



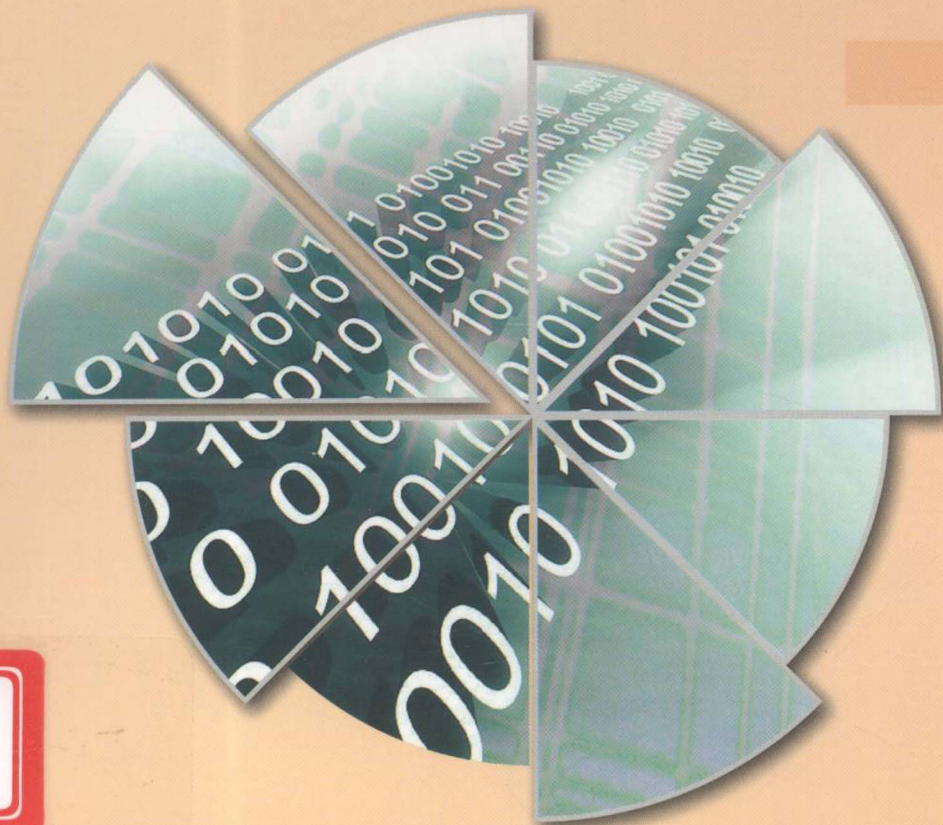
高等教育“十二五”规划教材



C YUYAN CHENGXU SHEJI SHIXUN JIAOCHENG

语言程序设计实训教程

杨文君 宋 丽 柴宝杰 主编
赵 杰 主审



科学出版社

高等教育“十二五”规划教材

C 语言程序设计实训教程

杨文君 宋 丽 柴宝杰 主编

赵 杰 主审

科学出版社

北 京

内 容 简 介

本书为《C 语言程序设计教程》的配套实训教材。基于 Visual C++ 6.0 集成开发环境, 本书共分为 9 章, 每章均包含学习目标、知识要点、实训案例、上机实训和同步练习。本书实训案例及上机实训题型丰富, 且来自实际生活, 有利于培养学生编程解决实际问题的意识和能力; 重点结合应用中的实训题目, 分析解题的思路并给出参考方法, 对实例中使用的知识点进行阐释, 以便学生尽快理解和掌握 C 语言基本操作, 提高学生 C 语言综合应用能力。

本书可作为高等院校本、专科学生 C 语言程序设计课程的教材和教学参考书, 亦可供各类 C 语言学习对象阅读参考。

图书在版编目(CIP)数据

C 语言程序设计实训教程/杨文君等主编. —北京: 科学出版社, 2010
(高等教育“十二五”规划教材)

ISBN 978-7-03-029802-7

I. ①C… II. ①杨… III. ①C 语言—程序设计—高等学校: 技术学校—教材 IV. ①TP312

中国版本图书馆 CIP 数据核字 (2010) 第 250368 号

策划: 姜天鹏 李洪旺

责任编辑: 王纯刚 隽青龙 / 责任校对: 刘玉靖

责任印制: 吕春珉 / 封面设计: 东方人华平面设计部

科 学 出 版 社 出版

北京东黄城根北街 16 号

邮政编码: 100717

<http://www.sciencep.com>

百 善 印 刷 厂 印刷

科学出版社发行

各地新华书店经销

*

2011 年 2 月第 一 版

开本: 787×1092 1/16

2011 年 2 月第一次印刷

印张: 9

印数: 1—3 000

字数: 213 000

定价: 18.00 元

(如有印装质量问题, 我社负责调换〈百善〉)

销售部电话 010-62140850 编辑部电话 010-62135517-2037

版权所有, 侵权必究

举报电话: 010-64030229; 010-64034315; 13501151303

前 言

C 语言是一种广泛应用的结构化程序设计的高级语言。由于该语言具有简练灵活的特点,深受广大编程者的喜爱。C 语言具有功能丰富、表达力强、使用灵活方便、便于大型程序开发、便于编写、可移植性好等优点,又由于 C 语言引入了反映计算机硬件特性的机制,使 C 语言也能编写直接控制计算机硬件设备的程序,所以 C 语言是一门既适合编写系统软件,又适合设计应用软件的高级语言。本书是配合《C 语言程序设计教程》编写的实训教材,本书内容基于 Visual C++ 平台,根据多年的教学经验编写。在编排程序设计的内容顺序方面,保持与《C 语言程序设计教程》的课程体系相吻合,做到循序渐进,系统学习,广泛实践,便于学生接受。各章编写结构为学习目标、知识要点、实训案例、上机实训及同步练习,实训案例及上机实训精选了部分典型例题及一些实际生活中的题目,把 C 语言程序设计的方法融入实践环节中,用大量的实训案例、上机实训及同步练习帮助读者掌握程序设计的思想,学会程序设计的方法,训练程序设计的能力,达到培养学生能够初步解决程序设计实际问题的教学要求。

全书共分为 9 章。内容包括: C 语言程序设计的基本知识,顺序结构,选择结构,循环结构,编译预处理,数组,函数,指针和结构体,最后附本书同步练习答案。本书是编者在多年来从事 C 程序设计教学工作的基础上,结合教学实践中的经验编写的。书中列举学生容易出现问题的典型例题及实际生活中的题目,便于学生深入掌握重点内容,提高实践操作技能。本书力求体系结构安排合理、重点突出、难度适中;在语言叙述上注重概念清晰、通俗易懂,适应计算机教学实际需要。本书不仅可与《C 语言程序设计教程》的教材配套使用,也可作为培养实验能力的教材单独使用,还可以作为自学教材及工程技术人员的参考书。

本书编写分工如下: 绪论、第 1 章和第 9 章由杨文君编写,第 2 章~第 4 章由宋丽编写,第 5 章~第 8 章由柴宝杰编写。赵杰、杨文君对全书进行统稿和定稿。本书得到黑龙江省新世纪教学改革项目的支持。

本书在编写和出版过程中,参阅了一些专家和同行的科研资料、专著和教材,在此一并表示感谢!由于作者水平有限,加之时间仓促,书中错误和不当之处在所难免,敬请广大读者和专家批评指正。

编 者
2010 年 11 月

目 录

绪论	1
0.1 学习目标	1
0.2 知识要点	1
第1章 C 语言程序设计的基本知识	7
1.1 学习目标	7
1.2 知识要点	7
1.2.1 C 语言的结构	7
1.2.2 标识符、常量和变量	8
1.2.3 数据类型	9
1.2.4 数据类型转换	12
1.2.5 运算符和表达式	13
1.3 实训案例	18
1.3.1 题目	18
1.3.2 分析	18
1.3.3 程序	18
1.3.4 程序中使用的知识点	19
1.4 上机实训	19
1.4.1 实训目的	19
1.4.2 实训内容	19
1.5 同步练习	21
第2章 顺序结构	25
2.1 学习目标	25
2.2 知识要点	25
2.2.1 C 语言语句	25
2.2.2 数据输出、输入函数	26
2.2.3 顺序结构程序设计	28
2.3 实训案例	29
2.3.1 题目	29
2.3.2 分析	29
2.3.3 程序	29



2.3.4 程序中使用的知识点	30
2.4 上机实训	30
2.4.1 实训要求	30
2.4.2 实训步骤	30
2.5 同步练习	31
第3章 选择结构	37
3.1 学习目标	37
3.2 知识要点	37
3.2.1 用 if 语句构成单分支选择结构	37
3.2.2 用 switch 语句构成多分支选择结构	40
3.3 实训案例	41
3.3.1 实训案例一	41
3.3.2 实训案例二	42
3.3.3 实训案例三	43
3.4 上机实训	44
3.4.1 实训要求	44
3.4.2 实训内容	44
3.5 同步练习	46
第4章 循环结构	53
4.1 学习目标	53
4.2 知识要点	53
4.2.1 while 循环结构	53
4.2.2 do-while 循环结构	54
4.2.3 for 循环结构	54
4.2.4 循环嵌套	55
4.2.5 break 语句和 continue 语句在循环体中的作用	56
4.3 实训案例	57
4.3.1 实训案例一	57
4.3.2 实训案例二	58
4.3.3 实训案例三	59
4.4 上机实训	61
4.4.1 实训要求	61
4.4.2 实训内容	61

4.5 同步练习	62
第5章 编译预处理	66
5.1 学习目标	66
5.2 知识要点	66
5.2.1 宏定义	66
5.2.2 文件包含	67
5.2.3 条件编译	68
5.3 实训案例	69
5.3.1 题目	69
5.3.2 分析	69
5.3.3 程序	69
5.3.4 程序中使用的知识点	69
5.4 上机实训	69
5.4.1 实训目的	69
5.4.2 实训内容	70
5.5 同步练习	71
第6章 数组	75
6.1 学习目标	75
6.2 知识要点	75
6.2.1 一维数组的定义及应用	75
6.2.2 二维数组的定义及应用	76
6.2.3 字符串与字符数组	77
6.2.4 二维字符数组	78
6.3 实训案例	79
6.3.1 题目	79
6.3.2 分析	79
6.3.3 程序	79
6.3.4 程序中使用的知识点	80
6.4 上机实训	80
6.4.1 实训目的	80
6.4.2 实训内容	80
6.5 同步练习	82



第7章 函数	86
7.1 学习目标	86
7.2 知识要点	86
7.2.1 函数的定义与调用	86
7.2.2 递归函数	87
7.2.3 数组作函数参数	88
7.2.4 变量的作用域和函数的作用范围	88
7.2.5 变量的存储类别	89
7.3 实训案例	90
7.3.1 题目	90
7.3.2 分析	90
7.3.3 程序	91
7.4 上机实训	91
7.4.1 实训目的	91
7.4.2 实训内容	92
7.5 同步练习	93
第8章 指针	100
8.1 学习目标	100
8.2 知识要点	100
8.2.1 指针的概念	100
8.2.2 指针变量的定义和引用	100
8.2.3 指针变量作为函数参数	101
8.2.4 指向一维数组的指针变量	101
8.2.5 字符串与字符串指针变量	101
8.2.6 指针与函数	102
8.2.7 指针数组	102
8.3 实训案例	102
8.3.1 题目	102
8.3.2 分析	102
8.3.3 程序	103
8.3.4 知识点	104
8.4 上机实训	104
8.4.1 实训目的	104

8.4.2 实训内容	104
8.5 同步练习	106
第9章 结构体	113
9.1 学习目标	113
9.2 知识要点	113
9.2.1 用 typedef 定义一种新类型	113
9.2.2 结构体类型	113
9.2.3 结构体变量作为函数参数及返回结构体类型值的函数	115
9.2.4 用指针处理链表	116
9.2.5 共用体	117
9.3 实训案例	118
9.3.1 题目	118
9.3.2 分析	118
9.3.3 程序	118
9.3.4 程序中使用的知识点	119
9.4 上机实训	119
9.4.1 实训目的	119
9.4.2 实训内容	119
9.5 同步练习	122
同步练习答案	128
参考文献	135

绪 论

0.1 学习目标

- 📖 掌握 C 语言运行环境
- 📖 掌握 C 语言的运行方法
- 📖 领会 C 语言程序设计的风格

0.2 知识要点

1. Visual C++ 的安装与启动

Visual C++ 6.0 是在 Windows 环境中工作的,本节介绍的是 Visual C++ 6.0 中文版。

为了能使用 Visual C++ 6.0 集成环境,必须事先在所用的计算机上安装 Visual C++ 6.0 系统。在安装后建议在桌面上设立 Visual C++ 6.0 的快捷方式图标,以方便使用。

双击桌面上 Visual C++ 6.0 图标,就能进入 Visual C++ 6.0 集成开发环境,屏幕上出现 Visual C++ 6.0 的主窗口,如图 0-1 所示。

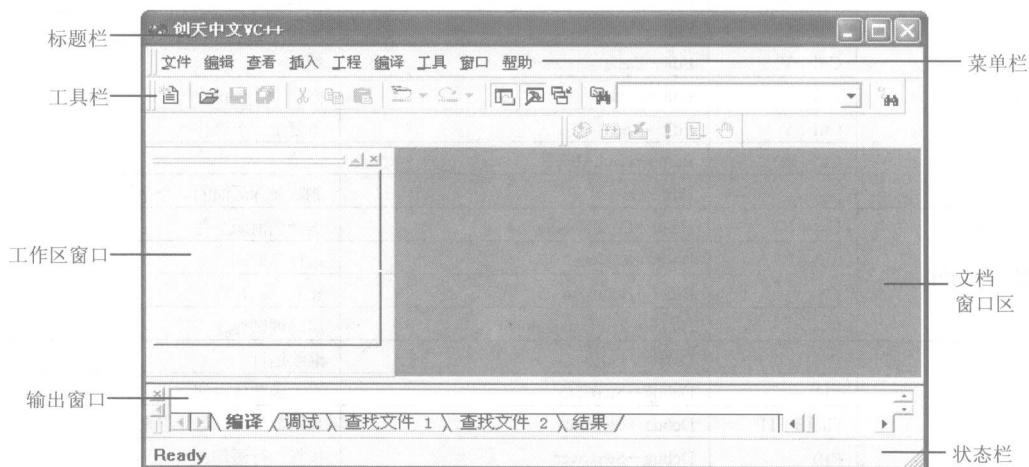


图 0-1 Visual C++ 6.0 主窗口

工具栏和菜单:用于提供用户操作的命令接口。菜单以文字和层次化的方式提供命令接口,工具栏由一系列按钮组成,这些按钮用一系列小的位图标志,工具栏以图标方式提供快速的命令选择。菜单和工具条在开发的不同进程有不同显示内容。当第一次打开,标准的工



具栏和菜单就会显示出来,随着开发的不同步骤,不同的工具条就会自动显示出来,菜单也会有所变化。

工作区窗口:这个窗口包含关于正在开发的项目的有关信息。在没有开发任何项目时,该窗口显示系统的帮助目录。当打开一个项目以后,工作区窗口将会显示关于当前项目的文件信息和类的信息。

文档窗口区:这个区域可以显示各种类型的文档,如源代码文件、头文件、资源文件等。可以同时打开多个文档。

输出窗口:输出窗口用来显示几种信息,可以通过选择不同的标签显示不同的信息。这些信息包括:编译连接结果信息(Build 标签)、调试信息(Debug 标签)、查找结果信息(Find in Files 标签)。其中查找结果信息有两个标签,可以显示两次在文件中查找指定内容的结果。

2. 常用功能键及其意义

为了使程序员能够方便、快捷地完成程序开发,开发环境提供了大量快捷方式来简化一些常用操作的步骤。Visual C++常用功能键如表 0-1 所示。

表 0-1 Visual C++常用功能键

操作类型	功能键	对应菜单	含义
文件操作	Ctrl+N	File→New	创建新的文件、项目等
	Ctrl+O	File→Open	打开项目、文件等
	Ctrl+S	File→Save	保存当前文件
编辑操作	Ctrl+X	Edit→Cut	剪切
	Ctrl+C	Edit→Copy	复制
	Ctrl+V	Edit→Paste	粘贴
	Ctrl+Z	Edit→Undo	撤消上一个操作
	Ctrl+Y	Edit→Redo	重复上一个操作
	Ctrl+A	Edit→Select All	全选
建立程序操作	Del	Edit→Del	删除光标后面的一个字符
	Ctrl+F7	Build→Compiler current file	编译当前源文件
	Ctrl+F5	Build→Run exe	运行当前项目
	F7	Build→Build exe	建立可执行程序
	F5	Build→Start Debugging	启动调试程序
调试	F5	Debug→Go	继续运行
	F11	Debug→Step into	进入函数体内部
	Shift+F11	Debug→Step out	从函数体内部运行出来
	F10	Debug→Step over	执行一行语句
	F9		设置/清除断点
	Ctrl+F10	Debug→Run to cursor	运行到光标所在位置
	Shift+F9	Debug→Quick Watch	快速查看变量或表达式的值
	Shift+F5	Debug→Stop debugging	停止调试

3. 输入与编辑源程序

(1) 新建一个源程序

在 Visual C++ 6.0 主菜单栏中选择“文件”|“新建”命令，如图 0-2 所示。

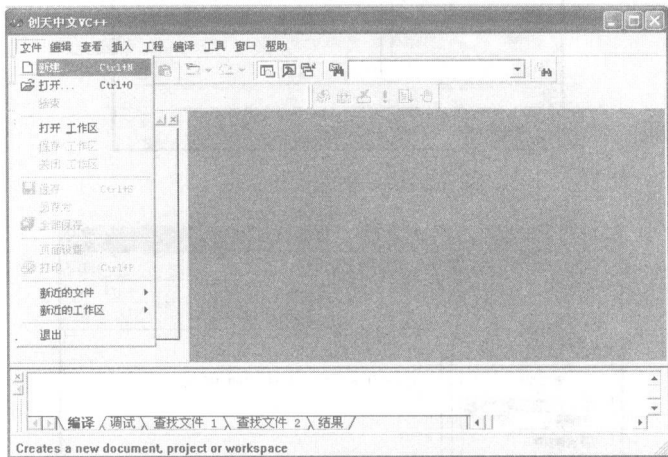


图 0-2 “文件”菜单下的“新建”命令

单击“新建”命令后屏幕上出现一个新建对话框。选择对话框的上方文件选项卡，在其下拉菜单中选择“C++ Source Files”项，如图 0-3 所示。然后在对话框右半部的文件中输入编辑的源程序文件名字（如 b.c）。

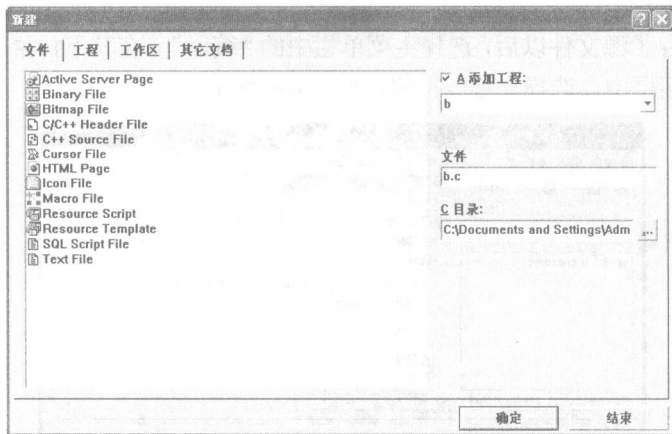


图 0-3 “新建”对话框

b.c 文件默认保存在 C 盘目录下。当然，我们也可以通过目录指定其文件的存储路径。单击“确定”按钮后，在编辑区输入源程序，如图 0-4 所示。

注意：源程序一定要在英文状态下输入，即字符标点都要在半角状态下，同时注意大小写，一般都用小写。

输入完毕后，在主菜单栏中菜单选择“文件”|“保存”命令，如图 0-5 所示。

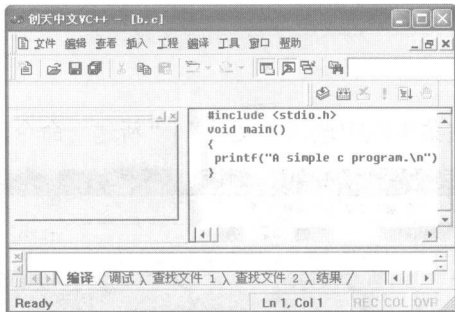


图 0-4 程序在编辑区的输入

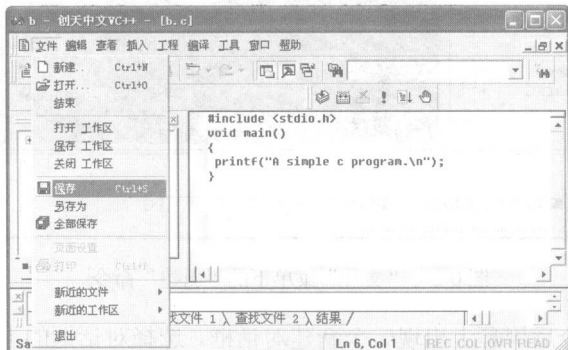


图 0-5 程序的保存

(2) 程序的编译

在编辑和保存了源文件以后，选择主菜单栏中的“编译”，在其下拉菜单中选择编译项。在弹出的提示对话框中，选择“是”按钮，如图 0-6 所示。

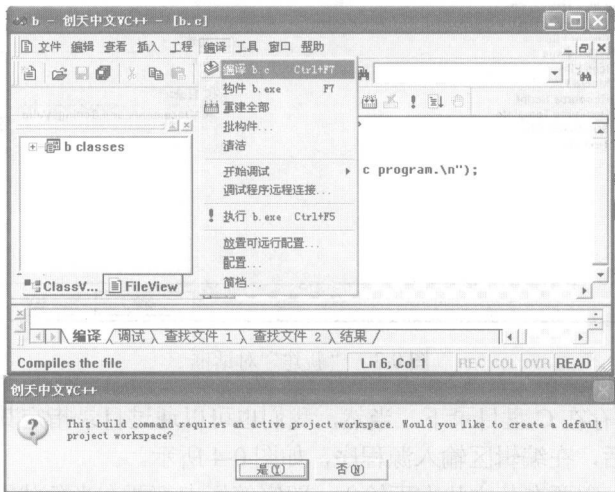


图 0-6 文件的编译

也可以不用选择主菜单的方法,而使用快捷键 **Ctrl+F7** 键来完成编译。编译系统检查源程序中有无语法错误,然后在主窗口下部的调试信息窗口输出编译的信息,如果无错,则生成目标文件 **b.obj**。否则会指出错误的位置和性质,提示用户改正错误。

(3) 程序的构建

在得到目标文件 **b.obj** 后,选择菜单“编译”|“构建”命令,如图 0-7 所示,表示连接并建立一个可执行文件。也可以使用快捷键 **F7**。

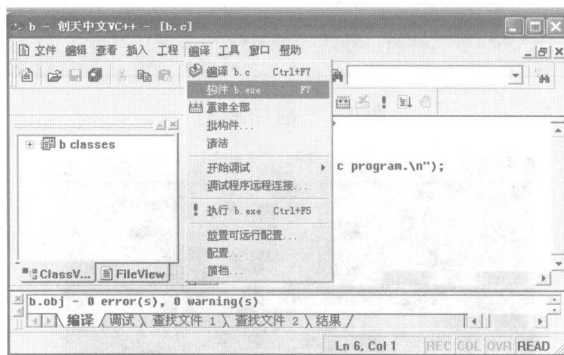


图 0-7 文件的构建

这时在调试输出窗口中显示构建时的信息,如图 0-8 所示。若无错误,则生成可执行的文件 **b.exe**。

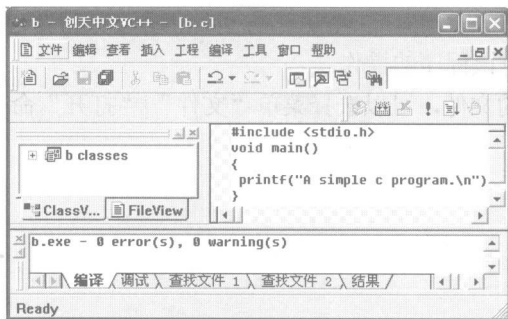


图 0-8 输出窗口显示连接成功后的信息

(4) 程序的执行

生成可执行的文件 **b.exe** 后,选择主菜单中的“编译”|“执行”命令,如图 0-9 所示,或直接按快捷键 **Ctrl+F5** 执行。

程序执行后,屏幕切换到输出结果的窗口,显示出运行结果,如图 0-10 所示。

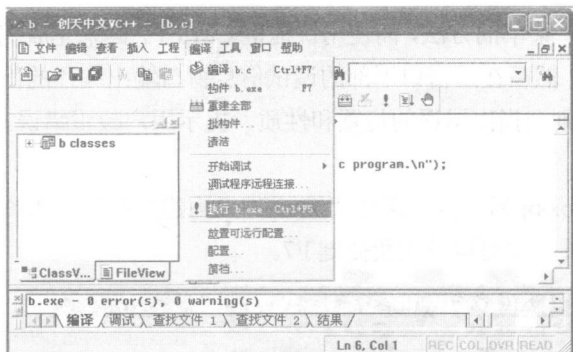


图 0-9 文件的执行

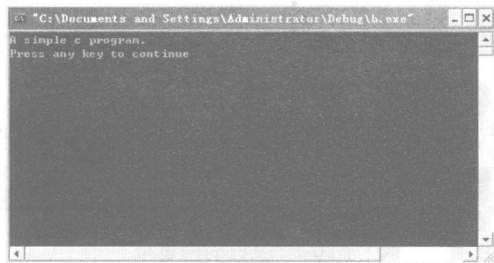


图 0-10 结果的输出窗口

(5) 关闭程序工作区

如果已完成一个程序的操作，不再对它进行其他处理，应当选择菜单“文件”|“关闭工作区”命令，以结束对该程序的操作。

若想打开一个已有的文件，可以选择菜单“文件”|“打开”命令。

第1章 C语言程序设计的基本知识

1.1 学习目标

- 📖 掌握 C 语言程序的结构
- 📖 了解基本类型及其常量的表示法
- 📖 掌握变量的定义及初始化方法
- 📖 掌握运算符与表达式的概念
- 📖 领会 C 语言的自动类型转换和强制类型转换的概念

1.2 知识要点

1.2.1 C语言的结构

(1) C 程序是由函数构成的。一个 C 源程序可以包含一个或多个函数，但有且仅有一个 main 主函数。函数是 C 语言的基本单位。函数分为两类：系统提供的库函数和用户根据需要编写的函数。

(2) C 程序总是从 main () 函数开始执行。

(3) 一个函数由两部分组成：函数的首部和函数体。

① 函数首部，也是函数的第一行。包含函数类型、函数名、函数属性、函数参数名（形式参数）、参数类型等。函数参数可以没有，但是函数后必须带一对圆括弧。如：main ()。

② 函数体。函数体是函数首部下面花括弧中的内容，包含声明部分和执行部分。声明部分用于定义函数中所使用的变量，执行部分由若干条语句组成。在某些情况下也可以没有声明部分，甚至也可以没有执行部分。

③ main 函数通常位于程序之首，实际上它位于程序的开头、最后及函数与函数之间均是合法的，但不管在什么位置，一个 C 程序总是从 main 函数开始执行的。

④ C 程序书写格式自由，一行内可以写几个语句，一个语句可以分写在多行上。例如：

```
int a,b;
a=5;b=8;    /*两个语句在一行*/
printf("a=%d,b=%d\n",
      a,b); /*一个语句在两行*/
```

⑤ C 语言本身没有输入输出语句，输入输出都由函数实现。

⑥ 可以用 /*...*/ 对 C 程序中的任何部分作注释，增强程序的可读性。

⑦ C 语言程序中每一条语句结束后都必须加一个分号。分号是 C 语言语句的组成部分。



1.2.2 标识符、常量和变量

1. 字符集

字符是 C 语言的最基本的元素, C 语言字符集由字母、数字、空白、标点和特殊字符组成(在字符串常量和注释中还可以使用汉字等其他图形符号)。由字符集中的字符可以构成 C 语言进一步的语法成分(如标识符、关键词、运算符等)。

组成 C 语言基本词汇的基本字符有:

(1) 数字 10 个 (0~9)。

(2) 英文字母大、小写各 26 个 (A~Z, a~z)。

(3) 空白符: 是指空格、制表符(跳格)、换行符(空行)的总称。空白符除了在字符、字符串中有意义外, 编译系统忽略其他位置的空白。空白符在程序中只是起到间隔作用。在程序的恰当位置使用空白符将使程序更加清晰, 增强程序的可读性。

(4) 标点符号、特殊字符: ! # % ^ & + - * / = _ ~ < > \ | . , ; : ? ' " () [] { }。

2. 标识符

标识符是起标识作用的一类符号。用来标识变量名、符号常量名、函数名、数组名、类型名等实体(程序对象)的有效字符序列。

(1) 标识符的特点

- 在 C 语言中, 一个合理的标识符只能由字母、数字和下划线 3 种字符组成, 且第一个字符必须为字母或下划线。
- ANSI C 没有限制标识符长度, 但各个编译系统都有自己的规定和限制。如 Turbo C 变量名最大长度为 32 个字符, MS C 则为 8 个字符。为了程序的可移植性以及阅读程序的方便, 建议变量名的长度最好不要超过 8 个字符。
- 标识符命名区分大、小写字母。
- 通常标识符命名时应当有一定的意义, 做到“见名知义”。
- 不同的 C 语言系统对标识符的有效字符个数有不同的规定。一般取 8 个字符; 也就是说, 两个超过 8 个字符的不同标识符, 当前 8 个字符完全一样时, 系统就认为它们是同一个标识符, 而不加以区别。

(2) 标识符的分类

C 语言的标识符可以分为三类。

① 关键字。所谓关键字是指它们用来表示 C 语言本身的特定成分, 具有相应的语义, 不能再另做他用。所有关键字都使用小写英文字母表示, 共 32 个:

auto	break	case	char	const	continue	default	do	double
else	enum	extern	float	for	goto	if	int	long
register	return	short	signed	sizeof	static	struct	switch	typedef