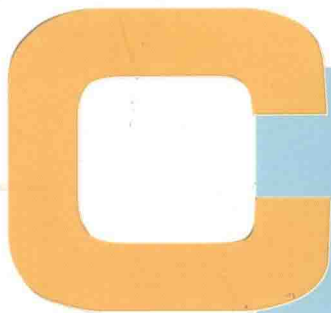




21世纪高等学校计算机科学与技术规划教材



主编 薛纪文 高晓娟 加云岗

语言程序设计教程



C Yuyan Chengxu Sheji Jiaocheng

主审 王会燃 石美红



北京邮电大学出版社
www.buptpress.com



21 世纪高等学校计算机科学与技术规划教材

C 语言程序设计教程

主 编 薛纪文 高晓娟 加云岗
主 审 王会燃 石美红

北京邮电大学出版社
• 北京 •

内 容 简 介

本书根据教育部计算机基础课程教学指导委员会制订的大纲,结合目前普通高等院校计算机基础教学的现状和全国计算机等级考试大纲,由西安工程大学计算机学院担任“C 语言程序设计”课程的一线教师组织编写。

全书共分为 13 章,主要内容包括:C 语言概述、基本数据类型与表达式(含位运算)、顺序结构、选择结构、循环结构、数组、函数、编译预处理、指针、结构体与链表、共用体和枚举、文件以及 C 语言在工程项目开发中的综合应用实例。

本书主线清晰、重点明确、叙述严谨、内容由浅入深。书中从计算机等级考试试题中选用了大量的经典试题作为例题和练习题,有很强的针对性,并有配套的教学课件。

本书适合作为高等院校的教学用书,也可作为参加全国计算机等级考试二级(C 语言)的辅导教材,同时也是工程技术人员较好的参考用书。

图书在版编目(CIP)数据

C 语言程序设计教程/薛纪文,高晓娟,加云岗主编. --北京:北京邮电大学出版社,2010.2
ISBN 978-7-5635-2167-8

I. ①C… II. ①薛… ②高… ③加… III. ①C 语言—程序设计—教材 IV. ①TP312

中国版本图书馆 CIP 数据核字(2010)第 009125 号

书 名	C 语言程序设计教程
主 编	薛纪文 高晓娟 加云岗
责任编辑	沙一飞
出版发行	北京邮电大学出版社
社 址	北京市海淀区西土城路 10 号(100876)
电话传真	010-62282185(发行部) 010-62283578(传真)
电子信箱	ctrd@buptpress.com
经 销	各地新华书店
印 刷	北京忠信诚胶印厂
开 本	787 mm×1 092 mm 1/16
印 张	25
字 数	626 千字
版 次	2010 年 2 月第 1 版 2010 年 2 月第 1 次印刷

ISBN 978-7-5635-2167-8

定价: 39.00 元

如有质量问题请与发行部联系

版权所有 侵权必究

前 言

“C 语言程序设计”是培养学生程序设计方法与技能的课程,它详细介绍了 C 语言程序设计的思想和方法。通过学习,学生可以掌握 C 语言程序设计的方法与技能,为后续专业课程的学习打基础,更为重要的是在教学实践中培养解决实际问题的能力。

本书编者根据多年来从事“C 语言程序设计”教学的丰富经验和教学过程中收集的典型案例,编写完成了本书。本书内容由浅入深、编写风格严谨。

全书共分为 13 章,第 1 章介绍了 C 语言程序设计的基础知识。第 2 章介绍了 C 语言基本数据类型与表达式,包含位运算。第 3 章~第 5 章介绍了结构化程序设计方法,包括顺序结构、选择结构和循环结构。第 6 章、第 9 章~第 12 章分别详细介绍了数组、指针、结构体、共用体、枚举和文件等类型及其相关应用。第 7 章详细介绍了函数及其使用。第 8 章介绍了编译预处理。第 13 章详细介绍了 C 语言在工程项目开发中的综合应用。

本书的特色有如下几点:

①为了满足学生参加全国计算机等级考试二级(C 语言)的需要,我们在等级考试的试题中选择了大量的经典试题进行分析和讲解。

②详细介绍了 C 语言在工程项目开发中的应用,对工程项目开发有很高的参考价值。

为了配合实验教学,我们还编写了配套的实验指导用书《C 语言程序设计实践教程》,其中提供了教材中全部的习题解答及模拟套题。

参加本书编写工作的有薛纪文、高晓娟、加云岗、黄国兵、张旭风、李莉、马骥等。

本书的出版要感谢西安工程大学计算机学院的全体教职工,他们前期的教学和实践经验为本书的完成奠定了基础。在本书的写作过程中还得到了王会燃教授和石美红教授的指导,在此表示衷心感谢。

书中内容符合计算机专业和非计算机专业程序设计课程的教学大纲,也兼顾了全国计算机等级考试二级(C 语言)大纲的有关要求,因此,本书既可作为高等院校的教学用书,也可作为等级考试的辅导用书。

尽管本书策划及写作时间较长,但由于编者水平所限,书中难免有不妥之处,恳请各位读者指正。

编 者

目 录

第1章 C语言概述	1
1.1 程序设计的相关概念	1
1.2 C/C++语言的产生及特点	3
1.3 C语言程序的结构特点	4
1.4 C语言中的词汇	6
1.5 C语言程序编写的基本规范	7
1.6 C语言程序的开发步骤及 Visual C++ 6.0 集成开发环境介绍	9
习题	11
第2章 基本数据类型和表达式	14
2.1 C语言的数据类型	14
2.2 标识符、常量和变量	15
2.3 整型数据	16
2.4 实型数据	19
2.5 字符型数据	21
2.6 算术表达式	23
2.7 赋值表达式	27
2.8 关系表达式和逻辑表达式	28
2.9 逗号表达式	30
2.10 位运算	31
习题	34
第3章 顺序结构	37
3.1 C语句概述	37
3.2 程序的3种基本结构	39
3.3 数据的输入/输出	41
3.4 顺序结构程序设计举例	50
习题	54
第4章 选择结构	59
4.1 if语句	59
4.2 switch语句	64
4.3 选择结构程序设计举例	67
习题	70
第5章 循环结构	76
5.1 概述	76
5.2 while语句及其循环结构	77
5.3 do...while语句及其循环结构	80
5.4 for语句及其循环结构	81

5.5	break、continue 及 goto 语句	83
5.6	循环的嵌套	86
5.7	循环结构程序设计举例	88
	习题	92
第 6 章	数组	96
6.1	一维数组的定义和引用	96
6.2	二维数组的定义和引用	103
6.3	字符数组	110
	习题	123
第 7 章	函数	129
7.1	函数概述	129
7.2	函数的定义与返回值	130
7.3	函数的调用	133
7.4	函数的嵌套调用	137
7.5	函数的递归调用	139
7.6	数组作为函数的参数	144
7.7	局部变量和全局变量	149
7.8	内部函数和外部函数	156
	习题	157
第 8 章	编译预处理	162
8.1	宏定义	162
8.2	文件包含	168
8.3	条件编译	170
	习题	173
第 9 章	指针	178
9.1	存储单元及其地址	178
9.2	变量的实质	179
9.3	指针和变量	181
9.4	指针和数组	188
9.5	指针和字符串	199
9.6	指针和函数	205
9.7	指针和指针	214
	习题	217
第 10 章	结构体与链表	228
10.1	概述	228
10.2	结构体类型的声明	229
10.3	结构体类型变量的定义	229
10.4	结构体变量的引用	232
10.5	结构体变量的初始化	233
10.6	结构体数组	234
10.7	指向结构体类型数据的指针	238

10.8	链表·····	245
10.9	用 typedef 定义类型·····	264
	习题·····	265
第 11 章	共用体和枚举·····	271
11.1	共用体类型·····	271
11.2	共用体类型应用举例·····	274
11.3	枚举类型·····	276
	习题·····	280
第 12 章	文件·····	282
12.1	文件概述·····	282
12.2	文件指针·····	283
12.3	文件的打开和关闭·····	284
12.4	文件的读写·····	286
12.5	文件的随机读写·····	288
12.6	文件检测函数·····	290
12.7	库文件·····	291
12.8	文件应用举例·····	292
	习题·····	295
第 13 章	C 语言工程应用与项目实践·····	301
13.1	概述·····	301
13.2	项目背景·····	302
13.3	RTU 实时多任务运行软件的设计与实现·····	307
13.4	项目实践总结·····	380
附录 1	ASCII 码表·····	382
附录 2	运算符的优先级和结合性·····	385
附录 3	C 语言标准库函数·····	386
	参考文献·····	392

第1章 C语言概述

【教学目的】

1. 了解 C 语言的发展过程及语言特点。
2. 掌握 C 语言程序的组成。
3. 掌握程序设计的概念、规范和步骤。
4. 掌握在 Visual C++ 6.0 集成开发环境下开发 C 语言程序的过程。

【教学重点】

C 语言程序的组成、程序设计的规范和步骤。

【教学难点】

C 语言程序的开发过程及开发环境的掌握。

本章从计算机算法、数据结构、程序等基本概念入手,介绍了程序设计语言的产生、发展,以及 C 语言的程序特点、程序代码的书写规范及对简单程序的认识,最后学习在 Visual C++ 6.0 集成开发环境下,编译、连接和运行 C 语言程序的过程。

1.1 程序设计的相关概念

1.1.1 基本概念

- 程序(Program):程序是控制计算机完成特定功能的一组有序指令的序列。
- 程序设计语言:编写程序所使用的语言称为程序设计语言,它是人与计算机之间进行信息交流的工具。例如,C 语言和 C++ 语言是两种不同的程序设计语言,是程序员编程的工具。程序设计语言的发展从低级到高级,经历了机器语言、汇编语言、高级语言和面向对象语言 4 个阶段。
- 语句(Statement):组成语言的基本单位,包含语法和语义两个方面。
- 语法(Syntax):定义构造语言的正确的表达式或句子所需的各种规则。
- 程序设计(Programming):利用程序设计语言,设计、书写及检查程序的过程,包括分析问题、确定解决方法、设计程序结构,最后使问题的内容和解题方案用计算机能够接收的指令或语句序列来实现。

1.1.2 算法

什么是算法(Algorithm)呢?我们可以把算法定义成解一类问题的精确的、有效的方法,它是一组有穷的规则,规定了解决某一特定类型问题的一系列运算,是对解题方案准确的、完整的描述。制定一个算法,一般要经过设计、确认、分析、编码、测试和调试等阶段。

1. 算法的特征

- 确定性:算法的每一种运算必须有确定的含义,该运算执行何种操作,目的明确,无二义性。

● 可行性:要求算法中有待实现的运算都是基本的,每种运算至少在原理上能用手工在有限的时间内完成。

● 输入:一个算法有零个或多个输入,在算法运算开始之前,给出算法所需数据的初值。

● 输出:算法运算的结果,一个算法需要产生一个或多个输出,输出是同输入有某种特定关系的量。

● 有穷性:一个算法应包含有限个操作步骤,即该算法是可达的。

只满足前4个特性的一组规则不能称为算法,而称为计算过程。例如,操作系统是计算过程的一个典型例子。操作系统用来管理计算机资源,控制作业的运行,没有作业运行时,计算过程并不停止,而是处于等待状态。

2. 算法的要求

对算法需满足正确性、可读性、健壮性、高效性和低存储量的要求。

3. 算法的描述

关于算法的描述有如下几种方法:

(1) 自然语言

自然语言容易理解,但是由于自然语言语义不严谨、有不确定性、对复杂的问题难以表达准确、不能被计算机识别和执行等因素,一般情况下不使用。

(2) 专用工具

专用工具有流程图、N-S图等,用流程图描述算法使用较多。流程图是人们对解决问题的方法、思路或算法的一种图形描述。流程图的优点有:

①采用简单规范的符号,画法简单。

②结构清晰,逻辑性强。

③便于描述,容易理解。

图 1-1 是常用的流程图标准化符号。图 1-2 描述了从键盘输入两个数据,并在显示器上输出两数之和的算法实现过程。

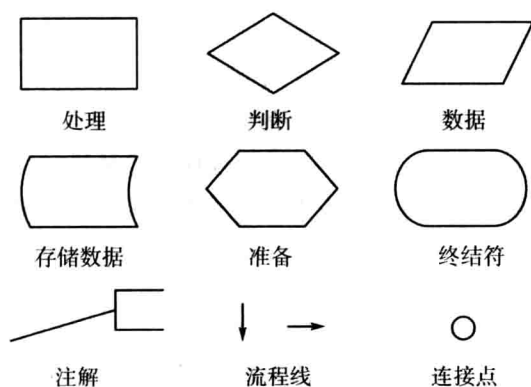


图 1-1 部分流程图标准化符号

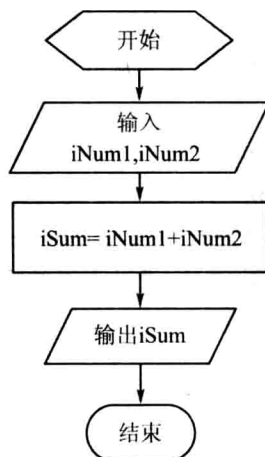


图 1-2 流程图样例

(3) 伪代码

伪代码是用代码的格式表示程序的执行过程,它不能在机器上通过编译,目的是为了展示程序的执行过程。

图 1-2 也可以用伪代码描述如下:

```
Begin
    Input iNum1,iNum2
    iSum = iNum1 + iNum2
    Output iSum
End
```

1.1.3 数据结构

数据结构(Data Structures)是在整个计算机科学与技术领域上广泛被使用的术语。它用来反映一个数据的内部构成,即一个数据由哪些成分数据构成,以什么方式构成,呈现怎样的结构。

数据结构有逻辑上的数据结构和物理上的数据结构之分。逻辑上的数据结构反映数据之间的逻辑关系,而物理上的数据结构反映数据在计算机内部的存储安排。

数据结构是信息的一种组织方式,其目的是为了提高算法的效率。它通常与一组算法的集合相对应,通过这组算法集合可以对数据结构中的数据进行某种操作。

数据结构主要研究数据的各种逻辑结构和存储结构,以及对数据的各种操作。因此,主要有3个方面的内容:数据的逻辑结构、数据的物理存储结构、对数据的操作(或算法)。通常,算法的设计取决于数据的逻辑结构,算法的实现取决于数据的物理存储结构。

数据结构是计算机专业教学计划中的核心课程之一,在计算机科学中是一门综合性的专业基础课,有兴趣的读者可以参考相关的资料进一步学习。

1.1.4 算法+数据结构=程序

1976年,著名的瑞士籍科学家沃思(N. Wirth,1984年图灵奖得主、1987年计算机先驱奖获得者)提出了一个经典的公式:

算法+数据结构=程序

此公式不仅指出了数据结构在计算机科学中的地位,也指出了它与算法的密切联系。算法与数据结构是相辅相成的,是缺一不可的两个方面。这就是说,既不能离开数据结构去抽象地分析求解问题的算法,也不能脱离算法去孤立地研究程序的数据结构。

1.2 C/C++ 语言的产生及特点

20世纪70年代以前,软件主要是用汇编语言进行编写的。由于汇编语言过于依赖硬件,所以可读性和移植性较差,而高级语言又难以实现对计算机硬件的直接操作,于是希望有一种既具有汇编语言特性,又具有高级语言特性的新语言。就在这样的背景下,C语言诞生了。

1978年由美国贝尔实验室正式发表了C语言,同时由B. W. Kernighan和D. M. Ritchie两人合著了著名的“The C Programming Language”一书,简称“K&R”标准,后来被多次改进,出现了多种C语言的版本。1989年,由美国国家标准协会(American National Standards Institute, ANSI)根据C语言产生以来的各种版本,对C语言制定了一个C语言的标准,称之为ANSI C。

C语言仍然是当今世界上最为流行的面向过程的程序设计语言之一。目前流行的C语言主要有以下几种:Microsoft C(MS C)、Borland Turbo C(Turbo C)、AT&T C、Quick C。

C语言有如下的特点:

①C语言同时具有汇编语言和高级语言的特性。在DOS环境下,C语言允许直接访问物理

地址,能进行“位”操作,可以直接对硬件进行操作,所以有时也将 C 语言称为中级语言。

- ②语言简洁、紧凑、方便、灵活。
- ③有 34 种丰富的运算符。
- ④生成的目标代码质量高、效率高。
- ⑤有较好的移植性。

1980 年由贝尔实验室在 C 语言的基础上,推出了 C++ 语言,它进一步扩充和完善了 C 语言体系。1994 年,第 1 个 C++ 标准 ANSI C++ 被制定出来。C++ 语言是一种面向对象的程序设计语言,它提供了一种与面向过程的程序设计不同的思维方式和编程方法。

由于篇幅所限,本书主要以 Turbo C 语言进行学习,想深入学习 C++ 语言的读者,可以参阅 C++ 相关的资料。

1.3 C 语言程序的结构特点

我们先来看一个简单的 C 语言程序。

【例 1.1】 从键盘输入两个数据,并在显示器上输出两数之和。

程序的流程图如图 1-2 所示,程序实现如下:

```
#include<stdio.h>    /* 包含的头文件 */
void main()          /* 主函数,返回类型为 void 空类型 */
{
    int iNum1,iNum2,iSum;    //变量的定义声明
    scanf("%d,%d",&iNum1,&iNum2);    //键盘接收两个数据并赋值给 iNum1 和 iNum2
    iSum = iNum1 + iNum2;
    printf("%d\n",iSum);    // 输出 iSum 变量的值
}
```

在 Visual C++ 6.0 环境下,经过编译、连接,然后运行程序,其运行结果如图 1-3 所示。

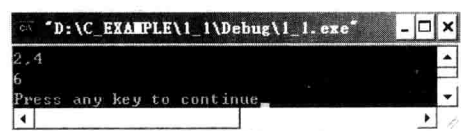


图 1-3 例 1.1 程序运行结果

对于 C 语言程序,一般有如下特点:

- ①一个 C 语言工程可以由一个或多个 C 语言源程序组成。
- ②构成 C 语言源程序的基本单位是函数,一个 C 语言源程序可由一个 main 函数和若干个其他的函数组成。其中,必须有一个且只能有一个 main 函数。任何函数(包括主函数 main)都是由函数声明和函数体两部分组成的。函数的标准结构如下:

```
函数返回值类型 函数名(函数的形式参数)    /* 函数头声明部分 */
{
    声明语句部分;    /* 函数体部分 */
    执行语句部分;
}
```

函数声明部分包括函数返回值类型、函数名、形式参数类型、形式参数(简称形参)。例如,例 1.1 中的 main 函数:

void	main	()
↓	↓	↓
函数返回值类型	函数名	函数的形参(本例中没有参数)

函数体即函数声明部分下面的用一对花括号括起来的部分{……}。

③C 语言程序总是从 main 函数开始执行,在 main 函数结束。

④C 语言程序的语句必须以分号(;)结束。在 C 语言中,分号是程序语句结束的标志,也是 C 语言语句的必要组成部分。但预处理命令、函数头和花括号{}之后不能加分号。

⑤源程序中可以有预处理命令(预处理命令将在后面的章节进行介绍),预处理命令通常应放在源程序的最前面,如#include"stdio.h"。

⑥C 语言的数据输入/输出是由库函数实现的,C 语言本身没有输入/输出语句。输入/输出操作是由库函数 scanf 和 printf 等函数来实现的。除此之外,C 语言还提供了大量的标准库函数,程序员可以根据自己的需要进行选择。

⑦在 C 语言中,用“/* 注释内容 */”的形式进行程序的注释,但在 Visual C++ 6.0 中调试程序时,允许 C 程序使用 C++程序的注释方法,即“//注释内容”。在“//”之后或“/*”和“*/”之间的所有字符都是注释内容,系统对注释内容不进行编译。注释可出现在程序中的任何位置。注释通常用于:版本、版权声明;函数接口说明;重要的代码行或段落提示。注释用来向用户提示或解释程序的意义,增加程序的可读性。在调试程序中,对暂不使用的语句也可用注释符括起来,使编译跳过不进行处理。

⑧C 语言源程序文件的扩展名为“c”。

【例 1.2】 注释参考样例。

```
/*
 * Copyright(c) 2008,西安工程大学计算机学院
 * All rights reserved.
 * 文件名称:l_1.c
 * 摘 要:本程序是一个样例程序,主要用来介绍一个简单的 C 程序
 * 运行环境:Visual C++ 6.0
 * 当前版本:1.1
 * 作 者:薛纪文
 * 完成日期:2009 年 9 月 15 日
 */

/*
 * 函数介绍:主函数实现两数相加,并输出运算结果
 * 输入参数:无
 * 输出参数:无
 * 返回值 :无
 */
#include<stdio.h>    /* 包含的头文件 */
void main()          /* 主函数,返回类型为 void 空类型 */
{
    int iNum1,iNum2,iSum;    //变量的定义声明
```

```
scanf("%d,%d",&iNum1,&iNum2);    //键盘接收两个数据并赋值给 iNum1 和 iNum2
iSum = iNum1 + iNum2;
printf("%d\n",iSum);    // 输出 iSum 变量的值
}
```

1.4 C 语言中的词汇

C 语言中使用的词汇可分为 6 类:标识符、关键字、运算符、分隔符、常量和注释符。

1. 标识符

标识符是指用来标识程序中用到的变量名、函数名、类型名、数组名、文件名以及符号常量名的有效字符序列。C 语言规定,标识符只能是字母 A~Z、a~z、数字 0~9、下划线(_)组成的字符串,并且其第 1 个字符必须是字母或下划线。例如:

以下标识符是合法的:

```
a,a1,x4,student,sum,temp,_pointer
```

以下标识符是非法的:

```
3x    // 不能以数字开头
-x    // 出现非法字符“-”(减号)
```

定义标识符的时候,还应注意遵守命名规则,下一节将详细介绍。

2. 关键字

关键字(也称为保留字)是由 C 语言规定的具有特定意义的标识符。ANSI C 一共只有 32 个关键字,根据关键字的作用,可将其分为数据类型关键字、控制语句关键字、存储类型关键字和其他关键字 4 类。值得注意的是,这些关键字均为小写,如表 1-1 所示。

表 1-1 C 语言中的关键字

ANSI C 关键字 (32 个)	数据类型关键字 (12 个)	char、double、enum、float、int、long、short、signed、struct、union、unsigned、void	用于声明变量、函数或其他数据结构的类型
	控制语句关键字 (12 个)	break、case、continue、default、do、else、for、goto、if、return、switch、while	用于表示一个语句的功能
	存储类型关键字 (4 个)	auto、extern、register、static	变量的存储类型
	其他关键字 (4 个)	const、sizeof、typedef、volatile	
Turbo C 扩充关键字 (11 个)		asm、_cs、_ds、_es、_ss、cdecl、far、huge、interrupt、near、pascal	

3. 运算符

C 语言中含有丰富的运算符,共有 34 种,可以实现其他高级语言难以实现的运算。C 语言中的运算符分类如表 1-2 所示。

表 1-2 C 语言中的运算符分类表

算术运算符 (7 个)	加(+)、减(-)、乘(*)、除(/)、求余(或称模运算,%)、 自增(++)、自减(--)	用于各类数值运算
关系运算符 (6 个)	小于等于(<=)、小于(<)、等于(==)、大于(>)、大于 等于(>=)、不等于(!=)	用于比较运算
逻辑运算符 (3 个)	与(&&)、或()、非(!)	用于逻辑运算
位运算符 (6 个)	位与(&)、位或()、位非(~)、位异或(^)、左移(<<)、 右移(>>)	按二进制位进行运算
赋值运算符 (11 个)	简单赋值(=) 复合算术赋值(+=、-=、*=、/=、%=) 复合位运算赋值(&=、 =、>=、<=)	用于赋值运算
条件运算符 (1 个)	?: (三目运算符)	用于条件求值
逗号运算符 (1 个)	,	用于把若干表达式组合成 一个表达式
指针运算符 (2 个)	取内容(*)、取地址(&)	用于指针运算
求字节数运算符 (1 个)	sizeof	用于计算数据类型所占的 字节数
特殊运算符	括号()、下标[]、成员(→、.)等	

4. 分隔符

C 语言中的分隔符有逗号、空格和冒号等。逗号主要用在类型声明和函数参数表中,分隔各个变量。空格多用于语句各单词之间,起分隔作用。在关键字、标识符之间必须要有一个或多个空格,否则将会出现语法错误。冒号作为语句标号、语句间的分隔符以及 switch 语句中 case 与语句序列之间的分隔符。

例如,将“int temp;”写成“ inttemp;”,C 编译器会把 inttemp 当成一个标识符处理。

5. 常量

C 语言中使用的常量可分为:数字常量、字符常量、字符串常量、符号常量和转义字符等,这些内容将在后面章节详细介绍。

6. 注释符

C 语言中的注释符有两种: //……和 /* …… */。

1.5 C 语言程序编写的基本规范

在程序设计过程中,不仅仅是最终完成了某项任务,程序就是好程序。要写出一个高质量的程序代码,还要注意很多细节的问题,如代码的书写规范、程序的执行效率和文档的编写等。本节仅就书写规范,参照一些有经验的程序员的建议,在这里归纳了一些基本的需要初学者掌握的

编码规范。

(1)C 语言程序书写格式

对于短语句,可以将多条语句书写在同一行,也可以将长语句分写在多行。注意分行时不能将一个单词分开,也不能将双引号引起来的字符串分开。“{”一般与该结构语句的第 1 个字母对齐,并单独占一行。强烈建议使用一行一条语句的风格,也就是说“一行代码只做一件事情”。

(2)C 语言程序书写采用缩进格式(锯齿形书写格式)

根据语句的从属关系,程序书写时采用缩进格式。具体的方法是:同一层次语句要左对齐,不同层次的语句要缩进若干个字符。这样程序层次清楚,便于阅读和理解。

(3)空行的使用

空行起着分隔程序段落的作用,空行用得得体(不过多也不过少),会使程序的布局更加清晰。

(4)添加程序的注释

前面介绍了注释是用来向用户解释程序意义的,可以增加程序的可读性。应大约 5 行代码至少有一行有注释,用词要简单、清楚。

一般来说,程序块的注释常采用“/* */”方法,行的注释一般采用“//.....”方法。

(5)命名规则

①Microsoft 公司的匈牙利命名法被广泛使用,此命名技术是由一位 Microsoft 的程序员查尔斯·西蒙尼(Charles Simonyi)提出的。匈牙利命名法是一种编程时的命名规范,其基本原则是“标识符=属性+类型+对象描述”,其中每一部分的名称都要求有明确的含义,命名要基于容易记忆、容易理解的原则。该命名规则的主要思想是“在变量和函数名前加入前缀,以增进人们对名称的理解”。匈牙利命名法中常用标识符组成参考如表 1-3 所示。

表 1-3 匈牙利命名法中常用标识符组成参考

属性	类型	对象描述举例
g_ 表示全局变量	a 表示数组	最大 Max
c_ 表示常量	ch 表示字符(char)	最小 Min
s_ 表示静态变量	n 表示短整型(short int)	初始化 Init
	i 表示整型(int)	临时变量 T(或 Temp)
	l 表示长整型(long int)	源对象 Src
	u 表示无符号(unsigned)	目的对象 Dest
	r 表示实型	
	sz 表示字符串	
	fn 表示函数	
	p 表示指针	
	f 表示文件	

【例 1.3】 采用匈牙利命名规则定义变量。

```
int iNumber;    // 前缀 i 表示 int 类型变量
char chName;    // 前缀 ch 表示字符类型变量
char * pPointer; // 前缀 p 表示指针类型变量
```

一般来说,名字中的前缀字母小写,其后每个英文单词的第 1 个字母大写,其他的小写。例如:

```
int iPageCount
```

说明:在实际的项目开发中,匈牙利命名法必须灵活运用。例如,单字符的名字也是有用的,常见的如 i、j、k、m、n、x 等,它们通常可用做函数内的局部变量或循环变量。

②标识符虽然可由程序员随意定义,但标识符是用于标识某个量的符号,因此命名应尽量有相应的意义,以便于阅读理解,做到“顾名思义”。例如,要定义某变量存放学生姓名,可以将其定义为 chName;若定义为 x1,这样的程序可读性就会很差。

③用户定义的标识符不能与关键字相同。

④在标识符中,C 语言严格区分大小写。例如,对于 area、Area 和 AREA,系统认定为 3 个不同的标识符。

⑤ANSI C 规定名字不准超过 8 个字符,现今的 C/C++ 不再有此限制。标识符的长度应当符合“min-length & max-information”原则,即短名称大信息。一般来说,长名称能更好地表达含义,所以函数名、变量名等长达十几个字符也不足为奇,但并不是越长越好,如变量名 max-Val 就比 maxValueUntil 好用。

规范的、高质量的代码需要我们遵守很多成文的或不成文的规范,本小节仅仅介绍了一小部分程序员好的建议,在这里我们抛砖引玉,更多的内容需要读者在实践的过程中用心去体会,也可参阅相关的资料进一步学习。

1.6 C 语言程序的开发步骤及 Visual C++ 6.0 集成开发环境介绍

1.6.1 C 语言程序的开发步骤

计算机是一种有效的信息处理机,我们利用它去处理信息来实际问题的时候,一般使用下面的开发步骤:

(1)分析问题

首先根据任务书的要求,对问题进行深入分析,弄清楚问题有什么要求,有哪些已知条件,从给出的已知条件出发,经过哪些必要的处理才能解决问题。

(2)设计算法

确定解决方法,抽象出一个适当的数学模型,并画出相应的算法流程图。

(3)编写程序

用 Turbo C 编写程序代码,并作静态分析。

(4)上机调试

利用 Visual C++ 6.0 集成开发环境调试(动态测试),消除语法错误或逻辑错误,直到功能实现。

(5)编写、整理文档资料

在项目开发中,规范的开发过程需要撰写相关的文档。文档主要有如下几种:

- 可行性分析报告:报告说明该项目在技术、经济等因素的可行性,评述可供选择的多种实施方案,并论证所选定方案的理由。

- 项目开发计划:制订具体实施计划,包括负责人员、开发进度、经费预算和所需的环境等。

- 需求说明书:对所开发项目的功能、性能、用户界面及运行环境等进行较详细的说明,它是在用户与开发人员双方对需求取得共同理解并达成协议的条件下编写的。

- 概要设计报告:说明模块的划分、系统的总体结构、输入输出以及接口设计、数据结构设计和出错处理设计等,这是详细设计的基础。

- 详细设计报告:描述流程、算法实现等。

- 测试计划:组织测试计划,出测试分析报告。
- 修改报告:对修正、更改的问题进行详细的记录。
- 用户操作手册:描述系统的功能、性能和用户界面,让用户了解操作的具体细节。

1.6.2 Visual C++ 6.0 集成开发环境介绍

Visual C++ 6.0 是由 Microsoft 公司推出的被广泛使用的可视化开发工具,是 Visual Studio 6.0 家族中的一员。Visual C++ 提供了一个集源程序编辑、代码编译与调试于一体的开发环境,这个环境称为集成开发环境。Visual C++ 6.0 依靠强大的编译器以及网络与数据库的开发能力,可以开发出功能强大的基于 Windows 平台的 32 位应用程序,也可以开发本教材将要学习的 Win32 控制台应用程序。Visual C++ 6.0 有 3 个版本:标准版、专业版和企业版。

1. Visual C++ 6.0 编辑环境

图 1-4 就是 Visual C++ 6.0 运行的主界面,图中标注了界面主要的组成部分,包括工具栏、菜单栏、标题栏、项目工作区、程序和资源编辑窗口以及状态栏等。

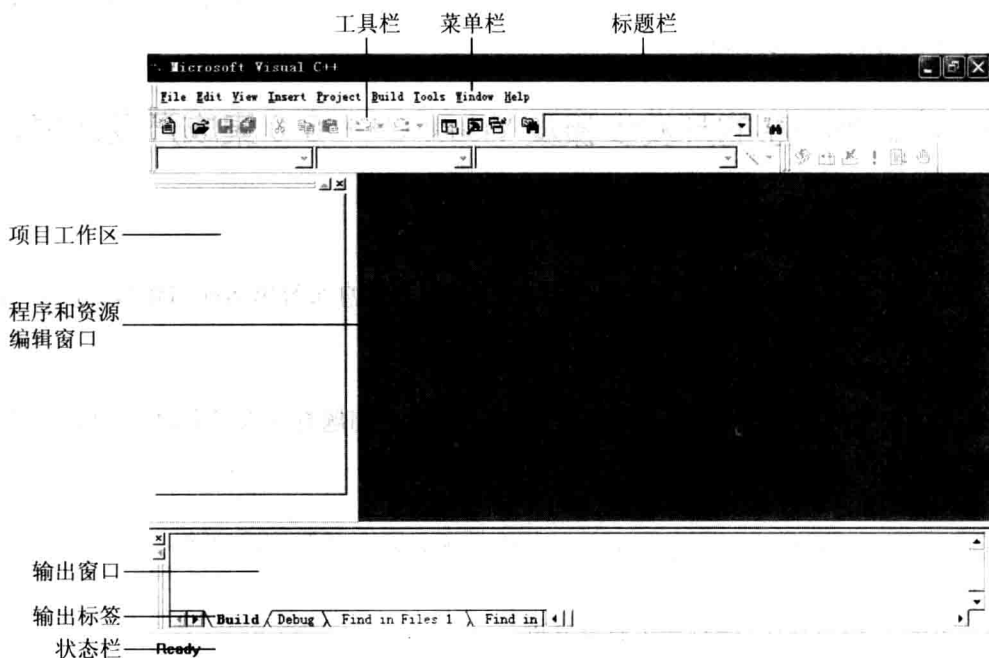


图 1-4 Visual C++ 6.0 主界面

2. 编辑源程序文件

- ① 编辑创建源程序,即将编写好的 C 语言源程序代码录入到计算机中,形成源程序文件。
- ② 在 Visual C++ 6.0 集成开发环境中的 C 语言源程序文件,其扩展名为“c”。“cpp”是 C++ 程序的扩展名。

3. 编译和连接

图 1-5 表示了 C 语言的上机过程,源程序需经过编辑、编译和连接才能最终产生可执行文件。

① 编译源程序就是由 Visual C++ 6.0 集成开发环境提供的编译器将源程序代码转换成目标代码的过程。编译过程主要进行词法分析和语法分析,在分析过程中如果发现错误,将错误信息显示在屏幕上。经过编译后的目标文件的扩展名为“obj”。