

高等学校计算机课程规划教材



C++与数据结构

简明教程

实验指导与习题集

陆明 葛卫民 吴虎统 王保旗 汪大菊 主编



天津大学出版社
TIANJIN UNIVERSITY PRESS

高等学校计算机课程规划教材

C++ 与数据结构简明教程 实验指导与习题集

陆明 葛卫民 吴虎统 王保旗 汪大菊 主编

内 容 提 要

本书与天津大学出版社出版的《C++与数据结构基础简明教程》配套。全书内容分为三部分：第1部分为实验指导，包括 Visual C++ 6.0 的使用和 20 个精心设计的实验，每个实验均包括实验目的、实验内容 with 要求、说明与提示等；第2部分为 C++ 习题与答案；第3部分为数据结构习题与答案。

本书除与《C++与数据结构简明教程》一书配套供高等院校非计算机专业学生使用外，也可作为各类人员学习、使用 C++ 和数据结构的参考书。

图书在版编目(CIP)数据

C++与数据结构简明教程实验指导与习题集/陆明等主编.
—天津:天津大学出版社,2012.9
高等学校计算机课程规划教材
ISBN 978-7-5618-4467-0

I. ①C… II. ①陆… III. ①C 语言—程序设计—高等学校—
—教学参考资料②数据结构—高等学校—教学参考资料 IV.
①TP312②TP311.12

中国版本图书馆 CIP 数据核字(2012)第 205996 号

出版发行	天津大学出版社
出 版 人	杨欢
地 址	天津市卫津路 92 号天津大学内(邮编:300072)
电 话	发行部:022-27403647
网 址	publish.tju.edu.cn
印 刷	廊坊市长虹印刷有限公司
经 销	全国各地新华书店
开 本	185mm × 260mm
印 张	15.5
字 数	387 千
版 次	2012 年 9 月第 1 版
印 次	2012 年 9 月第 1 次
定 价	28.00 元

凡购本书,如有缺页、倒页、脱页等质量问题,烦请向我社发行部门联系调换

版权所有 侵权必究

前 言

软件技术基础与 C++ 程序设计是一门实践性很强的课程,仅阅读教材、熟记语法规则是远远不够的。许多程序设计的初学者常有这样的感觉,语言的语法规则能够理解,看例题程序也能够基本看明白。一旦让自己编写程序,往往不知如何下手,编出来的程序会出现很多错误,以至不能正确运行。因此认为学习程序设计是件很难的事情。其实要想学好程序设计,既要熟悉、理解编程语言的语法规则,又要理解程序编写的思路,掌握正确的编程方法。只有自己能够编写出正确的程序,才算是真正学好了程序设计这门课。要做到这一点,就必须多做有针对性的练习,以加深对语法规则的深入理解并达到能够灵活运用程度。同时,通过阅读程序和亲自动手编程并上机实践来培养编程能力。

为此,作者曾经编写了《C++ 程序设计与数据结构基础实验指导 习题 解答》一书,作为《C++ 程序设计与数据结构基础》一书的配套教材,它在教学实践中收到了很好的效果,受到了读者的欢迎。本书是在《C++ 程序设计与数据结构基础实验指导 习题 解答》一书的基础上改编而来的,作为《C++ 与数据结构基础简明教程》的配套教材,目的是使读者在学习主教材内容的同时,提供实验方面的指导和有针对性的训练,以提高读者的学习效果和实际编程能力。

本书分三部分共 16 章。

第 1 部分是实验指导,包括 5 章。第 1 章以微软公司的 Visual C++ 6.0 为范例,介绍了编译环境的使用方法,通过实例说明了调试 C++ 源程序的基本步骤,给出了调试程序时常见的错误类型及说明。第 2 章给出了一些基本实验,主要是验证性实验,目的是熟悉主教材中前两章关于 C++ 程序基本语法要素、程序框架和基本语句等语法知识。第 3 章设计了一些有一定难度的实验。目的有两个:一是熟悉数组、指针、函数等比较复杂的语法现象;二是练习用面向过程的程序设计方法编写程序,掌握一些基本的算法。第 4 章设计了一些面向对象的编程实验,目的是熟悉面向对象程序设计的语法,初步掌握利用面向对象方法编写程序的能力。第 5 章给出了一些复杂的编程练习,目的是通过这些练习进一步巩固前面学习的语言知识,更重要的是训练学生利用数据结构的一些基本知识设计复杂程序的综合能力。在具体教学过程中,这些练习可以根据不同的教学对象选用。

第 2 部分是 C++ 语言知识的练习和解答,包括 8 章(第 6 章~第 13

章)。这一部分主要是针对主教材中关于 C++ 语言及程序设计基本方法的内容,提供了大量有针对性的练习。题目类型分为单项选择题、阅读程序题、程序填空题和编程题四种类型。目的是通过练习帮助读者更好地理解 and 掌握 C++ 语言的语法规则以及程序设计的基本方法和思路。

第 3 部分是关于数据结构的练习和解答,包括 3 章(第 14 章~第 16 章)。这部分内容主要为读者提供了针对主教材中数据结构基础知识部分的练习和解答。目的是通过有针对性的练习,进一步理解所学知识并检验学习效果。

本书第 1 部分由吴虎统、陆明、葛卫民、王保旗编写,第 2 部分由陆明、汪大菊、葛卫民编写,第 3 部分由陆明编写。全书由葛卫民、陆明统稿。此外,本书的编写还得到了天津大学计算机基础教学中心软件技术基础课程组全体任课教师的支持和帮助,同时也借鉴了一些已公开出版的著作中的内容,在此一并表示感谢。

由于作者水平所限,书中可能存在错误和不妥之处,敬请专家和读者指正。

编者

2012 年 7 月

目 录

第 1 部分 实验指导

第 1 章 Visual C++6.0 的使用	(1)
1.1 控制台应用程序的生成与调试	(1)
1.2 Visual C++6.0 编译、连接中常见的错误信息	(24)
第 2 章 基本实验	(33)
2.1 实验 1: Visual C++6.0 开发环境及使用	(33)
2.2 实验 2: 简单的 C++ 程序设计	(34)
2.3 实验 3: 程序测试初步	(35)
2.4 实验 4: 分支程序设计	(36)
2.5 实验 5: 循环程序设计	(39)
第 3 章 面向过程的程序设计方法	(43)
3.1 实验 1: 数组的应用	(43)
3.2 实验 2: 指针程序设计	(45)
3.3 实验 3: 函数程序设计	(50)
第 4 章 面向对象的程序设计方法	(54)
4.1 实验 1: 类和对象的基本概念和简单应用	(54)
4.2 实验 2: 类和对象的应用程序设计	(56)
4.3 实验 3: 结构、联合和枚举类型的使用	(58)
4.4 实验 4: 继承和派生	(61)
4.5 实验 5: 运算符重载	(65)
4.6 实验 6: 多态性与虚函数	(67)
4.7 实验 7: C++ 流文件的使用	(68)
第 5 章 综合性实验	(70)
5.1 实验 1: 线性结构	(70)
5.2 实验 2: 非线性结构	(73)
5.3 实验 3: 查找算法及应用	(74)
5.4 实验 4: 排序算法及应用	(75)
5.5 实验 5: 综合练习	(76)

第 2 部分 C++ 习题与答案

第 6 章 C++ 基础知识习题	(86)
6.1 单项选择题	(86)
6.2 阅读程序题	(88)
6.3 程序填空题	(91)
6.4 编程题(根据要求编写程序并上机调试)	(92)
第 7 章 控制结构习题	(93)
7.1 单项选择题	(93)
7.2 阅读程序题	(94)
7.3 程序填空题	(98)
7.4 编程题(根据要求编写程序并上机调试)	(105)
第 8 章 数组与指针习题	(106)

8.1	单项选择题	(106)
8.2	阅读程序题	(108)
8.3	程序填空题	(114)
8.4	编程题(根据要求编写程序并上机调试)	(121)
第9章	函数习题	(122)
9.1	单项选择题	(122)
9.2	阅读程序题	(124)
9.3	程序填空题	(134)
9.4	编程题(根据要求编写程序并上机调试)	(148)
第10章	类和对象习题	(150)
10.1	单项选择题	(150)
10.2	阅读程序题	(152)
10.3	程序填空题	(165)
10.4	编程题(根据要求编写程序并上机调试)	(175)
第11章	继承和派生习题	(177)
11.1	单项选择题	(177)
11.2	阅读程序题	(178)
11.3	程序填空题	(184)
11.4	编程题(根据要求编写程序并上机调试)	(186)
第12章	虚函数与运算符重载习题	(187)
12.1	单项选择题	(187)
12.2	阅读程序题	(188)
12.3	程序填空题	(193)
12.4	编程题(根据要求编写程序并上机调试)	(201)
第13章	I/O 流习题	(202)
13.1	单项选择题	(202)
13.2	阅读程序题	(203)
13.3	程序填空题	(205)
13.4	编程题(根据要求编写程序并上机调试)	(208)
第3部分 数据结构习题与答案		
第14章	数据结构基础知识及线性结构习题	(209)
14.1	单项选择题	(209)
14.2	简答题	(212)
14.3	程序填空题	(214)
第15章	非线性结构习题	(225)
15.1	单项选择题	(225)
15.2	简答题	(228)
第16章	查找和排序习题	(233)
16.1	单项选择题	(233)
16.2	简答题	(236)
16.3	程序填空题	(238)
参考文献		(242)

第 1 部分 实验指导

第 1 章 Visual C++ 6.0 的使用

Visual C++ 6.0 是 Microsoft 公司于 1998 年推出的基于 Windows 9x 和 Windows NT 的优秀集成开发环境。面向对象的程序设计方法与可视化开发环境的完美结合,使 Visual C++ 6.0 深受广大软件开发者的喜爱。Visual C++ 不仅是 C++ 语言的集成开发环境,而且还是开发 32 位 Windows 应用程序的强大工具。它所提供的一系列功能丰富的编程工具,使编写在 Windows 环境下运行的应用程序十分简捷、快速;同时,基于类库的设计方法又使编写面向对象的程序变得更为简单。另外,与 Visual Basic 相比,用 Visual C++ 开发出的应用程序的代码效率更高。

Visual C++ 6.0 软件包中包含了许多独立的组件,例如编辑器、编译器、连接器、调试器以及各种各样的辅助开发工具。在 Visual C++ 的软件包中还包含着一个名为 Developer Studio (开发者工作室)的集成应用程序。Developer Studio 是访问 Visual C++ 提供的开发工具的主要界面。它把 Visual C++ 的各种工具结合在一起,集成为一个有机的整体,通过一个由窗口、菜单、对话框、工具栏以及快捷键组成的和谐系统,使用户比较轻松地监控应用程序的创建过程。

如果计算机上已经安装了 Visual C++ 6.0,可通过用鼠标单击“开始|程序|Microsoft Visual Studio 6.0|Microsoft Visual C++ 6.0”,启动 Visual C++ 6.0 (图 1.1)。此时,屏幕上显示图 1.2 所示的 Visual C++ 6.0 的主窗口。由于设置不同,看到的窗口与图 1.2 所示的窗口可能有些差别。

下面简要介绍在 Visual C++ 6.0 中开发应用程序的过程。

1.1 控制台应用程序的生成与调试

为使大家能集中精力完成有关 C++ 语言方面的练习,而不考虑编写 Windows GUI (Graphic User Interface,图形用户界面)程序时的一些复杂情况,下面主要介绍在 Visual C++ 下生成简单的控制台(console)的应用程序,即生成字符方式应用程序的过程。

在 Visual C++ 中,可以使用编写字符方式程序时常用的技术编写控制台应用程序。具体地说,在程序中可以使用 C++ 的 `iostream` 类库的成员函数读取输入数据和显示输出等。对于一个编写好的 C++ 源程序,要想在 Visual C++ 环境下运行,需要经过若干步骤,下面给出各步骤的方法及注意事项。

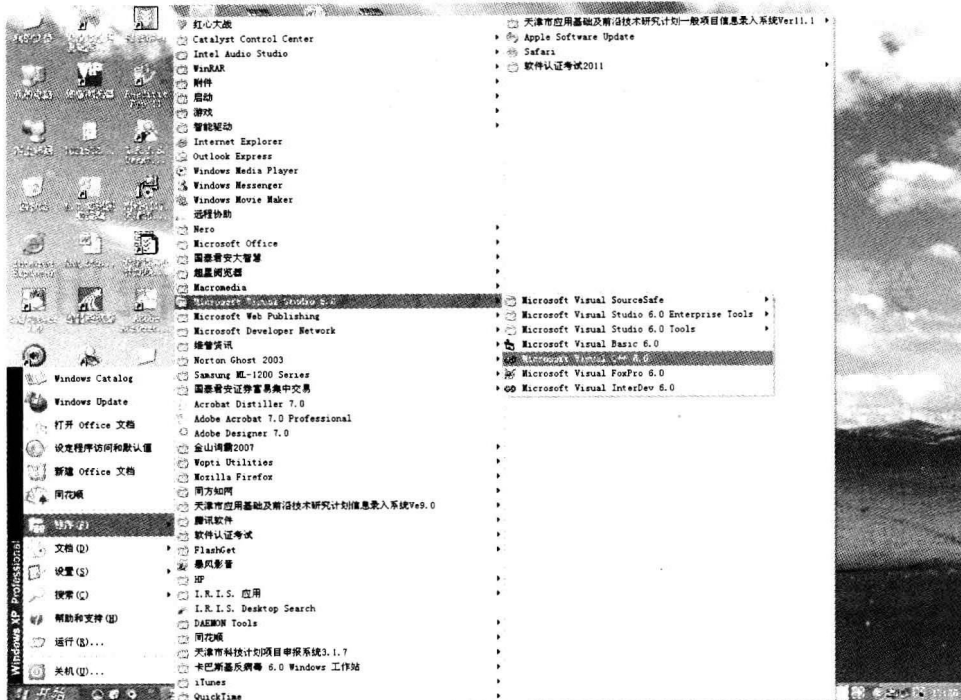


图 1.1 启动 Visual C++ 6.0

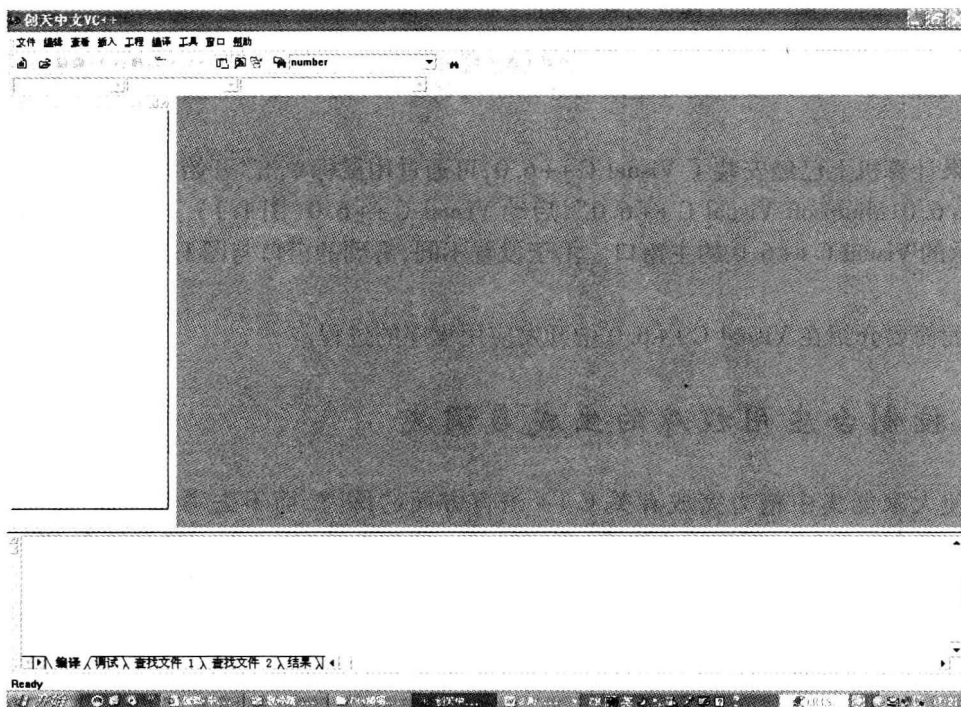


图 1.2 Visual C++6.0 的主窗口

1.1.1 生成项目

现在的应用程序,尤其是 Windows 应用程序,一般由多个文件(File)组成,其中包括源程序文件(.cpp)、头文件(.h)、资源程序文件等。为使组成程序的所有文件能够形成一个有机的整体,引入了项目(Project)的概念,即把一个应用程序作为一个项目。项目能够自动地将其包含的文件进行分类、管理,这就大大减轻了程序员的负担。

在 Visual C++ 中,项目又置于项目工作区(Workspace)的管理之下。一个项目工作区可以管理多个项目,甚至是不同类型的项目。这些项目之间相互独立,但共用一个项目工作区的环境设置。在 Visual C++ 中,文件、项目以及项目工作区之间的关系如图 1.3 所示。

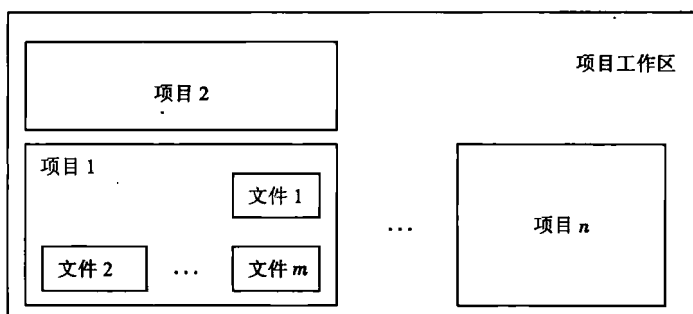


图 1.3 文件、项目、项目工作区之间的关系

为简单起见,本章介绍的所有程序实例都使用了只包含一个项目的项目工作区。假设现有一个名为 Welcome 的 C++ 程序。该程序的源程序代码如下:

//Welcome.cpp: Welcome 程序的 C++ 源程序代码

```
#include <iostream.h>
char Name[16];
void main()
{
    cout << "Enter your name:";
    cin.getline (Name, sizeof(Name));
    cout << "\nWelcome, " << Name << "\n";
    cout << "\nPress Enter to continue ...";
    cin.get();
}
```

为生成以上程序所需要的项目文件,可按以下步骤工作。

①执行 Visual C++ 主窗口中的“File|New”菜单命令或按 Ctrl + N 键打开 New 对话框,如图 1.4 所示。

②选择 New 对话框中的 Projects 标签,以便生成新项目。在生成新项目时,Developer Studio 将自动生成项目工作区,并将新项目加入其中。

③在 Projects 标签的项目类型列表框中选择“Win32 Console Application”(Win32 控制台应用程序)。

④在“Project Name:”文本框中输入项目名 Welcome,指示 Developer Studio 把新项目(以及

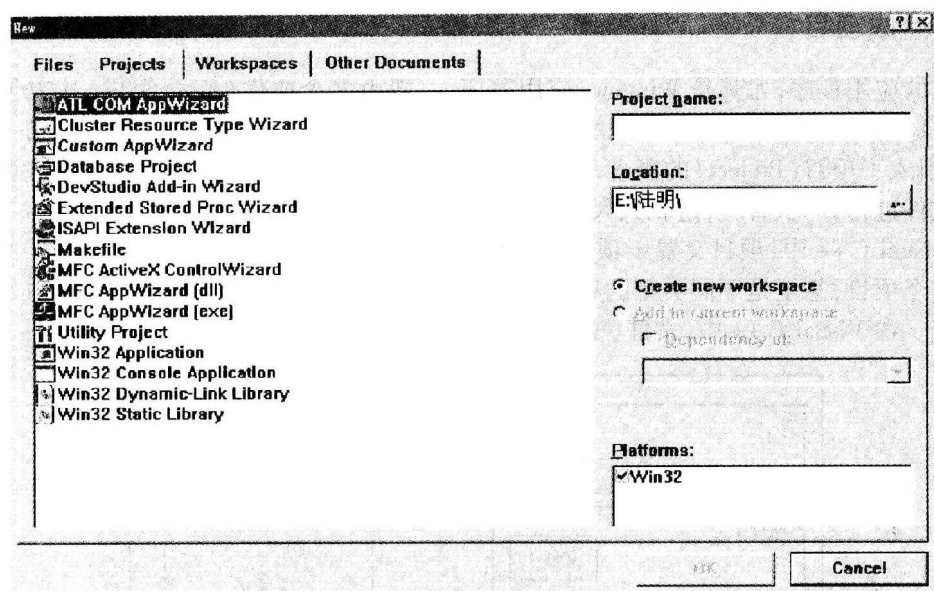


图 1.4 New 对话框

该项目所在的项目工作区)取名为 Welcome。

⑤在“Location:”文本框中指定保存项目文件的文件夹(即项目文件夹)的路径。既可以采用该文本框中给出的缺省文件夹路径,也可以单击文本框右侧带省略号(...)的按钮进行搜索,查找不同位置的项目文件夹,还可以直接输入项目文件夹的路径。如果指定的项目文件夹不存在,Developer Studio 会自动生成这个项目文件夹,同时还会在项目文件夹中生成一个或几个存放输出文件的子文件夹。

⑥在 Platforms 区中,检查是否选择了“Win32”。Win32 是 Windows 9x 和 Windows NT 下的 GUI 应用程序和控制台应用程序的 32 位 API 接口(Application Programming Interface)。选择 Win32 表示应用程序由 Win32 支持、可以调用 Win32 函数、能够作为 32 位程序在 Windows 9x 和 Windows NT 上运行(经过以上步骤后,New 对话框中 Projects 标签的设置情况如图 1.5 所示)。

⑦单击 New 对话框中的“OK”按钮,这时 Developer Studio 将显示 Win32 控制台应用程序向导。该向导只有一个对话框,如图 1.6 所示。

⑧本练习的目的是学习如何自己生成源程序文件,所以在 Win32 控制台应用程序向导对话框中选择“An empty project”选项,如图 1.6 所示。此选择表示只生成项目,但不向其中加入任何源程序文件。当然,也可以根据需要选择其他选项。

⑨单击 Win32 控制台应用程序向导对话框底部的“Finish”按钮,此时屏幕上出现 New Project Information 对话框,在该对话框中显示了将要建立的新项目的基本信息,如图 1.7 所示。

⑩单击 New Project Information 对话框下面的“OK”按钮,屏幕显示将切换到 Visual C++ 6.0 的主窗口。此时,Developer Studio 已经生成并打开了新的项目工作区 Welcome。在 Welcome 项目工作区中只包含了一个项目,名称也是 Welcome。与项目有关的信息会在工作区

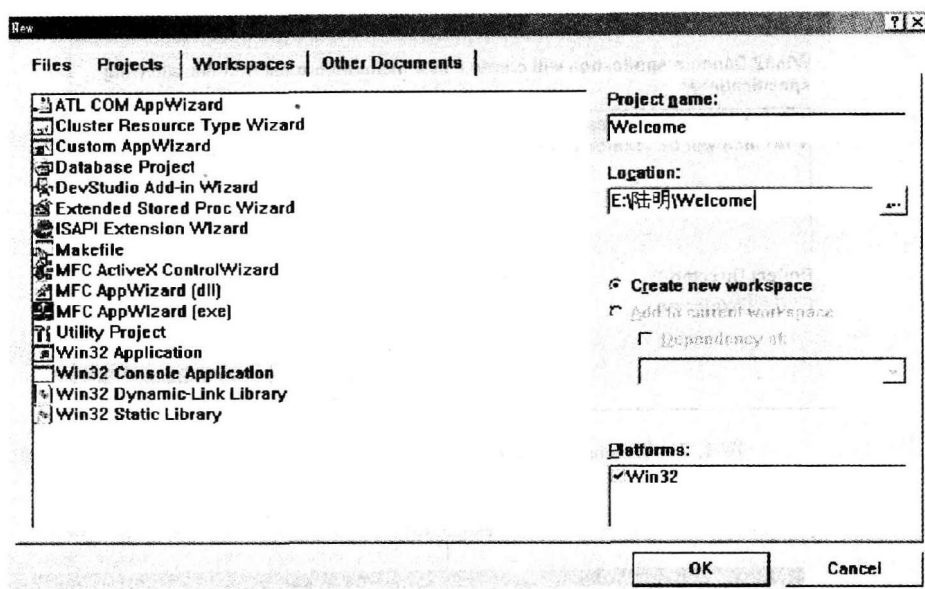


图 1.5 针对 Welcome 程序设置的 New 对话框的 Projects 标签

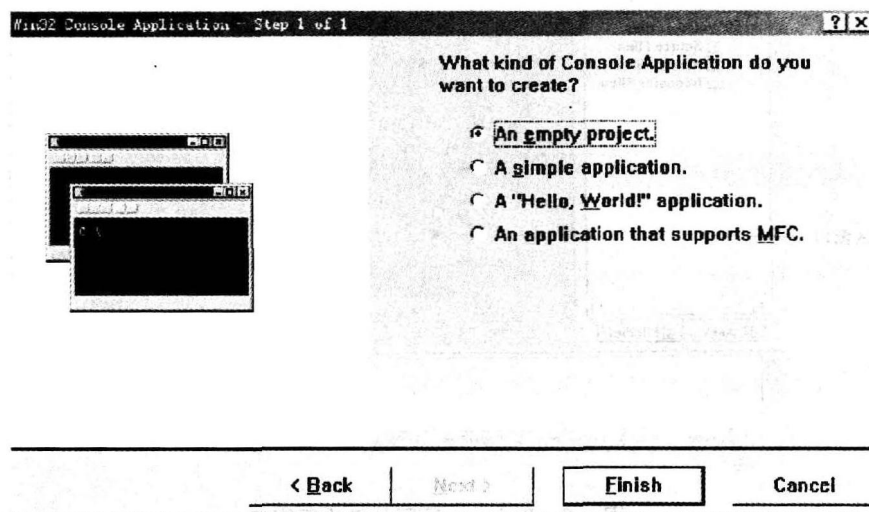


图 1.6 Win32 控制台应用程序向导

(Workspace)窗口中显示,如图 1.8 所示。如果屏幕上没有出现工作区窗口,可执行“View | Workspace”菜单命令将其打开。

在工作区窗口中包含了两个标签,即 ClassView 和 FileView(如果生成的是一个 Windows GUI 程序,那么在工作区窗口中还会增加一个 ResourceView 标签)。它们显示了与项目有关的各种信息。关于在各标签中显示的信息的意义将在后续章节介绍。

可以用“File | Save Workspace”菜单命令保存当前工作区;用“File | Close Workspace”菜单命令关闭当前工作区;用“File | Open Workspace”菜单命令打开一个已经存在的工作区。工作

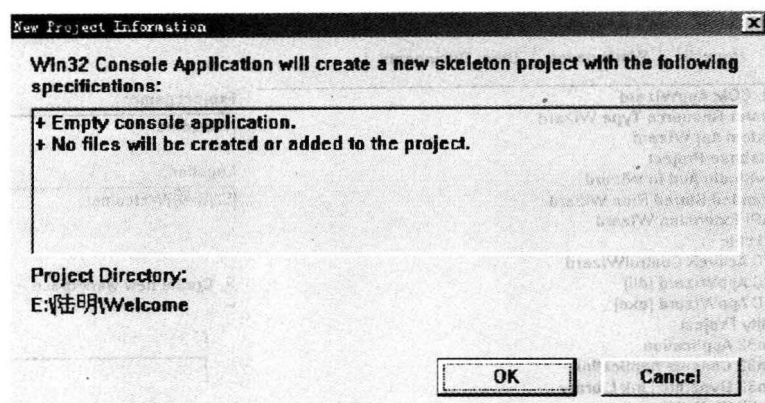


图 1.7 Welcome 项目的 New Project Information 对话框

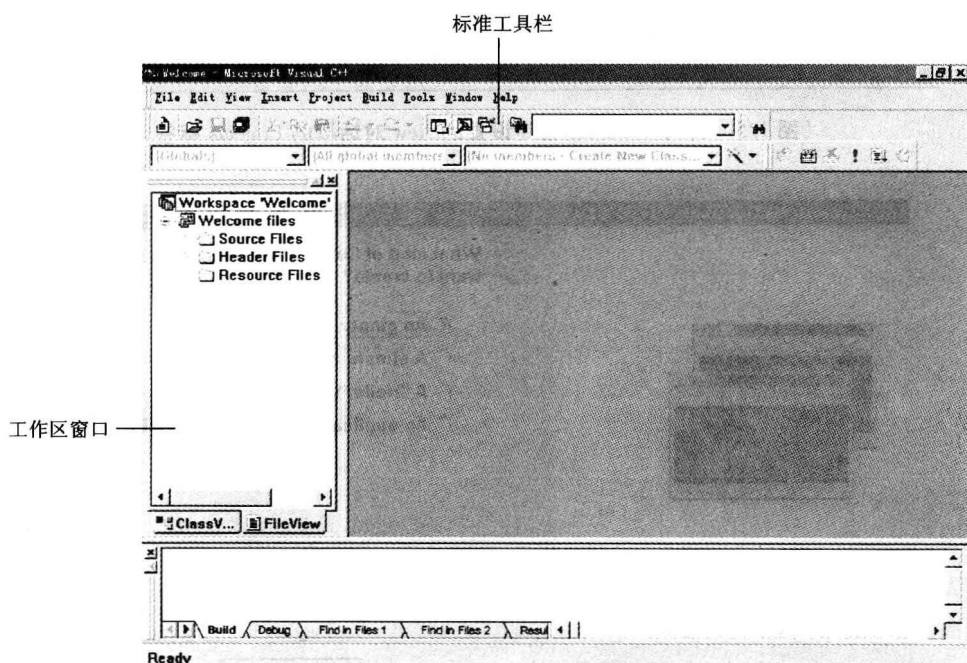


图 1.8 Welcome 项目的工作区窗口

区文件的扩展名是 .dsw。

至此,为生成项目而需要做的工作已经全部完成。

1.1.2 生成和编辑源程序文件

在生成项目后,下一步是生成和编辑源程序文件 Welcome.cpp。为此,执行“File|New”菜单命令,在打开的 New 对话框中选择 Files 标签,并在文件类型列表框中选择“C++ Source File”项,然后在“File”下面的文本框中输入源程序文件名 Welcome。确保选中“Add To Project:”复选框,在“Add To Project:”复选框下面的下拉列表框中给出了缺省的项目名 Welcome,

在“Location:”下面的文本框中显示的是生成 Welcome 项目时指定的项目文件的路径。保持这两个框中的内容不变,完成的 New 对话框的 Files 标签如图 1.9 所示。

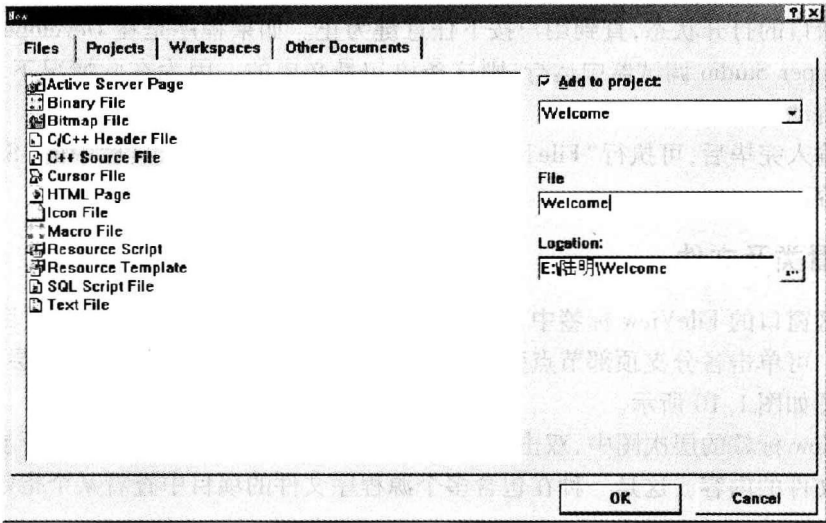


图 1.9 完成的 New 对话框的 Files 标签

单击 New 对话框中的“OK”按钮后,Developer Studio 将生成一个名为 Welcome.cpp 的源程序文件,并将此源程序文件加入到 Welcome 项目中,同时在文本编辑器窗口将该源程序文件打开。可以在文本编辑器窗口中输入编写好的 Welcome 源程序的代码,结果如图 1.10 所示。

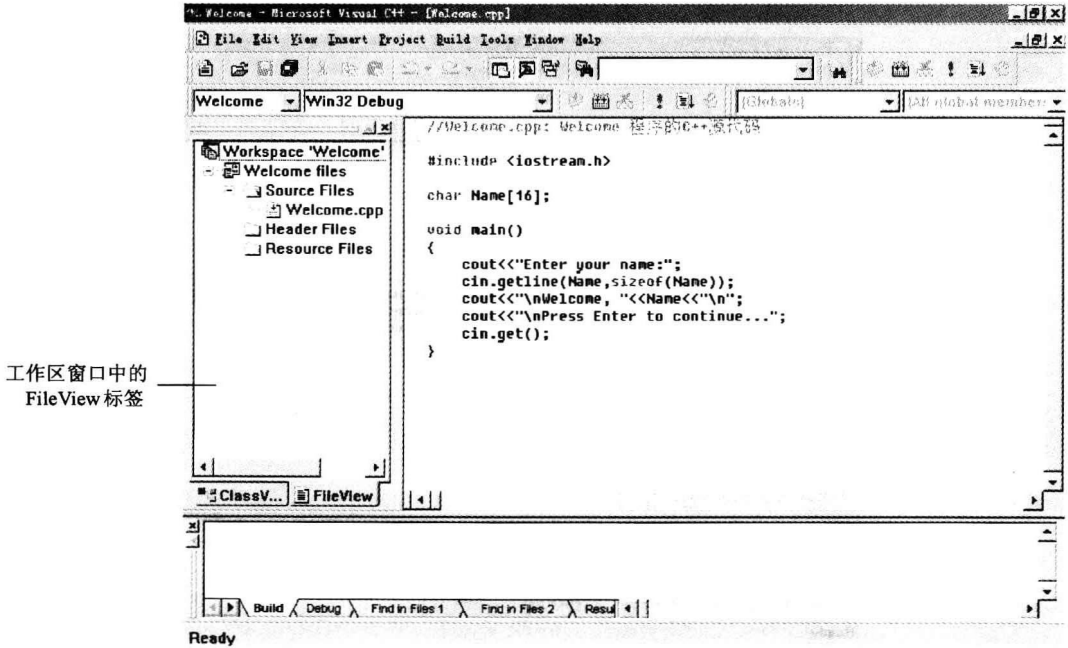


图 1.10 输入 Welcome 源程序代码

在 Welcome 程序运行时,程序中最后一条语句的作用是暂停程序的执行,等待用户按下 Enter 键。这样做的目的是使用户在关闭程序窗口前能够看到程序的输出结果。如果程序是在 Developer Studio 中运行,这条语句是多余的。因为 Developer Studio 会自动安排在程序执行后保持程序窗口的打开状态,直到用户按下任意键为止。如果程序是在 Developer Studio 外运行或在 Developer Studio 调试器中运行,则这条语句是必要的。因为在此情况下,程序执行结束后窗口会自动关闭。

源程序输入完毕后,可执行“File | Save”菜单命令或单击标准工具栏中的“Save”按钮,保存输入的内容。

1.1.3 查看类及文件

在工作区窗口的 FileView 标签中,以层次图方式显示了当前工作区中的项目以及项目中包含的文件。可单击各分支顶部节点旁的“+”(或“-”)标记,以展开或折叠层次图的分支。展开的层次图如图 1.10 所示。

在 FileView 标签的层次图中,双击其中的某个文件名,可以将这个文件打开或在编辑窗口中显示这个文件的内容。这是一种在包含多个源程序文件的项目中查看某个指定文件的简单方法。

在工作区窗口的 ClassView 标签中也显示了一个层次图,如图 1.11 所示。Globals 文件夹中列出了在 Welcome 程序中定义的所有全局符号。对于一个包含类的 C++ 程序,在 ClassView 标签的层次图中还将显示该程序定义的所有类及其成员。如果双击 ClassView 标签中的某个符号(如 Welcome 程序的 main() 或 Name),Developer Studio 会显示定义该符号的源程序文件,并将插入点放在定义处。

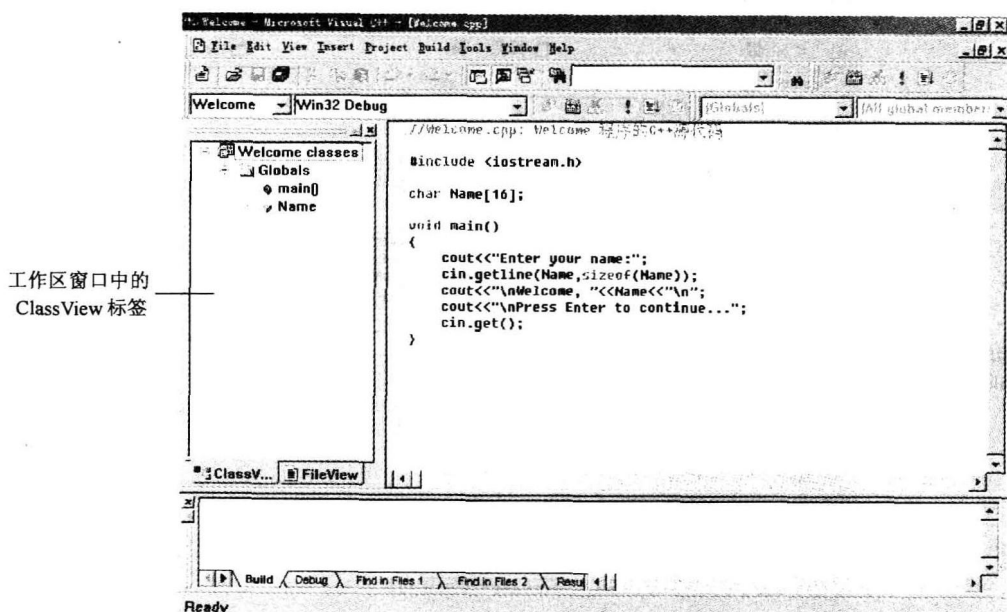


图 1.11 工作区窗口中的 ClassView 标签

1.1.4 访问联机帮助

如果想更全面地了解 Visual C++ 和 Windows 的工作机制,或想在开发过程中获得帮助,应启动 Visual C++ 6.0 的联机帮助。

Visual C++ 6.0 的帮助系统做得非常详尽。访问 Visual C++ 帮助系统的方法主要有两种:一是从“Help”菜单中选择执行 Contents、Search 或 Index 命令;二是在文本编辑窗口中把光标定位在一个需要查询的单词处,然后按 F1 键。打开的 Visual C++ 6.0 的联机帮助窗口如图 1.12 所示。

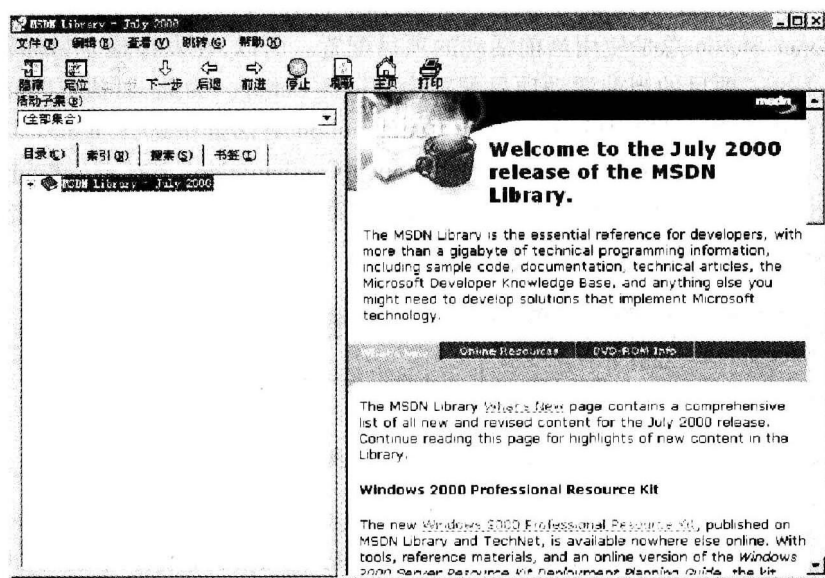


图 1.12 Visual C++ 的联机帮助窗口

Visual C++ 6.0 的帮助系统很像微软的 Internet Explorer,可以在此浏览以 HTML 文件组成的静态帮助文件。此外,帮助系统还提供了全文搜索功能。当用户在搜索标签中输入要搜索的单词时,帮助系统将在所有的帮助文件中进行全文搜索。同时,Visual C++ 6.0 的帮助系统通过“活动子集”下拉列表框还提供了很好的文件分类,用户可以缩小搜索范围。

在使用联机帮助前,必须保证已正常安装了 MSDN (Microsoft Developer Network)。安装 MSDN 需要额外的两张光盘。

1.1.5 改变项目配置

Developer Studio 中的项目可以产生两种可执行代码:一种是 Win32 Debug (调试版);另一种是 Win32 Release (发布版)。

在调试版中存在着大量的符号信息。这些信息用于记录程序中函数和变量的名字及其在源程序中的位置。通过这些符号信息,调试器可以将源程序中的每一行与可执行代码中的相应指令联系起来,以便于用户检测程序中的错误。调试版的体积较大,运行速度通常比发布版要低,主要在开发过程中使用。

发布版只包含了经过优化的可执行代码,其中没有符号信息。正因如此,它不能用于调试

程序。当认为程序可以为用户使用,可以提供发布版。

在系统生成可执行代码之前,可以改变影响文件编译方式的项目设置。通常情况下,系统为项目创建的缺省项目配置中指定的设置对于大多数项目而言是足够的,但如果需要对项目设置进行更改,可以执行“Project|Setting...”菜单命令,在打开的 Project Setting 对话框中完成设置。

1.1.6 建立目标程序

下面说明如何建立 Welcome 程序的调试版。

如果 Welcome 项目还未打开,应执行“File|Open Workspace”命令将它打开。在建立目标程序时,Developer Studio 总是使用当前活动的项目配置。所以,在发出建立目标程序的命令之前,应保证 Welcome 项目的当前活动项目配置为 Win32 Debug。为此,可执行“Build|Set Active Configuration”菜单命令,在弹出的 Set Active Configuration 对话框中选择 Win32 Debug,然后单击“OK”按钮。也可以通过 Build 工具栏中的“Active Configuration”下拉列表框进行设置,如图 1.13 所示。如果窗口中没有出现 Build 工具栏,可在菜单栏或某个工具栏上单击鼠标右键,并从弹出的快捷菜单中选择 Build,打开 Build 工具栏。

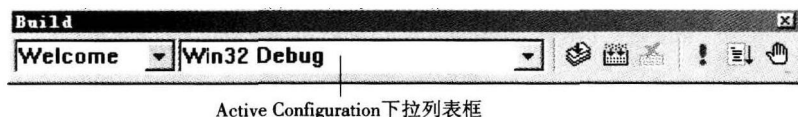


图 1.13 Build 工具栏

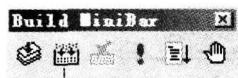


图 1.14 Build MiniBar 工具栏

要建立目标程序,可执行“Build|Build Welcome.exe”菜单命令或单击 Build 工具栏或 Build MiniBar 工具栏中的 Build 按钮,如图 1.14 所示。

Developer Studio 在建立目标程序时将自动弹出 Output 窗口(如果它原来没有出现的话),并在此窗口中显示目标程序建立过程中的一些相关信息,包括错误和警告信息。当建立目标程序的工作完成后,无论成功与否,Developer Studio 都会发生“嘟”的声响,并在 Output 窗口中显示错误信息和警告信息的总数。假设在 Output 窗口中显示的是“Welcome.exe—0 error(s),0 Warning(s)”,表示程序中没有错误和警告,目标程序的建立工作顺利完成。

在生成目标程序时,Developer Studio 会在项目文件夹中建立子文件夹 Debug,用来存放输出文件(Welcome.obj, Welcome.exe 等)。如果还要基于发布配置建立目标文件,则 Developer Studio 会在项目文件夹中建立子文件夹 Release,并在此文件夹中存放发布版的输出文件。由此可见,基于调试版和发布版生成的输出文件分别存放在不同的地方,它们不会相互覆盖。如果要改变存放输出文件的文件夹,只要执行“Project|Setting...”菜单命令,并在打开的 Project Setting 对话框的 General 标签中指定中间输出文件(如 Welcome.obj)和最终输出文件(如 Welcome.exe)所在的文件夹即可。