

C YUYAN CHENGXU SHEJI  
SHIYAN ZHIDAO YU XITI JIEDA

# C语言程序设计

## 实验指导与习题解答

主 编 / 甘 岚

副主编 / 刘媛媛 范 萍 雷莉霞



西南交通大学出版社

# C 语言程序设计实验指导与习题解答

主 编 甘 岚

副主编 刘媛媛 范 萍 雷莉霞

西南交通大学出版社

· 成 都 ·

图书在版编目(CIP)数据

C语言程序设计实验指导与习题解答 / 甘岚主编. —  
成都: 西南交通大学出版社, 2015.9  
ISBN 978-7-5643-4233-3

I. ①C… II. ①甘… III. ①C语言-程序设计-高等  
学校-教学参考资料 IV. ①TP312

中国版本图书馆CIP数据核字(2015)第199206号

---

C语言程序设计实验指导与习题解答

主编 甘岚

责任编辑 姜锡伟  
特邀编辑 穆丰  
封面设计 墨创文化

---

出版发行 西南交通大学出版社  
(四川省成都市金牛区交大路146号)

发行部电话 028-87600564 028-87600533

邮政编码 610031

网址 <http://www.xnjdcbs.com>

---

印刷 四川森林印务有限责任公司

成品尺寸 185 mm × 260 mm

印张 14.25

字数 353 千

版次 2015年9月第1版

印次 2015年9月第1次

书号 ISBN 978-7-5643-4233-3

定价 26.80 元

---

课件咨询电话: 028-87600533

图书如有印装质量问题 本社负责退换

版权所有 盗版必究 举报电话: 028-87600562

# 前 言

C 语言程序设计是一门实践性很强的课程，不断地进行编程实践是学好本课程的必要条件。因此，作为编者，我们必须对于实践环节的教学和训练足够重视。为了能让广大初学者在短期内迅速具备 C 语言编程能力，同时为课程相关实验提供配套指导，我们特地组织了理论和实践教学经验丰富的教师编写了本书。

本书共分为 11 章。第 1 章介绍了 Visual C++6.0 可视化工具的编辑、编译和调试方法以及使用注意事项等内容；第 2 章的内容包括常见的数据结构和一些常用算法的相关习题的解答；第 3 章给出了 C 语言基本的数据类型、运算符和表达式等相关习题和解答，并提供了一个配套实验；第 4、5、6 章的内容囊括了 C 语言中的三种常用程序结构，即顺序结构、选择结构和循环结构的相关练习和解答，并提供了 4 个相关实验；第 7 章介绍了数组的基本概念以及一维、二维、多维和字符数组的定义及使用，提供了相关习题的解答，并给出了数值数组和字符数组两个实验；第 8 章介绍了函数的定义和使用方法以及编译预处理的相关知识，提供了相关的练习和习题解答，并给出了函数定义和函数传址调用的相关实验；第 9 章的内容主要是介绍了指针的概念及使用方法，对相关习题进行了解答，并提供了指针与数组及其应用的相关实验；第 10 章介绍了构造型数据类型的相关概念，给出了相关的习题的解答，并提供了一个配套实验；第 11 章主要是文件使用的习题解答和相关实验等内容。

本书内容丰富，实用性强，在实验设计上，注重题目的代表性、典型性，并力求对问题的阐述详尽细致，为初学者更好地服务。每个实验中亦会提出思考问题，让学生在完成实验之后可以灵活应用所学知识作出解答，提升编程能力。与此同时，本书注重程序代码设计的良好风格，让初次学习程序设计的学生从一开始就认识和培养良好的程序设计风格。本书的全部实验及例题已在 Visual C++6.0 平台调试通过，可直接引用，非常符合 C 语言程序设计课程实践性很强的教学特色，不仅可以作为自学者的参考资料，而且可以作为其他各类 C 语言教材的辅助教材，也可供各类报考 C 语言考试的读者参考。

本书由华东交通大学甘岚教授负责组织编写并统稿。刘媛媛、范萍和雷莉霞参与编写。由于作者水平有限，加之时间仓促，书中不当之处在所难免，敬请读者批评指正。

编 者

2015 年 5 月

# 目 录

1 C 语言程序设计概述	1
1.1 知识介绍	1
1.2 VC6.0 介绍	1
1.2.1 编制并运行一个简单程序	1
1.2.2 VC6 集成开发环境使用参考	7
1.2.3 调试程序	11
1.2.4 编译与链接过程中常见的出错提示	16
1.3 实验部分	20
1.3.1 实验 1: VC++6.0 下新建一个简单 C 源程序	20
1.3.2 实验 2: VC++6.0 下新建一个简单 C 程序工程	24
1.4 习题答案	29
2 数据结构与算法概述	30
2.1 知识介绍	30
2.2 习题答案	31
3 基本数据类型、运算符和表达式	35
3.1 知识介绍	35
3.2 实验部分	37
3.2.1 实验 1 数据类型的应用	37
3.2.2 实验 2 上机实验解答	41
3.3 练习解答	46
3.4 习题答案	47
4 顺序结构	49
4.1 知识介绍	49
4.2 实验部分	51
4.2.1 实验 1 简单 C 程序编程	51
4.2.2 实验 2 上机实验解答	56
4.3 练习解答	61
4.4 习题答案	61
5 选择结构	64
5.1 知识介绍	64

5.2 实验部分	65
5.2.1 实验 1 if 语句	65
5.2.2 实验 2 switch 语句	72
5.2.3 实验 3 上机实验解答	74
5.3 练习解答	82
5.4 习题答案	82
6 循环结构	85
6.1 知识介绍	85
6.2 实验部分	88
6.2.1 实验 1 循环结构	88
6.2.2 实验参考	94
6.3 练习解答	99
6.4 习题答案	99
7 数 组	105
7.1 知识介绍	105
7.2 实验部分	108
7.2.1 实验 1 数值数组	108
7.2.2 实验 2 字符数组	112
7.2.3 实验参考	117
7.3 练习解答	133
7.4 习题答案	134
8 函数与编译预处理	141
8.1 知识介绍	141
8.2 实验部分	145
8.2.1 实验 1 简单函数的定义及调用	145
8.2.2 实验 2 函数的传地址调用	150
8.2.3 实验习题解答	152
功能说明: 通过调用数制转换函数把整数转换成二进制字符输出。	158
8.3 练习解答	159
8.4 习题答案	160
9 指 针	166
9.1 知识介绍	166
9.2 实验部分	168
9.2.1 实验 1 指针与数组	168
9.2.2 实验 2 指针的其他应用	172
9.2.3 实验习题解答	174
9.3 练习解答	182

---

9.4 习题答案	182
10 构造型数据类型	189
10.1 知识介绍	189
10.2 实验部分	192
10.2.1 实验 1 构造型数据类型	192
10.2.2 实验习题解答	195
10.3 练习解答	200
10.4 习题答案	200
11 文 件	203
11.1 知识介绍	203
11.2 实验部分	204
11.2.1 实验 1 文件的基本操作	204
11.2.2 实验习题解答	208
11.3 练习解答	216
11.4 习题答案	216
参考文献	219



# 1 C 语言程序设计概述

## 1.1 知识介绍

(1) 程序设计语言按照与计算机硬件的联系程度分为三类，即机器语言、汇编语言和高级语言。前两类依赖于计算机硬件，有时统称为低级语言，而高级语言与计算机硬件关系较小。

(2) 按照结构化程序设计的观点，任何算法功能都可以通过由组成程序模块的三种基本结构的组合（顺序结构、选择结构和循环结构）组成。

(3) 面向对象的特点是：抽象、继承、封装、多态。

(4) 编译程序的工作过程一般也可以划分为五个阶段：词法分析，语法分析，语义分析，中间代码生成、优化与目标代码生成。

(5) 运行一个 C 源程序，需要如下几个步骤：输入编辑源程序→编译源程序→链接库函数→运行目标程序。

(6) C 程序的集成开发工具基本特点：符合标准 C，各系统具有一些扩充内容，能开发 C 语言程序。

## 1.2 VC6.0 介绍

Visual C++6.0，简称 VC 或者 VC6.0，是微软推出的一款 C 编译器，是能将“高级语言”翻译为“机器语言（低级语言）”的程序。

### 1.2.1 编制并运行一个简单程序

#### 1. 编制并运行程序的“四步曲”

让我们用 VC6 先来编制一个最简单的程序，并让它运行（执行）而得出结果，以此来作为了解 VC6 的开端。这个程序的功能仅仅是向屏幕上输出一个字符串“Hello World”。程序虽小，但与编制运行大程序的整个过程是相同的，都包含着如下所谓的“四步曲”：

(1) 编辑（把程序代码输入，交予计算机）。

(2) 编译（转成目标程序文件“.obj”）。编译就是把高级语言变成计算机可以识别的二进制语言。因为计算机只认识 1 和 0，所以编译程序把人们熟悉的语言换成计算机能识别的二进制的语言。编译程序把一个源程序翻译成目标程序的工作过程分为五个阶段：词法分析，语法分析，语义检查，中间代码生成、优化和目标代码生成。其中，主要是进行词法分析和语法分析，又称为源程序分析，分析过程中发现有语法错误，给出提示信息。

(3) 链接（成可执行程序文件“.exe”）。链接是将编译产生的“.obj”文件和系统库文件



链接装配成一个可以执行的程序文件。由于在实际操作中可以直接点击 Build 从源程序产生可执行程序,可能有人就会质疑:为何要将源程序翻译成可执行文件的过程分为编译和链接两个独立的步骤,不是多此一举吗?之所以这样做,主要是因为在一个较大的复杂项目中,有很多人共同完成一个项目(每个人可能承担其中一部分模块),其中有的模块可能是用汇编语言写的,有的模块可能是用 VC 写的,有的模块可能是用 VB 写的,还有的模块可能是购买(不是源程序模块而是目标代码)或已有的标准库模块。因此,各类源程序都需要先各自编译成目标程序文件(二进制机器指令代码),再通过链接程序将这些目标程序文件连接装配成可执行文件。

#### (4) 运行(可执行程序文件)。

上述四个步骤中,第一步的编辑工作是最繁杂而又必须细致地由程序员在计算机上来完成的,其余几个步骤则相对简单,基本上由计算机自动完成。

## 2. 工程(Project)以及工程工作区(Project Workspace)

在开始编程之前,必须先了解工程(Project,也称“项目”,或称“工程项目”)的概念。工程具有两种含义:一种是指最终生成的应用程序;另一种则是指为了创建这个应用程序所需的全部文件的集合,包括各种源程序、资源文件和文档等。绝大多数较新的开发工具都利用工程来对软件开发过程进行管理。

用 VC6 编写并处理的任何程序都与工程有关(都要创建一个与其相关的工程),而每一个工程又总与一个工程工作区相关联。工作区是对工程概念的扩展。一个工程的目标是生成一个应用程序,但很多大型软件往往需要同时开发数个应用程序,VC 开发环境允许用户在一个工作区内添加数个工程,其中有一个是活动的(缺省的),每个工程都可以独立进行编译、链接和调试。

实际上,VC6 是通过工程工作区来组织工程及其相关元素的,就好像是一个工作间(对应于一个独立的文件夹,或称子目录),以后程序所牵扯到的所有的文件、资源等元素都将放入到这一工作间中,从而使得各个工程之间互不干扰,使编程工作更有条理,更具模块化。最简单情况下,一个工作区中存放一个工程,代表着某一个要进行处理程序(我们先学习这种用法)。但如果需要,一个工作区中也可以存放多个工程,其中可以包含该工程的子工程或者与其有依赖关系的其他工程。

可以看出,工程工作区就像是一个“容器”,由它来“盛放”相关工程的所有有关信息。当创建新工程时,同时要创建这样一个工程工作区,而后则通过该工作区窗口来观察与存取此工程的各种元素及其有关信息。创建工程工作区之后,系统将创建出一个相应的工作区文件(.dsw),用来存放与该工作区相关的信息;另外,还将创建出的其他几个相关文件是工程文件(.dsp)以及选择信息文件(.opt)等。

编制并处理 C++ 程序时要创建工程,VC6 已经预先为用户准备好了近 20 种不同的工程类型以供选择,选定不同的类型意味着让 VC6 系统帮着提前做某些不同的准备以及初始化工作(例如,事先为用户自动生成一个所谓的底层程序框架,又称框架程序,并进行某些隐含设置,如隐含位置、预定义常量、输出结果类型等)。在工程类型中,有一个为“Win32 Console Application”,它是我们首先要掌握的,是用来编制运行 C++ 程序方法中最简单的一种。此种类型的程序运行时,将出现并使用一个类似于 DOS 的窗口,提供对字符模式的各种处理与支

持。实际上，它提供的只是具有严格的采用光标而不是鼠标移动的界面。此种类型的工程小巧而简单，但已足以解决并支持本课程中涉及的所有编程内容与技术，使我们把重点放在程序的编制而非界面处理等方面。至于 VC6 支持的其他工程类型（其中有许多还将涉及 Windows 或其他的编程技术与知识），有待在今后的不断学习中来逐渐了解、掌握与使用。

### 3. 创建工程并输入源程序代码

为了把程序代码输入给计算机，需要使用 VC6 的编辑器。如前所述，首先要创建工程以及工程工作区，而后才能输入具体程序完成所谓的“编辑”工作（注意，该步工作在四步骤中最繁杂，而又必须细致地由人工来完成）。

#### (1) 新建“Win32 Console Application”工程。

选择菜单 File 下的 New 项，会出现一个选择界面，在属性页中选择 Projects 标签后，会看到近 20 种的工程类型，我们只需选择其中最简单的一种“Win32 Console Application”，而后在右上处的“Location”文本框和“Project name”文本框中填入工程相关信息所存放的磁盘位置（目录或文件夹位置）以及工程的名字即可，如图 1-1 所示。

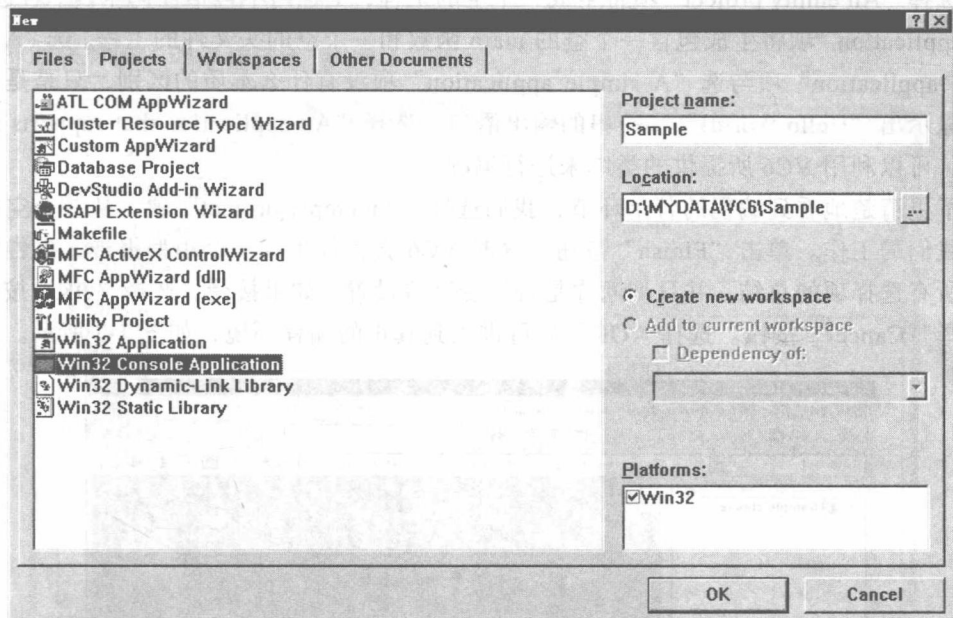


图 1-1 新建一个名为 Sample 的工程（同时自动创建一工作区）

在图 1-1 中，“Location”文本框中填入“D:\MYDATA\VC6”，这表示准备在 D 磁盘的“\MYDATA\VC6”文件夹，即子目录下存放与工程工作区相关的所有文件及其相关信息。当然，也可通过点击其右部的“...”按钮去选择并指定这一文件夹即子目录位置。然后在“Project name”文本框中填入如“Sample”的工程名（注意，名字一般根据工程性质确定，此时 VC6 会自动在其下的“Location”文本框中用该工程名“Sample”为你建立一个同名子目录，随后的工程文件以及其他相关文件都将存放在这个目录下）。

选择“OK”按钮进入下一个选择界面。这个界面主要是询问用户想要创建一个什么类型的工程，如图 1-2 所示。

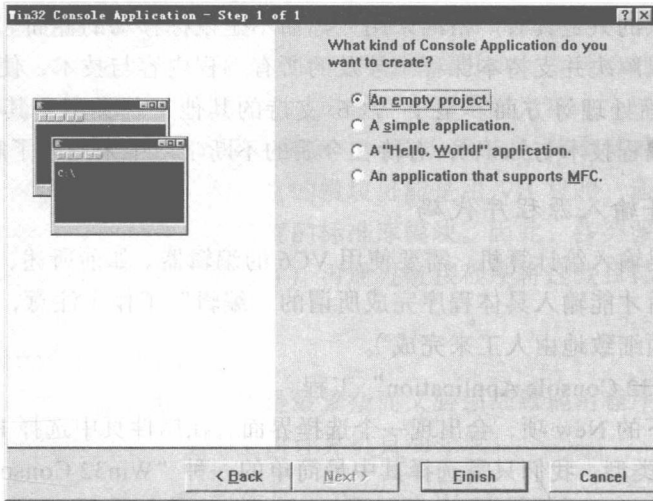


图 1-2 选择创建一个什么样的工程

若选择“An empty project”项将生成一个空的工程，工程内不包括任何东西。若选择“A simple application”项将生成包含一个空的 main 函数和一个空的头文件的工程。选“A ‘Hello World!’ application”项与选“A simple application”项没有什么本质的区别，只是是否需要包含有显示出“Hello World!”字符串的输出语句。选择“An application that supports MFC”项的话，可以利用 VC6 所提供的类库来进行编程。

为了更清楚地看到编程的各个环节，我们选择“An empty project”项，从一个空的工程开始我们的工作。单击“Finish”按钮，这时 VC6 会为你生成一个小型报告，报告的内容是刚才所有选择项的总结，并且询问你是否接受这些设置。如果接受，选择“OK”按钮；否则，选择“Cancel”按钮。选择“OK”后可进入到真正的编程环境，如图 1-3 所示。

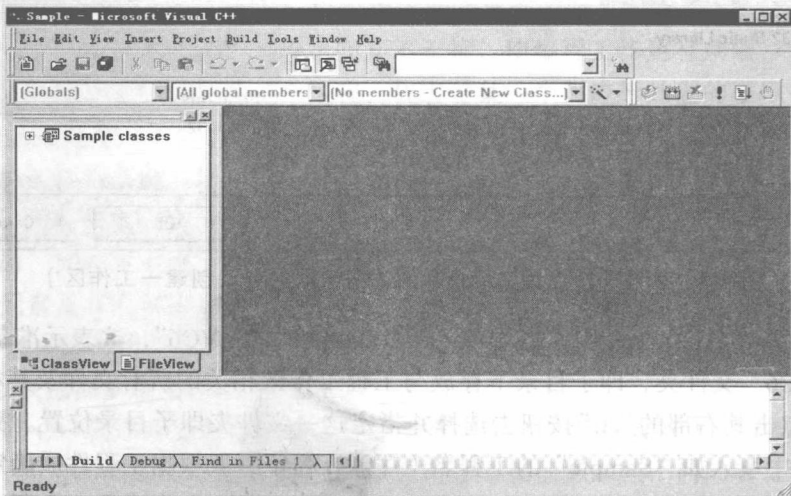


图 1-3 刚完成创建 Sample 工程的 VC6 集成开发环境窗口

(2) 在工作区窗口中查看工程的逻辑架构。

注意屏幕中的 Workspace 窗口，该窗口中有两个标签，一个是 ClassView，一个是 FileView。

ClassView 中列出的是这个工程中所包含的所有类的有关信息，当然我们的程序将不涉及类，这个标签中现在是空的。点击 FileView 标签后，将看到这个工程所包含的所有文件信息。点击“+”图标打开所有的层次会发现有三个逻辑文件夹：Source Files 文件夹中包含了工程中所有的源文件；Header Files 文件夹中包含了工程中所有的头文件；Resource Files 文件夹中包含了工程中所有的资源文件。所谓资源就是工程中所用到的位图、加速键等信息，在我们的编程中不会牵扯到这一部分内容。现在，FileView 中也不包含任何东西。

逻辑文件夹是逻辑上的，它们只是在工程的配置文件中定义的，在磁盘上并没有物理地存在这三个文件夹。我们也可以删除自己不使用的逻辑文件夹；或者根据我们项目的需要，创建新的逻辑文件夹来组织工程文件。这三个逻辑文件夹是 VC 预先定义的，就编写简单的单一源文件的 C 程序而言，我们只需要使用 Source Files 一个文件夹就够了。

(3) 在工程中新建 C 源程序文件并输入源程序代码。

下面该轮到生成一个“Hello.c”的源程序文件，而后通过编辑界面来输入所需的源程序代码。选择菜单“Project”→“Add To Project”→“New”项，在出现的对话框的“Files”标签（选项卡）中，选择“C++ Source File”项，在右中处的 File 文本框中为将要生成的文件取一个名字，我们取名为“Hello”（其他遵照系统隐含设置，此时系统将使用“Hello.c”的文件来保存所键入的源程序），如图 1-4 所示。

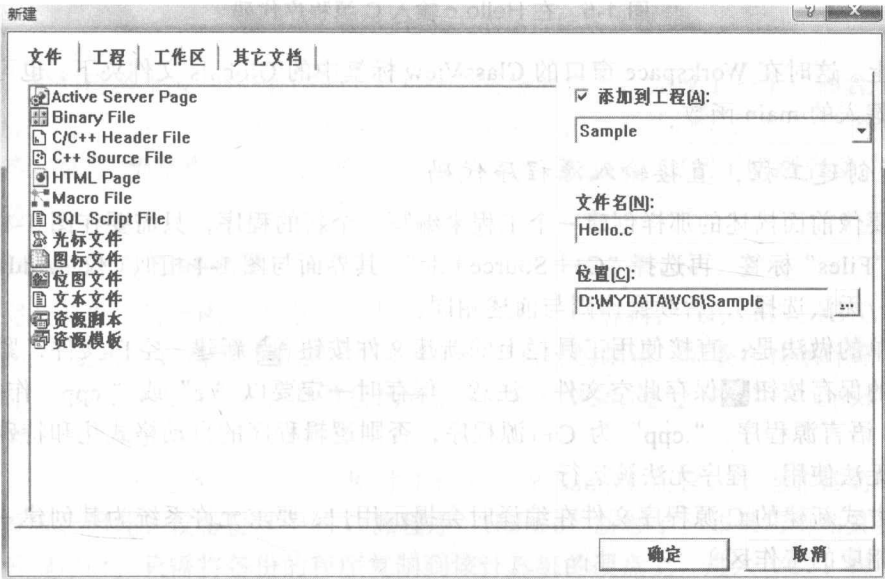


图 1-4 选择在工程 Sample 中新建一名为 Hello.c 的 C 源程序文件

而后选择“确定”按钮，进入输入源程序的编辑窗口（注意，窗口出现呈现“闪烁”状态的竖线为输入位置光标），此时，通过键盘输入你所需要的源程序代码：

```
#include <stdio.h>
void main()
{
    printf("Hello World!\n");
}
```

可通过 Workspace 窗口中的 FileView 标签, 看到 Source Files 文件夹下文件 Hello.c 已经被加了进去, 此时的界面情况如图 1-5 所示。

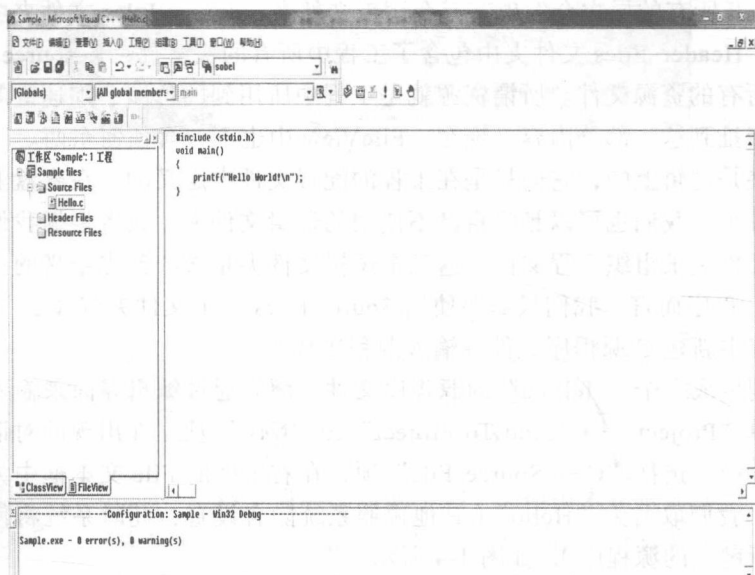




图 1-5 在 Hello.c 输入 C 源程序代码

实际上, 这时在 Workspace 窗口的 ClassView 标签中的 Globals 文件夹下, 也可以看到我们刚才所键入的 main 函数。

#### 4. 不创建工程, 直接输入源程序代码

不需要像前面描述的那样创建一个工程来编写一个新的程序, 只需要在图 1-1 所示的界面中, 选“Files”标签, 再选择“C++ Source File”, 其界面与图 1-4 相似(仅“Add to project”是暗淡的, 无法选择), 后续操作则与前述相同。

最简单的做法是: 直接使用工具栏上的新建文件按钮  新建一空白文件, 紧接着单击工具栏上的保存按钮  保存此空文件。注意, 保存时一定要以“.c”或“.cpp”作为扩展名, “.c”为 C 语言源程序, “.cpp”为 C++源程序, 否则逻辑程序的自动格式化和特殊显示等很多特性将无法使用, 程序无法被运行。

这种方式新建的 C 源程序文件在编译时会提示用户, 要求允许系统为其创建一个默认的工程(含相应的工作区)。

#### 5. 编译、链接后运行程序

程序编制完成(即所谓“四步曲”中第一步的编辑工作得以完成)之后, 就可以进行后三步的编译、链接与运行了。所有后三步的命令项都处在菜单命令“Build”之中。注意, 在对程序进行编译、链接和运行前, 最好先保存自己的工程(使用“File”→“Save All”菜单项)以避免程序运行时系统发生意外而使自己之前的工作付之东流, 并且, 应让这种做法成为自己的习惯。

若想单独编译, 首先选择执行菜单第一项“Compile”, 此时将对程序进行编译。若编译中发现错误(error)或警告(warning), 将在 Output 窗口中显示出它们所在的行以及具体的



出错或警告信息，可以通过这些信息的提示来纠正程序中的错误或警告（注意，错误是必须纠正的，否则无法进行下一步的连接；而警告则不然，它并不影响下一步工作，当然最好还是能把所有的警告也“消灭”掉）。当没有错误与警告出现时，Output 窗口所显示的最后一行应该是“Hello.obj - 0 error (s), 0 warning (s)”。

编译通过后，可以选择菜单的第二项 Build 来进行链接生成可执行程序。在链接中出现的错误也将显示到 Output 窗口中。链接成功后，Output 窗口所显示的最后一行应该是“Sample.exe - 0 error (s), 0 warning (s)”。最后就可以运行（执行）我们所编制的程序了，选择“Execute”项（该选项前有一个深色的感叹号标志“!”，实际上也可通过单击窗口上部工具栏中的深色感叹号标志“!”来启动执行该选项），VC6 将运行已经编好的程序，执行后将出现一个结果界面（所谓的类似于 DOS 窗口的界面），如图 1-6 所示。其中的“Press any key to continue”是由系统产生的。该窗口使得用户可以浏览输出结果，直到按下了任一个键盘按键时为止（那时又将返回到集成界面的编辑窗口处）。

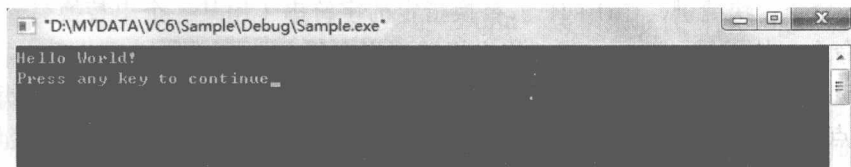


图 1-6 程序 Hello.c 的运行结果界面

至此，我们已经生成并运行（执行）了一个完整的程序，完成了一个“回合”的编程任务。此时应执行“File”→“Close Workspace”菜单项，待系统询问是否关闭所有的相关窗口时，回答“是”，则结束了一个程序从输入到执行的全过程，回到了刚刚启动 VC6 的那一个初始画面。

## 6. 及时备份自己的创作

(1) 完全备份。对于刚才工作的工程 Sample 而言，只需将“D:\MYDATA\VC6”下的文件夹 Sample 复制到 U 盘或打包成一个文件后放到自己的邮箱。需要在其他计算机上继续完成该工程时，将该文件夹复制到该计算机的硬盘上，进入 VC6，通过“File”→“Open Workspace”菜单项将该工程打开即可。

(2) 只备份 C 源程序文件。对于刚才工作的工程 Sample 而言，工程非常简单，没有什么专门的设置，因此，仅备份其中的 C 源程序“Hello.c”就够了。如果需要在其他计算机上继续完成该程序时，只需将备份的程序复制到该计算机的硬盘上，进入 VC6，根据前面的讲述，新建“Win32 Console Application”（如图 1-5 所示的界面）工程，然后通过“Project”→“Add to Project”→“Files”菜单项将“Hello.c”添加到新建的工程中。

还有一种最简单的做法是：直接使用工具栏上的文件打开按钮  打开“Hello.c”文件。

## 1.2.2 VC6 集成开发环境使用参考

### 1. VC6 的常用菜单命令项

#### (1) File 菜单。

New 命令：打开“New”对话框，以便创建新的文件、工程或工作区。

**Close Workspace 命令：**关闭与工作区相关的所有窗口。

**Exit 命令：**退出 VC6 环境，将提示保存窗口内容等。

### (2) Edit 菜单。

**Cut 命令：**快捷键“Ctrl + X”。将选定内容复制到剪贴板，然后再从当前活动窗口中删除所选内容。与“Paste”命令联合使用可以移动选定的内容。

**Copy 命令：**快捷键“Ctrl + C”。将选定内容复制到剪贴板，但不从当前活动窗口中删除所选内容。与“Paste”命令联合使用可以复制选定的内容。

**Paste 命令：**快捷键“Ctrl + V”。将剪贴板中的内容插入（粘贴）到当前鼠标指针所在的位置。注意，必须先使用“Cut”或“Copy”命令使剪贴板中具有准备粘贴的内容。

**Find 命令：**快捷键“Ctrl + F”。在当前文件中查找指定的字符串。顺便指出，可按快捷键“F3”寻找下一个匹配的字符串。

**Find in Files 命令：**在指定的多个文件中查找指定的字符串。

**Replace 命令：**快捷键“Ctrl + H”。替换指定的字符串（用某一个串替换另一个串）。

**Go To 命令：**快捷键“Ctrl + G”。将光标移到指定行上。

**Breakpoints 命令：**快捷键“Alt + F9”。弹出对话框，用于设置、删除或查看程序中的所有断点。断点将告诉调试器应该在何时何地暂停程序的执行，以便查看当时的变量取值等现场情况。

### (3) View 菜单。

**Workspace 命令：**如果工作区窗口没显示出来，选择执行该项后将显示出工作区窗口。

**Output 命令：**如果输出窗口没显示出来，选择执行该项后将显示出输出窗口。输出窗口中将随时显示有关的提示信息或出错警告信息等。

### (4) Project 菜单。

**Add To Project 命令：**选择该项将弹出子菜单，用于添加文件或数据链接等到工程中去。例如，子菜单中的“New”选项可用于添加“C++ Source File”或“C/C++ Header File”文件，而子菜单中的“Files”选项则用于插入已有的文件到工程中。

**Settings 命令：**为工程进行各种不同的设置。当选择其中的“Debug”标签（选项卡），并通过在“Program arguments:”文本框中填入以空格分割的各命令行参数后，则可以为带参数的 main 函数提供相应参数（对应于 void main (int argc, char\* argv[ ]) {…}形式的 main 函数中所需各 argv 数组的各字符串参数值）。注意，在执行带参数的 main 函数之前，必须进行该设置，当“Program arguments:”文本框中为空时，意味着无命令行参数。

### (5) Build 菜单。

**Compile 命令：**快捷键“Ctrl + F7”。编译当前处于源代码窗口中的源程序文件，以便检查是否有语法错误或警告，如果有的话，将显示在“Output”输出窗口中。

**Build 命令：**快捷键“F7”。对当前工程中的有关文件进行链接，若出现错误的话，也将显示在 Output 输出窗口中。

**Execute 命令：**快捷键“Ctrl + F5”。运行（执行）已经编译、链接成功的可执行程序（文件）。

**Start Debug 命令：**选择该项将弹出一个子菜单，含有用于启动调试器运行的几个选项。例如其中的“Go”选项用于从当前语句开始执行程序，直到遇到断点或遇到程序结束；“Step



Into”选项开始单步执行程序，并在遇到函数调用时进入函数内部再从头单步执行；“Run to Cursor”选项使程序运行到当前鼠标光标所在行时暂停其执行（注意，使用该选项前，要先将鼠标光标设置到某一个你希望暂停的程序行处）。执行该菜单的选择项后，就启动了调试器，此时菜单栏中将出现“Debug”菜单（取代了“Build”菜单）。

#### （6）Debug 菜单。

启动调试器后才出现该“Debug”菜单（而不再出现“Build”菜单）。

Go 命令：快捷键“F5”。从当前语句启动继续运行程序，直到遇到断点或遇到程序结束而停止（与“Build”→“Start Debug”→“Go”选项的功能相同）。

Restart 命令：快捷键“Ctrl + Shift + F5”。重新从头开始对程序进行调试执行（当对程序做过某些修改后往往需要这样做）。选择该项后，系统将重新装载程序到内存，并放弃所有变量的当前值（而重新开始）。

Stop Debugging 命令：快捷键“Shift + F5”。中断当前的调试过程并返回正常的编辑状态（注意，系统将自动关闭调试器，并重新使用“Build”菜单来取代“Debug”菜单）。

Step Into 命令：快捷键“F11”。单步执行程序，并在遇到函数调用语句时，进入该函数内部，并从头单步执行（与“Build”→“Start Debug”→“Step Into”选项的功能相同）。

Step Over 命令：快捷键“F10”。单步执行程序，但当执行到函数调用语句时，不进入该函数内部，而是一步直接执行完该函数后，接着再执行函数调用语句后面的语句。

Step Out 命令：快捷键“Shift + F11”。与“Step Into”配合使用，当执行进入到函数内部，并单步执行若干步之后，若发现不再需要进行单步调试的话，通过该选项可以从函数内部返回（到函数调用语句的下一语句处停止）。

Run to Cursor 命令：快捷键“Ctrl + F10”。使程序运行到当前鼠标光标所在行时暂停其执行（注意，使用该选项前，要先将鼠标光标设置到某一个你希望暂停的程序行处）。事实上，这相当于设置了一个临时断点，与“Build”→“Start Debug”→“Run to Cursor”选项的功能相同。

Insert/Remove Breakpoint 命令：快捷键“F9”。本菜单项并未出现在“Debug”菜单上（在工具栏和程序文档的上下文关联菜单上），放在此处是为了方便大家掌握程序调试的手段，其功能是设置或取消固定断点——程序行前有一个圆形的黑点标志，表示已经为该行设置了固定断点。另外，与固定断点相关的还有“Alt + F9”（管理程序中的所有断点），“Ctrl + F9”（禁用/使能当前断点）。

#### （7）Help 菜单。

通过该菜单来查看 VC6 的各种联机帮助信息。

#### （8）上下文关联菜单。

除了主菜单和工具栏外，VC6 开发环境还提供了大量的上下文关联菜单，用鼠标右键单击窗口中很多地方都会弹出一个关联菜单，里面包含有与被单击项目相关的各种命令，建议大家在工作时可以试着多点点鼠标右键，说不定会发现很多有用的命令，从而大大加快一些常规操作的速度。

## 2. VC6 的主要工作窗口

### （1）Workspace 窗口。

Workspace 窗口显示了当前工作区中各个工程的类、资源和文件信息，当新建或打开一

个工作区后, Workspace 窗口通常就会出现三个树视图: ClassView (类视图), ResourceView (资源视图) 和 FileView (文件视图), 如果在 VC6 企业版中打开了数据库工程, 还会出现第四个视图 DataView (数据视图)。如同前面所述, 在 Workspace 窗口的各个视图内单击鼠标右键可以得到很多有用的关联菜单。

ClassView 视图显示当前工作区中所有工程定义的 C++ 类、全局函数和全局变量, 展开每一个类后, 可以看到该类的所有成员函数和成员变量, 如果双击类的名字, VC6 会自动打开定义这个类的文件, 并把文档窗口定位到该类的定义处, 如果双击类的成员或者全局函数及变量, 文档窗口则会定位到相应函数或变量的定义处。

ResourceView 视图显示每个工程中定义的各种资源, 包括快捷键、位图、对话框、图标、菜单、字符串资源、工具栏和版本信息, 如果双击一个资源项目, VC6 就会进入资源编辑状态, 打开相应的资源, 并根据资源的类型自动显示出“Graphics”“Color”“Dialog”“Controls”等停靠式窗口。

FileView 视图显示了隶属于每个工程的所有文件。除了 C/C++ 源文件、头文件和资源文件外, 我们还可以向工程中添加其他类型的文件, 例如 Readme.txt 等, 这些文件对工程的编译连接不是必需的, 但将来制作安装程序时会被一起打包。同样, 在 FileView 中双击源程序等文本文件时, VC6 会自动为该文件打开一个文档窗口, 双击资源文件时, VC6 也会自动打开其中包含的资源。

在 FileView 中对着一个工程单击鼠标右键后, 关联菜单中有一个“Clean”命令, 在此特地要解释一下它的功能。VC6 在建立 (Build) 一个工程时, 会自动生成很多中间文件, 例如预编译头文件、程序数据库文件等, 这些中间文件加起来的大小往往有数兆, 很多人在开发一个软件期间会使用办公室或家里的数台机器, 如果不把这些中间文件删除, 在多台机器之间使用软盘拷贝工程就很麻烦。“Clean”命令的功能就是把 VC6 生成的中间文件全部删除, 避免了手工删除时可能会出现误删或漏删的问题。另外, 在某些情况下, VC6 编译器可能无法正确识别哪些文件已被编译过了, 以至于在每次建立工程时都进行完全重建, 很浪费时间, 此时使用“Clean”命令删除掉中间文件就可以解决这一问题。

应当指出, 承载一个工程的本质上还是存储在工作文件夹下的多个文件 (物理上), 在 Workspace 窗口中的这些视图都是逻辑意义上的, 它们只是从不同的角度去自动统计总结了工程的信息, 以方便和帮助我们查看工程, 更有效地开展工作。如果开始时不习惯且工程很简单 (学习期间很多时候都只有一个.c 文件), 则你完全没有必要去搭理这些视图, 只需要在该文件内容窗口中工作。

## (2) Output 窗口。

与 Workspace 窗口一样, Output 窗口也被分成了数栏, 其中前面 4 栏最常用。在建立工程时, Build 栏将显示工程在建立过程中经过的每一个步骤及相应信息, 如果出现编译连接错误, 那么发生错误的文件名、行号、错误类型编号和描述都会显示在 Build 栏中, 用鼠标双击一条编译错误, VC6 就会打开相应的文件, 并自动定位到发生错误的那一条语句。

工程通过编译连接后, 运行其调试版本, Debug 栏中会显示出各种调试信息, 包括 DLL 装载情况、运行时警告及错误信息、MFC 类库或程序输出的调试信息、进程中止代码等。

两个 Find in Files 栏用于显示从多个文件中查找字符串后的结果, 当你想看看某个函数或变量出现在哪些文件中时, 可以从“Edit”菜单中选择“Find in Files...”命令, 然后指定