

- 走进编程殿堂
- C++语言的基础知识
- 结构化编程基础
- 计算机解题基础知识
- 函数与数组
- 解题实例
- 结构和文件
- 课程设计实例

高等学校文科类专业大学计算机规划教材

丛书主编 卢湘鸿

程序设计及应用

刘振安 编著



清华大学出版社

高等学校文科类专业大学计算机规划教材

丛书主编 卢湘鸿

程序设计及应用

刘振安 编著

清华大学出版社

北 京

内 容 简 介

本书分为基础、提高和课程设计3个部分。基础部分重在引入结构化编程和计算机解题基础知识,结合趣味性程序,解析学习难点。提高部分通过典型算法,简要介绍基于过程编程的基本方法,为编写实用程序打下基础。课程设计部分给出了难易不同的设计实例,以便学生巩固所学知识,扩大教师教学选择的余地。

本书语言生动,叙述清楚,通俗易懂,实例丰富,不仅可作为高等学校文科类专业程序设计课程的教材,也可作为自学材料。

版权所有,翻印必究。举报电话:010-62782989 13501256678 13801310933

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

本书防伪标签采用特殊防伪技术,用户可通过在图案表面涂抹清水,图案消失,水干后图案复现;或将表面膜揭下,放在白纸上用彩笔涂抹,图案在白纸上再现的方法识别真伪。

图书在版编目(CIP)数据

程序设计及应用/刘振安编著. —北京:清华大学出版社,2006.8

(高等学校文科类专业大学计算机规划教材/卢湘鸿主编)

ISBN 7-302-13017-5

I. 程… II. 刘… III. 程序设计—高等学校—教材 IV. TP311.1

中国版本图书馆 CIP 数据核字(2006)第 047869 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: 焦 虹

文稿编辑: 孙建春

印 刷 者: 北京市清华园胶印厂

装 订 者: 三河市李旗庄少明装订厂

发 行 者: 新华书店总店北京发行所

开 本: 185×260 印张: 10.5 字数: 234 千字

版 次: 2006 年 8 月第 1 版 2006 年 8 月第 1 次印刷

书 号: ISBN 7-302-13017-5/TP·8261

印 数: 1~4000

定 价: 16.00 元

序

能够满足社会与专业本身需求的计算机应用能力已成为合格的大学毕业生必须具备的素质。

文科类专业与信息技术的相互结合、交叉、渗透,是现代科学技术发展趋势的重要方面,是不可忽视的新学科的一个生长点。加强文科类专业的计算机教育是培养能够满足信息化社会对文科人才要求的重要举措,是培养跨学科、复合型文科通才的重要环节。因此,使用一定层次、一定内容的计算机科学与技术知识来武装文科类专业(包括哲学、经济学、法学、教育学、文学、历史学等学科和管理学中的一些专业)的学生(包括研究生、本科生和高职高专生),开设具有文科专业特色的计算机课程是十分必要的。

为了指导文科类专业的计算机教学工作,教育部高等教育司组织制定了《高等学校文科类专业大学计算机教学基本要求(2003年版)》(下面简称《基本要求》)。

《基本要求》把文科类计算机教学的知识结构分为两大部分:一是大学计算机公共基础课;二是在开设计算机公共基础课之后,体现专业特色或与专业教学相结合的后续课程。

计算机公共基础课是为了满足文科类专业的共同需要,而后续课程是为了满足各文科专业的不同需要。

公共基础课由①计算机基础知识(软、硬件平台)、②微机操作系统及其使用、③多媒体知识和应用基础、④办公软件应用、⑤计算机网络基础、⑥Internet基本应用、⑦电子政务基础、⑧电子商务基础、⑨数据库系统基础和⑩程序设计基础等10个模块构筑。这些内容都是每个文科学生应该掌握的,可为他们在与专业紧密结合的信息技术应用方向上深入学习打下基础。根据这些模块可以编著3本书:一是由模块①~⑥组成计算机应用基础;二是由模块⑤~⑧组成网络应用;三是由模块⑨~⑩组成数据库与程序设计。

后续课程在深度上超过了公共基础部分的相应模块,或者是开拓了新的应用领域。它在更大程度上决定了学生应用计算机技术解决本专业实际问题的能力与水平。故建议至少设置16门后续课程。包括:微机组装与维护、计算机网络技术及应用、网页设计基础、动态网页设计——服务器端应用开发、电子政务应用、电子商务应用、法律事务信息处理基础(电子法务概论)、多媒体技术及应用、三维建模与动画设计、数据库基础及其应用、社会统计学分析及应用(非财经类)、经济统计与分析软件应用基础(财经类)、信息处理基础、管理信息系统、财经信息化基础、程序设计及应用。

显然,包括文科在内的大学非计算机专业的计算机教学有着广阔的前景。

清华大学出版社推出的高等学校文科类专业大学计算机规划教材,就是根据《基

本要求》编写而成的。它可以满足文科类专业计算机公共基础课及其后续课程的教学需要。

信息技术的发展日新月异,文科类专业的计算机教学还处在探索和不断完善的阶段,因此对教材中的不足或错误,需要同行和读者批评指正。

卢湘鸿

于北京中关村科技园

卢湘鸿 北京语言大学信息科学学院计算机科学与技术系教授、教育部高等学校文科计算机基础教学指导委员会副主任。负责教育部高等教育司组织制定的《普通高等学校文科类专业计算机基础课程教学大纲(2000年版)》和《高等学校文科类专业大学计算机教学基本要求(2003年版)》的统稿工作。

• II •

前 言

考虑到高等院校中文科计算机课程的教学重在实用,所以本书在介绍“面向过程”的程序设计时,采取了引入 C++ 特性的方法,以降低学习面向过程编程的难度,使学生在掌握 C++ 语言知识的同时,也初步掌握 C 语言的基本知识。另外,本书不要求学生已经学过其他语言。

本书以够用为原则,尽量采取简化程序设计的措施,以便使程序员能集中精力设计更好的面向过程的程序。本书专门介绍了常用的逻辑求解问题、递推和递归以及调试等知识,也提供一些趣味性程序,以提高学习兴趣。

为了便于各校根据本校特点选择教学内容,全书分为基础部分、提高部分和课程设计三个部分。基础部分共 4 章。第 1 章“走进编程的殿堂”,重在引入编程概念。第 2 章“C++ 语言的基础知识”,重点引入 C++ 语言基本数据类型和表达式。第 3 章“结构化编程基础”,结合实例简要介绍基于过程编程的基本方法。第 4 章“计算机解题基础知识”,用实际的例子说明使用计算机解决具体问题的思路。为了简化学习和提高学习兴趣,还通过典型例题和错误分析来研究容易出现错误,解析学习的难点。同时,为了方便学习和实验,书中还简要介绍了基本的程序调试方法。其实,第 6 章 6.1 节的解题和算法描述以及第 8 章 8.4 节中有关测试与调试知识的简介,也属于基础部分的内容,应该酌情讲授。

提高部分共 3 章。第 5 章讲解函数与数组。一般认为函数与数组对文科学生来讲较难理解,但本书不涉及指针、重载、引用、传址、多维数组等概念,仅以够用为原则,主要让学生掌握函数在程序设计中的地位,所以还是比较容易理解的。第 6 章是解题实例。这一章将结合典型算法简要介绍基于过程编程的基本方法。第 7 章讨论结构和文件,属于可选教学内容。本章只是简要介绍结构和结构数组的基本概念和使用方法,没有介绍结构指针,所以内容较浅,文件也仅仅介绍作为字符流的基本使用方法。其实,这些知识属于课程设计内容的一部分,也为编写实用程序打下基础。

第 8 章提供了课程设计实例。为提供选择余地,本章给出了难易不同的设计实例,同时也介绍了软件测试的基础知识。

总之,本书这样编排的目的是为了减少教学和学习的难点,让学生把主要精力放在算法的设计上。

这本教材是在文科计算机基础教学指导委员会的支持下编写的,这只是一种尝试,不妥之处在所难免,希望读者给予批评指正。

刘振安

2006 年 2 月

目 录

第 1 部分 基础部分

第 1 章 走进编程的殿堂	3
1.1 似曾相识燕归来	3
1.2 程序的编辑、编译和运行的 基本概念	4
1.2.1 密切配合的三个主要 步骤	4
1.2.2 编写控制台程序	4
1.3 工欲善其事,必先利其器	8
1.3.1 菜单栏	9
1.3.2 工具栏	9
1.4 打好编程基本功	10
1.4.1 自然语言与计算机语言 之间的鸿沟	10
1.4.2 面向过程与结构化程序 设计	11
1.5 C++ 更容易编制面向过程的 程序	15
1.6 本书的结构及教学建议	15
1.6.1 基础部分	15
1.6.2 提高部分	16
1.6.3 课程设计部分	17
实验 1 练习使用集成环境	17
习题 1	17

第 2 章 C++ 语言的基础知识

2.1 C++ 的基本数据类型	18
2.1.1 小荷才露尖尖角	18
2.1.2 标识符	20
2.1.3 变量	21
2.1.4 基本数据类型	22
2.1.5 常量	23
2.1.6 命名注意事项	25
2.2 C++ 的表达式	26

2.2.1 运算表达式和运算符	26
-----------------------	----

2.2.2 赋值运算符与赋值 表达式	27
-----------------------------	----

2.2.3 逗号运算符与逗号 表达式	28
-----------------------------	----

2.3 典型例题及错误分析

2.3.1 典型例题	28
------------------	----

2.3.2 初学者最容易出现的 语法错误	29
-------------------------------	----

2.3.3 容易出现的其他错误	30
-----------------------	----

2.4 活捉臭虫保平安

2.4.1 一个简单的示例程序	31
-----------------------	----

2.4.2 编译程序	32
------------------	----

2.4.3 排错	34
----------------	----

2.4.4 基本调试命令简介	35
----------------------	----

实验 2 练习调试程序

习题 2

第 3 章 结构化编程基础

3.1 程序语句	39
----------------	----

3.2 关系运算	41
----------------	----

3.3 逻辑运算	42
----------------	----

3.4 结构化程序设计概述	44
---------------------	----

3.5 控制选择结构	44
------------------	----

3.5.1 用 if 语句实现选择结构 设计	44
---------------------------------	----

3.5.2 用 switch 语句实现选择 结构设计	48
-------------------------------------	----

3.6 循环控制结构设计	49
--------------------	----

3.6.1 while 语句	49
----------------------	----

3.6.2 do~while 语句	50
-------------------------	----

3.6.3 for 语句	51
--------------------	----

3.6.4 break 语句、continue 语句及 goto 语句	52
--	----

3.7 典型例题及错误分析	55
---------------------	----

3.7.1 典型例题	55
------------------	----

3.7.2 错误分析	55	6.2 递推算法	95
实验 3 求水仙花数	58	6.2.1 基础知识	95
习题 3	58	6.2.2 递推实例	96
第 4 章 计算机解题基础知识	60	6.3 递归算法	99
4.1 枚举法	60	6.3.1 递归与递推的比较	99
4.1.1 重复运算	60	6.3.2 图解递归执行过程 实例	101
4.1.2 分支运算	61	6.4 查找算法	101
4.1.3 逻辑思维的计算机表示	62	6.4.1 线性查找	102
4.1.4 使用枚举法解题的思路	63	6.4.2 二分查找	102
4.1.5 参考程序	64	6.5 冒泡排序	104
4.2 逻辑问题求解实例	66	6.5.1 图解排序过程	104
4.2.1 赛车问题	66	6.5.2 算法分析	104
4.2.2 新郎新娘问题	68	6.5.3 算法设计	105
4.3 小结	69	6.5.4 参考程序	105
实验 4 编程调试实验	70	实验 6 编程实验	106
习题 4	70	习题 6	106
 第 2 部分 提高部分		 第 7 章 结构和文件	108
第 5 章 函数与数组	73	7.1 结构	108
5.1 典型 C++ 程序结构	73	7.1.1 结构定义及其变量的 初始化	108
5.1.1 环环相扣的函数	74	7.1.2 结构数组	109
5.1.2 函数类型和返回值	74	7.2 文件操作	111
5.1.3 函数原型显身手	76	7.2.1 写文件	111
5.1.4 函数调用形式	77	7.2.2 读文件	112
5.1.5 函数的形式参数和实在 参数	79	7.3 综合实例	115
5.1.6 函数的返回区	80	实验 7 结构和文件编程实验	116
5.2 变量的存储类型	80	习题 7	117
5.3 一维数组	83	 第 3 部分 课程设计部分	
5.4 几个最常用的字符串函数	85	第 8 章 课程设计实例	121
5.5 正确使用库函数和数组	87	8.1 鸡兔同笼	121
实验 5 函数编程实验	89	8.2 求解百鸡问题	123
习题 5	89	8.3 比赛评分程序	125
第 6 章 解题实例	91	8.3.1 直接算法	125
6.1 解题和算法描述	91	8.3.2 间接算法	126
6.1.1 计算机解题	91	8.4 职工信息档案	128
6.1.2 常用过程设计算法描述 方法	94	8.4.1 设计要求	128
		8.4.2 算法分析	129

8.4.3	参考程序	133		
8.4.4	测试程序	140	附录 B	C 语言关键字
8.5	测试与调试知识简介	145		152
8.5.1	软件测试	145	附录 C	C 语言的 printf 格式输出函数
8.5.2	程序的测试与调试	147		153
			附录 D	C 语言的 scanf 格式输入函数
				156
附录 A	按字母表顺序排序的 C 和 C++		参考文献	158
	保留字	151		



第 1 部分

基础部分

第 1 章 走进编程的殿堂

本章首先以一个典型而简单的 C++ 程序为例,说明使用集成环境编写、编译和运行一个 C++ 程序的全过程,并引入程序设计的一些基本概念。其实,大家会发现很多似曾相识的东西,结合学过的知识,就会轻松步入程序设计的殿堂。

1.1 似曾相识燕归来

下面是一个典型而简单的 C++ 程序,程序运行后输出“欢迎光临!”。

【例 1.1】 演示输出的示例程序。

```
/* 功能:输出一句话
   程序名:mycpp.cpp */
#include <iostream>           //包含头文件
using namespace std;         //使用命名空间
void main()                  //主程序
{                             //主程序开始
    cout<< "欢迎光临!";      //输出信息
}                             //主程序结束
```

1. 从主函数开始

用 C++ 语言编写的程序称为 C++ 语言的源程序(简称 C++ 程序),以 .cpp 作为文件扩展名。main 称为主函数。一个程序中必须有一个且只能有一个名为 main 的主函数, C++ 编译系统假设程序中定义有 main 函数。如果没有定义,程序将无法执行。

为了简单起见,主函数使用 void 的形式,现在暂不去探讨 void 的具体含义,仅把它们看作一种标记。main 函数后面有一对花括号“{}”,符号“{”和“}”必须成对出现。

2. 灵活的注释方式

程序中以“//”开始的内容是注释,有效范围至本行结束。注释的目的是方便别人读懂程序,注释的内容在编译时不产生目标代码。所谓目标代码,就是程序可以执行的代码。另外一种注释形式是沿用 C 语言的语法,即从“/*”开始,到“*/”结束。【例 1.1】的程序中用它注释两行,它更适合注释占多行的情况。

3. 简单方便的输出语句

cout 用来将信息输出到屏幕上。语句

```
cout<<"欢迎光临!";
```

将“欢迎光临!”输出到屏幕上。它就是 C++ 的“语句”,语句以分号作为结束。就像自然语言中的句子一样,语句是 C++ 程序的最小独立单元。

4. 形影不离的头文件和命名空间

```
#include <iostream>           //包含头文件
using namespace std;          //使用命名空间
```

iostream 是头文件, std 是命名空间。这里暂不涉及它们的含义, 只要记住一般的程序都要具有这两条语句即可。第 2 章将详细介绍它们的含义。

1.2 程序的编辑、编译和运行的基本概念

本节简要介绍如何编辑【例 1.1】的程序, 并运行这个程序, 获得预定输出的全过程。

1.2.1 密切配合的三个主要步骤

用 C++ 语言写成的程序称为源程序, 源程序必须经过 C++ 编译程序翻译成机器语言才能执行。要得到一个用 C++ 语言设计的、名为 mycpp.exe 的可执行文件, 其过程可分为如下几步:

- (1) 先使用编辑器编辑一个 C++ 程序 mycpp.cpp, 可称其为 C++ 的源程序。
- (2) 然后使用 C++ 编译器对这个 C++ 程序进行编译, 产生文件 mycpp.obj。
- (3) 使用连接程序(又称为 Link), 将 mycpp.obj 变成 mycpp.exe 文件。

现在一般都使用集成环境。所谓集成环境, 就是将 C++ 语言的编辑、编译、连接和运行程序都集成为一个综合环境。由于目前 Windows 应用软件界面的标准化, 加之菜单名称和作用的规范以及 Windows 程序使用的普及, 大大降低了集成开发环境使用的困难, 集成环境的帮助文件更给学习带来方便。下面简要介绍如何利用微软公司的 Microsoft Visual C++ 6.0(为叙述方便, 将其简称为 VC)来编制并运行一个 C++ 程序。

1.2.2 编写控制台程序

本程序是在 DOS 环境下运行的程序。VC 提供了编制控制台程序的方法, 控制台程序兼容过去的 DOS 程序。可以利用 VC 提供的向导工具, 制作 C++ 程序源文件。

1. 设计一个工程

VC 是以工程的形式进行管理的, 它为每个工程建立一个与工程名称同名的目录, 所有的文件均建立在这个目录之下。在设计一个程序时, 必须首先为该程序建立一个工程。工程的名称可以跟程序名相同, 也可以不同。下面使用相同的名称, 说明建立的过程。

(1) 如图 1.1 所示, 从【开始】菜单的【程序】中选择 Microsoft Visual Studio 6.0 的 Microsoft Visual C++ 6.0 菜单项启动并进入开发环境。

(2) 选中 File 菜单的 New 命令, 弹出 New 对话框, 并选中 Projects 选项卡。图 1.2 是 Projects 选项卡示意图。

(3) 在左边栏中选择 Win32 Console Application, 在右边 Project name 输入框中输入工程名称 mycpp。系统自动把这个工程置于原来设置的工作目录之下, 例如原设置工作目录是在 F 盘的 PROG\MENKE\CH1 下面, 这里自动加上工程文件目录 mycpp。如

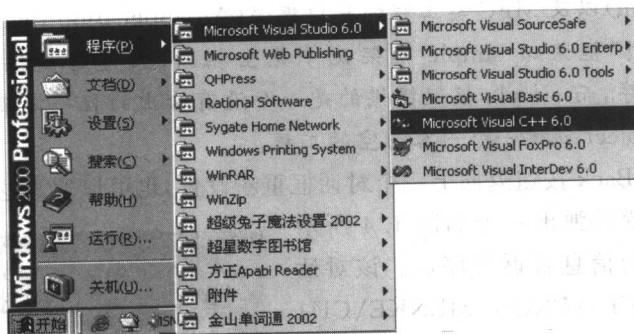


图 1.1 启动 Visual C++ 6.0 开发环境

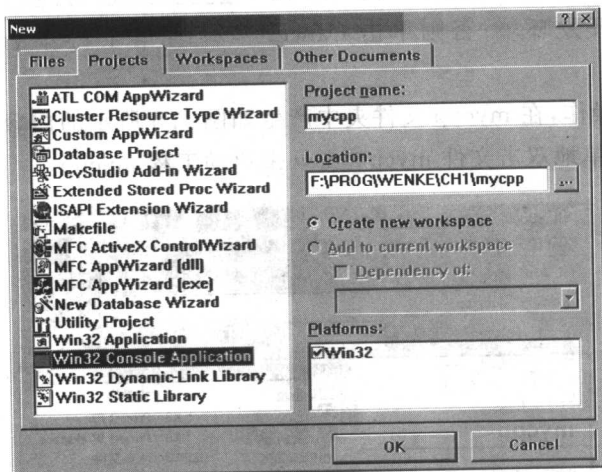


图 1.2 Projects 选项卡示意图

果想改变工作目录,在该输入框中输入即可。也可以单击右边 3 个点标记的按钮,在弹出的对话框中选择所要目录。

(4) 选择完毕。单击 OK 按钮即可进入 Win32 Console Application 制作向导的第一步,这时出现如图 1.3 所示的 Win32 Console Application 向导。

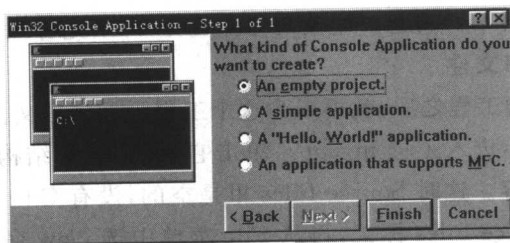


图 1.3 Win32 Console Application 向导

注意: 图 1.3 中的向导提供了四种工程内容,确切地说只有三种:即工程内容为空 (An empty project)、简单应用工程框架(两个,即 A simple application 和 A "Hello

World!” application)以及 MFC 类工程的接口框架(An application that supports MFC)。其中“Hello World!”框架与 simple 框架几乎完全相同,只是它提供一个输出“Hello World!”的演示程序,而 simple 框架提供的是一个没有输出的程序框架。后面将要练习使用 An empty project 创建一个内容为空的工程。

(5) 可以单击 Back 按钮返回上一个对话框重新设置,也可以取消这次操作。单击按钮 Finish,完成设置后弹出一个如图 1.4 所示的对话框,把创建的信息告诉程序员。该对话框表明已在给定的 F:\PROG\MENKE\CH1 目录下创建了一个空的工程 mycpp,这个工程已经具有目录 myfile。如果满意,单击 OK 按钮确认,返回集成开发环境;否则单击 Cancel 按钮撤销。

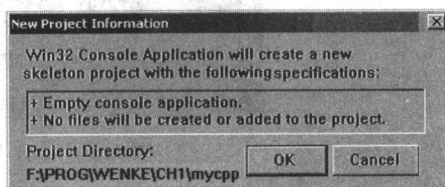


图 1.4 生成的工程信息

(6) 通过以上操作,在 mycpp 文件夹中产生如图 1.5 所示的内容。以后可以不直接启动 VC,而通过简单地双击文件 mycpp.dsw 来启动工程 mycpp。

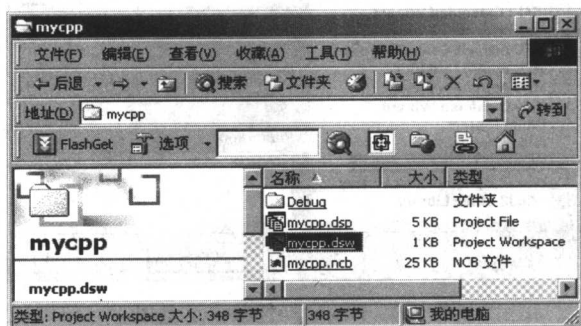


图 1.5 mycpp 文件夹下的内容

注意:需要说明的是,VC 是以工程的形式进行管理的,它为每个工程建立一个与工程名称同名的目录,所有的文件均建立在这个目录之下。虽然可以在建立过程中撤销工程,但这个目录已经建立,重新建立工程时,不能再使用这个名称。由此可见,在同一个工作目录之下,工程不能同名。要想继续使用这个名称,必须先将这个目录删除。

2. 为工程添加程序文件

产生名为 mycpp 的工程之后,就可以为这个工程添加程序文件,以便编写程序。

(1) 选中 FileView,进入工程 mycpp。单击它,展开树形结构。选中 mycpp files 结点,展开向导建立的内容。这时 Source Files 里是空的,没有 C++ 程序的源文件名称,如图 1.6 所示。选中 Source Files 标记,再从 File 菜单中选 New 命令,弹出 New 对话框。首先看到的是 Files 选项卡的内容,如图 1.7 所示。

(2) 选中 C++ Source File 选项,在右方 File 输入框中输入 C++ 程序文件名,这里选择的 CPP 程序文件名与工程同名,即输入 mycpp。因为系统默认的后缀是 .cpp,所以可以不使用后缀。

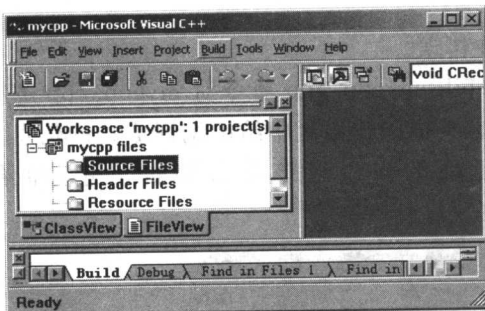


图 1.6 空项目内容示意图

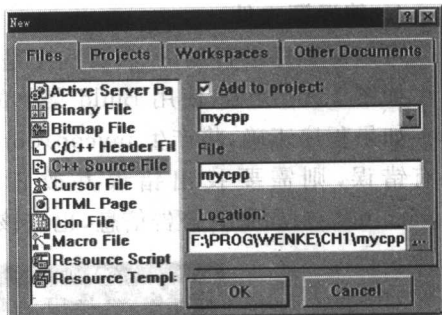


图 1.7 Files 选项卡

注意：如果要产生 C 文件，只要使用后缀 .C 即可。

(3) 单击 OK 按钮，返回集成环境。单击 Source Files 的展开标记 +，其下将出现如图 1.8 所示的 C++ 源程序文件名 mycpp.cpp。双击 mycpp.cpp，光标进入右边的编程区，以便编写 mycpp.cpp 的文件内容。

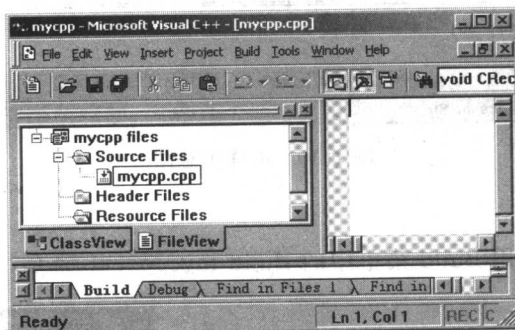


图 1.8 展开 Source Files 示意图

3. 编写源文件

在如图 1.8 右边所示的编程区中编写程序。图 1.9 是编写程序的示意图。

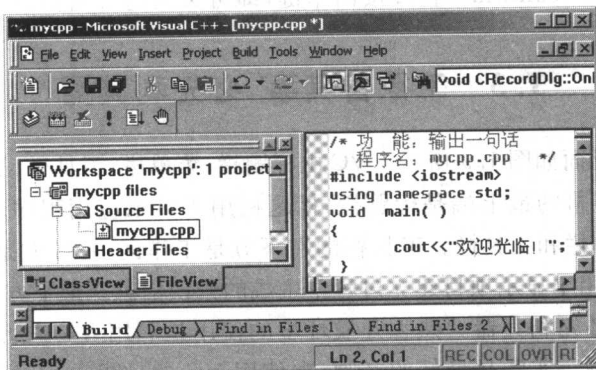


图 1.9 编写程序示意图

4. 编译源文件

可以对程序进行编译,看看程序是否存在错误。

如图 1.10 所示,使用 Build 菜单的 Compile mycpp.cpp 命令对 mycpp.cpp 进行编译。如果程序正确,将产生 mycpp.obj,同时在下方的输出区中给出编译成功的信息。如果有错误,则需要找出错误并改正之。正确的程序应该没有警告信息,即给出 0 warning(s)。如果有警告信息,也应该排除它们。

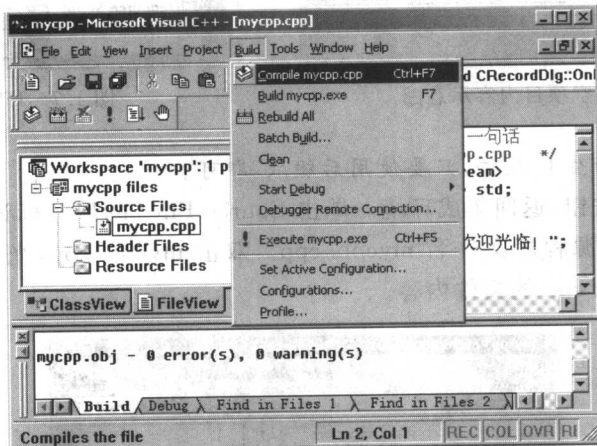


图 1.10 编译示意图

5. 运行文件

如图 1.10 所示,使用 Build 菜单中的 Build mycpp.exe 命令,产生 mycpp.exe 文件,同时给出如下信息:

mycpp.exe - 0 error(s), 0 warning(s)

运行 Build 菜单中的 Execute mycpp.exe 命令,得到如图 1.11 所示的运行界面和结果。按照提示 Press any key to continue,直接按回车键,即可关闭这个窗口。

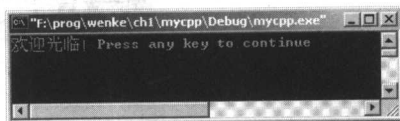


图 1.11 运行界面和结果示意图

1.3 工欲善其事,必先利其器

典型开发环境界面如图 1.9 所示。VC 由标题栏、菜单栏、工具栏、工作区、编程区、输出区和状态栏组成。屏幕的最上端是标题栏,标题栏用于显示应用程序名和所打开的文件。标题栏的下面是菜单栏和工具栏。工具栏的左下方是工作区,右下方是编程区。再下面是输出区,输出区主要用于显示项目建立过程中生成的错误信息。最下方是状态栏。

“工欲善其事,必先利其器。”熟悉开发环境的使用是利用 VC 进行实际的程序开发工作最基本的前提之一。实践是最好的老师,在使用 VC 的过程中,会发现它有许多有趣的特性,还可以定制个性化的编译器,使它更好地为自己服务。另外,VC 提供了强大的在线帮助。通过 Help 菜单可以查找各种联机帮助信息。