.3.1 匈牙利符号表示法

在大的软件公司，可能若干个人同时在做一个项目，如果每个人都采用自己的一套编写代码的方式来编写程序，则自己的代码很难被其他人读懂，结果将会引起一些不必要的麻烦，这样就需要提出一个编写代码的标准方式。Microsoft 公司委托 Charles Simonyi 创立了一套编写 Microsoft 代码的规范，称为匈牙利符号表示法。现在这套命名法已成为大多数程序员编写代码的规范。

匈牙利符号表示法包括了关于变量、函数、类型和常量、类以及参数的有关约定。表 1.1 给出了匈牙利符号表示法使用的前缀代码，用于指明数据类型。

表1.1 匈牙利符号表示法的前缀代码

前缀代码	数据类型
c	字符
by	字节（无符号字符）
n	短整数和整数
i	整数
x,y	短整数（通常用于 x 坐标和 y 坐标）
cx,cy	短整数（通常表示 x 和 y 的长度，c 表示计数）
b	布尔型（整数）

(续表)

前缀代码	数据类型
w	WORD (无符号字和无符号短整数)
l	LONG (长整数)
dw	DWORD (无符号长整数)
s	字符串
sz, str	以 0 字节终止的字符串
lp	32 位的长整数指针
fn	函数指针
h	编号 (常用于 Windows 对象)
cb	用于定义对象 (结构) 大小的整数
g, g_	全局变量
m_	类成员变量
msg	消息

1. 变量的命名

应用匈牙利表示法, 变量名可用表 1.1 中的前缀代码来表示。此外, 当一个变量由一个或几个子名构成时, 每一个子名都要以大写字母开头。例如:

```
char *szName;      //以 0 字节终止的字符串
int *lpiCount;     //一个指向 int 类型的 32 位指针
BOOL m_bFlag;      //一个 BOOL 型类成员变量
DWORD dwMinWidth;  //一个 32 位 DWORD 类型变量
long g_lLength;    //一个 32 位 long 型的全局变量
```

2. 函数的命名

函数的命名和变量的命名基本相同, 但没有前缀, 子名的第一个字母要大写, 带下划线的函数名是无效的匈牙利表示法。例如:

```
int GetLength(ix1, iy1, ix2, iy2);
char *StringCopy(char *szDes, char *szScr);
```

3. 类型和常量的命名

所有的类型和常量都是大写字母, 但名字中可以允许使用下划线。例如, 类型的命名如下所示:

```
typedef unsigned short USHORT;
```

常量的命名如下所示:

```
#define MAXLENGTH 1024
#define DATA_IN_EVENT 100
```