



清华大学计算机基础教育课程系列教材

Visual C++

面向对象与可视化程序设计习题解析与编程实例

黄维通 编著



清华大学出版社

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





清华大学计算机基础教育课程系列教材

Visual C++

面向对象与可视化程序设计习题解析与编程实例

黄维通 编著



清华大学出版社

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

JSS57 60

(京)新登字 158 号

内 容 简 介

本书是“清华大学计算机基础教育课程系列教材”中的《Visual C++ 面向对象与可视化程序设计》一书的配套教材。书中全面具体地对该教材中各章的习题做了详细解答,对每一道题都进行了必要的分析和代码注释,力求通过实例让读者全面掌握面向对象与可视化程序设计的思路 and 开发技巧。同时,为让读者更好地掌握“Visual C++ 面向对象与可视化程序设计”的思想和方法,还增加了部分紧扣相关知识点的典型实例。书后附有使用 API 工具、MFC 工具和 Debug 工具开发、调试应用程序的基本步骤。

本书涉及的知识点包括 Visual C++ 的基础知识及开发环境、Windows 编程基础知识、应用 API 函数进行可视化编程的基本方法和应用 MFC 进行可视化编程的思想方法。

本书通俗易懂,重点突出,偏重应用。不仅适合作为大专院校理工科学生学习 Visual C++ 面向对象编程的辅助教材,还适合于初步掌握 Visual C++ 的编程人员作为习题教材,同时也可供有关科研及开发人员参考。

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++ 面向对象与可视化程序设计习题解析与编程实例

作 者: 黄维通 编著

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: 世界知识印刷厂

发行者: 新华书店总店北京发行所

开 本: 787×1092 1/16 **印张:** 24 **字数:** 550 千字

版 次: 2000 年 11 月第 1 版 2000 年 11 月第 1 次印刷

书 号: ISBN 7-900630-58-9

印 数: 0001~6000

定 价: 38.00 元(含盘)

序

计算机科学技术的发展不仅极大地促进了整个科学技术的发展,而且明显地加快了经济信息化和社会信息化的进程。因此,计算机教育在各国备受重视,计算机知识与能力已成为 21 世纪人才素质的基本要素之一。

清华大学自 1990 年开始将计算机教学纳入基础课的范畴,作为校重点课程进行建设和管理,并按照“计算机文化基础”、“计算机技术基础”和“计算机应用基础”三个层次的课程体系组织教学:

第一层次“计算机文化基础”的教学目的是培养学生掌握在未来信息化社会里更好地学习、工作和生活所必须具备的计算机基础知识和基本操作技能,并进行计算机文化道德规范教育。

第二层次“计算机技术基础”是讲授计算机软硬件的基础知识、基本技术与方法,从而为学生进一步学习计算机的后续课程,并利用计算机解决本专业及相关领域中的问题打下必要的基础。

第三层次“计算机应用基础”则是讲解计算机应用中带有基础性、普遍性的知识,讲解计算机应用与开发中的基本技术、工具与环境。

以上述课程体系为依据,设计了计算机基础教育系列课程。随着计算机技术的飞速发展,计算机教学的内容与方法也在不断更新。近几年来,清华大学不断丰富和完善教学内容,在有关课程中先后引入了面向对象技术、多媒体技术、Internet 与互联网技术等。与此同时,在教材与 CAI 课件建设、网络化的教学环境建设等方面也正在大力开展工作,并积极探索适应 21 世纪人才培养的教学模式。

为进一步加强计算机基础教学工作,适应高校正在开展的课程体系与教学内容的改革,及时反映清华大学计算机基础教学的成果,加强与兄弟院校的交流,清华大学在原有工作的基础上,重新规划了“清华大学计算机基础教育课程系列教材”。

该系列教材有如下几个特色:

1. 自成体系:该系列教材覆盖了计算机基础教学三个层次的教学内容。其中既包括所有大学生都必须掌握的计算机文化基础,也包括适用于各专业的软、硬件基础知识;既包括基本概念、方法与规范,也包括计算机应用开发的工具与环境。

2. 内容先进:该系列教材注重将计算机技术的最新发展适当地引入教学中来,保持了教学内容的先进性。例如,系列教材中包括了面向对象与可视化编程、多媒体技术与应用、Internet 与互联网技术、大型数据库技术等。

3. 适应面广：该系列教材照顾了理、工、文等各种类型专业的教学要求。

4. 立体配套：为适应教学模式、教学方法和手段的改革，该系列教材中多数都配有习题集和实验指导、多媒体电子教案，有的还配有 CAI 课件以及相应的网络教学资源。

本系列教材源于清华大学计算机基础教育的教学实践，凝聚了工作在第一线的任课教师的教学经验与科研成果。我希望本系列教材不断完善，不断更新，为我国高校计算机基础教育做出新的贡献。



1999 年 12 月

注：周远清，现任教育部副部长，原清华大学副校长、计算机专业教授。

前言

随着计算机技术的飞速发展,社会对人才的计算机应用与开发水平的要求也日益提高,为适应此形势,高校的计算机基础教学内容也在不断的改革之中。作为清华大学 985 重点攻关项目的“计算机基础系列课程改革”的组成部分之一的教材建设,也正是为适应此形势而进行的,本书是该项目中教材建设的组成部分。

目前,C 语言已经成为高校理工科学生的必修或选修课程,但 C 语言是面向过程的编程语言,随着软件工程技术不断发展,应用面向对象的编程技术已经成为当今软件开发的重要手段之一,尤其是 Visual C++ 的出现,大大推进了面向对象与可视化编程技术的应用与发展。因此,掌握“面向对象与可视化程序设计”的内容与方法已经成为对大学生的计算机应用与开发能力的要求之一。

本书作为清华大学出版社出版的《Visual C++ 面向对象与可视化程序设计》一书的配套教材,提供了《Visual C++ 面向对象与可视化程序设计》教材中各章习题的详细解答与分析。为让读者更好地掌握面向对象与可视化程序设计的思想和方法,还增加了部分紧扣相关知识点的典型实例。书后附有使用 API 工具、MFC 工具和 Debug 工具开发、调试应用程序的基本步骤。

本书涉及的知识点包括 Visual C++ 的基础知识及开发环境、Windows 编程基础知识、应用 API 函数进行可视化编程的基本方法和应用 MFC 进行可视化编程的思想方法。

本书特点是对《Visual C++ 面向对象与可视化程序设计》一书中涉及的知识点以样例的形式进行全面复习,每一道题都进行了必要的分析;同时,所有代码均给出详细的注释,力求通过实例让读者全面掌握面向对象与可视化程序设计的思路 and 开发技巧。本书的习题都是根据教学特点精心安排的,所有的习题和例题都在 Windows 98 及 Visual C++ 6.0 的环境下调试运行通过。

本书面向各大专院校本科生、研究生及科研院所的计算机开发与应用人员,既适合于初学者作为辅导教材,也适合于初步掌握 Visual C++ 的编程人员作为习题教材,同时也可供有关科研及开发人员参考。

本书在编写过程中得到清华大学计算机与信息管理中心王行言教授的具体指导和帮助,在此表示衷心的感谢。刘哲博士对本书中的例题调试做了大量具体的工作,此外,参加例题调试及审核工作的还有徐宏生、刘宏、刘启仑、章志林、李浩等同志,清华大学园丁园的张志东、林常明两位同志为本书稿的录入做了大量的工作。

由于作者水平有限,缺点和错误在所难免,恳请读者批评指正。

谢谢选择阅读本书的读者!

黄维通

2000 年 5 月于清华园

目 录

第 1 章	C++ 基础知识	1
第 2 章	Visual C++ 6.0 开发环境	4
第 3 章	Windows 编程基础	10
第 4 章	Windows 的图形设备接口及 Windows 绘图	12
第 5 章	文本的输出方法与字体的设置	43
第 6 章	Windows 应用程序对键盘与鼠标的响应	71
第 7 章	资源在 Windows 编程中的应用	106
第 8 章	Windows 标准控件在可视化编程中的应用	143
第 9 章	文件及剪贴板操作	201
第 10 章	MFC 基础知识	219
第 11 章	编辑框类及其应用	225
第 12 章	菜单类及其应用	257
第 13 章	滚动条类及其应用	287
第 14 章	按钮类与列表框类及其应用	312
第 15 章	工具条类及其应用	334
第 16 章	文档-视图结构及其应用	349
附录 A	使用 API 工具开发应用程序的基本步骤	364
附录 B	使用 MFC 工具开发应用程序的基本步骤	367
附录 C	VC++ 6.0 Developer Visual Studio 调试工具简介	372

第 1 章

C++ 基础知识

【1-1】 C++语言的基本数据类型包含哪些？

解答：C++语言基本数据类型有 bool(布尔型)、char(字符型)、int(整型)、float(浮点型,表示实数)、double(双精度浮点型,简称双精度型)。除了 bool 型外,这些数据类型又可以分为两大类:即整型和浮点型。char 从本质上说也是整数类型,它是长度为 1 字节的整型数,通常用来存放字符的 ASCII 码。

【1-2】 C++语言中函数的作用是什么？

解答：在面向对象的程序设计中,函数有着非常重要的作用,函数是模块划分的基本单位,是程序设计中的基本抽象单元,是对功能的抽象。一个复杂的系统往往需要划分为若干个子系统,然后分别进行开发和调试。通常我们将相对独立的、经常使用的功能抽象为函数。使用函数时可以只考虑函数的功能和使用方法而不必关心它的具体实现。这样有利于代码的重用,可以提高程序的开发效率,增强程序的可靠性,方便程序的维护。

【1-3】 C++语言中的指针有哪些类型？它们都如何定义？

解答：C++语言中所有的数据类型基本上都可以定义为指针类型,定义的方法很简单:只要在声明变量时在变量名前加“*”即可。

例如:定义一个指向整型数的指针变量 nAge 的形式为:int * nAge;定义一个指向浮点型数据的指针变量 fScore 的形式为:float * fScore;定义一个指向对象 Student 的指针变量 studStudent1 的形式为:Class Student * studStudent1;等等。同样我们也可以定义一个指向指针变量的指针变量 pChar 的形式为:Char * * pChar,此变量指向指针变量 * pChar。

【1-4】 在C++语言中如何进行“类”的定义？

解答：类是C++语言的精华,是进行封装和数据隐藏的工具。通过它把逻辑上相关的实体联系起来,并具备从外部对这些实体进行访问的手段。定义一个类的语法格式如下:

```
Class 类名:基类名
{
private:
```



```
    私有成员数据及函数;  
protected:  
    保护成员数据及函数;  
public:  
    公共成员数据及函数;  
}类的对象声明;
```

【1-5】 构造函数和析构函数的功能是什么？如何创建构造函数和析构函数？

解答：构造函数是一个特殊的成员函数，它主要用来为对象分配内存空间，对类的成员进行初始化并执行对象的其他内部管理操作。构造函数的特点是它的名字同类名相同，当定义该类的对象时，自动调用该函数完成对该对象的初始化操作，构造函数可以重载。构造函数的创建和其他的函数基本相同，但有以下特点：无返回值，可以带参数也可以不带参数，在实际的应用中如果没有给类定义构造函数，则系统自动生成一个默认的构造函数，该函数没有参数，只是简单的把对象中的每个实例变量初始化为 0。例如：

```
Class My- Class  
{  
    float a,b;  
    public:  
    My- Class(...)           //构造函数,可以带有参数也可以不带参数  
    {  
        ...                 //函数体  
    }  
    ...  
}
```

析构函数也是类中的特殊成员函数，与定义它的类具有相同的名字，但要在前面加上一个波浪号(~)。析构函数没有参数也没有返回值。它不能重载，因此一个类中只能有一个析构函数。析构函数执行与构造函数相反的操作，通常用来释放分配给对象的存储空间。当程序超出对象的作用域时，或者当对一个类指针进行 delete 操作时，系统将自动调用析构函数。例如：

```
Class My- Class  
{  
    float a,b;  
    public:  
    My- Class(...)           //构造函数,可以带有参数也可以不带参数  
    {  
        ...                 //函数体  
    }  
    ~My- Class()            //析构函数,不可以带参数  
    {  
        ...                 //函数体  
    }  
}
```

```
    }  
    ...  
}
```

【1-6】 C++语言中“派生”的含义是什么？

解答：“派生”实质上是指在保持原有类特性的基础上，进行更具体、更详细的类的定义。派生类从基类中继承所有的公共部分并可以增加数据成员和成员函数，对原有基类的属性和方法进行修改和扩充。类的继承和派生的层次结构，可以说是人们对自然界中事物进行分类分析的认识过程在程序设计中的体现。例如：我们现在有了一个交通工具类，其中包含了关于交通工具的共有的信息，现在我们用它来描述汽车显然是不够的，因为汽车有属于自己的特殊的信息，如果从头来定义汽车类那么有许多代码显然是浪费的。我们把汽车类定义称为交通工具类的派生类，自动继承了交通类的信息，然后只编写描述汽车特殊属性的代码即可。这样大大提高了工作效率和代码效率。

【1-7】 C++语言中重载的作用是什么？

解答：重载是C++语言的一个重要特性，它包含函数重载和操作符重载。所谓函数重载是指同一个函数体对应着多个函数的实现。即允许同一个程序中可以声明多个名称相同的函数，这些函数分别完成不同的功能。使用函数重载可以减轻用户的记忆负担，并使程序的结构简单、易懂。操作符重载是指将语言中已有的操作符赋予新的功能，但与该操作符的本来含义不冲突。同样也达到了减轻程序员的记忆负担和使程序结构简单易懂的效果。

第 2 章

Visual C++ 6.0 开发环境

【2-1】 在VC++ 6.0环境中如何创建项目文件？

解答：在VC++ 6.0环境中创建项目文件一般遵循如下步骤：

(1) 首先运行VC++ 6.0。在VC++ 6.0开发环境中单击 File 菜单中的 New 选项（如图 2-1 所示）。

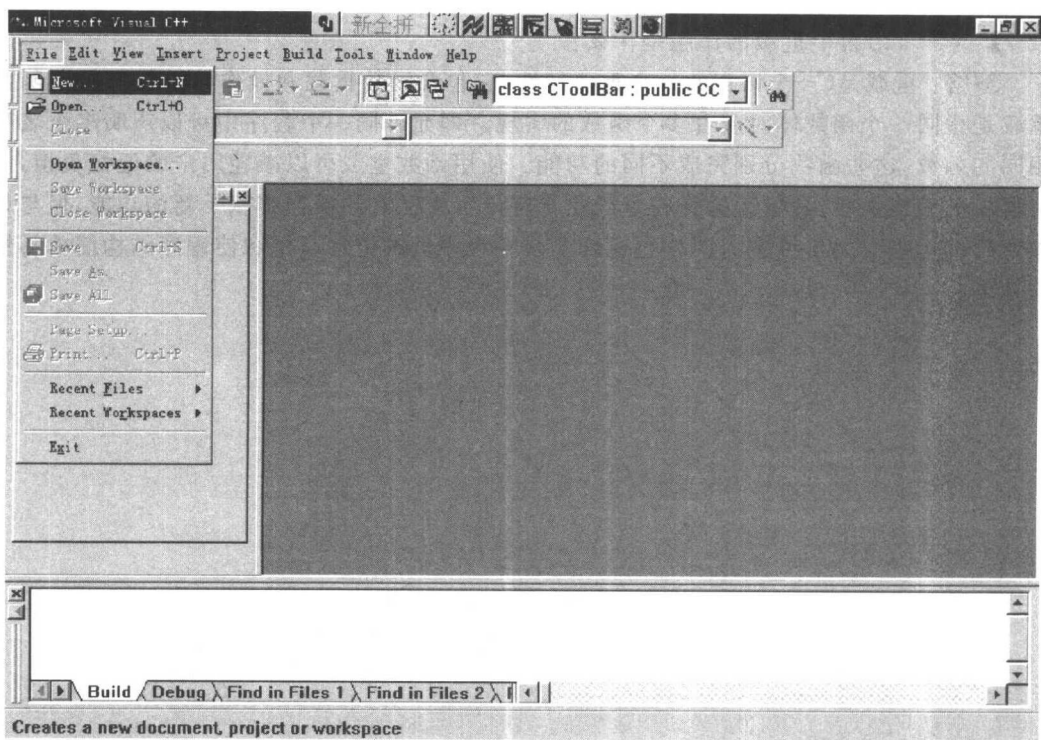


图 2-1 VC++ 6.0开发环境

(2) 此时弹出 New 对话框，在此对话框中选中 Projects 选项卡，该窗口中列出了一系列可以创建的项目类型（如图 2-2 所示）。

VC++ 6.0可以创建的项目类型如表 2-1 所示。

表 2-1 VC++ 6.0 可以创建的 Project 类型

Project 类型	说明	Project 类型	说明
ATL COM Appwizard	创建 ATL 应用程序	Database Project	创建数据库文件
Win32 Dynamic-Link Library	创建 Win32 动态链接库	DevStudio Add-in Wizard	创建自动化宏
Custom AppWizard	创建自定义的 AppWizard	ISAPI Extension Wizard	创建 Internet 服务器或过滤器
Makefile	创建 Make 文件	MFC ActiveX ControlWizard	创建 ActiveX 控件程序
MFC AppWizard(dll)	创建 MFC 动态链接库	MFC AppWizard(exe)	创建 MFC 可执行程序
Win32 Application	创建 Win32 应用程序	Win32 Console Application	创建 Win32 控制台应用程序
Cluster Resource Type Wizard	通过它可以创建两种 Project 类型 (Resource DLL 和 Cluster Administrator Extension DLL)	Utility Project	通过它可以创建一个效用 Project, 该 Project 只作为其他 Project 的一个容器, 减少子 Project 的联编时间, 但它本身并不包含任何文件
Win32 Static Library	创建 Win32 静态库		

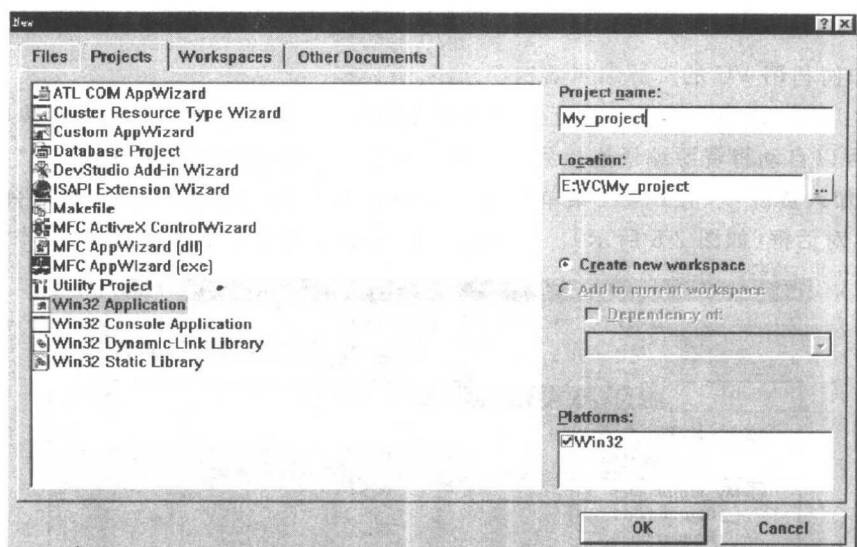


图 2-2 Project 选项卡

我们这门课中常用的项目类型有 Win32 Application 和 MFC AppWizard[exe] 这两种。

(3) 假设在图 2-2 中选择 Win32 Application 后, 单击 Location 编辑框右侧带省略号“...”的按钮, 可以选择该项目文件保存的路径, 如“E:\VC\”。在 Project Name 编辑框中可以输入项目名字。例如: 在图 2-2 中的项目名字为 My_project。

(4) 还可以选择程序运行的平台(Platforms), 默认类型为 Win32。

(5) 如果要添加新的项目到已打开的项目中, 如添加到 My.. project 中, 请选择 Add to current workspace 选项(如图 2-3 所示), 否则 Visual C++ 6.0 将会为你自动创建包含新项目的新的工作区。如果要使新项目成为已有项目的子项目, 请选中 Dependency of 复选框并指定项目名。

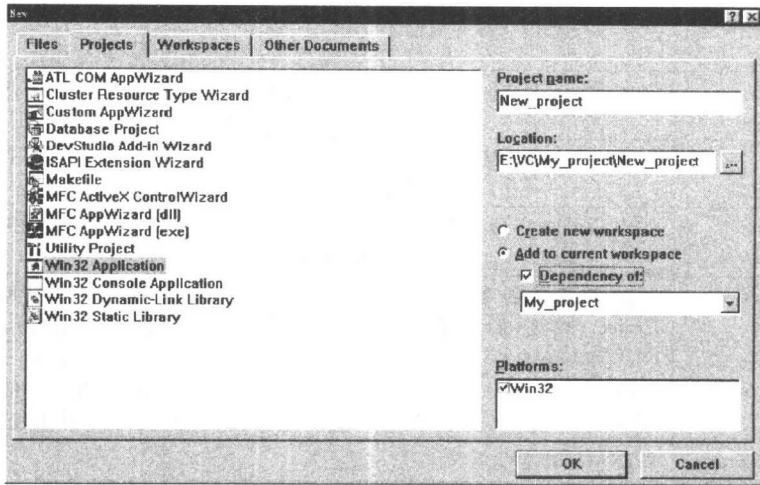


图 2-3 添加一个新的项目

【2-2】 如何利用 VC 的资源编辑器?

解答: VC++ 6.0 提供了具有强大功能的资源编辑器来帮助用户完成资源的创建和修改, 并可以自动将资源编译连接到应用程序中去。主要的操作步骤如下:

(1) 加入资源: 单击 Insert 菜单中的 Resource 选项(如图 2-4 所示)。此时弹出 Insert Resource 对话框(如图 2-5 所示)。在 Insert Resource 对话框中列出了 VC++ 6.0 可以

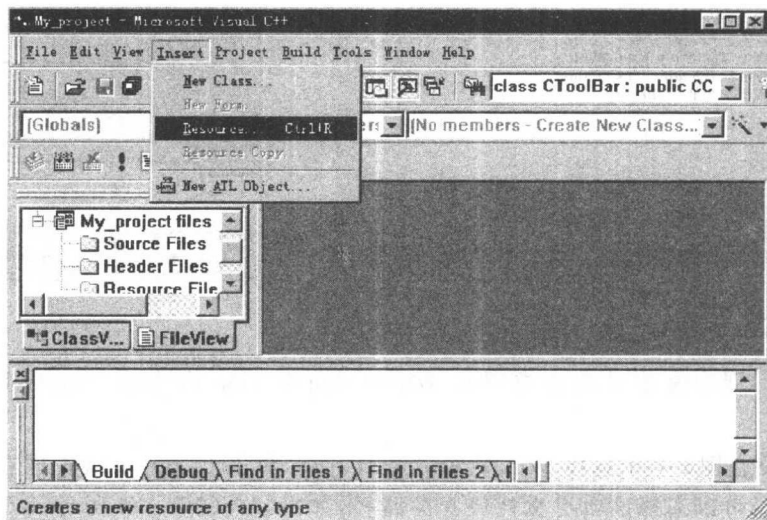


图 2-4 加入资源的操作

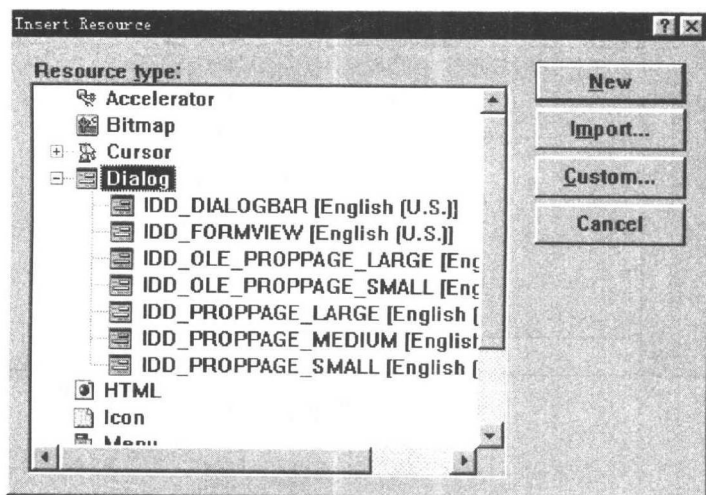


图 2-5 Insert Resource 对话框

创建的资源种类,如 Menu,Dialog 等等。选中一种资源,单击右侧的 New 按钮,可以创建一个新的资源。单击 Import 按钮,可以将一个已存在的资源加入到当前的项目中。在图 2-5 中选择了 Dialog 资源。

(2) 编辑资源:加入资源后就可以编辑它们了。例如,当加入了一个对话框资源后,在 Workspace 窗口的 Resource View 选项卡中,出现了加入的资源(如图 2-6)。双击加入的对话框资源,在右侧的窗口中会出现一个空的对话框和一个工具条,这个工具条中有各种控件,选择需要的控件后,可以在对话框的合适位置用鼠标加入控件。同时在窗口的下方出现了一个辅助工具条,用来帮助我们完成诸如控件定位等工作。在图 2-7 中加入了一个编辑框,图 2-7 中左边两列所示的即为各种供创建的控件图标。

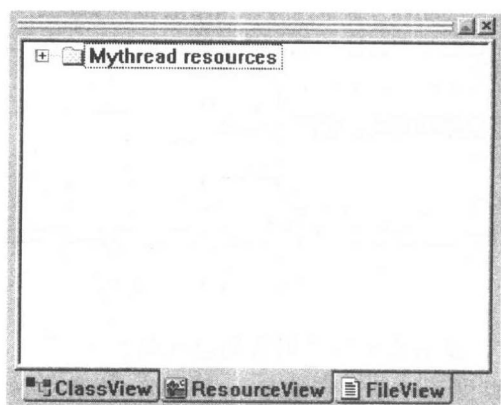


图 2-6 Resource View 选项卡

(3) 修改属性。在需要修改属性的控件上单击鼠标右键,此时弹出快捷菜单(如图 2-8所示)。选择 Properties 选项,在弹出的 Dialog Properties 对话框中可以修改控件的 ID、Caption 等属性(如图 2-9 所示)。

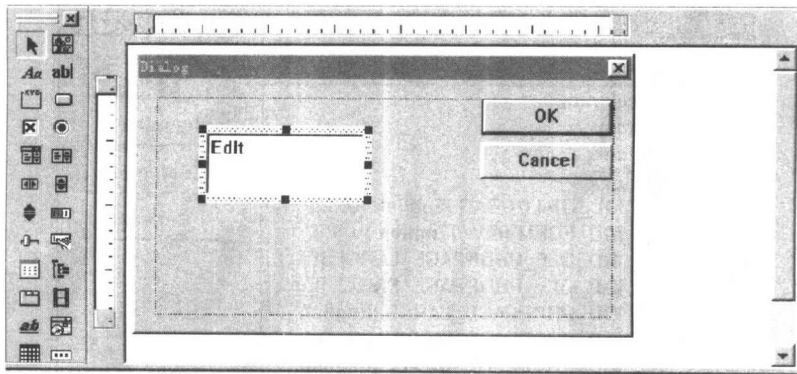


图 2-7 增加 Edit 对话框

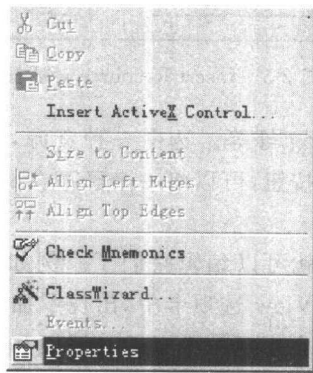


图 2-8 快捷菜单

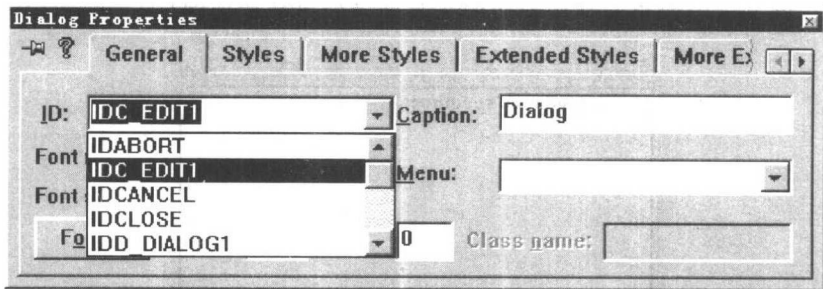


图 2-9 Dialog Properties 对话框

【2-3】 如何充分利用 VC 提供的丰富的联机帮助功能？

解答：VC++ 6.0 提供了一个全新的基于 HTML 的联机帮助系统。每一个主题都用一个独立的 HTML 文档来表达，而且所有的主题都组织起来，根据 InfoViewer 索引文件 (IVI)，形成了 InfoViewer 标题文件 (IVT)。InfoViewer 窗口使用了从 Microsoft Internet Explorer 引用过来的地址码，所以看起来有一些像人们熟悉的 WEB 浏览器。该浏览器可以从 CD-ROM 上访问 IVT 文件，也可以从硬盘上读取帮助文件，还可以从 Internet 上读取帮助文件。VC++ 6.0 帮助系统提供的帮助文件主要包括以下五个方面：

- (1) Visual C++ Programmer's Guide: VC++ 6.0 程序员指南;
- (2) Visual C++ User's Guide: VC++ 6.0 用户指南;
- (3) Visual C++ Tutorials: VC++ 6.0 教程;
- (4) Microsoft Foundation Class and Templates: 微软基础类库和模板参考信息;
- (5) Languages and Libraries for Visual C++: VC++ 6.0 语言和库文件信息。

可以有四种方式启动帮助信息:

(1) 分类。当从 Help 菜单中选择 Contents 时,出现的帮助窗口列出了上面五个方面的信息(如图 2-10 所示)。

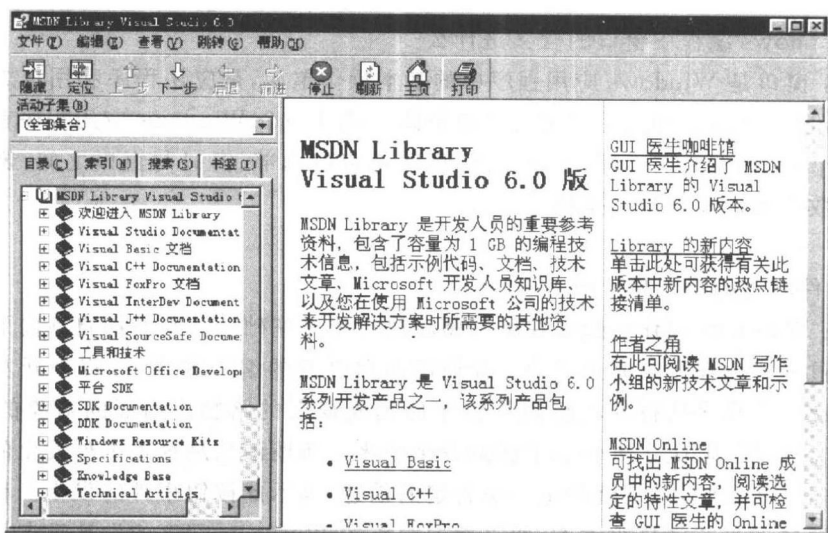


图 2-10 VC++ 6.0 的 Help 界面

(2) 按主题索引关键字。当在 Help 菜单中选择 Index 选项时,只要输入一个关键字,帮助窗口中就会列出包含该关键字的主题和文章。

(3) 按关键字检索。当从 Help 中选择 Search 选项时,只要输入一组词,就可以看到包含这些词的主题和文章。

(4) 按 F1 键。这是对程序员最好的方式。只要把光标移到函数、宏或者类名上,然后按下 F1 键,就会启动帮助系统。

第 3 章

Windows 编程基础

【3-1】 Windows 编程中窗口的含义是什么？

解答：窗口是 Windows 应用程序中的基本操作单元，是应用程序和用户之间交互的接口环境，也是系统管理应用程序的基本单位。编写一个 Windows 应用程序首先应创建一个或多个窗口，随后应用程序的运行过程即是窗口内部、窗口与窗口之间、窗口与系统之间进行数据处理和数据交换的过程。

【3-2】 事件驱动的特点是什么？

解答：Windows 操作是建立在事件驱动执行程序基础上。与利用自顶向下设计和执行的传统过程式编程方式完全不同。在使用面向过程的编程方式时，程序开发者是处于完全支配状态。程序执行的流程遵从程序员预先规定的路径。而面向对象的 Windows 程序的执行顺序则取决于事件和消息的发生顺序。所谓消息是描述事件发生的信息。当我们单击一个按钮时，系统就产生一条特定的消息，表示此按钮事件的发生。程序的执行顺序是由顺序产生的消息驱动的，所以面向对象的 Windows 程序的设计重点是编写事件和消息的处理程序。程序员可以根据消息类型编写程序以处理接受的消息，或者发出其他消息以驱动其他程序，但是不必预先确定消息产生的次序。事件驱动的这些特点对于编写交互式的程序很有用处。

【3-3】 Windows 应用程序中的消息传递是如何进行的？请举例说明。

解答：Windows 为应用程序提供称为消息队列的保留区。由 Windows 以及由其他应用程序向用户的应用程序发送的全部消息都存储在此队列里，等待调用。SDK 程序以特定的循环调用 Windows 消息，称为消息循环，如下代码表示此循环一直运行直到程序接受终止执行消息时才停止。

```
while(GetMessage(&msg,0,0,0))
{
    TranslateMessage(&msg);
    DispatchMessage(&msg);
}
```

在 while 循环内部，由循环的每次迭代调用 Windows API 函数 GetMessage(&msg, 0,0,0)以得到消息队列中的下一个消息，并把它存在特定的 MSG 结构体变量 msg 中。