



海峡两岸高校应用型系列教材

# C 程序设计

## 实践教程

主 编 张鸣华  
副主编 寿宇文 曾台盛  
主 审 孔令德

The Practice Course

高等教育出版社

Course of C Programming

The Practice Course of C Programming

The Practice Course of C Programming



海峡两岸高校应用型系列教材

# C程序设计实践教程

C Chengxu Sheji Shijian Jiaocheng

主 编 张鸣华  
副主编 寿宇文 曾台盛  
主 审 孔令德

高等教育出版社·北京

## 内容提要

本书以 Microsoft Visual C++ 6.0、Dev-C++ 为开发环境,讲授编写程序的思路和方法。全书共分为 9 章,主要包括 C 语言入门,C 程序设计基础,分支语句,循环语句,函数,数组,指针,数据的组织处理及文件。全书以案例为背景,通过问题驱动的方式,由浅入深地引导读者进行理性思考和有效实践。本书引入 ACM 程序设计竞赛培训的学习方法,引导读者自主学习,培养其应用程序设计及解决问题的能力。特别注重为后继的课程和编程实践打好基础。

本书的配套资源丰富,为读者提供了“程序设计在线测试平台”及无需网络支撑的“程序设计单机测试系统”。为教师提供了 PPT 课件和案例源代码。还提供了考试和练习使用的管理系统。

本书可作为高等院校理工科相关的程序设计类课程教材或教学参考书,也可供对程序设计感兴趣的读者学习参考。

## 图书在版编目(CIP)数据

C 程序设计实践教程 / 张鸣华主编. -- 北京: 高等教育出版社, 2014. 8

ISBN 978-7-04-040604-7

I. ①C… II. ①张… III. ①C 语言-程序设计-高等学校-教材 IV. ①TP312

中国版本图书馆CIP数据核字(2014)第159371号

策划编辑 唐德凯	责任编辑 陈哲	特约编辑 谷玉春	封面设计 李小璐
版式设计 杜微言	插图绘制 杜晓丹	责任校对 刘丽娟	责任印制 赵义民

出版发行 高等教育出版社  
社 址 北京市西城区德外大街 4 号  
邮政编码 100120  
印 刷 北京印刷一厂  
开 本 787 mm×1092 mm 1/16  
印 张 17.75  
字 数 400 千字  
购书热线 010-58581118

咨询电话 400-810-0598  
网 址 <http://www.hep.edu.cn>  
<http://www.hep.com.cn>  
网上订购 <http://www.landaco.com>  
<http://www.landaco.com.cn>  
版 次 2014 年 8 月第 1 版  
印 次 2014 年 8 月第 1 次印刷  
定 价 36.90 元 (含光盘)

本书如有缺页、倒页、脱页等质量问题,请到所购图书销售部门联系调换  
版权所有 侵权必究  
物料号 40604-00

C 语言程序设计是计算机及相关专业重要的基础课程之一。程序设计类课程教学长期以来有着多种不同的教学模式和理念,产生了不同的效果。本书作者长期承担程序设计类课程的教学任务;带领学生参加 ACM/ICPC 竞赛;组织学生进行多种创新与实践活动,积累了丰富的经验,取得了一定的成绩。师生共同创建了程序设计在线测试平台(简称 OJ),开发了考试中用到的系列软件,奠定了在这平台上的教学理念和教学方式:实践与思考。

本书根据学生在实践中的学习方式和学习顺序来安排内容。此外,本书还参考了目前国内外的程序设计教材和程序设计竞赛训练材料。本书以案例为背景,以问题驱动为教学模式,引导读者自主学习。

本书的内容安排,循序渐进,具体内容如下。

第 1 章介绍 C 语言的特点、开发平台、程序设计流程、OJ 平台的使用,引导读者以实践的方式入门。

第 2 章是程序设计基础,介绍数据表达方式、数据的输入与输出、赋值语句,引导读者用顺序结构实现常规的数学计算编程。

第 3 章介绍逻辑思维及分支程序设计,包括关系运算、关系表达式以及逻辑运算符和逻辑表达式,还介绍了 ASCII 码,拓展程序设计的范围。

第 4 章介绍循环问题的程序设计。使用大量流程图表达循环问题的算法,使读者容易掌握思考和解决这类问题的方法。

第 5 章介绍自定义函数及使用方法,引导读者掌握程序模块化的设计方法,还由浅入深地引导读者学习递归函数,培养抽象思维能力。函数的思想和方法在后继的章节中将不断地应用。

第 6 章介绍数组,强调了循环处理数组中数据元素的方法,培养读者处理批量数据的能力。

第 7 章介绍指针,用浅显的例子帮助读者理解指针即是地址,用实际应用展开指针的使用。

第 8 章介绍结构体与共用体,培养数据的组织与处理能力。通过链表的学习,让读者能够综合使用各种程序设计的方法,为学习数据结构奠定基础。

第 9 章介绍文件的使用,为需要深入学习的读者打下基础。

本书每章中的进阶部分是本章内容的深入或来自 ACM 程序设计竞赛类问题及后继课程中的算法设计问题,这有利于加强读者的思维训练、引导读者自主学习、培养读者解决问题的能力。而且,本书中进阶部分的内容不影响进度,可以滞后学习。

本书各章都提供了大量习题,这些习题知识面广、趣味性强、思维层次深。本书光盘中的“程序设计单机测试系统(Offline Judge)”是一个可独立运行的绿色软件,与 OJ 有相似的功能,可以

对用 C、C++ 编写的代码进行测评,测评结果与 OJ 上的一致。光盘中还含有 200 多道例题与习题供读者练习,无需网络支撑。

本书提供多个附录供读者参考。主要有关键字、常见错误、程序调试、库函数、常用字符与 ASCII 码对照表、常用 C 语言库文件、预处理。

本书由张鸣华主编,其中张鸣华编写了第 1、2、3、5、6 章,曾台盛编写了第 4、8、9 章,寿宇文编写了第 7 章。全书由张鸣华统稿,孔令德教授审阅。本书的代码由王文辉负责校对,本书关联的在线测评系统由王文辉开发、许建众开发了考试时的 IP 绑定功能模块、吴振深完善后台测试数据管理功能与多线程考试功能模块、陈松金开发了考试成绩管理功能模块、吴俊开发了考试时监控 U 盘和网络功能及考试点名功能模块、余宇航完善了 OJ 考试与练习两平台切换的功能模块等,配书的程序设计单机测评系统由曾坚坚、王吉生开发。

本书的各个部分都由多名不同层次的学生试读,编者根据他们提出的问题不断修改。本书的电子版已经试用过一个学期,由张标汉、惠苗、董园、孙丽丽、魏晶晶、李渭提出意见并加以修改。本书的习题由以上教师负责编写。

需要特别指出的是,我们的教学改革、ACM 系列活动及教材编写的过程都得到福州大学吴英杰老师的指导。广西师范学院的邓育林老师也为本书提出了宝贵意见。

在此,对以上的各位参与者、支持者表示衷心的感谢。

书中大部分的例题和习题后面有用括号指示的 4 位数字,例如,(smu1112),指的是该题在三明学院程序设计在线测试平台上基础区的编号。又如(smp1112),指的是问题区的编号。三明学院在线测试平台的网址是:<http://218.5.241.13:8060/oj>。

本书提供了相关的教学资料,有需要的老师请联系 [sm\\_zmh@163.com](mailto:sm_zmh@163.com)。由于作者水平有限,本书难免存在疏漏和不足,敬请广大读者批评指正。

编 者

2014 年 4 月

<b>第 1 章 C 语言入门</b> .....	1	<b>第 3 章 逻辑思维及分支程序设计——</b>	
1.1 学习 C 语言 .....	1	<b>分支语句</b> .....	38
1.2 使用 Visual C++ 6.0 编写		3.1 关系运算符和关系表达式 .....	38
程序 .....	2	3.2 逻辑运算符和逻辑表达式 .....	39
1.3 使用 Dev-C++ 开发程序 .....	5	3.3 分支结构 .....	40
1.4 C 语言的特点 .....	7	3.3.1 if 语句 .....	40
1.4.1 C 语言的结构 .....	7	3.3.2 switch 语句 .....	44
1.4.2 程序风格 .....	8	3.4 条件表达式 .....	46
1.5 程序流程图 .....	9	3.5 ASCII 码 .....	47
1.6 设计程序 .....	10	3.6 案例及分析 .....	49
1.7 在线测试(OJ)平台上的数据		3.7 进阶 .....	51
处理 .....	11	习题 .....	55
习题 .....	16	<b>第 4 章 循环问题设计——循环语句</b> .....	60
<b>第 2 章 程序设计基础</b> .....	17	4.1 循环结构 .....	60
2.1 数据类型 .....	17	4.1.1 for 语句 .....	60
2.2 变量与常量 .....	19	4.1.2 while 语句 .....	63
2.3 const 修饰符 .....	20	4.1.3 do-while 语句 .....	64
2.4 运算符和表达式 .....	21	4.1.4 3 种循环语句的比较 .....	65
2.5 赋值语句 .....	23	4.1.5 break 语句 .....	67
2.6 库函数 .....	25	4.1.6 continue 语句 .....	68
2.7 输入与输出函数 .....	25	4.2 循环的嵌套 .....	69
2.7.1 printf( ) 函数 .....	25	4.3 案例及分析 .....	70
2.7.2 scanf( ) 函数 .....	28	4.4 进阶 .....	74
2.7.3 字符数据的输入输出 .....	31	习题 .....	80
2.8 案例及分析 .....	32	<b>第 5 章 过程封装与递归思想——</b>	
2.9 进阶 .....	34	<b>函数</b> .....	84
习题 .....	36	5.1 自定义函数 .....	84

5.1.1 自定义函数的结构 .....	84	7.2 指针与数组间的关系 .....	156
5.1.2 自定义函数的返回值 .....	85	7.3 指针变量作为函数参数 .....	158
5.1.3 自定义函数的调用与传递 参数 .....	85	7.4 函数值为指针类型 .....	161
5.1.4 函数的定义与声明的区别和 使用 .....	87	7.5 指针数组 .....	162
5.2 全局变量与局部变量 .....	88	7.6 双重指针 .....	163
5.3 存储类型 .....	90	7.7 案例及分析 .....	168
5.4 函数的嵌套调用 .....	92	7.8 进阶 .....	171
5.5 函数的递归调用 .....	93	习题 .....	173
5.6 案例及分析 .....	94	<b>第 8 章 数据的组织与处理——结构体、 共用体、枚举与链表</b> .....	179
5.7 进阶 .....	98	8.1 用 typedef 声明的类型名 .....	179
习题 .....	103	8.2 结构体与共用体 .....	180
<b>第 6 章 批量数据处理——数组</b> .....	108	8.2.1 结构的定义 .....	180
6.1 一维数组 .....	108	8.2.2 结构类型变量的使用 .....	183
6.1.1 一维数组的定义 .....	108	8.2.3 结构数组 .....	186
6.1.2 一维数组的引用 .....	109	8.2.4 共用体 .....	189
6.2 二维数组 .....	113	8.3 指针与结构体 .....	191
6.2.1 二维数组的定义 .....	113	8.4 返回值是结构体指针的函数 .....	192
6.2.2 二维数组的引用 .....	114	8.5 链表 .....	194
6.3 字符数组 .....	116	8.5.1 单链表的结构 .....	194
6.3.1 字符数组的定义 .....	116	8.5.2 单链表的建立与输出 .....	195
6.3.2 字符数组的使用 .....	117	8.5.3 单链表结点的插入 .....	197
6.3.3 字符串函数 .....	119	8.5.4 单链表结点的删除 .....	200
6.4 案例及分析 .....	120	8.6 枚举数据类型 .....	203
6.5 进阶 .....	126	8.7 案例及分析 .....	205
习题 .....	134	8.8 进阶 .....	213
<b>第 7 章 间接访问——指针</b> .....	150	习题 .....	219
7.1 指针 .....	150	<b>第 9 章 流的输入 / 输出——文件</b> .....	223
7.1.1 变量的地址 .....	150	9.1 认识文件 .....	223
7.1.2 指针变量的定义 .....	151	9.2 文件的打开与关闭 .....	225
7.1.3 指针的动态内存分配 .....	153	9.3 读写文件的有关函数 .....	228
7.1.4 指针的应用 .....	154	9.4 二进制文件 .....	232
		9.5 随机读写文件 .....	234

---

9.6 案例及分析 .....	235	附录 C 程序调试 .....	248
9.7 进阶 .....	237	附录 D 库函数 .....	256
习题 .....	242	附录 E 常用字符与 ASCII 码对照表 .....	264
<b>附录</b> .....	<b>245</b>	附录 F 常用 C 语言库文件 .....	266
附录 A 关键字 .....	245	附录 G 预处理 .....	268
附录 B 常见错误 .....	246	<b>参考文献</b> .....	<b>272</b>



## 导学

请读者先在计算机上安装 Microsoft Visual C++ 6.0 或 Dev-C++, 输入本章的程序并运行。理解 C 语言的特点和作用;认识程序的基本结构;理解简单的程序;用 freopen 将输入、输出的数据放在文件中处理。试着编写简单的程序,再通过 OJ 平台或本书提供的单机测评系统,在上面提交 a+b 的代码,熟悉学习环境。

## 1.1 学习 C 语言

计算机做任何一件事情都离不开人们为它编写的程序。用于程序设计的语言有很多,其中 C 语言有着悠久的历史 and 众多应用,同时也是一门很好的程序设计入门语言。事实上,不同的高级语言有着很多相似之处,只要对入门的程序设计语言打好基础,就可以很容易掌握其他语言的应用。

本书学习 C 语言的同时,不仅要学习编写 C 语言的语法规则,重要的是要学会程序设计的方法和思路。设计一个程序通常包括下列一些步骤。

### 1. 定义程序目标

首先,读者对程序执行什么功能要有一个清晰的认识。考虑程序需要什么信息、进行什么操作和计算,以及程序将要报告的信息。这是程序设计规划的阶段。

### 2. 设计程序

设计程序要考虑程序的输入、输出是什么;用什么样的数据结构组织数据;用什么样的方法设计程序;用什么语句来实现。也就是说,程序设计 = 算法 + 数据结构 + 代码。读者可以用流程图表示程序设计过程,使思路更加清晰。

### 3. 编写代码

我们决定用 C 语言编写代码,要确定一个编程环境,选择一个程序运行平台。

C 语言的开发平台很多,常用的 C 语言集成开发环境 (IDE) 有 Microsoft Visual C++, Borland C++, Watcom C++, Borland C++, Borland C++ Builder, Borland C++ 3.1 for DOS, Watcom C++ 11.0 for DOS, GNU DJGPP C++, Lccwin32 C Compiler 3.1, Microsoft C, High C, Turbo C, Dev-C++, C-Free, win-tc 等。由于 C 语言是 C++ 语言的子集,所以本书以 Microsoft Visual C++ (以下简称 VC) 平台为主进行讲解。

用 C 语言设计出来的源代码是形如 \*.C 形式的文件,可以在 VC 平台上编辑,也可以直接用

记事本等文档编辑器生成。比如,让计算机输出一句“`How are you.`”,可以这样写:

#### 【程序清单 1.1】

```
1. //How are you.C
2. #include <stdio.h>
3. void main( )
4. {
5.     printf("How are you.\n");
6. }
```

注意:每行前面的行号是便于阅读而加上的,不属于代码中的内容。

#### 4. 编译程序

\*.C 格式的文件要编译成 \*.obj 的文件,再链接成可执行的文件 \*.exe 才能在机器上运行。这个工作由编译器完成。编译器是一个程序,其工作是将源代码转换为可执行的代码。可执行的代码是用计算机的机器语言表示的代码。机器语言由二进制代码表示的详细指令组成。C 编译器将 C 语言转换成特定的机器语言,还从 C 的库中向最终程序加入代码。

编译器还检查用户的程序是否为有效的 C 语言程序。如果编译器发现错误,就将错误报告给用户;如果没有发现错误就生成可执行文件。

#### 5. 运行程序

运行程序在 VC 中通过选择菜单或工具栏中的“运行”命令完成。运行时通过输入程序要求的数据,就可以得到运行后的结果。

#### 6. 调试程序

程序可以运行还不能保证结果是正确的。可能会有一些编写的错误或是设计的错误,这种错误在计算机中被称为错误(Bug)。调试就是要发现并修正程序错误,即 Debug 的过程。

#### 7. 维护程序

程序在运行时,可能会发现问题,这时要对程序进行修改,或增加新的功能。这项工作就是维护。

## 1.2 使用 Visual C++ 6.0 编写程序

Visual C++ 6.0,简称 VC 或者 VC 6.0,是微软公司推出的一款 C++ 编译器(将“高级语言”翻译为“机器语言”的程序)。Visual C++ 是一个功能强大的可视化软件开发工具,有特别强大的 Debug 功能。自 1993 年 Microsoft 公司推出 Visual C++ 1.0 版本后,随着其新版本的不断问世,Visual C++ 已成为专业程序员进行软件开发的首选工具。实际应用中,VC 占用内存资源少,是学习 C 语言程序设计的良好平台。对于初学者,建议安装和使用 Visual C++ 6.0。本书中的主要代码都是在 VC 中调试的。

请先安装 Visual C++ 6.0。注意,如果 Windows 7 的操作系统不支持 Visual C++ 6.0,可以改

装 Visual C++ 2010 版本。

VC 最好在操作系统安装后就安装,特别要在 .net 安装之前先安装,否则使用时容易卡机。

**【例 1.1】** 输出“Hello. C.” (smu1001)。

具体操作步骤如下。

### 1. 建立工程

从 Windows 的“开始”|“程序”栏选择 Microsoft Visual Studio 6.0 菜单中的 Microsoft Visual C++ 6.0 菜单项,启动 Visual C++ 集成开发环境。在菜单项中选中“文件”|“新建”命令,就出现如图 1.1 所示的界面。在“工程”选项卡中选择“Win32 Console Application”选项,在“工程名称”文本框中输入自己定义的项目名称(这里使用 Hello),在“位置”文本框中输入保存项目的路径,单击“确定”按钮进入如图 1.2 所示的界面,选择“一个空工程”单选按钮,单击“完成”按钮,进

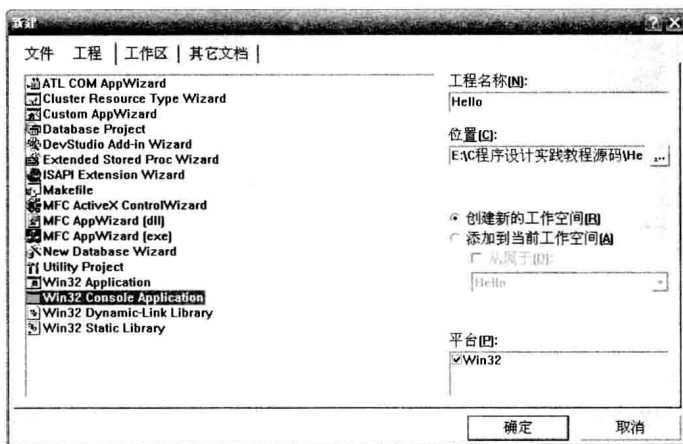


图 1.1 创建一个项目

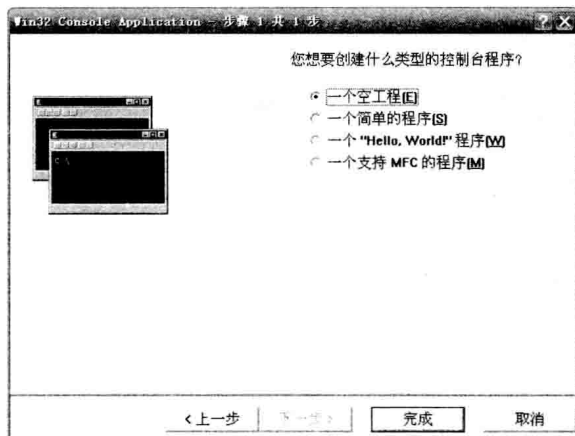


图 1.2 创建控制台程序

入下一个页面,再单击“确定”按钮,工程就建好了,此时在工程文件夹 Hello 中产生一个 Hello.dsw 的工程文件。

## 2. 添加文件

单击“文件”|“新建”命令,在弹出的“新建”对话框中选择“C++ Source File”,在“文件名”文本框中输入 1\_1.c(注意文件名一定要有扩展名 .c,否则就成为 C++ 的文件),再单击“确定”按钮,如图 1.3 所示。一个 C 的文件就建立好了,同时打开了这个文件的编辑窗口。

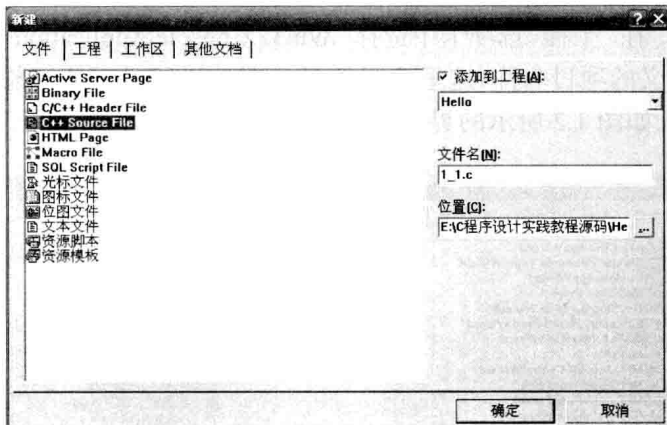


图 1.3 建立文件

图 1.4 是开发应用程序时一般的 Developer Studio(即 VC6.0 的 IDE)窗口示意图。Developer Studio 窗口由标题栏、菜单栏、工具栏、工作区窗口、编辑窗口、输出窗口、状态栏等部分组成。

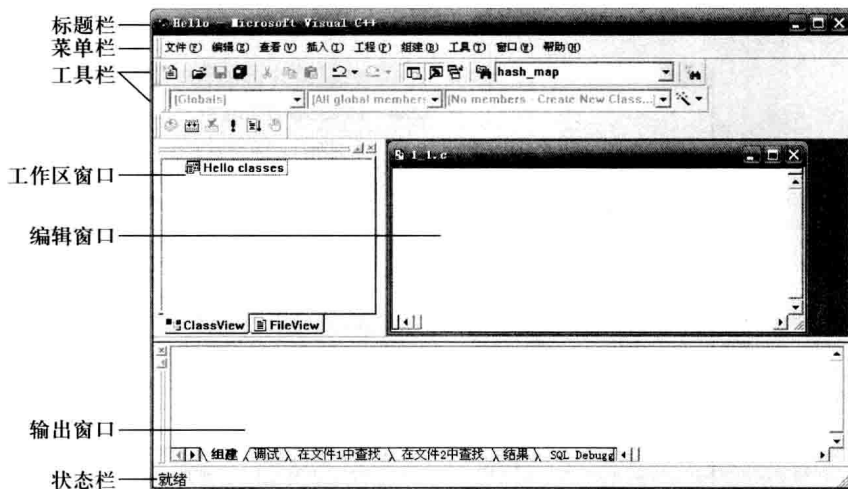


图 1.4 Visual C++ 6.0 集成开发环境

在工作区有一个类视图 (ClassView) 和一个文件视图 (FileView)。展开文件视图 (FileView) 后会出现 3 个文件的目录, 依次为:

- ① Source Files: 用来存放一般的程序文件。
- ② Header Files: 用来存放头文件 (含 .h)。
- ③ Resource Files: 用来存放资源文件。

展开 Source Files 可以看到刚才建的文件 1\_1.c, 如图 1.5 所示。

### 3. 输入程序

在编辑窗口输入【程序清单 1.2】中的内容, 如图 1.6 所示。



图 1.5 工作区的展开图

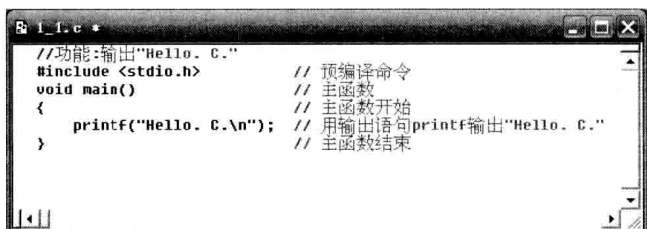


图 1.6 代码编辑框

#### 【程序清单 1.2】

- ```

1. // 功能: 输出 "Hello. C."
2. #include <stdio.h>           // 预编译命令
3. void main ( )              // 主函数
4. {                          // 主函数开始
5.     printf ("Hello. C.\n"); // 用输出语句 printf 输出 "Hello. C."
6. }
```

#### 4. 运行程序





在  工具栏中,  (Compile <Ctrl+F7> 组合键) 表示编译, 即编译 1\_1.c, 生成 1\_1.obj 文件, 如果成功会在输出框中的组建窗口显示: 1\_1.obj - 0 error(s), 0 warning(s);  (Build <F7>) 表示链接, 即将 1\_1.obj 链接成可执行文件 Hello.exe, 如果成功会在输出框中的组建窗口显示 Hello.exe - 0 error(s), 0 warning(s);  (BuildExecute) 表示链接和执行, 即执行 Hello.exe 文件, 并在弹出窗口显示结果, 如图 1.7 所示。

图 1.7 运行结果

## 1.3 使用 Dev-C++ 开发程序

Dev-C++ 是一个开发 C/C++ 程序的免费集成开发工具, 用户可以在网络上下载和安装。在 Dev-C++ 上开发程序有如下几个主要操作。

### 1. 启动 Dev-C++

从“开始”|“所有程序”菜单栏中找到 Dev-C++ 并运行,这样就启动了 Dev-C++ 程序,如图 1.8 所示。

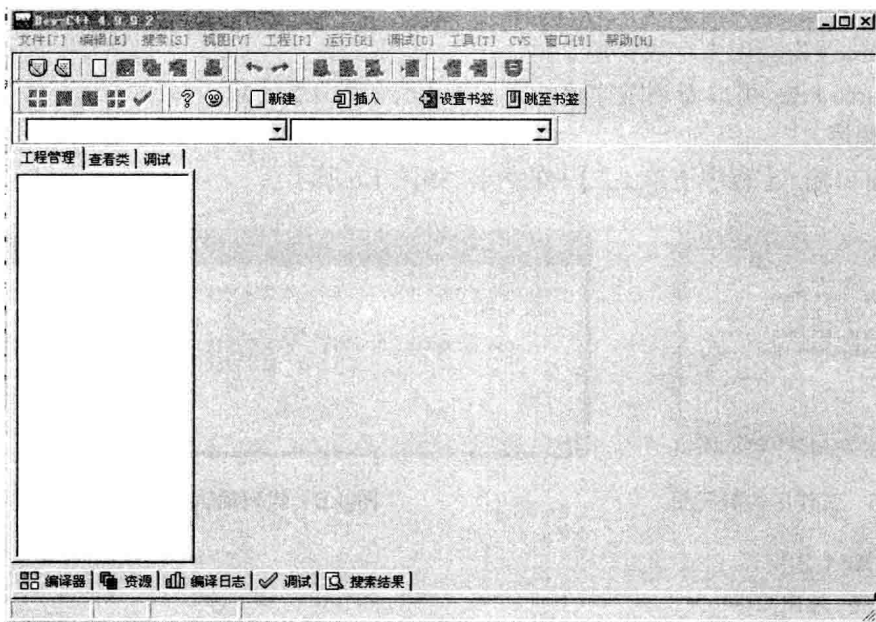


图 1.8 Dev-C++ 启动界面

### 2. 源程序的创建和编辑

在 Dev-C++ 中创建一个源程序的步骤如下。

- (1) 在菜单栏中选择“文件”|“新建”|“源代码”,也可以按〈Ctrl+N〉组合键进入。
- (2) 在新出现的空白窗口中输入自己的程序源代码。

#### 【程序清单 1.3】


```
1. #include <stdlib.h>           // 使用 system ("pause"); 的头文件
2. #include <stdio.h>           // 预编译命令
3. int main ( )                 // 主函数,在 Dev 中主函数的类型不能是 void,因此需要返回值
4. {                             // 主函数开始
5.     printf("Hello. C.\n");    // 用输出语句 printf 输出 "Hello. C."
6.     system("pause");         // 为了能看到输出结果而加的
7.     return 0;
8. }
```

(3) 保存源程序,可以按〈Ctrl+S〉组合键,也可从文件菜单中选择“保存”命令。存盘的文件类型选 C source files (\*.c)。

### 3. 编译源程序


编译程序的方法有多种,可以按〈Ctrl+F9〉组合键,也可以从菜单栏中选择“运行”|“编译”命令。还可以从工具按钮中选择  按钮。如果没有错误,会在“Compile Progress”对话框中显示 Errors 和 Warnings 均为 0 的状态,程序就可以运行了,如图 1.9 所示。

### 4. 运行程序

要运行已编译成功的程序,也有多种不同的方法。可以按〈Ctrl+F10〉组合键,也可以选择“运行”|“运行”命令,或单击工具栏中的按钮 。这时就可以看到控制台窗口显示结果。

### 5. 打开和修改已有程序

假如想修改以前编写好并已存盘的源程序,就需要用 Dev-C++ 重新打开它,在完成编辑修改后,再编译运行。

要打开已有源程序进行修改,同样也有多种方法。可以按〈Ctrl+O〉组合键,或者选择菜单“文件”|“打开”工程或文件命令,或者单击工具栏上的按钮 。这样就可以进行修改、重新编译和运行了。

通过前面的讲解注意到,用 Visual C++ 6.0 和 Dev-C++ 开发设计,有以下两点区别。

(1) Dev-C++ 的主函数 main() 只能是 int 类型,而 Visual C++ 6.0 的主函数可以是 int 类型,也可以是 void 类型。本书后续代码中的 main() 函数大都使用 int 类型。

(2) Dev-C++ 的运行窗口要用“system("pause");”才可以停留、显示运行结果,而 VC 会在运行窗口中提示:“按任意键继续”(英文是:“Press any key to continue”)。

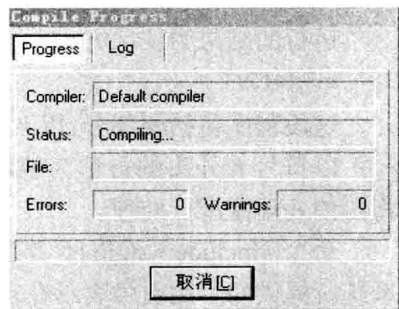


图 1.9 Compile Progress 对话框

## 1.4 C 语言的特点

### 1.4.1 C 语言的结构

【例 1.2】求圆的面积(与 smu1006 相似)。

【程序清单 1.4】

```

1.  #include <stdio.h>           // 预编译命令
2.  #include <cstdlib>          // 使用 system("pause"); 的头文件
3.  #define PI 3.1415926       // 宏定义
4.  int main( )                // 主函数,在 Dev 中主函数的类型不能是 void
5.  {                           // 主函数开始
6.      double r=1.0,s;
7.      s = PI*r*r;
8.      printf("s=%lf\n",s);    // 用输出语句 printf 输出圆的面积

```

```
9.     system("pause");           // 为了能看到输出结果而加的语句,在 Dev 环境下要加
10.     return 0;
11. }
```

运行的结果为:

```
s=3.141593
```

这段程序包括两部分:以 # 开头的部分和主函数 `int main() {return 0;}` 部分。其中,

以符号 # 开头的行称为编译预处理。C 编译器内置了预处理器,源代码中以“#”开始的行将由预处理器进行处理。

命令 `#include <stdio.h>` 是指添加头文件“`stdio.h`”到新建的程序中去。这个头文件是由 C 系统所提供的。用引号表示在自己定义的位置,用“`<stdio.h>`”(两端有尖括号)表示这个头文件应该从标准位置寻找,一般放在 `include` 的文件夹内。使用它的原因是因为它包含了 `printf()` 的信息。C 语言中提供了一些可以被直接拿来使用、能够完成某些特定功能的库函数,分别声明于不同的头文件中。

`#define PI 3.1415926` 是宏定义。规定在这个程序中圆周率 `PI` 代表 `3.141 592 6`。

`int main() { return 0;}` 是每一个 C 或 C++ 程序都要有的,称为主函数。程序运行时,从 `int main() {…}` 的第一个语句开始执行。用户程序的主要部分写在 `main` 函数里。

`printf()` 是输出语句,“`\n`”表示换行。

“`/**`”表示后面的内容是注释部分,同样可以使用“`/* 注释内容 */`”进行注释。注释内容不参与程序运行。为了使程序便于阅读和维护,增加注释是非常必要的。

“`return 0;`”意思是返回 0 给操作系统,表示主函数 `int main()` 内的程序运行结束,正常返回操作系统。

注意:# 部分的使用见附录 G 预处理。

## 1.4.2 程序风格

程序要有很好的可读性,易于修改和交流,因此要有优美、逻辑清晰的风格。一般要遵循下列规则。

(1) 花括号“`{}`”成对出现;“`{`”后不写代码;除了 `do{}while()`、结构体之类的语句外,“`}`”后不加代码;右花括号与所在的程序块左对齐。

(2) 左花括号后的代码要缩进一个 Tab 位。

(3) 运算符两端各加一个空格。

(4) 适当增加注解,程序开始加上该程序说明的注解,循环块的右括号后加上指明哪个循环的注解等。

为了美观和突出功能模块的意思,还可以考虑以下几点。

(1) 如果一行有多条语句,这些语句间最好用空格分开。

(2) 有时可以在较大功能块之间加一个空行。



在 VC 上编写代码时,可以选中要调整的部分,按下 $\langle \text{Alt}+\text{F8} \rangle$ 组合键,VC 会自动调整缩进的尺度。

## 1.5 程序流程图

为了表达程序设计的方法(即算法),常使用程序流程图。常用的流程图符号如图 1.10 所示。还有其他的流程图符号,读者可以从 Microsoft PowerPoint 的自选图形中的流程图中,或从 Microsoft Visio 中的流程图中或其他地方得到。

例如,如果  $x > 0$ ,则  $y = 1$ ,否则  $y = -1$ 。可以用图 1.11 表示。

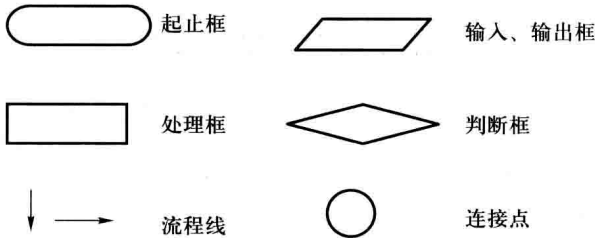


图 1.10 常用程序流程符号

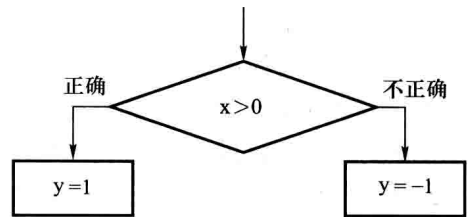


图 1.11 判断  $y$  值的流程图

【例 1.3】按照如图 1.12 所示流程图中的步骤操作,给出最终的执行结果。

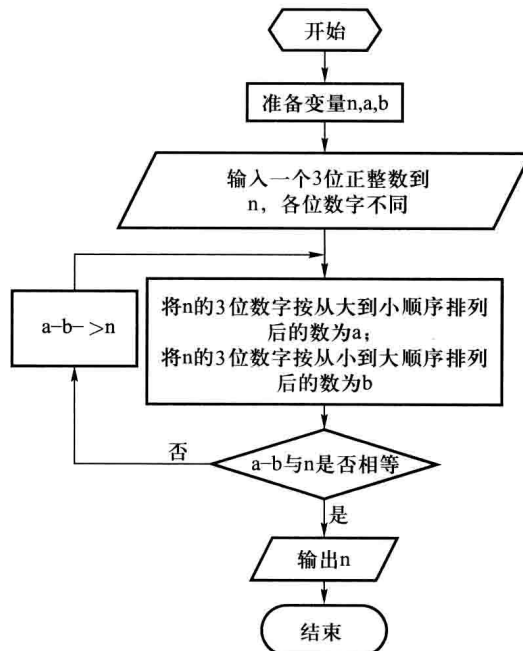


图 1.12 寻找黑洞数的流程图