

清华推荐教材与自学用书



Visual C++ 程序设计基础

戴 锋 编著

- ◆ 可完全不懂C语言
- ◆ 直接学习 Visual C++ 程序设计
- ◆ 轻松开启程序设计之路
- ◆ 牢固打造程序设计基础



清华大学出版社
<http://www.tup.tsinghua.edu.cn>



Visual C++程序设计基础

戴 锋 编著

清华大学出版社

(京)新登字 158 号

内 容 简 介

Visual C++是 Microsoft 公司的开发工具中综合性最高、最复杂的产品。本书从基础入手,结合 Windows 下的 Visual C++编程环境详细讲解了 C++程序设计语言,并在此基础上对使用 MFC 类库进行 Windows 编程作了简单介绍。本书从一开始就在图形界面环境下讲述 C++语言程序设计,彻底摆脱了以往程序设计学习过程中必须的从字符界面到图形界面的过程。全书结构清晰、文笔流畅,配有丰富的例子程序。

本书是在 Windows 环境下学习 C++语言的入门书籍,可作为 C++语言的自学教材、培训教材,也可供高等院校非计算机专业的师生作教学参考之用。

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。

书 名: Visual C++程序设计基础

著 者: 戴 锋

责任编辑: 杨洪军

出 版 者: 清华大学出版社(北京清华大学学研大厦, 邮编 100084)

<http://www.tup.tsinghua.edu.cn>

印 刷 者: 北京市振华印刷厂

发 行 者: 新华书店总店北京发行所

开 本: 787×1092 1/16 印张: 25.25 字数: 611 千字

版 次: 2001 年 4 月第 1 版 2001 年 4 月第 1 次印刷

书 号: ISBN 7-302-04359-0 /TP · 2560

印 数: 0001~5000

定 价: 29.00 元

前 言

有了 C 语言在面向过程的时代中的辉煌作基础，继承了 C 语言衣钵的 C++ 语言，在面向对象的时代同样成为了最受欢迎的程序设计语言。如今，绝大多数优秀的系统软件和应用软件的开发语言都是 C++ 语言。在各种操作系统下 C++ 语言的开发平台都是必备的，C++ 语言已经成为了面向对象时代的程序设计语言的标准。

彻底掌握 C++ 语言，不仅是软件工程师份内的事，很多业余电脑爱好者也跃跃欲试。然而，C++ 语言是公认的最难学习的计算机语言，在面向过程的时代里 C 语言就被认为是“入门容易得道难”，而 C++ 语言更是“入门也难，得道更难”。总结其原因，不外乎有以下 3 点：

- 面向对象同面向过程的思维方法完全不同，习惯了使用诸如 Basic、Pascal、C 等面向过程语言进行编程的程序员，在思维上早已形成定式，即使学过 C++ 语言，也常常只会用 C++ 语言写出面向过程风格的程序来。令人遗憾的是，现在的计算机语言教学仍然保持着先学 Pascal，再学 C，最后学 C++ 的传统，使得很多人难以真正掌握面向对象方法的精髓，也就难以真正学会使用 C++ 语言进行程序设计。
- C++ 语言的祖先 C 语言本身就是一种中级语言，语法中涉及很多同系统底层有关的操作、运算，C++ 语言更是在此基础上变本加厉。如果 C 语言的初学者对于像指针这样的复杂运算不能完全掌握，仍旧可以用 C 语言的其他功能完成一些简单的程序，但是在 C++ 语言中如果不掌握指针，想学会面向对象的方法就无从谈起。
- 由于 C++ 语言诞生在字符界面时代，经典的 C++ 语言教材都是基于传统的字符界面下的 C++ 语言进行讲解的。然而如今图形界面的操作系统已经成为主流，想要在像 Windows 这样的图形界面操作系统下使用 C++ 语言编程，几乎所有仅仅学过经典 C++ 语言的人都会有茫然不知所措的感觉。

本书针对这些特点，从全新的角度讲解 C++ 语言，力求让大多数人在短时间内迅速掌握 C++ 语言，并能顺利地过渡到 Windows 图形界面环境下的编程工作中。其特色如下：

- 在 C++ 语言的众多版本中，选择 Microsoft 公司的 Visual C++ 作为教学语言。Visual C++ 系列无疑是 Windows 操作系统下最重要的 C++ 语言开发平台，从一开始就使用 Visual C++ 作为编程环境，除了能掌握 C++ 语言本身外，更为将来彻底掌握 Visual C++ 和 MFC 编程打下基础。
- 所有的例子均是在图形界面环境下编写的，同用户交互的输入/输出界面都是 Windows 标准的图形界面。也就是说，这是一本从一开始就在图形界面下讲授 C++ 语言的书，使用户摆脱了以前必须的从字符界面到图形界面的过渡。在学完本书

后，用户实际上已经能够在 Windows 环境下使用 Visual C++进行一些简单的软件开发工作了。

- 本书是将 C++语言作为完整的程序设计语言来讲解的，用户无须有 C 语言基础。由于将 C++语言作为完整的程序设计语言来讲解，就能从一开始就将面向对象的观点和方法融会其中，最大限度地让用户避免受到面向过程的思想的影响，写出真正的面向对象的程序来。

本书的用户对象

本书可供稍有程序基础知识的人士作为自学 C++语言的教材，也可供高等院校计算机专业的师生作教学参考之用。

目 录

第 1 章 欢迎进入 Visual C++ 的编程世界.....	1
1.1 C++ 程序设计语言与 Visual C++	1
1.1.1 程序设计语言	1
1.1.2 Visual C++ 的特色.....	2
1.2 Visual C++ 的集成开发环境	3
1.2.1 启动 Visual C++.....	3
1.2.2 Visual C++ 的主窗口	3
1.2.3 项目工作台窗口	3
1.2.4 源代码编辑器	4
1.2.5 编译器和链接程序	5
1.2.6 资源编辑器	5
1.2.7 调试器	5
1.3 第 1 个 Visual C++ 程序	6
1.3.1 使用 MFC AppWizard 建立一个新项目	6
1.3.2 添加代码	8
1.3.3 编译	8
1.3.4 运行与调试	9
1.4 练习题	9
第 2 章 C++ 程序设计基础知识.....	11
2.1 数在计算机中的表示.....	11
2.1.1 数制与不同数制间的转换.....	11
2.1.2 计算机如何存储数据	16
2.2 信息在计算机中的表示.....	17
2.3 数据类型	18
2.3.1 经典的 C++ 基本数据类型	18
2.3.2 Visual C++ 自定义的数据类型	19
2.4 常量	20
2.4.1 整型常量	20
2.4.2 实型常量	21
2.4.3 字符型常量	21
2.4.4 字符串型常量	22
2.4.5 符号常量	22

2.5 变量	23
2.5.1 变量的命名	23
2.5.2 变量的声明	24
2.5.3 变量的赋值	25
2.5.4 变量初始化	27
2.5.5 常数型变量	27
2.5.6 匈牙利变量命名法	27
2.6 运算符与表达式	28
2.6.1 算术运算符与算术表达式	28
2.6.2 关系运算符与关系表达式	30
2.6.3 逻辑运算符与逻辑表达式	31
2.6.4 条件运算符与条件表达式	33
2.6.5 位运算符与位运算表达式	34
2.6.6 赋值运算符与赋值表达式	35
2.6.7 逗号运算符与逗号表达式	35
2.6.8 sizeof 运算符	36
2.6.9 运算符优先级总表	36
2.7 类与对象的基本概念	37
2.7.1 类、属性与方法	37
2.7.2 对象	38
2.8 C++语言程序的基本结构	38
2.9 C++语言的编译预处理	40
2.9.1 #define 语句	40
2.9.2 #undef 语句	41
2.9.3 #include 语句与头文件	41
2.9.4 #ifdef ... #else ... #endif 语句	41
2.10 练习题	43
第3章 结构化程序设计	47
3.1 C++语言的基本语句	47
3.1.1 变量或对象声明语句	47
3.1.2 表达式语句	48
3.1.3 复合语句	48
3.1.4 函数调用语句	49
3.1.5 对象的数据成员访问、成员函数调用语句	50
3.1.6 空语句	51
3.1.7 注释	52
3.2 程序设计流程图	53
3.3 顺序结构程序设计	54
3.4 选择结构程序设计	55

3.4.1 if 语句.....	55
3.4.2 switch 语句.....	60
3.5 循环结构程序设计.....	62
3.5.1 for 语句	63
3.5.2 while 语句	66
3.5.3 do...while 语句	67
3.5.4 循环的嵌套	68
3.5.5 break 语句和 continue 语句.....	69
3.6 Visual C++的程序调试.....	70
3.6.1 调试器界面	71
3.6.2 程序调试实例	73
3.7 练习题	73
第 4 章 数组	77
4.1 一维数组	77
4.1.1 一维数组的声明	77
4.1.2 访问一维数组中的元素	78
4.1.3 数组元素的初始化	79
4.2 多维数组	81
4.2.1 多维数组的声明	81
4.2.2 访问多维数组中的元素	81
4.3 数组元素排序	86
4.3.1 选择排序法	86
4.3.2 冒泡排序法	88
4.4 数组元素查找	91
4.4.1 顺序查找法	91
4.4.2 二分查找法	92
4.5 传统的 C++语言的字符串.....	93
4.5.1 字符型数组	93
4.5.2 字符串的运算	94
4.6 练习题	96
第 5 章 函数	99
5.1 函数的定义与使用.....	99
5.1.1 函数的定义	99
5.1.2 函数的参数	102
5.2 变量的作用域	110
5.2.1 局部变量和全局变量	110
5.2.2 动态变量和静态变量	113
5.3 递归	113

5.4	内部函数与外部函数.....	117
5.5	标准库函数	119
5.5.1	调用 Visual C++的标准库函数.....	119
5.5.2	常用的标准库函数	119
5.6	练习题	121
第 6 章	指针.....	125
6.1	指针的概念	125
6.2	指针型变量	126
6.2.1	指针型变量的声明	127
6.2.2	指针型变量的引用	127
6.2.3	指针型变量作为函数参数.....	127
6.2.4	指针值作为函数参数的返回值.....	129
6.3	指针的运算	130
6.3.1	指针的赋值运算	130
6.3.2	指针的算术运算	132
6.3.3	指针的关系运算	133
6.4	数组与指针	133
6.4.1	数组的实质	133
6.4.2	通过指针引用数组元素	134
6.4.3	指向多维数组的指针	136
6.4.4	指针型变量的数组	138
6.4.5	字符串与指针	138
6.5	指针的指针	139
6.6	引用型变量	140
6.6.1	引用型变量的定义	140
6.6.2	引用型变量的引用	142
6.6.3	引用型变量作为函数参数.....	142
6.6.4	引用型变量作为函数的返回值.....	143
6.7	函数的指针	143
6.8	练习题	145
第 7 章	复杂数据类型	149
7.1	结构体	149
7.1.1	结构体和结构体变量	149
7.1.2	结构体变量的初始化和引用.....	151
7.1.3	结构体数组	154
7.1.4	结构体指针与引用	157
7.1.5	位域(位成员).....	159
7.2	共用体	160

7.2.1 共用体和共用体变量	160
7.2.2 共用体变量的引用	161
7.3 枚举类型	162
7.3.1 枚举类型和枚举型变量	162
7.3.2 枚举型变量的引用	164
7.4 自定义数据类型	167
7.5 选择题	168
第 8 章 动态内存分配	171
8.1 动态内存分配的概念	171
8.2 进行动态内存分配	172
8.2.1 new 运算符	172
8.2.2 delete 运算符	173
8.2.3 动态内存分配函数	175
8.3 动态内存分配实例——链表	176
8.3.1 动态数据结构与链表	176
8.3.2 链表的建立	177
8.3.3 链表结点的插入	177
8.3.4 链表结点的删除	181
8.3.5 链表元素的查找	182
8.4 练习题	182
第 9 章 类	185
9.1 类的概念	185
9.2 类的定义	185
9.2.1 类的结构	185
9.2.2 类的数据成员	188
9.2.3 类的类成员	189
9.2.4 类的成员函数	189
9.2.5 构造函数和析构函数	191
9.2.6 在 Visual C++ 项目中添加一个类	194
9.3 生成类的实例——对象	198
9.3.1 生成类的实例：对象	198
9.3.2 对象指针和对象的动态生成	200
9.3.3 对象数组	203
9.3.4 类与函数参数	204
9.3.5 类与函数返回值	207
9.4 类的内联成员函数	208
9.5 this 指针	210
9.6 类的静态成员	211

9.7 练习题	213
第 10 章 重载与类的友元	217
10.1 函数的重载	217
10.2 运算符重载	222
10.2.1 运算符与函数的对应关系	223
10.2.2 重载单目运算符	224
10.2.3 重载双目运算符	227
10.2.4 类型转换成员函数	230
10.3 类的友元	231
10.3.1 类的友元函数	231
10.3.2 类的友元类	237
10.4 练习题	237
第 11 章 继承与多态	241
11.1 类的继承	241
11.1.1 类继承的基本知识	241
11.1.2 子类的构造与析构	248
11.1.3 成员函数的覆盖	250
11.1.4 从多个父类继承	254
11.2 多态	255
11.2.1 用父类的指针指向子类对象	256
11.2.2 虚拟成员函数	256
11.2.3 虚拟析构函数	262
11.2.4 纯基类	263
11.2.5 多态用法实例	264
11.3 练习题	272
第 12 章 模板	277
12.1 函数的模板	277
12.2 类的模板	281
12.2.1 类模板的定义	281
12.2.2 派生类和类模板	287
12.3 练习题	289
第 13 章 C++语言的异常处理	291
13.1 异常处理的概念	291
13.2 C++语言异常处理的结构	292
13.2.1 throw 与 catch	292
13.2.2 异常处理器的嵌套	295
13.2.3 放置异常处理器	296
13.3 Win32 异常及其处理	300

13.4 练习题	303
第 14 章 Visual C++ 的文件操作	307
14.1 文件	307
14.1.1 文件的概念	307
14.1.2 文件的分类	308
14.2 使用 CFile 类读写文件	309
14.2.1 CFile 类概述	309
14.2.2 CFile 类的构造函数	309
14.2.3 打开文件	310
14.2.4 关闭文件	311
14.2.5 读写文件	311
14.2.6 文件的定位	318
14.2.7 文件的状态	321
14.3 使用 CStdioFile 类读写文件	321
14.3.1 构造函数	321
14.3.2 文件的打开和关闭	322
14.3.3 文件的读写	322
14.3.4 其他操作	323
14.4 内存文件和 CMemFile 类	324
14.4.1 CMemFile 类的构造函数	324
14.4.2 内存文件的读写	324
14.4.3 其他操作	326
14.5 C++流简介	326
14.5.1 C++流的概念	326
14.5.2 文件流类	327
14.6 练习题	328
第 15 章 使用 MFC 类库编写简单的 Windows 应用程序	331
15.1 Windows 编程与 MFC	331
15.1.1 Windows 应用程序编程接口	331
15.1.2 MFC 类库与 Visual C++ 的 ClassWizard	332
15.2 MFC 体系结构	333
15.3 基于对话框的 MFC 应用程序	335
15.3.1 计算机与用户的交互工具: 对话框	335
15.3.2 使用 AppWizard 生成基于对话框的 MFC 应用程序	335
15.3.3 添加控件	337
15.3.4 编写控件命令处理代码	341
15.3.5 使用多个对话框	345
15.3.5 其他控件简介	354

15.3.6 Windows 的公用对话框	361
第 16 章 文档-视图结构简介	367
16.1 文档-视图结构概述	367
16.2 窗口框架	370
16.2.1 CFrameWnd 类	370
16.2.2 菜单	370
16.2.3 工具栏	374
16.2.4 状态栏	375
16.3 文档和视图	378
16.3.1 CView 类	378
16.3.2 CDocument 类	380
16.3.3 打印和打印预览	389

第 1 章 欢迎进入 Visual C++ 的编程世界

本章主要介绍 C++ 语言和 Visual C++ 编程环境，并通过一个实例讲解使用 Visual C++ 编程环境的方法。

学习重点

- 程序设计语言的分类和特点
- 熟悉 Visual C++ 集成开发环境
- 用 AppWizard 建立一个项目的方法

1.1 C++ 程序设计语言与 Visual C++

1.1.1 程序设计语言

计算机系统由硬件和软件组成。硬件是计算机的“躯体”，是看得见的电子线路；软件则是计算机的“灵魂”，它控制着计算机的电路，完成一定的功能。

现代的软件理论认为，计算机软件是计算机程序和与程序配套的技术文档、用户文档的统一体，但从功能上说，控制计算机硬件电路完成各种任务的只有计算机程序。所以，想要利用计算机解决各种实际问题，最重要的是要学会编写计算机程序。

人们使用计算机程序设计语言编写计算机程序。计算机程序设计语言有 3 种：机器语言、汇编语言和高级语言。其中机器语言和汇编语言是面向具体硬件的编程语言，被通称为低级语言。低级语言同硬件紧密相连，不同的 CPU 上使用的低级语言是不一样的，用低级语言在一种 CPU 系统上编写的程序不能在另一种 CPU 上运行。在低级语言中，机器语言是计算机唯一能直接识别的语言，它由二进制机器码组成，很难学习、记忆；汇编语言是机器语言的变种，它采用某些符号代表难记的机器码，相对于机器语言而言，学习起来较为容易。低级语言的优点在于运行速度快，但缺点也是很明显的：低级语言程序不容易编写、不容易调试，可移植性很差。所以，除了某些对时序要求很高的场合外，一般不提倡使用低级语言编写程序。

相对于低级语言的计算机程序设计语言是高级语言，它接近于人类自然语言，基本上符合人类通常的思维习惯。例如，要计算 5 乘以 6 再加上 7 的值，如果使用汇编语言来写程序，则需要写以下 3 条语句：

```
mov ax,5  
mul 6  
add ax,7
```

但在高级语言中只要写成

```
x=5 * 6 + 7
```

就可以了，非常直观。另外，高级语言采用了人类语言中的某些元素来描述控制过程和程序结构，使高级语言具有易学易用的优点。

高级语言接近于人类语言，显然它不能被机器直接识别的。要运行用高级语言编写的程序，要么必须通过编译软件将其“翻译”成机器语言程序然后再运行；要么通过解释软件边解读程序，边告诉计算机该做什么。这两种运行高级语言程序的方法分别称为编译执行和解释执行。使用高级语言编写的程序文件被称为“源代码文件”，通常称作“源程序”或“源代码”。

一般地说，采用编译执行的方法运行程序在速度上要比采用解释执行的方法快，但采用解释执行的方法比较容易进行程序调试。目前，各种高级语言普遍采用了编译执行的方法，只有少数几种采用了解释执行的方法。

相对于低级语言而言，高级语言的缺点在于程序运行速度慢，并且对系统的资源(内存空间)占用大。但是，随着硬件技术的提高、硬件价格的下降，这些缺点早已不是程序设计中最主要的矛盾，所以，现代的应用软件开发使用的基本上是高级语言。

从计算机诞生到现在的近 50 年间，计算机科学家设计出了大量不同的高级语言，以满足不同领域的需要。在早期的高级程序设计语言中，著名的有面向初学者的 Basic 语言、工程计算用的 Fortran 语言、具有数学般严谨风格的 Pascal 语言和高效灵活的 C 语言等，其中影响面最广的当属 C 语言。

C 语言最初的版本是美国贝尔实验室的 Dennis Ritchie 等在 1972 年完成的。设计 C 语言的初衷是为了改写 UNIX 操作系统，由于 UNIX 系统的成功，C 语言也就随之流行开来。C 语言的主要特点在于语法简洁、灵活，运行效率高，甚至可同低级语言相媲美，再加上丰富的运算符和数据结构，使得 C 语言成为深受广大程序员喜爱的高级程序设计语言。

近年来，随着面向对象方法的兴起，又出现了如 C++、Smalltalk、Java 等面向对象的程序设计语言，而 C++ 是其中的佼佼者。现在使用的大多数应用软件是用 C++ 语言开发出来的。

C++ 语言是在 C 语言的基础上增加面向对象的程序设计元素而成的，它继承了 C 语言高效灵活的特性，从而吸引了大量的 C 程序员转向 C++ 编程。很多著名的软件厂商都推出了 C++ 语言的开发环境，在 Windows 操作系统下比较著名的有 Microsoft 公司的 Visual C++ 和 Inprise 公司的 C++ Builder。本书选用 Visual C++ 作为环境来介绍 C++ 语言。

1.1.2 Visual C++ 的特色

Visual C++ 是 Microsoft 公司推出的功能最强大、最复杂的语言产品之一，它是目前为止在 Windows 环境下进行大型软件开发的首选。其特点如下：

- Visual C++ 的语法符合 ANSI C++ 标准，并在此基础上针对 Windows 操作系统增加了一些语句；
- 集成了 MFC(Microsoft Foundation Class)类库，MFC 封装了 Windows API 函数和消息，使程序员可以使用 MFC 高效率地开发出各种应用程序；
- 提供了 MFC AppWizard，可方便地生成程序框架；

- 提供了基于 MFC 的 ClassWizard，通过它可以轻松地完成对各种 MFC 类的使用与维护。

1.2 Visual C++的集成开发环境

现代的计算机语言产品都提供了集成开发环境。Visual C++的集成开发环境提供了源程序编辑、编译、调试和运行等功能，是进行程序设计的工作场所。

1.2.1 启动 Visual C++

Visual C++是 Microsoft Visual Studio 的一部分，安装 Microsoft Visual Studio 很简单，只要照着安装过程中的提示，即可顺利完成安装工作。建议使用所有的默认设置安装 Microsoft Visual Studio。安装完 Microsoft Visual Studio 后，Windows【开始】菜单中的【程序】子菜单中会出现 Microsoft Visual Studio 子菜单。

要启动 Visual C++，只需选择【开始】|【程序】|Microsoft Visual Studio|Microsoft Visual C++ 6.0 命令即可。

1.2.2 Visual C++的主窗口

Visual C++的主窗口如图 1.1 所示，这是在 File 菜单中选择 Open Workspace 命令打开一个项目后所显示的典型窗口。

在 Visual C++主窗口中，除了通常的菜单、工具栏和状态栏外，还有项目工作台窗口、正文编辑窗口和输出窗口。

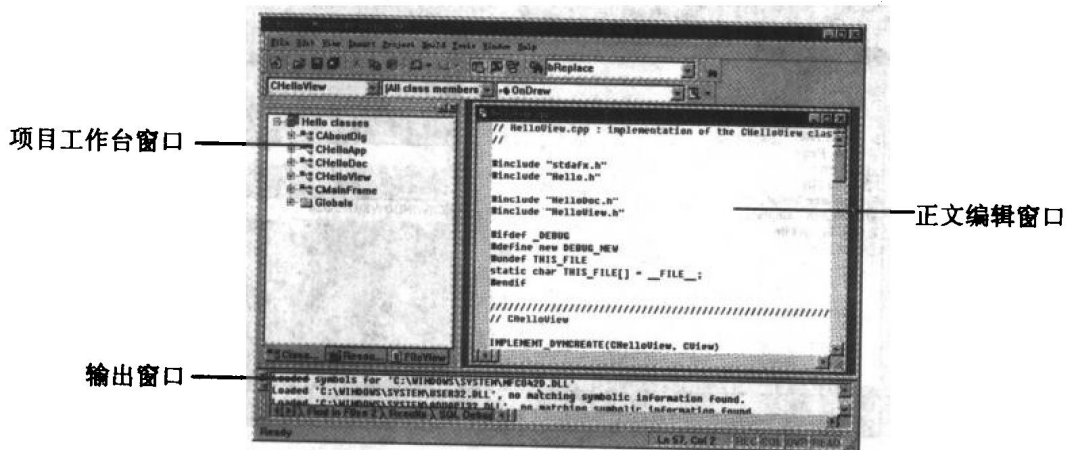


图 1.1 Visual C++的主窗口

1.2.3 项目工作台窗口

Visual C++以项目(project)为单位对开发中的软件进行管理，一个项目通常包括所有的源程序、资源文件和其他的支持文件。项目工作台是管理这些文件的界面，通过在项目工作台上操作，用户可以调出任何在当前项目中所需要的文件进行编辑、修改。在项目工作

台窗口上有 3 个标签, 分别是 ClassView(类)、ResourceView(资源)和 FileView(文件)。单击 ClassView 标签, 则在项目工作台窗口中将使用树型结构显示当前项目中所有的类和类成员(在第 9 章中将详细介绍类的概念); 单击 ResourceView 标签, 则显示以资源类型组织的所有资源; 单击 FileView 标签, 则当前项目中用到的所有的文件将按照文件的类型显示在项目工作台窗口中。

项目工作台窗口是可停靠(Docking)的窗口, 通常停靠在主窗口框架上, 成为窗口框架的一部分。如果要使项目工作台窗口成为普通的 MDI 子窗口, 必须先在项目工作台窗口上右击以弹出快捷菜单, 然后在快捷菜单上选择 Docking View 命令。

1.2.4 源代码编辑器

源代码编辑器是 Visual C++ 自带的一个很出色的文本编辑器, 它不仅可以编辑 C++ 源程序, 也可以编辑文本文件。在编辑 C++ 源程序时, 正文编辑窗口中将根据 C++ 语法对文本加以不同的颜色, 这是程序员非常喜欢的设计。

选择 File | New 命令, 将出现 New 对话框。单击对话框上的 Files 标签, 此时对话框的外观如图 1.2 所示。在对话框左边的文件类型列表框中包括了多种 Visual C++ 能识别的文件类型, 其中的 C/C++ Source File(C/C++ 源文件)、C/C++ Header File(C/C++ 头文件)和 Text file(文本文件)等是 Visual C++ 的源代码编辑器可以编辑的。选择 C++ Source File 选项, 然后在右边的 File 和 Location 文本框中分别输入文件名和路径, 选中 Add to project 复选框并在该复选框下面的下拉列表中选择当前的项目, 即可创建一个新的 C++ 源程序文件并将其加入到项目中, 同时, 将增加一个正文编辑窗口, 用于编辑新建的 C++ 源程序文件。

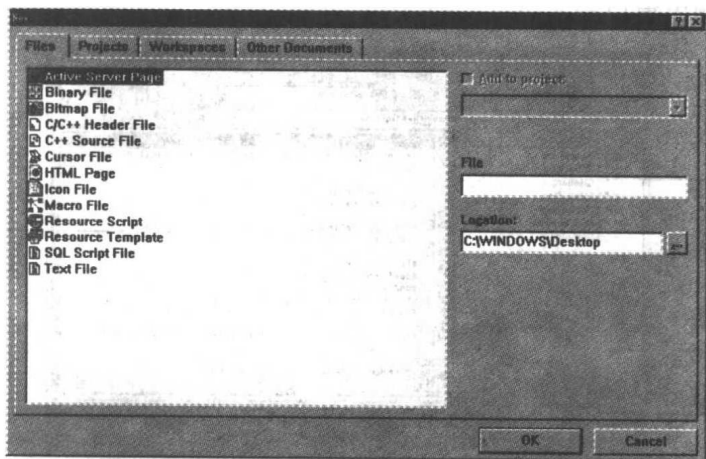


图 1.2 建立新文件

如果要编辑一个在当前项目中已经存在的 C++ 源程序文件, 先单击项目工作台窗口中的 FileView 标签, 然后在不同类型的文件列表中选择想要编辑的文件并双击即可。

源代码编辑器的用法同普通的文本编辑器完全一样, 只要掌握如【记事本】这样简单的文本编辑器的用法, 就不难适应 Visual C++ 的源程序编辑环境。