高等学校计算机专业教材精选·算法与程序设计

C程序设计案例教程

林小茶 编著

清华大学出版社

高等学校计算机专业教材精选·算法与程序设计

C程序设计案例教程

林小茶 编著

清华大学出版社
北 京

内 容 简 介

案例教学是学生们喜闻乐见的一种教学方式，但是，将案例教学应用于程序设计还是有一定难度的。本书对这种方式进行了有益的尝试，在不违反教学规律的情况下，先给出案例，然后进行说明和讲解。在内容的编排上，则更多地考虑了初学者的要求；在选择实例时，尽量选择能够解决实际问题的实例。

本书主要内容包括认识C语言、顺序结构程序设计、选择结构程序设计、基础知识深化、循环结构程序设计、函数、数组、指针、结构体、联合体与枚举以及文件等。

本书既适合作为大学低年级需要掌握一门程序设计语言的学生教材，也适合作为C语言自学者的教材或参考书。

图书在版编目(CIP)数据

C程序设计案例教程/林小茶编著. —北京：清华大学出版社，2015
高等学校计算机专业教材精选·算法与程序设计
ISBN 978-7-302-37932-4

Ⅰ. ①C… Ⅱ. ①林… 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207771号

责任编辑：张　民　薛　阳
封面设计：傅瑞学
责任校对：时翠兰
责任印制：何　芊

出版发行：清华大学出版社
网　　址：http://www.tup.com.cn，http://www.wqbook.com
地　　址：北京清华大学学研大厦A座　　**邮　　编**：100084
社 总 机：010-62770175　　**邮　　购**：010-62786544
投稿与读者服务：010-62776969，c-service@tup.tsinghua.edu.cn
质量反馈：010-62772015，zhiliang@tup.tsinghua.edu.cn
课件下载：http://www.tup.com.cn,010-62795954
印 刷 者：北京富博印刷有限公司
装 订 者：北京市密云县京文制本装订厂
经　　销：全国新华书店
开　　本：185mm×260mm　　**印　　张**：19.25　　**字　　数**：445千字
版　　次：2015年4月第1版　　**印　　次**：2015年4月第1次印刷
印　　数：1～2000
定　　价：35.00元

产品编号：039597-01

前　言

尽管有不少同行认为C语言作为程序设计的入门语言已经有些过时，但是事实上很多人仍以其为入门的程序设计语言，除了C语言的众多优点，最主要的还是因为C语言的实用性。众所周知的操作系统UNIX、MS-DOS、Microsoft Windows及Linux等都是用C语言编写的。C语言具有高效、灵活、功能丰富、表达力强和移植性好等特点。

本书在内容的编排上主要考虑如下几点。

第一，突出案例讲解的方法。本书采取的写作方法是：首先给出案例，然后再逐步表述其中牵涉的概念和思想。例如，在第3章的开始给出了求一个圆的面积的程序段：

```
if(r>=0)
    printf("面积=%lf\n",PI*r*r);                    /*输出圆的面积*/
else
    printf("半径输入错误!");                        /*提示用户输入错误*/
```

告诉读者这是一种最简单的分支语句，并不仔细研究其语法。有点英语常识的读者就能看懂这段程序，此时只要了解这是分支语句就已足够。随后才会逐步深化分支语句的使用。

第二，案例的选择符合初学者的要求。如果案例太复杂，会给初学者带来困扰。将前面求一个圆的面积的例子稍加扩充，可以演变成循环语句的简单案例：

```
#include "stdio.h"                                  /*求10个圆的面积*/
#define PI 3.141596
int main()
{   int i,r;                                        /*定义变量i*/
    i=1;                                            /*设i的初值为1*/
    while(i<=10)                                    /*i小于等于10时,做循环*/
    {   printf("请输入半径:");
        scanf("%d",&r);                             /*接收半径*/
        if(r>=0)
            printf("面积=%lf\n",PI*r*r);            /*输出圆的面积*/
        else
            printf("半径输入错误!");                /*提示用户输入错误*/
        i++;                                        /*i的内容增值1*/
    }
    return 0;
}
```

第三，强调如何编写好的程序。在本书的很多地方强调要努力编写一个好的程序，而不要花心思在一些小的程序设计技巧上。例如，告诫学习者避免使用像i++++++i这样的表达式，而不是花大量的篇幅去分析这个表达式到底等于多少，资深程序员绝对不会这样

用，避免给自己和阅读程序的人带来困惑。类似地，本书在介绍运算符的优先级和结合性时也提出使用括号表示运算顺序是最好的方法，而不要求学习者去记忆每个运算符的优先级。

第四，与大部分同类教材不同，提前了对文件内容的讲解。在第5章循环结构程序设计中第一次加入了对文件的介绍，目的是尽早地提出文件的概念。因为大部分教材将文件内容放在最后，在教学过程中，由于学时有限，文件内容就被放弃了。本书尝试将文件的内容提前，尤其适合对文件的使用要求比较高的专业。

作为本书的姐妹篇，将同时出版本教程的习题解答和实验指导书，给出本教材中所有习题的参考答案，供读者学习时借鉴和参考。

编者水平有限，书中不足在所难免，敬请广大读者批评指正。

编　者

2014年12月于北京

目　录

第1章　认识C语言

首先，通过三个最简单的程序来认识C语言。

1.1　C语言源程序的基本结构

1.1.1　“欢迎”等三个源程序

例1.1　在屏幕上显示“欢迎”字样。

```
/*-------------------------显示“欢迎”-------------------------*/
#include "stdio.h"
int main()
{
    printf("欢迎!\n");                    /*调用库函数显示*/
    return 0;
}
```

运行结果：

```
欢迎!
```

例1.2　“泰囧”电影票50元一张，编写一个程序计算买58张票需要花多少钱。

```
/*-------------------------“泰囧”电影票计算---------------------*/
#include "stdio.h"
int main()
{
    int count,sum;                          /*定义两个整型变量*/
    count=50;                               /*50赋值给count*/
    sum=count*58;                           /*计算结果并赋值给sum*/
    printf("总共需要花 %d 元钱!\n",sum);      /*显示计算结果*/
    return 0;
}
```

运行结果：

```
总共需要花 2900 元钱!
```

例1.3　假设有两个单位(如信息管理学院和计算机学院)需要分别买电影票，两个单位的人数分别为58和76。编写一个程序计算两个单位分别需要花多少钱。

```
/*---------"泰囧"电影票计算(使用函数)-----*/
#include "stdio.h"
int fun_sum(int x);                    /*自定义函数说明*/
int main()                             /*main函数定义*/
{   printf("信息管理学院总共需要花 %d元钱!\n",fun_sum(58));
    printf("计算机学院总共需要花 %d元钱!\n",fun_sum(76));
    return 0;
}
int fun_sum (int x)                    /*(自定义)求一个单位的总票价*/
{   return 50*x;                       /*返回计算结果*/
}
```

运行结果：

```
信息管理学院总共需要花 2900 元钱!
计算机学院总共需要花 3800 元钱!
```

1.1.2 关于程序的基本概念

1.1.1 节中已经展示了三个 C 语言源程序，那么究竟什么是程序呢？

对于计算机来说，程序就是由计算机指令构成的序列。计算机按照程序中的指令逐条执行，就可以完成相应的操作。更准确一点，计算机执行由指令构成的程序，对提供的数据进行操作。例如，可以编写程序计算买“泰囧”电影票需要花多少钱。

计算机程序的操作对象是“数据”。这里的数据不是简单的阿拉伯数字，而是包括各种现代计算机能够处理的字符、数字、声音、图像等。

实际上计算机自己不会做任何工作，它所做的工作都是由人们事先编好的程序来控制的。程序需要人来编写，就像例 1.1～例 1.3，是人们编写好的三个小程序，使用的工具就是程序设计语言，当然，此处使用的是 C 语言。不同的程序设计语言书写的方法是不同的，就像中国人说中文，美国人说英语一样，但是可以表达相同的意思。

目前，通用的计算机还不能识别自然语言，而只能识别特定的计算机语言。

计算机语言一般分为高级语言和低级语言。C 语言属于高级语言。

高级语言是一种比较接近自然语言和数学语言的程序设计语言。例如，“count * 58”就是计算两个数的乘积，只是由于计算机的键盘上没有乘号，用星号代替了！

低级语言直接依赖计算机硬件，不同的机型所使用的低级语言是完全不一样的。高级语言则不依赖计算机硬件，因此，需要在高级语言和低级语言之间搭建一个桥梁，从高级语言到机器语言要经过编译程序进行“翻译”，而高级语言几乎为每一种机器都创建了各自的编译程序，从而可以将用高级语言编写的程序几乎不加修改地运行在不同的计算机平台上。

编译程序分为两种，一种是解释系统，另一种是编译系统。解释系统是对高级语言编写的程序翻译一句执行一句；而编译系统是将高级语言编写的程序文件全部翻译成机器语言，生成可执行文件以后再执行。高级语言几乎在每一种机器上都有自己的编译程序。C 语言的编译程序属于编译系统。

1.1.3 源程序基本结构学习

下面通过对例 1.1～例 1.3 进行说明，帮助读者了解 C 程序的基本组成。

(1) 例 1.1～例 1.3 中的第一行，分别是：

```
/*-------------------------显示“欢迎”--------------------------*/
/*------------“泰囧”电影票计算-------------------------*/
/*--------“泰囧”电影票计算(使用函数)-----*/
```

用/*和*/括起来的是注释行。注释行用于说明程序的功能和目的，如果想做一个好的程序员，必须习惯为程序写出详细的注释。按照惯例，一般要在程序的最开始说明整个程序的目的和功能，并在必要时，为每一行代码写出注释，以增加可读性。注意，使用/*和*/括起来的语句并不一定在一行，可以是多行。

“/*调用库函数显示*/”也是注释行，说明“printf("欢迎!\n");”语句是在屏幕上显示“欢迎!”字样。

(2) “#include "stdio.h""是预处理命令，凡是以#开始的语句都是预处理命令。这些命令是在编译系统翻译代码之前需要由预处理程序处理的语句。“#include"stdio.h""”语句是请求预处理程序将文件 stdio.h 包含到程序中来，作为程序的一部分。文件 stdio.h 中是一些重要的定义，没有它，“printf("欢迎!\n");”语句不能通过编译系统的“翻译”。也就是说，每个 C 语言的源程序都必须包含“#include "stdio.h""”语句。

(3) 每个 C 程序都必须包含一个主函数 main()，也只能包含一个主函数。用{}括起来的部分是一个程序模块，在 C 语言中也称为分程序，每个函数中都至少有一个分程序。C 程序的执行是从主函数中的第一句开始，到主函数中的最后一句结束。

(4) 分号“;”是 C 语言的执行语句和说明语句的结束符。

(5) C 语句在书写上采用自由格式。书写 C 语句时不含行号，不硬性规定从某列开始书写，但是好的程序员应该学会使用缩进格式，例如“printf("欢迎!\n");”语句在 main 函数内部，书写时不能与 main 对齐，而是向右移动了几格。

(6) C 语言的关键字和特定字使用小写字母。main 是关键字，include 是特定字，都必须用小写。

(7) printf 是 C 语言提供的标准输入输出库函数，它的功能是将用两个双引号括起来的内容“欢迎!”输出到标准输入输出设备显示器上，并输出一个换行。

因此例 1.1 的运行结果是：

```
欢迎!
```

练习 1.1 请修改下列程序使之能通过编译：

```
/*-------------------------显示“欢迎”--------------------------
int Main()
{
    printf("欢迎!\n")                    /*调用库函数显示*/
    return 0;
}
```

错误分析：第一个错误是注释行缺少了“*/”；第二个错误是主函数的名字拼写错误，经常有初学者犯错误，认为大小写都行，恰恰相反，C 编译对大写和小写是非常敏感的，Main 中的 M 应该小写；第三个错误是“printf("欢迎!\n")”一句缺少了分号。

(8) 例 1.2 中“int count,sum;”语句定义两个整型变量 count 和 sum，人们称之为变量的数据类型定义。那么变量是什么呢？变量是由程序命名的一块计算机内存区域，用来存储一个可以变化的数值。

图 1.1 显示的是“int count,sum;”语句定义的两个变量。每个变量保存的是一个特定的数据类型的数值，这两个存储空间的数据类型为整型，通常使用整数，如 2,10,1000 都是整数，int 是类型说明符，后面还会学到 char、float、double 等类型说明符号。C 语言中规定，任何变量都要经过数据类型的定义，以便在程序运行时分配相应的存储空间。也就是说，有了定义语句“int count, sum;”，程序运行时才会有图 1.1 显示的两个空间。

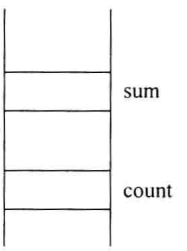

图 1.1　内存示意图

(9) 例 1.2 中的 58 和 50 以及“return 0;”中的 0 都是直接常量。直接常量又称无名常量或文字常量。常量是在程序执行过程中不会变化的数值，直接常量就是在代码中直接书写的数值，没有名字。

(10) 例 1.2 中使用了赋值运算符＝。

```
count=50;                          /* 50 赋值给 count */
sum=count * 58;
```

注意：这里的“＝”与数学上的等号在概念上完全不同。“count＝50;”表示将 count 中的内容变为 50，“sum＝count * 58;”表示将 sum 中的值变为 count 的值乘以 58，由于前一条语句已经将 count 的值设置为 50，所以 sum 中的值等于 50 乘以 58，即 2900。

“＝”的功能是将右边表达式的值送到左边的变量中，使表达式的值中的内容与常量相等。

(11) 例 1.2 中还使用了运算符 * 。 * 是 C 语言的算术运算符“乘号”，因为键盘上没有数学乘号，只好用星号代替，“sum＝count * 58;”表示将 count 乘以 58，并将结果赋值到 sum 变量中。

(12) printf 是一个标准输出函数。它执行格式化输出，其格式是：

```
printf("格式信息", 数据参数 1, 数据参数 2,…);
```

其中，数据参数可有可无。

例 1.1 中有语句“printf("欢迎!\n");”，是直接在屏幕上显示“欢迎!”，然后换行，“\n”表示换行。这条语句没有数据参数，直接显示双引号里的内容。只不过，冠以斜杠“\”的字符是一些特殊的字符，\n 表示换行。

例 1.2 中有语句“printf("总共需要花 %d 元钱!\n",sum);”，与例 1.1 相比，除了%d，其他的内容也是要直接显示的。“%d”是转换说明，它规定了后面的数据参数显示的格式为十进制输出，也就是规定显示 sum 时用十进制格式。而如果数据参数有两个，则转换说明也应该有两个。

练习 1.2　如果希望例 1.2 的显示结果是

```
50 * 58=2900
```

应该如何修改程序呢？

将 printf("总共需要花 %d 元钱!\n",sum);语句改为

```
printf("%d * %d=%d!\n",count,58,sum);
```

这里，由于显示了三个数，需要有三个转换说明和三个数据参数。

(13) 例 1.3 中定义了两个函数，主函数 main 及 fun_sum 函数。一般情况下，除了 main 函数以外，其他由程序员定义的函数被称为自定义函数，因此例 1.3 中的自定义函数是 fun_sum。

fun_sum 函数的功能是根据不同的人数计算一个单位的总票价，main 函数中有两次对 fun_sum 函数的调用，一次是“printf("信息学院总共需要花 %d 元钱!\n",fun_sum(58));”，另一次是“printf("计算机学院总共需要花 %d 元钱!\n",fun_sum(76));”。

注意：程序中函数的排列顺序并不决定函数的执行顺序，执行顺序是通过函数调用来决定的。

本例中对自定义函数 fun_sum 进行了函数定义、函数调用和函数说明，这些内容将在专门的章节讨论，读者不必为看不懂例 1.3 中的细节而烦恼。

1.2 程序的调试

1.2.1 调试步骤

C 语言的编译程序属于编译系统。要完成一个 C 程序的调试，必须经过编辑源程序、编译源程序、连接目标程序和运行可执行程序 4 个步骤。简单一点，可将 4 个阶段称为编辑、编译、连接、运行。

C 的源程序就是符合 C 语言语法的程序文本文件，文本文件又称为源程序文件，扩展名为.c。许多文本编辑器都可以用来编辑源程序，例如 Windows 的写字板、Windows 的记事本以及 Word 等，要注意的是 C 源程序的存储格式必须是文本文件，在保存的时候要选择文本文件格式。

编辑完成以后是编译，对编辑好的文本文件进行成功编译后将生成目标程序，目标程序文件的主文件名与源程序的主文件名相同，扩展名是.obj。编译程序的任务是对源程序进行语法和语义分析，若源程序的语法和语义都是正确的，才能生成目标程序，否则，应该回到编辑阶段修改源程序。

编译成功以后，目标文件依然不能运行，需要将目标程序和库函数连接为一个整体，从而生成可执行文件。可执行文件的扩展名是 exe。

最后一步就是运行可执行文件了，可执行程序要装入内存执行。如果在运行过程中发现可执行程序不能达到预期的目标，必须重复“编辑、编译、连接、运行”4 个步骤。

目前，大多数的 C 语言编译程序都将这 4 个功能集成在一个全屏幕环境下，给程序员的调试工作带来了极大的方便。

调试过程如图 1.2 所示。

1.2.2 在 Visual C++ 6.0 调试环境下调试第一个程序

本教材的所有程序使用 Visual C++ 6.0 调试，在此有必要简单地介绍一下使用 Visual C++ 6.0 调试 C 程序的步骤和方法。尽管 Visual C++ 6.0 是 C++ 的版本，但是 C++ 是在 C 语言的基础上扩展的，所以 C 程序也能够在该环境下正确调试。

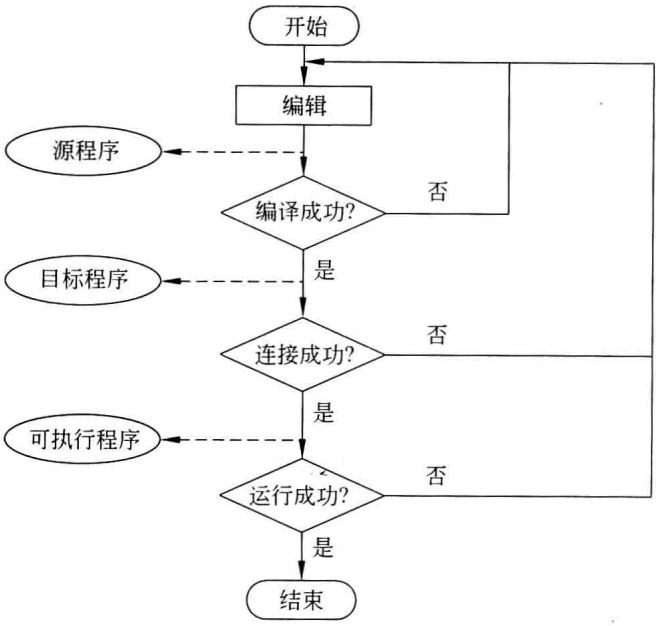

图 1.2　C 程序调试过程示意图

Visual C++ 6.0 是全屏幕编辑环境，编辑、编译、连接、运行都可以在它的控制下完成。

(1) 第一步：启动 Visual C++ 6.0。

在 Windows 环境下单击【开始】按钮，然后选择弹出菜单中的【程序】→Microsoft Visual Studio 6.0→Microsoft Visual C++ 6.0 命令。

(2) 第二步：建立一个新的工作空间。

选择主菜单中 File 中的 New(快捷键 Ctrl + N)命令，调出 New 对话框，并在该对话框中单击 Workspaces 标签，然后在右边的工作区命名框中输入要建立的工作区的名字(例如 MyWorkspace)，并单击 OK 命令按钮，如图 1.3 所示。新的工作区被建立以后，就作为用户当前的工作区。

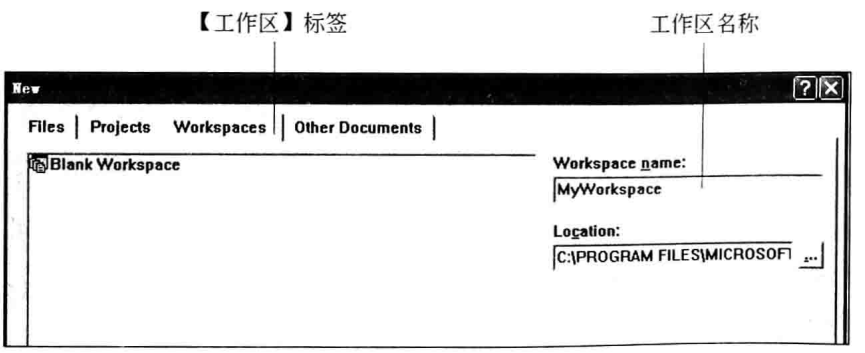

图 1.3　建立工程工作区的 New 对话框

(3) 第三步：建立一个新的工程。

选择主菜单中 File 中的 New 命令，调出 New 对话框，并在该对话框中单击 Projects 标签，在所列出的若干类型的“工程”中选择 Win32 Console Application，然后在右边的工程命名框中输入要建立的工程名(例如 MyProject)，并选择 Add to current workspace 单选钮，