



高职高专“十一五”规划教材

计算机类

C语言程序设计 项目化教程

王 侠 陈祥章 主编



冶金工业出版社
www.cnmp.com.cn

高职高专“十一五”规划教材·计算机类

C 语言程序设计 项目化教程

主 编 王 侠 陈祥章
副主编 韩永印 徐 涛
主 审 徐 超



北 京
冶金工业出版社
2009

内 容 简 介

本书力求突破高职高专旧的教学框架,用全新的方法组织编写。本书始终以应用为目的,从应用入手,采用项目教学和任务驱动的方式。书中内容精练,循序渐进,重点突出,易于理解。本书在体系结构安排上尽可能地将概念、知识点与实例结合起来,通过将学习内容设计成生活化和趣味化的活动思维过程,让学生在发现问题和解决问题的过程中,系统学习并掌握基础知识。本书将成绩管理等系统作为贯彻全书的内容,包括 C 语言基础知识、顺序程序设计、选择结构程序设计、循环结构程序设计、数组、函数、结构体、指针和课程设计 9 个项目。

本书实务性强,适合作为高职高专院校计算机专业的教材,也适合作为广大计算机编程爱好者学习 C 语言的自学参考书。

图书在版编目(CIP)数据

C 语言程序设计项目化教程/王侠,陈祥章主编. —北京:冶金工业出版社,2009.7

ISBN 978-7-5024-5033-5

I. C… II. ①王…②陈… III. C 语言—程序设计—高等学校:技术学校—教材 IV. TP312

中国版本图书馆 CIP 数据核字(2009)第 126160 号

出版人 曹胜利

地 址 北京北河沿大街嵩祝院北巷 39 号,邮编 100009

电 话 (010)64027926 电子信箱 postmaster@cnmip.com.cn

责任编辑 刘 源

ISBN 978-7-5024-5033-5

北京天正元印务有限公司印刷;冶金工业出版社发行;各地新华书店经销

2009 年 7 月第 1 版,2009 年 7 月第 1 次印刷

787mm×1092mm 1/16; 14.25 印张; 338 千字; 222 页; 1-3000 册

29.00 元

(本书如有印装质量问题,本社发行部负责退换)

前 言

近年来我国高等职业教育发展迅猛,目前,高等职业院校已占全国高等学校半数以上,高职学生数已超过全国大学生的半数。发展高职、培养大量技术型和技能型人才,是国民经济发展的迫切需要,是高等教育大众化的要求,是促进社会就业的有效措施。为了打破旧的框架,写出独具一格式的、能体现高职高专教育新理念和教学特点的教材,我们本着体系得当,循序渐进、台阶要小、分解难点,正确选择典型任务和项目,选好切入点,以及注重通俗易懂、实例丰富、易于理解的原则,在编写此书的过程中力求做到从实际到理论,从具体到抽象,从个别到一般,从零散到系统,培养学生的学习能力、工作能力和创造能力。

总结以往的C语言教材,发现普遍存在几个问题。一是过分讲语法,把简单的问题复杂化。二是太重系统性,长篇累牍,大段文字,致使书篇幅太厚,老师讲课时候不好运用。三是例题多是讲数学上的小知识,没有实际的项目,致使学生不知C语言到底是干什么,与现实生活有什么联系。四是一般前三章堆砌知识,第四章才能见到程序的影子。

本书在编写上突出了以下4个方面的特点:

(1) 精简教学内容。考虑到各学校的课时限制和高职高专学生的实际情况,内容上不贪多求全,合理舍去不常使用的内容,但对于C语言的基本内容予以细致的介绍,做到重点突出,易于理解。

(2) 项目为导向,任务为驱动。通过项目和任务培养学生学生分析问题、解决问题能力和团队协作精神,围绕项目和任务将语法和规则渗透教学中,增强课程内容与职业岗位能力要求的相关性。

(3) 教学重点难点适中,增加趣味性。采用简单易懂的实例和项目降低教学难度,强调实用性和趣味性,激发学生的学习积极性,并使学生在解决问题的过程中获得成就感,学习有信心。

(4) 项目贯穿整个知识点。将成绩管理等系统贯穿整本书,用此方法可使学生明确学习目标,尽早体会较大程序的编写过程。

本书是面向高职高专的学生的,建议教师在组织教学的时候,将重点放在程序结构的分析上,淡化枯燥的C语言程序设计语法规则上,着重培养学生的动手能力。

本书由王侠、陈祥章任主编,韩永印、徐涛任副主编,徐颖梅、车延雪、刘君、徐雅琴、刘海梅参加编写。全书由徐超统稿。

由于编者水平所限,书中如有不足之处敬请使用本书的师生与读者批评指正,以便修订时改进。如读者在使用本书的过程中有其他意见或建议,恳请向编者(bjzhangxf@126.com)踊跃提出宝贵意见。

编 者

目 录

项目一 C语言基础知识..... 1	项目三 选择结构程序设计..... 53
第一部分 任务学习..... 1	第一部分 任务学习..... 53
一、C程序的基本结构——设计 简单加法器..... 1	一、双分支选择语句——体重 测量仪改进版..... 53
二、C程序的编译与运行的过程 ——输出卡通画..... 4	二、单分支选择语句——身高 预测..... 56
三、常量和变量——简单数学 计算..... 9	三、逻辑运算符——电梯控制 程序..... 59
四、整型数据——简单数学计算..... 14	四、多分支选择语句 if...else——体重 测量仪改进版..... 63
五、实型数据——计算定期存款 本利之和..... 18	五、嵌套 if 语句——求解分段 函数..... 66
六、字符型数据——破译密码..... 20	六、switch 语句——模拟自动 饮料机..... 68
七、算术运算符及其表达式——求解 一元二次方程..... 24	七、switch 语句中多个 case 语句 相同情况的处理..... 72
八、赋值运算符..... 28	第二部分 项目学习..... 73
第二部分 项目学习..... 30	项目学习五 菜单设计..... 73
项目学习一 系统功能分析..... 30	第三部分 实验学习..... 75
项目学习二 数据结构定义..... 31	一、实验目的..... 75
第三部分 实验学习..... 33	二、实验条件..... 75
一、实验目的..... 33	三、实验内容..... 75
二、实验条件..... 33	项目四 循环结构程序设计..... 77
三、实验内容..... 33	第一部分 任务学习..... 77
项目二 顺序程序设计..... 34	一、while 语句——超市收费系统..... 77
第一部分 任务学习..... 34	二、for 语句——神探抓贼..... 81
一、普通输出语句——设计菜单..... 34	三、do...while 语句——密码 验证..... 85
二、格式输入输出语句——解决 一些简单的数学问题..... 36	四、break 语句和 continue 语句..... 89
三、格式输入输出语句——打印 成绩报表..... 40	第二部分 项目学习..... 91
四、综合运用 1——交换数据..... 42	项目学习六 信息录入..... 91
五、综合运用 2——编写一个 体重测量仪..... 44	项目学习七 成绩计算..... 92
六、单个字符输入输出函数的使用..... 46	第三部分 实验学习..... 93
第二部分 项目学习..... 47	一、实验目的..... 93
项目学习三 菜单设计..... 47	二、实验条件..... 93
项目学习四 信息录入和显示..... 49	三、实验内容..... 93
第三部分 实验学习..... 50	项目五 数组..... 96
一、实验目的..... 50	第一部分 任务学习..... 96
二、实验条件..... 51	
三、实验内容..... 51	

一、定义与引用—维数组——求超市商品利润..... 96	一、结构体变量的定义和引用——学生信息输入输出..... 160
二、数据统计——餐饮服务调查打分..... 100	二、结构体数组的定义和初始化——通信录..... 163
三、数据查询——查找成绩最高分..... 105	第二部分 项目学习..... 166
四、数据修改——修改学生成绩..... 107	项目学习十二 数据结构定义..... 166
五、数据插入与删除——插入学生信息..... 110	第三部分 实验学习..... 167
六、数据排序——商品销售排名..... 111	一、实验目的..... 167
七、二维数组——评出冠军..... 114	二、实验条件..... 168
第二部分 项目学习..... 117	三、实验内容..... 168
项目学习八 成绩排序..... 117	项目八 指针 169
项目学习九 信息查找..... 118	第一部分 任务学习..... 169
项目学习十 信息修改..... 119	一、认识变量的地址和指针变量——变量输出比较..... 169
第三部分 实验学习..... 120	二、指针变量作为函数参数——比较大小..... 172
一、实验目的..... 120	三、通过指针访问数组——求总分, 最大值, 最小值.. 174
二、实验条件..... 120	四、通过指针访问字符串——字符串复制..... 178
三、实验内容..... 121	第二部分 项目学习..... 182
项目六 函数 124	项目学习十三 学生成绩管理系统的指针实现... 182
第一部分 任务学习..... 124	第三部分 实验学习..... 188
一、函数的定义——绘制一个图形..... 124	一、实验目的..... 188
二、函数的一般调用——学生成绩分布显示..... 127	二、实验条件..... 188
三、函数的嵌套调用——弦截法求方程根..... 136	三、实验内容..... 188
四、函数的递归调用——猜年龄... 138	项目九 课程设计 190
五、变量作用域——求长方体表面积..... 141	一、课程设计的目的..... 190
六、变量存储类别——求长方体表面积(改)..... 144	二、课程设计的内容要求..... 190
七、函数和常用算法综合编程——学生成绩分析..... 147	三、课程设计的时间安排..... 191
第二部分 项目学习..... 154	四、课程设计成绩评定..... 191
项目学习十一 学生成绩管理系统..... 154	五、课程设计报告格式..... 191
第三部分 实验学习..... 158	六、课程设计的题目..... 191
一、实验目的..... 158	七、企业人事管理系统设计实例..... 196
二、实验条件..... 158	附录 A ASCII 码对照表 218
三、实验内容..... 158	附录 B 键盘常用 ASCII 码 219
项目七 结构体 160	附录 C 最常见的 20 种 VC++ 编译错误信息集 220
第一部分 任务学习..... 160	参考文献 222

项目一 C 语言基础知识

本项目主要可以分成三个部分。第一部分通过 C 程序实例分析，使读者从宏观上整体把握和了解 C 程序的基本结构、C 程序的开发调试过程，并且理解 C 语言基本数据类型的表示形式、存储格式及相关的运算，掌握变量定义的方法及意义，能够灵活运用各种表达式，为 C 语言编程打下基础，对 C 程序开发设计有较深入的感性认识，进而强化对计算机语言和程序设计的理解，为进一步学习 C 语言程序设计打下基础。第二部分是通过对项目的学习使学习者尽早了解编写应用程序的过程，逐步掌握编程的方法。第三部分通过实验学习加强动手编程的能力。

【学习目标】

- (1) 理解 C 程序的基本结构。
- (2) 掌握 C 程序的编译、连接与运行的过程。
- (3) 理解常量和变量的含义。
- (4) 熟悉基本的数据类型。
- (5) 掌握 C 语言各种运算符和表达式的应用。

第一部分 任务学习

一、C 程序的基本结构——设计简单加法器

【问题提出】

编写程序需要遵循一定的格式吗？是什么样的格式呢？

【工作任务】

阅读一个加法器程序，初步的认识 C 语言，并且理解 C 语言程序的基本结构。

【思路指导】

这是认识 C 语言的第一个程序，通过读程序要注意基本格式，标点符号，对齐方式几个方面，通过读第一个程序掌握一个 C 语言程序的基本结构。

【解决方案】

```
#include<stdio.h>          /*预处理命令*/
int sum(int x,int y)      /*定义 sum 函数*/
{
    int z;                /*声明定义变量 z*/
    z=x+y;                /*计算 z 等于 x 与 y 的和值*/
    return (z);          /*将 z 的值返回，通过 sum 带回调用处*/
}
main()                    /*主函数*/
{
    int a,b,s;            /*声明定义变量 a,b 和 s*/
    printf("请输入加数和被加数:\n"); /*提示输入两个数*/
    scanf("a=%d,b=%d",&a,&b); /*输入变量 a 和 b 的值*/
}
```

```

s=sum(a,b);          /*调用 sum 函数, 将得到的值赋给 s*/
printf("s=%d\n",s); /*输出变量 s 的值*/
}

```

【运行情况】

加法器程序运行结果如图 1-1 所示。

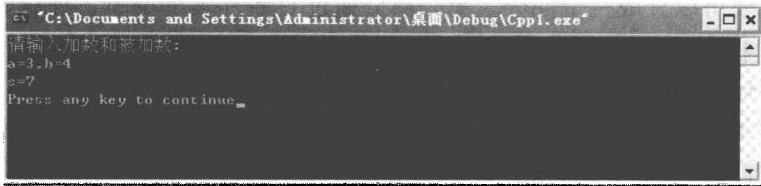


图 1-1 加法器程序运行结果

【归纳分析】

(1) 通过这个加法器程序, 可以把 C 程序的结构用一个表格表示出来, 见表 1-1。

表 1-1 C 程序的结构

预处理命令		#include<stdio.h>	
自定义函数 sum。除了主函数外, 在使用其他自定义函数前, 应对这些自定义函数进行函数原型说明		int sum(int x,int y)	
一对大括号内的是函数体, 计算两个数之和		{ int z; z=x+y; return (z); }	
主函数	函数首部	main()	
	函数体开始	{	
	定义变量	声明语句: 定义 3 个变量	int a,b,s;
	功能语句	输出“请输入加数和被加数:”	printf("请输入加数和被加数:\n");
		通过键盘输入两个数	scanf("a=%d,b=%d",&a,&b);
		调用函数	s=sum(a,b);
		输出两数之和 s	printf("s=%d\n",s);
函数体结束	}		

可以把加法器程序改写成只有主函数的实现方式, 学习者可按照上表分析一下这个程序的结构。

```

#include <stdio.h>          /*预处理命令*/
void main()                /*主函数*/
{
    int a,b,sum;           /*声明定义变量 a,b 和 sum*/
    printf("请输入加数和被加数:\n"); /*提示输入两个数*/
    scanf("a=%d,b=%d",&a,&b); /*输入变量 a 和 b 的值*/
    sum=a+b;              /*计算 sum 等于 a 与 b 的和值*/
    printf("sum=%d\n",sum); /*输出变量 sum 的值*/
}

```

通过上面两个加法器程序可以看出: 任何函数(包括主函数 main())都是由函数说明和函数体两部分组成。其一般结构如下:

```

[函数类型] 函数名(函数参数表) /*函数说明部分*/
{
    说明语句部分;              /*函数体部分*/
}

```


功能语句部分;

}

(2) `#include<stdio.h>`这一行是预处理命令,以`#`开头的语句称为预处理命令,也可以写成`#include"stdio.h"`,通常还会有`#include"math.h"`和`#include"string.h"`等。通过观察这个程序可以看到在程序中用到了库函数 `printf`(输出函数)和库函数 `scanf`(输入函数),C语言系统把所有的输入输出函数的信息都存放在`#include<stdio.h>`文件中,所以在使用这两个函数的时候要用预编译命令`#include`将 `stdio.h` 这个“头文件”(`stdio.h`, `math.h` 和 `string.h` 等文件通常放在程序的开头,称为头文件,扩展名 `h` 是 `head` 的缩写)包含到用户源文件中。程序中用到 `printf` 函数和 `scanf` 函数,如果在写程序的时候没有出现`#include<stdio.h>`这一行预处理命令,运行的时候将会出现如下的错误提示: `error C2065: "printf": undeclared identifier, error C2065: "scanf": undeclared identifier`。虽然C语言提供了丰富的标准库函数可以让使用者直接使用,但是在使用之前必须在程序的开头加上预处理命令。

(3) 通过观察这个程序大家可以看到有 `main` 函数、`sum` 函数、`printf` 函数和 `scanf` 函数,所以说函数是C语言程序的基本单位,并且一个C程序必须要有一个 `main()` 函数。在程序运行的过程中一个C程序必须从 `main()` 函数开始(不管 `main` 函数是放在什么位置)并且以 `main()` 函数结束。`main()` 函数可以返回一个值,也可以不返回值,如果某个函数没有返回值,那么在它的前面有一个关键字 `void`。

(4) 学习者在初学写程序的时候,要先了解C程序书写的格式,大家可以从以下几个方面注意:

1) C程序的每一条语句后都必须加分号“;”,“;”代表本条语句结束。

2) C程序的书写格式自由,对于短语句可以多条语句书写在同一行,也可以将长语句分写在多行,但需要注意,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。虽然书写格式是自由的,但是还是提倡一行一条语句的风格,这样可以提高程序的可读性。

3) 配对使用的符号不要出现遗漏,例如注释符号、函数体的起止标识符(花括号)和复合语句的花括号等。

4) 关键语句可以使用注释,但注释不是必需的,C系统不对注释符进行编译。C语言的注释格式为: `/*.....*/`。“`/*`”和“`*/`”必须成对使用,且“`/*`”和“`*/`”以及“`/*`”和“`/*`”之间不能有空格,否则都会出错。注释的位置,可以单占1行,也可以跟在语句的后面。

5) 根据语句的从属关系,程序书写时采用缩进格式,使程序语句的层次结构清晰,提高程序的可读性。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。

【技能训练】

分析下面程序的结构并和加法器程序进行比较。

输入两个整数,计算两者中较大的数,并输出。

```
#include <stdio.h>          /*预处理命令*/
void main()                 /*主函数*/
{                             /*main函数体开始*/
    int a,b,c;              /*声明部分定义变量*/
    scanf("%d,%d",&a,&b);    /*scanf是输入函数,&是"取地址"*/
    c=max(a,b);            /*调用max,将调用结果赋给c*/
```


④ 系统显示“新建工程信息”对话框，单击“确定”按钮，完成工程项目的创建，如图 1-4 所示。



图 1-2 建立新工程项目“新建”对话框

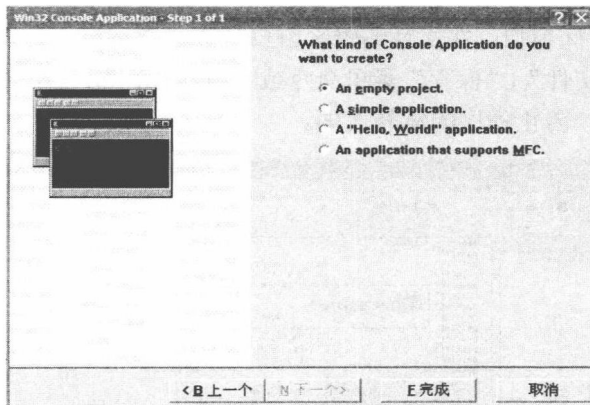


图 1-3 Win32 Console Application Step 1 of 1 对话框

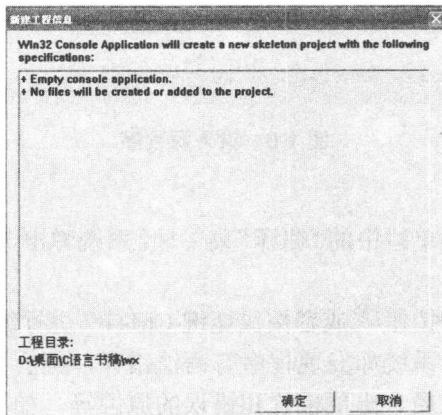


图 1-4 “新建工程信息”对话框

2) 建立新项目中的文件。

① 选择“文件”|“新建”菜单命令，弹出“新建”对话框。

② 选择“文件”选项卡，单击 C++ Source File 选项，在“文件”文件框中输入文件名，单击“确定”按钮，如图 1-5 所示。系统自动返回 Visual C++ 6.0 主窗口。

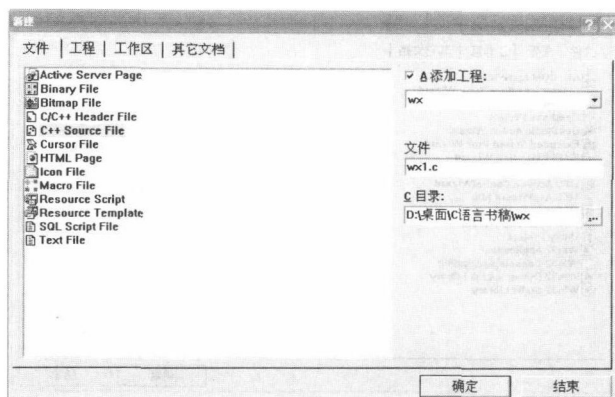



图 1-5 建立新项目文件“新建”对话框

③ 显示文件编辑区窗口，在文件编辑区窗口中输入源程序文件，如图 1-6 所示。编辑完源程序之后选择“文件”|“保存”菜单命令或者直接单击  按钮保存。在编辑程序的时候要及时地进行保存，防止程序的意外丢失。

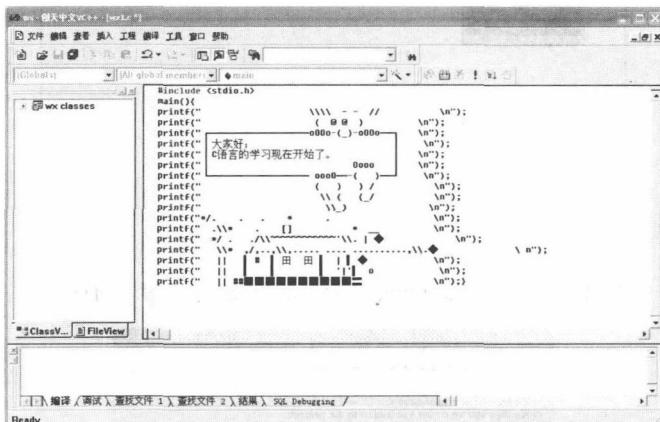



图 1-6 输入源程序

(3) 编译。

方法一 选择主窗口菜单栏中的“编译”菜单项；系统弹出下拉菜单，选择“编译 wx1.c”菜单命令。

方法二 直接单击编译按钮  或者按快捷键 Ctrl+F7 进行编译。

注意：在编译过程中，系统如发现程序有语法错误，则会在输出区窗口中显示错误和警告，给出错误和警告的性质、出现位置和错误的原因等。如果双击某条信息，编辑区窗口左侧出现一个箭头，指示错误的程序行。用户可据此对源程序进行相应的修改，并重新编译，直到通过为止。

如图 1-7 所示，这个程序在编译的过程中出现一个错误，双击错误信息的时候会看到编辑区窗口左侧出现一个箭头，值得注意的是如果当前语句缺少分号，运行环境会将箭头

指向缺少分号语句的下一行。在输出区窗口可以看到错误的原因是缺少分号，对程序重新进行修改，直到没有错误。

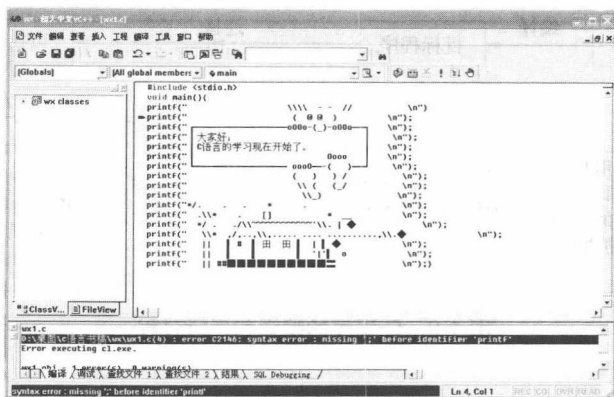


图 1-7 错误信息指示

(4) 连接。


方法一 选择主窗口菜单栏中的“编译”菜单项；系统弹出下拉菜单，选择“构件 wx1.exe”菜单命令。

方法二 直接单击连接按钮或者按快捷键 F7 进行连接。

如果在连接的过程中，系统发现错误，将会在输出窗口显示所有的错误和警告信息，处理的方法和编译的过程一样，继续进行修改直到没有错误信息为止。

(5) 运行。

方法一 选择主窗口菜单栏中的“编译”菜单项；系统弹出下拉菜单，选择“执行 wx1.exe”菜单命令。

方法二 直接单击运行按钮或者按快捷键 Ctrl+F5 进行运行。

运行之后会出现一个黑屏显示结果，按任意键就会回到编辑窗口。

【运行情况】

卡通画程序运行结果如图 1-8 所示。

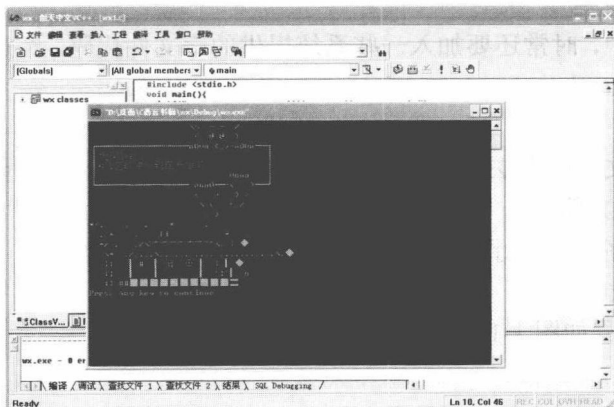


图 1-8 卡通画程序运行结果

【归纳分析】

(1) 了解 C 程序的编译和运行的过程, 如图 1-9 所示。

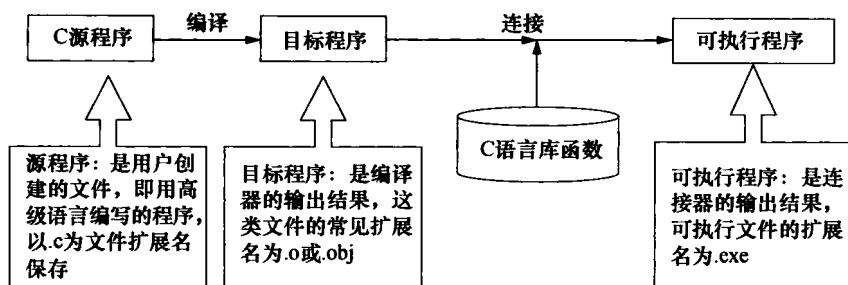


图 1-9 C 程序的编译和运行过程

1) 编辑。

① 编辑是指将编写好的 C 语言源程序(用高级语言或者汇编语言编写的程序称为源程序)代码录入到计算机中, 形成源程序文件, 源文件的编辑可以用任何文字处理软件完成。源程序不能直接在计算机上执行, 需要用“编译程序”将源程序翻译为二进制形式的代码。计算机只能识别和执行 0 和 1 组成的二进制的指令, 而不能识别用高级语言编写的指令。

② 常用的编译器有 Turbo C 2.0、Turbo C++ 3.0 和 Visual C++ 6.0。在本书中用 Visual C++ 6.0 集成环境对 C 程序进行编辑、编译、连接和运行。

2) 编译。

① 编译源程序就是由 C 系统提供的编译器将源程序文件的源代码转换成目标代码的过程。

② 编译过程主要进行词法分析和语法分析, 如果在分析过程中如果发现错误, 会将错误信息显示在屏幕上通知用户。经过编译后的目标文件的扩展名为.obj。目标代码尽管已经是机器指令, 但是还不能运行, 因为目标程序还没有解决函数调用问题, 需要将各个目标程序与库函数连接, 才能形成完整的可执行的程序。

3) 连接。

① 连接过程是将编译过程中生成的目标代码进行连接处理, 生成可执行程序文件的过程。

② 在连接过程中, 时常还要加入一些系统提供的库文件代码。经过连接后生成的可执行文件的扩展名为.exe。

4) 运行。

① 运行可执行文件的方法很多, 可在 C 系统下执行“运行”命令, 也可以在操作系统下直接执行可执行文件。

② 可执行的程序文件运行后, 将在屏幕上显示程序执行的结果。

(2) C 程序的开发过程。

C 程序的开发过程如图 1-10 所示。

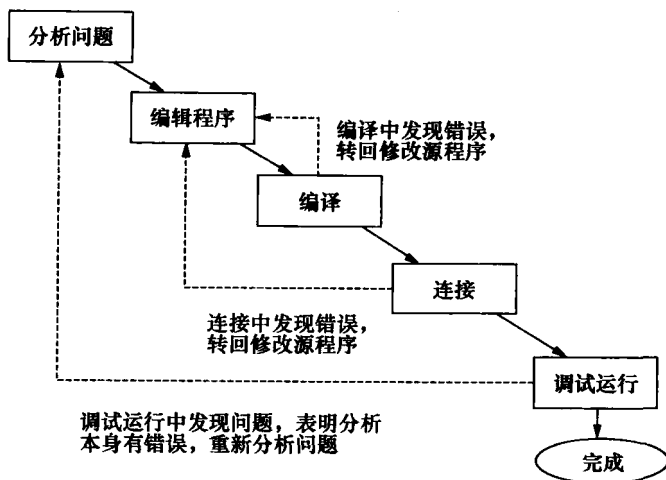


图 1-10 C 程序的开发过程

【技能训练】

用 Visual C++ 6.0 环境运行下面程序。在屏幕上显示两个短句“Programming is fun.”和“ And Programming in C is even more fun!”，每行显示一句。

```

#include <stdio.h>
void main()
{
    printf("Programming is fun.\n");
    printf(" And Programming in C is even more fun!\n");
}

```

三、常量和变量——简单数学计算**【问题提出】**

理解常量和变量的概念。

【工作任务】

已知圆半径 r ，求圆周长 c 和圆面积 s 的值。

【思路指导】

(1) 对于初学者，可以把所有的数据进行详细的分析，见表 1-2。

表 1-2 数据分析

名称	作用	数据类型	变量/常量	算法
r	半径：需要输入的数据	整型	变量	<code>scanf("%d",&r);</code>
c	周长：需要输出的数据	实型	变量	<code>c=2*PI*r;</code>
s	面积：需要输出的数据	实型	变量	<code>s=PI*r*r;</code>
PI	计算中用到的圆周率	实型	常量	<code>#define PI 3.1416</code>

(2) 流程图。

程序设计流程图如图 1-11 所示。

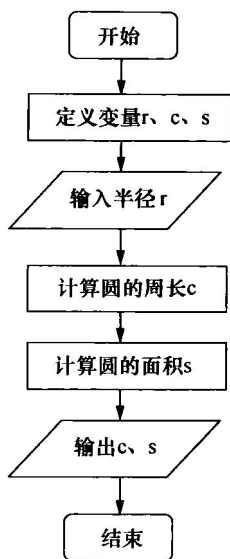


图 1-11 设计流程图——常量和变量

【解决方案】

```
#include<stdio.h>
#define PI 3.1416
void main()
{
    int r;
    float c,s;
    printf("请输入圆的半径:");
    scanf("%d",&r);
    c=2*PI*r;      /*编译时用 3.1416 替换 PI*/
    s=PI*r*r;     /*编译时用 3.1416 替换 PI*/
    printf("c=%6.2f,s=%6.2f\n",c,s);
}
```

【运行情况】

求圆周长和圆面积的程序运行结果如图 1-12 所示。

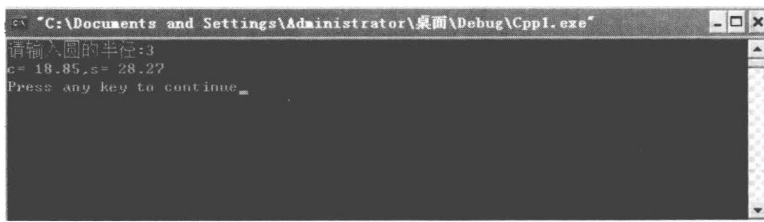


图 1-12 求圆周长和圆面积的程序运行结果

【归纳分析】

- (1) 常量是在程序运行过程中值保持不变的量，常量不占用存储空间。如 12、0、-3 为整型常量，4.6、-1.23 为实型常量，'a'、'd' 为字符常量。
- (2) 在 C 语言中可以用标识符来表示常量名、变量名、函数名、类型名和文件名等。

标识符是由字母、数字、下划线组成且由字母或下划线开头的字符串。如 `_sum`、`sum`、`stu_name` 和 `price` 等为合法的标识符。通常采用具有一定意义的英文单词、缩写作为标识符，做到“见名知意”。例如，`name/xm`(姓名)、`sex/xb`(性别)、`age/nl`(年龄)、`salary/gz`(工资)。

在使用标识符的时候有几点要注意：1)C语言区分大小写。如 `Price` 和 `price`，系统会认为是两个不同的标识符。2)关键字不能用作标识符。如 `int`、`printf` 不可以作为标识符，因为它们已经有了固定的含义。C语言有 32 个关键字。数据类型关键字：`auto`、`char`、`double`、`extern`、`float`、`int`、`long`、`register`、`short`、`static`、`struct`、`typedef`、`union`、`unsigned`。控制语句关键字：`break`、`case`、`continue`、`default`、`do`、`else`、`for`、`goto`、`if`、`return`、`switch`、`while`。

(3) 本程序中 `#define PI 3.1416` 语句执行的功能是把 3.1416 这个常量用 `PI` 来表示，即用一个标识符代表一个常量。这时，此标识符与此常量是等价的，通常把这类常量称为符号常量。格式为：`#define 标识符 数据`，如例子中 `#define PI 3.1416`，在程序中所有的 `PI` 的值都是 3.1416。在编译时，将 `PI` 自动替换成 3.1416。符号常量其实是一个宏(micro)，这个过程称为宏替换(micro substitution)。`#define PI=3.14` 是错误的。

(4) 变量是指在程序运行过程中，其值可以被改变的量，其实就是一块用于存放数据的内存单元，该内存单元的值是可以改变的量。为什么要使用变量呢？编写程序时，常常需要将数据存储在内存中，方便后面使用这个数据或者修改这个数据的值，使用变量可以引用存储在内存中的数据，并随时根据需要显示数据或执行数据操作。变量有 3 个基本要素：变量名、变量数据类型和变量的值，变量命名遵循标识符命名规则。

(5) 变量的实质是内存中的一个存储单元，每个变量在计算机中对应相应长度的存储空间，因此在使用变量前应向系统申请存储单元，这一过程就是定义变量的过程。变量定义语句可以出现在变量使用之前的任何位置，程序设计时不违背“先定义，后使用”的原则即可。该程序中的 `int r;`和 `float c,s;`就分别定义了变量 `r`、`c` 和 `s`。所以变量定义的实质是按照变量说明的数据类型为变量分配相应空间的存储单元，即变量在使用之前首先定义它的名字，并说明它的数据类型。

变量定义的一般格式为：数据类型 变量名 1,变量名 2,……,变量名 n;，例如：

```
int r;
float radius, length, area;
```

在使用变量的时候有几点需要注意：1)允许在一个类型说明符后，说明多个相同类型的变量，各变量名之间用逗号间隔，类型说明符与变量名之间至少用一个空格间隔。2)最后一个变量名之后必须以“;”结尾。3)变量说明必须放在变量使用之前，一般放在函数体的开头部分。

(6) 在定义变量的同时进行赋初值的操作为变量初始化。变量初始化的一般格式为：数据类型 变量名 1=初值 1,变量名 2=初值 2,……,变量名 n=初值 n;。

在定义的同时对部分变量赋初值：

```
float radius=2.5, length, area;
```

在定义的同时对全部变量赋初值：

```
float radius=2.5, length=2.5, area=2.5;(正确)
float radius=length= area=2.5;(错误)
```