



高等学校“十一五”规划教材



C++ 语言程序设计教程

C++ Yuyan Chengxu Sheji Jiaocheng

杨卫平 主编

中国矿业大学出版社

China University of Mining and Technology Press

高等学校“十一五”规划教材

C++ 语言程序设计教程

主 编	杨卫平	
副主编	樊铭渠	郑效田
参 编	王正杰	翟海霞
	连 剑	翟 永

中国矿业大学出版社

内 容 提 要

本教程面向零起点读者,本着深入浅出、循序渐进、轻松入门的原则,详细讲解了 Visual C++ 语言的基础知识和基本概念、程序设计的基本控制结构、面向对象的基本思想及设计方法、MFC 应用程序的建立以及常用控件的使用、数据库编程等内容,并给出了丰富多彩的例题、习题和应用实例。

本书是 Visual C++ 语言程序设计的入门教程,可以作为高等院校非计算机专业学生学习第一门计算机程序设计语言的教材,也可供读者自学。

图书在版编目(CIP)数据

C++ 语言程序设计教程/杨卫平主编. —江苏徐州:

中国矿业大学出版社, 2006. 8

ISBN 7 - 81107 - 341 - 2

I. C… II. 杨… III. C 语言—程序设计—教材

N. TP312

中国版本图书馆 CIP 数据核字(2006)第 061315 号

书 名 C++ 语言程序设计教程

主 编 杨卫平

责任编辑 何 戈

责任校对 杜锦芝

出版发行 中国矿业大学出版社

(江苏省徐州市中国矿业大学内 邮编 221008)

网 址 <http://www.cumtp.com> E-mail: cumtpvip@cumtp.com

排 版 中国矿业大学出版社排版中心

印 刷 中矿大印发科技有限公司

经 销 新华书店

开 本 787×1092 1/16 印张 16 字数 400 千字

版次印次 2006 年 8 月第 1 版 2006 年 8 月第 1 次印刷

定 价 26.00 元

(图书出现印装质量问题,本社负责调换)

前 言

目前,高等院校特别是工科院校,越来越重视培养学生利用计算机解决本专业及相关领域中实际问题的能力。随着各学科专业课程计算机应用水平的提高,对计算机基础课程的教学提出了新的要求。因此,选择学习一门合适的计算机高级语言,缩短从程序设计入门到开发应用软件的过程,为后续课程的学习打下良好的基础,具有十分重要的意义。

在众多的高级语言中,Visual C++ 语言是使用最广泛的一种面向对象的程序设计语言,它从C语言发展演变而来,具有C语言高效、灵活的优点,还支持面向对象的方法。面向对象的方法使程序能够比较直接地反映问题的本来面目,使软件开发人员能够利用人类认识事物所采用的一般思维方法进行软件开发,并且可以大大提高程序的可重用性,使得软件的开发和维护都更为方便。

本书作为一本 Visual C++ 语言的入门教材,详细介绍了 Visual C++ 语言的基础知识,讲述了面向对象的方法以及 Windows 程序设计的基础知识,针对初学者的特点,本书采用了循序渐进、重点突出、难点分解、实例说明、图表助学、每章对难点和重点进行小结等方法,尽可能地做到深入浅出,力求让读者能够轻松入门,尽快学懂学会。书中的源程序均在 Visual C++ 6.0 编程环境中调试通过。

全书可分为四个部分:第一部分(第一章),详细讲解了 Visual C++ 语言的数据类型、运算符、表达式等基础知识和基本概念;第二部分(第二、第三、第四章),重点讲解了程序设计的基本控制结构、函数、数组、指针等内容,采用给出各控制结构执行流程图、列举大量例题、注释重点难点语句等多种形式,力争让零起点读者尽快掌握程序设计的基本方法;第三部分(第五章),通过对类和对象的基本概念、类的组合、继承、多态和类模板等的讲解,掌握面向对象程序设计的基本思想和方法;第四部分(第六、第七章),讲解了 MFC 方式下建立 Windows 应用程序的一般方法和步骤,通过应用实例,引导读者轻松入门,从而解决了初学者难以在短时间掌握开发 Windows 应用程序的难题。

本书全部内容建议使用 70 学时讲述,其中课堂讲述 40 学时,上机实验 30 学时。建议每章的讲课学时数分配如下:第一章 6 学时,第二章 8 学时,第三章 4 学时,第四章 6 学时,第五章 6 学时,第六章 6 学时,第七章 4 学时。

在本书编写过程中,得到了山东科技大学信电学院许多老师的热情帮助,他们在部分章节的撰写和书稿校对方面付出了辛勤劳动,在此一并表示衷心感谢。

由于编者水平有限,加之时间仓促,书中错误之处在所难免,敬请广大读者批评指正。

编者

2006 年 5 月于青岛

目 录

第一章 Visual C++ 6.0 语言基础	1
第一节 Visual C++ 6.0 语言概述	1
一、计算机语言的发展	1
二、Visual C++ 6.0 语言的发展	2
三、Visual C++ 6.0 语言的快速入门	2
第二节 一个简单 Visual C++ 6.0 控制台程序实例	3
一、一个简单实例	3
二、VC 的程序编写规则	8
三、常用 VC 源程序文件类型	9
第三节 数据类型	9
一、VC 的基本字符集和词汇集	11
二、基本数据类型	12
三、常量与变量	13
第四节 运算符和表达式	17
一、算术运算符和算术表达式	17
二、赋值运算符和赋值表达式	19
三、关系运算符和关系表达式	19
四、逻辑运算符和逻辑表达式	19
五、位运算符和表达式	19
六、条件运算符和条件表达式	21
七、逗号运算符和表达式	22
八、其他运算符	22
九、运算符优先级和结合性	22
十、数据类型转换	22
十一、表达式的书写	23
第五节 VC 控制台程序中数据的输入输出	24
一、数据输入输出的概念	24
二、数据的输入	24
三、数据的输出	25
本章小结	25
习题一	25

第二章 Visual C++ 基本语句	27
第一节 顺序结构	27
一、语句	27
二、表达式语句	27
三、复合语句	28
四、顺序结构	28
第二节 选择结构	29
一、if 语句	29
二、if 语句的嵌套	32
三、多重 if-else 语句	35
四、switch 语句	37
第三节 循环结构	41
一、while 循环语句	41
二、do...while 循环语句	44
三、for 循环语句	45
四、continue 语句	47
五、break 语句	49
六、循环嵌套	51
本章小结	53
习题二	53
第三章 函数	55
第一节 函数的定义	55
一、函数的定义	55
二、函数原型声明	56
第二节 函数的调用	57
一、函数的调用	57
二、函数的返回值	59
第三节 函数的参数传递	60
一、形式参数和实际参数	60
二、函数的传值调用	61
三、函数的引用调用	62
第四节 函数的嵌套调用	63
第五节 函数的递归调用	65
第六节 函数的重载	67
第七节 C++ 系统函数	68
一、数据转换函数	69
二、数学函数	69
本章小结	71

习题三	71
第四章 数组、指针与字符串	73
第一节 数组	73
一、一维数组的定义和初始化	74
二、一维数组的使用	75
三、二维数组	78
四、多维数组	80
第二节 指针	82
一、指针的定义及访问	82
二、指针的运算	84
三、指针与数组	85
四、指针与函数	89
五、动态存储分配	91
第三节 字符串	93
一、字符数组	93
二、指针访问字符串	95
三、字符串处理函数	97
第四节 结构	99
一、结构的定义	99
二、结构的访问	100
三、结构数组	102
四、结构指针	104
本章小结	107
习题四	107
第五章 类与对象	109
第一节 面向对象设计的基本思想	109
一、程序设计方法的演变	109
二、面向对象程序设计的基本特征	109
第二节 类与对象的基本概念	110
一、类与结构的比较	110
二、类定义	112
三、类的使用	112
四、数据成员	113
五、函数成员	113
第三节 构造函数和析构函数	114
一、构造函数	115
二、拷贝构造函数	117

三、析构函数	119
第四节 友元	119
一、为本类声明一个友元函数	119
二、将一个类说明为另一个类的友元	120
三、成员函数是友元函数	121
第五节 类的组合	122
一、既有类的再利用	122
二、组合对象的构造函数和析构函数	122
第六节 类的继承	125
一、继承的基本概念	125
二、派生类的构造函数和析构函数	129
第七节 多态性	131
一、多态性的基本概念	131
二、函数重载	132
三、虚函数	133
四、虚拟析构函数	136
五、运算符重载	138
第八节 模板	141
一、函数模板	141
二、类模板	143
本章小结	146
习题五	146
 第六章 MFC 编程简介	148
第一节 MFC 基础知识	148
一、窗口	148
二、事件驱动	149
三、句柄	149
四、消息	149
五、MFC 中类的组织	150
第二节 基于对话框的应用程序	152
一、应用程序向导的使用	152
二、MFC 程序的基本组成	157
三、对基于对话框程序的分析	159
第三节 Windows 标准控件的使用	161
一、按钮控件	161
二、编辑框控件	167
三、静态控件	170
第四节 资源的使用	173

一、菜单资源	173
二、工具条资源	175
第五节 编程实例——一个简单计算器的实现	177
一、界面设计	177
二、功能设计	178
三、实现代码	178
本章小结	190
习题六	191
第七章 Visual C++ 数据库编程	192
第一节 Visual C++ 数据库编程概述	192
一、数据库系统的基本概念	192
二、Visual C++ 开发数据库技术的特点	193
三、Visual C++ 开发数据库技术	194
四、SQL 语言简介	195
第二节 MFC 访问 ODBC 数据源	196
一、ODBC 概述	196
二、MFC App Wizard 使用 ODBC 的一般过程	198
三、CDatabase 类	202
四、CRecordSet 类	202
五、CRecordView 类	203
第三节 数据库操作	204
一、数据库连接	204
二、编辑记录	204
第四节 用 ADO 开发数据库应用程序	206
一、ADO 简介	206
二、使用 ActiveX 开发应用程序	210
三、开发 ADO 对象应用程序	213
第五节 MFC ODBC 数据库管理应用举例	220
一、实例概述	220
二、实例实现过程	220
本章小结	244
习题七	244
参考文献	245

第一章 Visual C++ 6.0 语言基础

【本章重点】 简单 Visual C++ 6.0 控制台程序;VC 的基本数据类型以及常量、变量的含义和用法;各种运算符的运算规则、优先级、结合性;各种表达式的含义和用法。

【本章难点】 VC 基本数据类型、常量、变量、运算符、表达式的概念及在程序中的用法。

C++是目前最受欢迎的程序设计语言之一,它既可以用来编写系统软件,也可以用来编写应用软件。由于其完全支持面向对象程序设计,所以通过C++的学习可以快速掌握面向对象程序设计的思路。本章是学习本书的基础,首先介绍C++语言的发展和特点,随后通过一个简单的实例介绍 Visual C++ 6.0 控制台程序的建立过程及文件类型,然后讲解VC的基本数据类型、运算符、表达式及程序的编写规则,最后介绍 VC 控制台程序中数据的输入与输出。

第一节 Visual C++ 6.0 语言概述

一、计算机语言的发展

计算机语言(Computer Language)是用于人与计算机之间通讯的语言,是人与计算机之间传递信息的媒介。20世纪40年代当计算机刚刚问世的时候,程序员必须手动控制计算机,使用机器语言来编程。后来为了减小编程的难度,人们用英文缩写的标识符代替原来的二进制符号指令,于是就产生了汇编语言。汇编语言的实质和机器语言是相同的,都是直接对硬件操作,只不过容易识别和记忆,所以汇编语言有时又称为伪机器语言。机器语言和汇编语言被称为“低级语言”。随着程序规模的不断扩大,汇编语言越来越不能满足人们的需要,20世纪50年代出现了“高级语言”,它接近人们习惯使用的自然语言和数学语言,使人们易于学习和使用,如C、BASIC、PASCAL等,后来又出现了许多升级版本。发展到目前,我们常用的高级语言有Microsoft Visual Studio. net、Delphi. net、ASP. net等,Microsoft Visual Studio. net 2003版集成了Microsoft Visual Basic. net、Microsoft Visual C++. net、Microsoft Visual C#. net、Microsoft Visual J++. net等。高级语言是目前绝大多数编程者的选择,和汇编语言相比,它将许多相关的机器指令合成为单条指令,大大简化了程序中的指令,同时由于省略了很多细节,编程者也就不需要有太多的专业知识,更易编写,而且程序的平台无关性使程序可以方便地被移植到网络上的不同机器、不同平台。未来的程序设计语言会是什么样的,现在还不能确切预知,但可以肯定的是未来的程序设计语言功能会更强大、编程会更容易,编程人员也许不用再写具体的编程语句,只需简单地描述一下程序的功能,计算机就会给我们编写出程序来。从计算机编程语言的发展过程我们可以看出,编程语言的功能越来越强大,程序员的编程工作越来越容易。

二、Visual C++ 6.0 语言的发展

在 1970 年, AT&T 贝尔实验室的 Ken Thompson 根据 BCPL (Basic Combind Programming Language) 语言设计出较先进的并取名为 B 的语言。1972 年美国的 Dennis Ritchie 在 B 语言的基础上设计发明了另一种语言, 取名为 C 语言。随着微型计算机的日益普及, 出现了许多 C 语言版本, 如 Turbo C、ANSI C、Borland C 等。微软推出 Windows 操作系统后, 推出 C 语言的可视化编程版本 Visual C++ 1.0 一直到 Visual C++ 6.0, 目前最新的是 Visual C++ .NET。

三、Visual C++ 6.0 语言的快速入门

刚刚起步学习计算机, 学习的第一个计算机语言就是 Visual C++ 6.0 语言, 那么怎样学才能学会? 如何快速入门 Visual C++ 6.0 语言?

在回答这几个问题前, 先浅谈一下 Visual C++ 6.0 语言的重要性。

首先, Visual C++ 6.0 语言的语法结构很简洁精妙, 写出的程序也很高效, 能够让编程人员深入系统底层, 编写底层硬件的驱动程序, 所以, 如果想在程序设计方面有所建树, 就必须去学习它。

其次, 学会 Visual C++ 6.0 语言, 再学习别的语言会非常轻松, 如履平地, 可以说, 如果掌握了 Visual C++ 6.0 语言, 就等于掌握了很多门语言, 因此招聘程序员时, 许多招聘单位都要求应聘者精通 Visual C++ 语言。

再者, 对于专业以硬件方面为主的学生来说, 以后的课程还要用到它, 如数据结构、单片机编程等, 所以为了顺利学习以后的课程, 也必须去学好它。

那么对于初学编程者, 怎样学才能学会? 如何快速入门呢? 根据编者多年学习计算机语言的体会, 提出以下几点建议:

(1) 初学者开始一定要调整好自己的心态, 不要急于求成, 不用怕学不会。不少人在学习 Visual C++ 6.0 之前, 可能听到了不少人说 Visual C++ 6.0 非常难学。的确, 相对于 Visual Basic 或 Delphi 来说, VC 是比较难入门并且不容易学会, 但肯定能学会。刚开始学习时, 应顺着教材一步一步地走, 先记忆教材上的知识, 等到以后学习的内容逐渐增多, 再逐渐加深理解, 遵循由易入门、逐渐深入的学习方法。

(2) 勤学多练。不管是初学者还是已经能熟练运用者, 在学习新的内容时总会遇到不容易懂的知识点, 这时可以采取的方法是多看几遍书, 多问几个人, 多上机练习几次, 难点最终会被克服。

(3) 会丢包袱。Visual C++ 6.0 语言的功能非常强大, 内容也非常多, 初学者一定不要把它的的功能都学会, 只需要先掌握基本的概念和方法, 再把常用的部分和大的编程框架学会学懂就可以了, 以后需要实现新的软件功能时, 再去查找相应的参考资料来编程实现。

第二节 一个简单 Visual C++ 6.0 控制台程序实例

一、一个简单实例

控制台程序是一类功能和操作都比较简单的程序,下面我们建立一个简单的 Visual C++ 6.0 控制台程序,操作方法如下:

(1) 单击“开始→所有程序→Microsoft Visual Studio 6.0→Microsoft Visual C++ 6.0”,启动 Visual C++ 6.0,如图 1-1 所示。启动后显示 Visual C++ 6.0 开发平台,如图 1-2 所示。



图 1-1 从开始菜单启动 Microsoft Visual C++ 6.0 开发环境

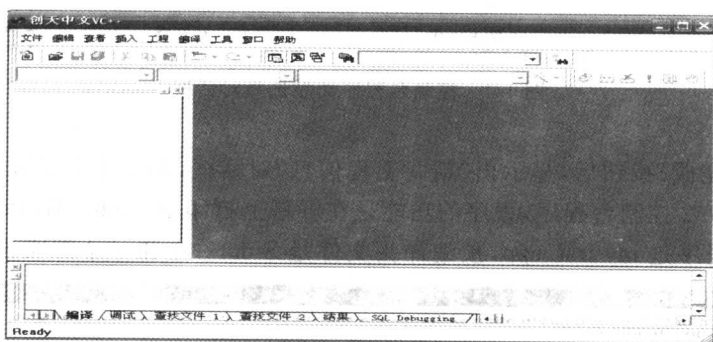


图 1-2 Visual C++ 6.0 开发平台

(2) 单击“文件→新建”菜单命令,显示出新建对话框,如图 1-3 所示。在新建对话框单击“工程”标签,选中“Win32 Console Application”项,然后在“工程”框中键入控制台应用程序项目名称,在这里取名为 myfirst,工程的所有文件保存在“位置”框中的文件夹内,该工程的文件保存在“C:\Program Files\Microsoft Visual Studio\MyProjects\myfirst”文件夹。为了便于个人工程的管理,最好建立保存自己工程的文件夹,可在“位置”下的编辑框中直接键入文件夹名称,也可单击“位置”框右方的按钮选择一个已有的文件夹。

(3) 单击“确定”按钮,显示出“Win32 Console Application—step 1 of 1”对话框,选择控

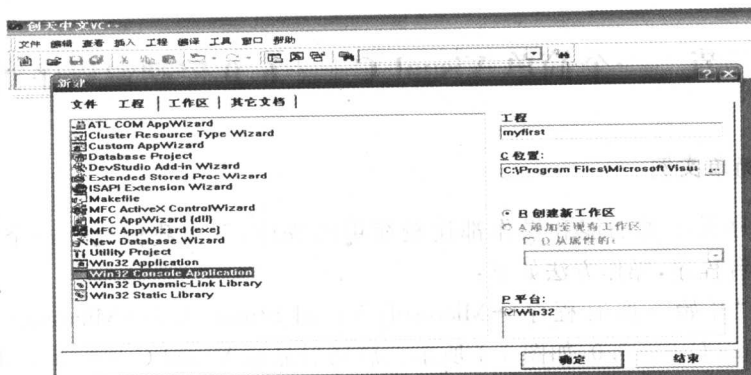


图 1-3 新建对话框

制台程序类型,如图 1-4 所示。选中“A "Hello, World!" application”项,然后单击“完成”按钮。

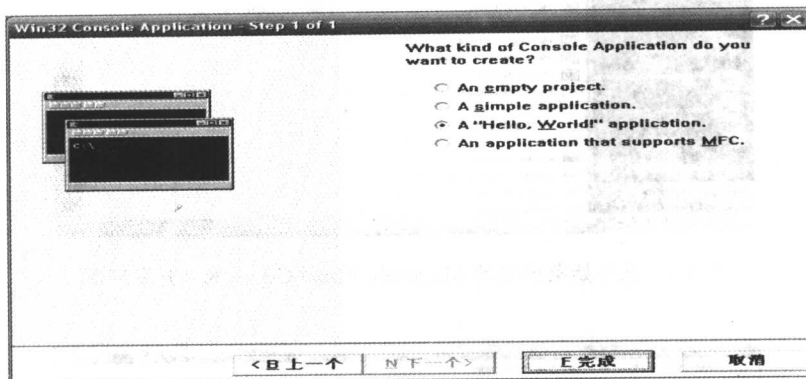


图 1-4 控制台程序类型选择对话框

(4) 单击“完成”按钮后,显示出“新建工程信息”对话框,如图 1-5 所示。新建的工程是一个简单的 Win32 控制台程序,程序的功能是在屏幕上打印出“Hello,World!”后退出,并显示该工程的主程序是 myfirst.cpp,预编译头文件是 Stdafx.h 和一个源文件 Stdafx.cpp,对

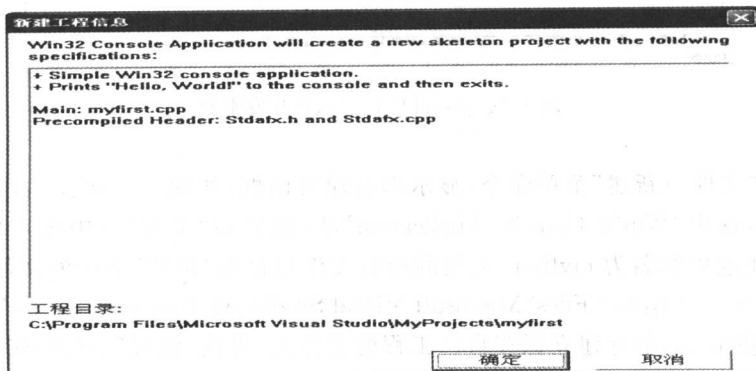


图 1-5 新建工程信息对话框

话框下方显示存放当前工程的文件夹是：“C:\Program Files\Microsoft Visual Studio\MyProjects\myfirst”。如果要建立该工程，则单击“确定”按钮，系统将自动创建此工程；若不想建立该工程，则单击“取消”按钮，系统将撤销此工程的建立。

(5) 单击“确定”按钮，系统进入 myfirst 工程开发平台，在工作区内显示出工程 myfirst 的两个视图，当前显示的是 FileView，即文件视图，如图 1-6 所示，可以看到当前工程中的所有文件：在 Source Files 文件夹即源文件夹中有 myfirst.cpp 和 Stdafx.cpp 两个源文件，源文件是存放程序代码的文件；在 Header Files 文件夹即头文件夹中有 Stdafx.h 这个头文件，头文件一般存放预定义的常量、函数等；在 Resource Files 文件夹即资源文件夹中目前没有文件，资源文件一般是存放图形、图标、菜单的文件；还有一个文档文件 ReadMe.txt，文档文件一般用来说明系统中各类文件的作用、软件使用说明等。用鼠标双击相应的文件，就可把该文件打开，例如用鼠标双击 myfirst.cpp 后，窗口显示 myfirst.cpp 文件的内容，如图 1-7 所示，它只有几行，每行的作用如下：

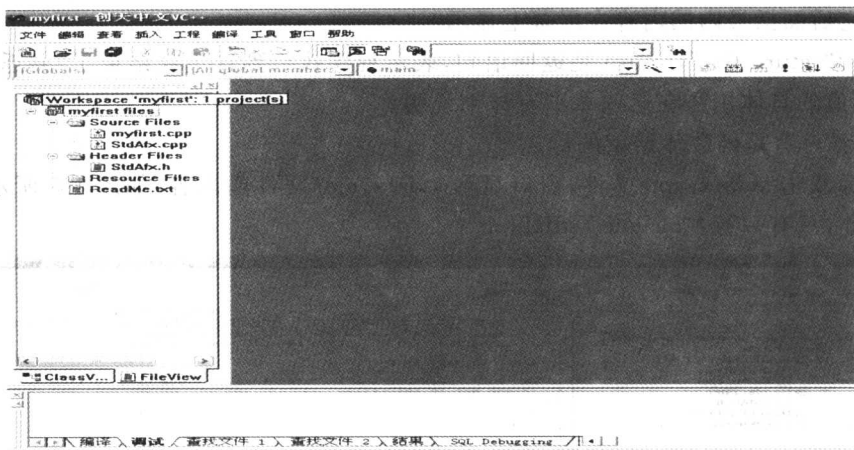


图 1-6 myfirst 工程开发平台文件视图

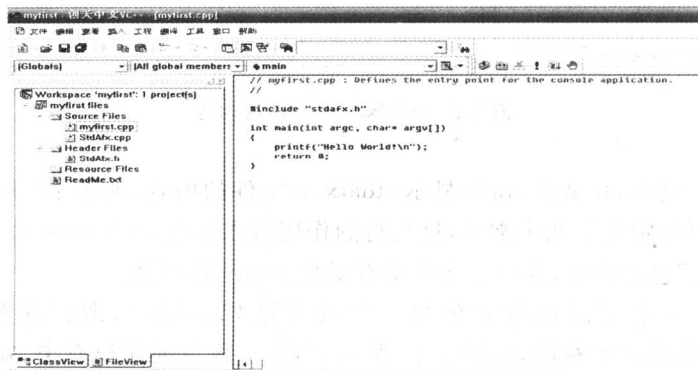


图 1-7 myfirst.cpp 文件的内容

第一行“// myfirst.cpp : Defines the entry point for the console application”，是注释行，“//”是注释行开始的标志，本句注释说明文件 myfirst.cpp 的主要作用是定义了控制台程序

的入口点,也就是说程序从这个文件开始执行。注释行在程序执行时被忽略,但对编程人员来说很重要,它使程序易读,并为程序以后的修改提供极大的便利。

第二行“//”,是一行空注释行。

第三行“#include "stdafx.h"”,#include 是一个预编译指令,告诉系统在编译后面的程序之前先编译 stdafx.h 文件的内容。

第四行“int main(int argc, char * argv[])”,声明了系统的主函数 main,函数 main 是控制台程序执行的开始点,每一个控制台程序都必须有且只能有一个 main 函数。函数是程序的基本功能单位,程序的功能就是由一个或多个函数实现的,main 前面的 int 定义了 main 函数运行结束后要返回一个整形数,main 后面的(int argc, char * argv[])是 main 函数的参数。

第五行“{”,是函数体开始的标志。

第六行“printf("Hello World!\n");”,printf 是一个系统已经定义好的函数即系统函数,功能是把它后面括号内的内容在屏幕上显示。

第七行“return 0;”,return 是返回语句,表示程序执行到这里就结束了,并返回一个整数零,返回值的具体用法在第三章中有详细说明。

第八行“}”,是函数体结束的标志。

用鼠标双击 stdafx.cpp 文件,窗口显示 stdafx.cpp 文件的内容,如图 1-8 所示,它除了几行注释外,只有一行#include "stdafx.h"。

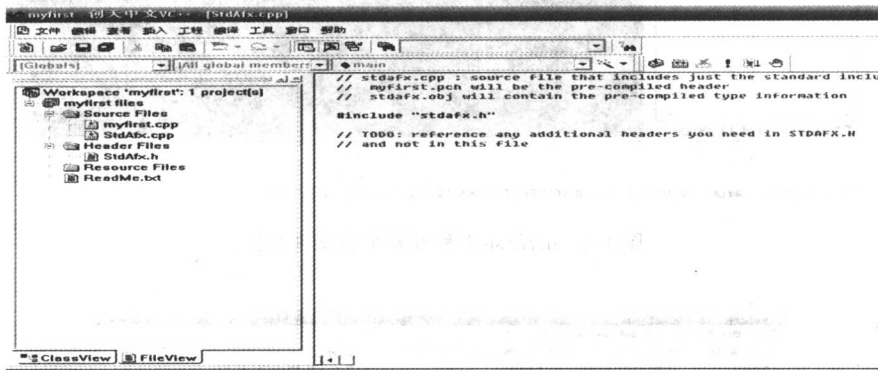


图 1-8 stdafx.cpp 文件的内容

用鼠标双击 stdafx.h 文件,窗口显示 stdafx.h 文件的内容,如图 1-9 所示,这个文件主要是用#define 语句定义了几个常量,这几行的作用可以忽略,#include <stdio.h>是把系统头文件“stdio.h”包含进来,这个头文件里有函数 printf 的声明。

工作区内显示出工程 myfirst 的另一个视图是 ClassView,即类视图。用鼠标单击 ClassView 按钮,则显示类视图,如图 1-10 所示,当前工程目前还没有类,在 Globals 文件夹即全局文件夹中只有一个主函数 main 的声明。用鼠标双击 main 函数,可以打开 main 函数所在的源文件,并定位到 main 函数的位置,如图 1-11 所示。

以上就是 VC 自动生成的各类文件,单击编译→执行 myfirst.exe 文件菜单(图 1-12)或同时按 Ctrl+F5 键,系统对 myfirst 工程进行编译、连接,同时在输出窗口中在线显示出编

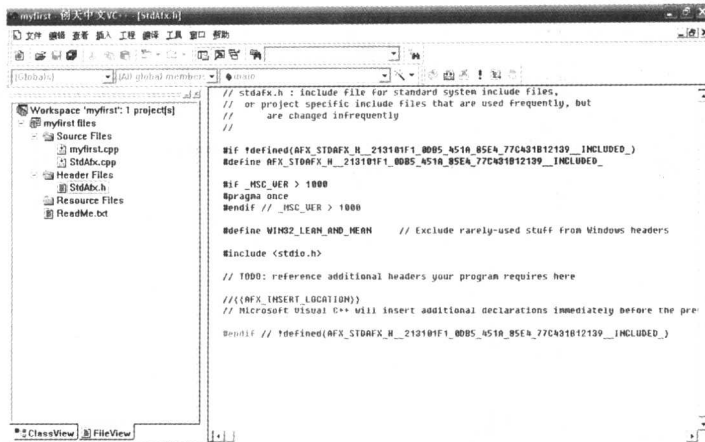


图 1-9 stdafx.h 文件的内容

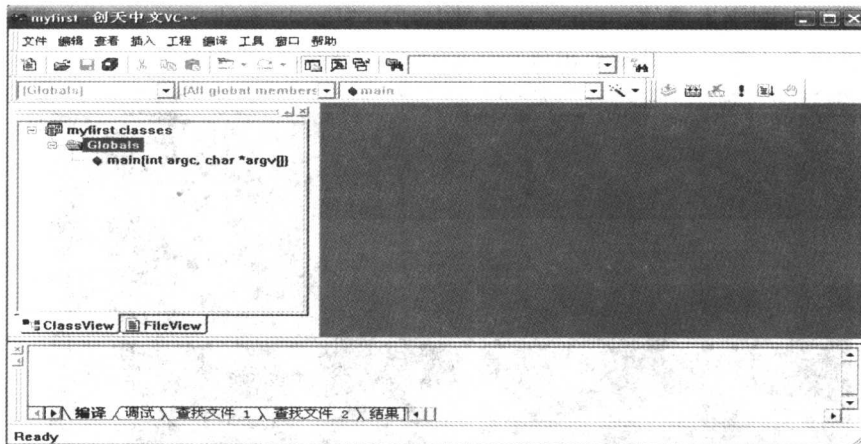


图 1-10 myfirst 工程开发平台类视图

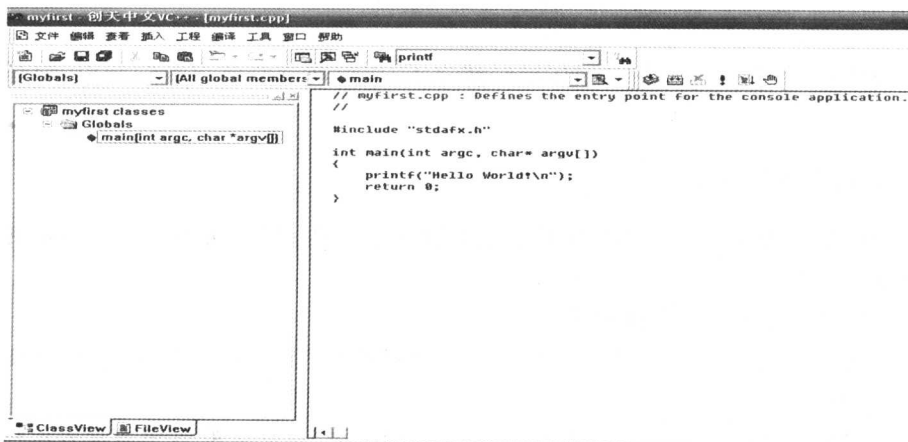


图 1-11 鼠标双击 main 函数后打开 main 函数所在的文件

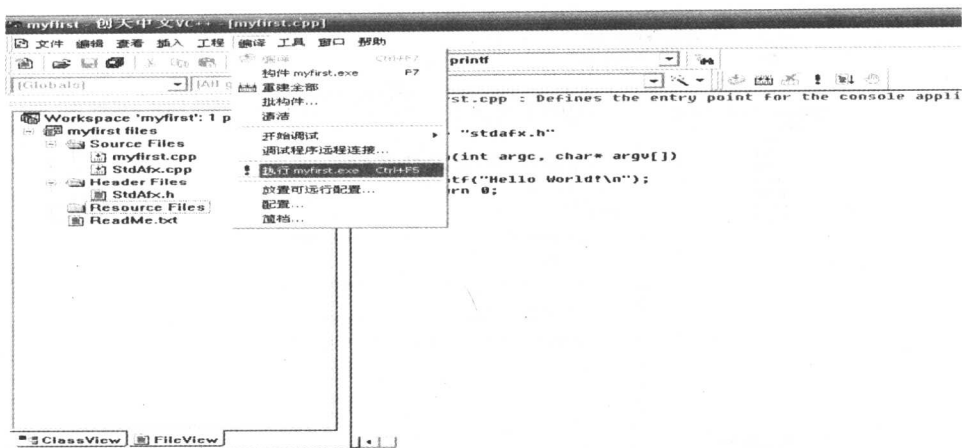


图 1-12 编译→执行 myfirst.exe 文件菜单

译的内容,并运行刚刚生成的 myfirst.exe(图 1-13)。“Press any key to continue”是系统自动加上去的,在键盘上按任意键则退出该程序的运行,返回 VC 开发平台。

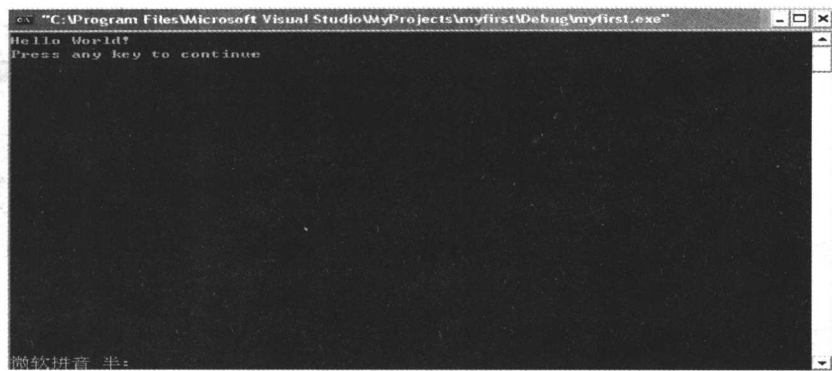


图 1-13 运行结果

到此为止,一个完整的控制台程序已经建立了,以后将逐步学习新的知识,编写用户自己的程序。

二、VC 的程序编写规则

前面的控制台程序例子中的程序代码全是系统自动生成的,从中可以看出在书写程序时应遵循以下规则:

- (1) 各个单词之间要用空格或其他分界符分开。
- (2) 一条语句占一行,每一条语句都必须以“;”结尾,但预处理命令、函数头和花括号“{”之后不加分号。
- (3) “{”都是成对出现的,相匹配的大括号在同一列,括起来的语句通常表示程序的某一层结构,内部再嵌套“{”时,应进行缩进排列,以便看起来层次清晰,增加程序的可读性。缩进是指程序在书写时不要将程序的每一行都由第一列开始,在适当的地方加进一些空