21世纪高等学校计算机规划教材

21st Century University Planned Textbooks of Computer Science

C语言程序设计与应用

Programming and Application of The C Language

张小东 郑宏珍 主编

- 全面的基础圈点，轻松构筑程序框架
- 精要的解析方法，平滑实现基础转换
- 生动的案例分析，提高探索创新技能

高校系列

人民邮电出版社
POSTS & TELECOM PRESS

21世纪高等学校计算机规划教材

21st Century University Planned Textbooks of Computer Science

C语言程序设计与应用

Programming and Application of The C Language

张小东 郑宏珍 主编

高校系列

人民邮电出版社
北京

图书在版编目（CIP）数据

C语言程序设计与应用 / 张小东，郑宏珍主编. —北京：人民邮电出版社，2009.4
21世纪高等学校计算机规划教材
ISBN 978-7-115-19319-3

Ⅰ. C… Ⅱ. ①张…②郑… Ⅲ. C语言－程序设计－高等学校－教材 Ⅳ. TP312

中国版本图书馆CIP数据核字（2009）第016014号

内 容 提 要

本书是以最基本的工程实践为基础，以教育部考试中心最新公布的全国计算机等级考试大纲（二级C语言）为依据编写的教材。全书共分9章，包括简单C程序设计、简单判定性问题求解、循环结构及应用、模块化设计与应用、相同类型数据集合、深入模块化设计与应用、构造数据类型、综合设计与应用、数据永久性存储等内容。

本书注重教材的可读性和实用性，从计算机工程角度展开讲解、探索和论述。每章开头都有关键字和难点提示，每章结尾安排本章小结，并从知识层面和方法层面对本章进行总结；从日常生活或实际工程中所遇到的问题着手，典型例题一题多解，由浅入深，循序渐进，强化知识点、算法、编程方法与技巧；还将程序测试、程序调试、软件的健壮性和代码风格、结构化设计与模块化程序设计方法等软件工程知识融入其中。

本书可作为高等学校公共课教材，也可作为全国计算机等级考试参考书及C语言自学教材。

21世纪高等学校计算机规划教材

C语言程序设计与应用

◆ 主　　编　张小东　郑宏珍
责任编辑　蒋　亮

◆ 人民邮电出版社出版发行　　北京市崇文区夕照寺街14号
邮编　100061　　电子函件　315@ptpress.com.cn
网址　http://www.ptpress.com.cn
北京隆昌伟业印刷有限公司印刷

◆ 开本：787×1092　1/16
印张：21
字数：631千字　　2009年4月第1版
印数：1－3 000册　　2009年4月北京第1次印刷

ISBN 978-7-115-19319-3/TP

定价：39.80元

读者服务热线：(010)67170985　印装质量热线：(010)67129223
反盗版热线：(010)67171154

本书编审人员

主　编：张小东　郑宏珍

主　审：初佃辉

副主编：郭长勇　张　华　闫建恩

郭浩岩　张维刚　王　伟

出版者的话

现今社会对人才的基本要求之一就是应用计算机的能力。在高等学校，培养学生应用计算机的能力，主要是通过计算机课程的改革，即计算机教学分层、分类规划与实施；密切联系实际，恰当体现与各专业其他课程配合；教学必须以市场需求为导向，目的是培养高素质创新型人才。

人民邮电出版社经过对教学改革新形势充分的调查研究，依据目前比较成熟的教学大纲，组织国内优秀的有丰富教学经验的教师编写了一套体现教学改革最新形势的“21世纪高等学校计算机规划教材——高校系列”。在本套教材的出版过程中，我社多次召开教材研讨会，广泛听取了一线教师的意见，也邀请众多专家对大纲和书稿做了认真的审读与研讨。本套教材具有以下特点。

1．覆盖面广，突出教改特色

本套教材主要面向普通高等学校（包括计算机专业和非计算机专业），是在经过大量充分调研的基础上开发的计算机系列教材，涉及计算机教育领域中的所有课程（包括专业核心骨干课程与选修课程），适应了目前经济、社会对计算机教育的新要求、新动向，尤其适合于各专业计算机教学改革的特点特色。

2．注重整体性、系统性

针对各专业的特点，同一门课程规划了组织结构与内容不同的几本教材，以适应不同教学需求，即分别满足不同层次计算机专业与非计算机专业（如工、理、管、文等）的课程安排。同时本套教材注重整体性策划，在教材内容的选择上避免重叠与交叉，内容系统完整。学校可根据教学计划从中选择教材的各种组合，使其适合本校的教学特点。

3．侧重培养应用能力

目前社会对人才的需要更侧重于其应用能力，包括须具备扎实的计算机基础理论、良好的综合素质和实践能力。本套教材注意通过实践教学与实例教学培养解决实际问题的能力和知识综合运用的能力。

4．教学经验丰富的作者队伍

高等学校在计算机教学和教材改革上已经做了大量的工作，很多教师在计算机教育与科研方面积累了相当多的宝贵经验。本套教材均由有丰富教学经验的教师编写，并将这些宝贵经验渗透到教材中，使教材独具特色。

5．配套资源完善

所有教材均配有PPT电子教案，部分教材配有实践教程、题库、教师手册、学习指南、习题解答、程序源代码、演示软件和素材等，以方便教与学。

我社致力于优秀教材的出版，恳切希望老师们在使用的过程中，将发现的问题及时反馈给我们，以便再版时修改。

前　言

目前市面上有关C语言方面的图书，内容大都按传统思路组织，从C语言的历史讲起，然后是数据类型、运算符、表达式、常量、变量、控制结构、数组、指针、函数、结构体与共用体、文件等，由最基本、最本质并较为抽象的内容开始，如单词、句法、语法、程序段等，逐渐复杂化，逐步进行讲解。表面上看，这样对抽象知识学习过程，并不适合于C语言的教学，学生对知识点的理解不充分，不全面，甚至是不懂，只是靠着死记硬背而获得一个不错的分数，学完之后不会使用，就连简单的命题也可能解决不了！不可否认，这跟我国的传统应试教育模式有很大的关系，但就事论事地讲，跟我们的教材编写也有一定的关系。

本着学以致用的原则，本书从第1章开始就教学生学习写简单应用程序，每一个知识点都糅进应用当中去，不做单纯的知识点堆积，不但把前几章那些“简单的”知识点分散到各个章节，避免机械式的记忆，而且把难点也分散了。全书以应用为主线，用到了才讲；讲，就是为了能更好地用！为了培养学生规范地使用语言，先从问题的规范描述开始，然后分析问题，建立模型，实现求解，最后测试通过。从严格的科学研究与工程应用的角度出发，进行C语言的学习与研究，并将这种思想渗透到每个实例中！明确的学习目的与目标也是学习的主要动力，本书要帮助学生建立的学习目的就是能够实现算法设计解决相关命题，目标是提高自己的学习能力与动手能力。

全书在提供丰富而有趣的经典实例时，还精心设计了两个相对完整的应用：计算器与学生成绩档案管理。计算器属于算法研究4大类问题之一——计算类问题，其中包括很多经典的数值运算算法，如应用泰勒（Taylor）公式去求解三角函数等。学生成绩档案管理属于非数值运算处理，而计算机处理的信息中绝大部分是非数值信息，因此本例在实际应用中具有代表性。学生成绩档案管理系统从最简单的单个学生成绩分类开始到用多维数组存储学生基本信息与成绩信息，利用冒泡排序与选择排序按不同科目、不同成绩进行排序，再到更有聚合力的组织方式——结构体、链表，最终能够将这些数据永久性存储到文件中为止，完全贯彻实用、实践和工程应用的理念。通过这两个实例的学习让学生对C语言程序设计有一个更全面的认知，能够综合运用所学知识去解决较为实际的问题。

全书由张小东、郑宏珍主编，第1章由全体参编人员共同编写，第2章由张小东、王伟编写，第3章由闫健恩编写，第4章由郭长勇编写，第5章由郭浩岩编写，第6章由张小东编写，第7章由张华编写，第8章由张小东、张华编写，第9章由张维刚编写。初佃辉教授在百忙之中审阅了全部初稿，对本书提出了很多宝贵意见；在书稿的录入、校对及例题和习题的调

试过程中，郭长勇、闫健恩、张华及其他参编同志做了大量的工作。在此向他们表示衷心的感谢。

因编者水平有限，书中疏漏之处在所难免，恳请读者批评指正。

作者 E-mail 地址为 z_xiaodong7134@163.com、hit_gcy@163.com 和 zhua547@163.com。

编者

2009 年 2 月

目　录

第1章

Chapter 1

简单C程序设计

内容提示

关键词

- C语言的基本构成、函数、语句、注释、文件包含
- 编程风格
- 问题求解、流程图
- 数据类型、简单变量、运算符
- Visual C ++ 6.0编译环境

难点

- 画流程图
- 优先级与结合性
- printf的格式控制

C 语言从诞生至今，已经度过了 30 多年的辉煌历程，以紧凑的代码、高效的运行、强大的功能和灵活的设计与使用而闻名于世，受到众多编程人员的青睐。下面就让我们步入 C 的世界，揭开它“古老”而神秘的面纱，一起享受编程所带来的苦与乐。

1.1　C 程序的构成

“说”是学好语言的最佳方法之一，C 语言也不例外。下面就让我们先“说”出第一个程序，跟奇妙的 C 打个招呼，了解一下它的目前状况。

1.1.1　简单的 C 程序实例

先来看一个简单的 C 语言程序。程序清单 1-1 中包含了 C 语言程序的一些基本特征。仔细阅读程序中的每一行，凭着对英文单词的理解，尝试猜测一下程序中每一行所起的作用，总结出程序所能完成的功能，然后与后续给出的程序逐行解读进行对比，衡量一下差距所在。

程序清单 1-1　dream.c

```
/* 一个简单的 C 程序实例 */
/*
 purpose: I have a dream
 author : Zhang Weigang
 created: 2008/06/30 21:18:08
*/
#include <stdio.h>
#include <stdlib.h>
void main()
{
    int nNumber1;
    nNumber1 = 1;
    printf("Hello C language!\n");
    printf("I have a dream that one day I will be skillful in C programming!\n");
    printf("I have a dream that one day I will be a famous computer scientist!\n");
    printf("I have a dream that one day I will solve the Goldbach's conjecture problem and"
           "prove that %d + %d = %d by using C!\n", nNumber1, nNumber1, 2);
    printf("I hava a dream today!\n");
}
```

把程序清单 1-1 当中的内容输入计算机，保存成以 .c 结尾的文件，命名为“dream.c”，然后使用相应的 C 语言编译器（本书使用的是微软公司出品的 Visual C++ 6.0 编程应用软件，将在 1.3 节中详细介绍）进行编译、链接和运行后，就会在计算机屏幕上看到该程序的输出结果，如图 1-1 所示。

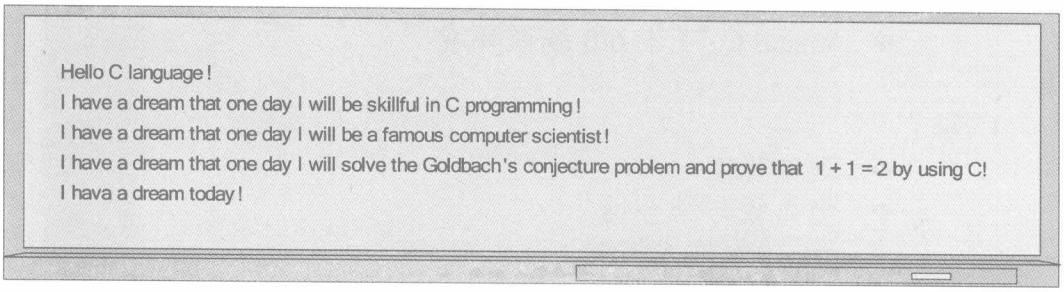


图 1-1　dream.c 程序运行结果

结果并不奇特，绝大部分内容都在程序清单中出现过。但是，程序清单中的那些“\n”和“%d”却消失得无影无踪了，取而代之的是换行效果和整数数值，这是怎么回事呢？

在继续解释上述程序实例之前，先给出本书中程序清单及运行结果的体例。如程序清单 1-1 所示，C 程序代码中 C 语言的关键字（如 int）和函数名（如 main、printf）等使用不同颜色突出显示，注释文字采用小号字体显示。

1.1.2 阅读C程序

阅读理解是驾驭语言最重要的练习之一。流程图可以为设计和分析 C 语言程序提供有力的帮助。通过流程图将程序当中的关键步骤提取出来，有助于我们阅读和理解程序清单 1-1。

1. 流程图描述

程序清单 1-1 对于刚开始接触 C 语言的读者来说，看上去比较复杂，可是如果从程序执行的结果来说，它又是简单的。细心的读者可能会发现，由显示器输出的结果就是由每个 printf 开始，包含在双引号之间的内容，显示顺序与 printf 在代码中的排列次序相同。这是不是暗示了程序的一种执行次序呢？事实上，由于当前计算机的构造都是延续使用了著名的冯·诺依曼结构体系，所以在单 CPU 的机器中程序都是按顺序执行的。

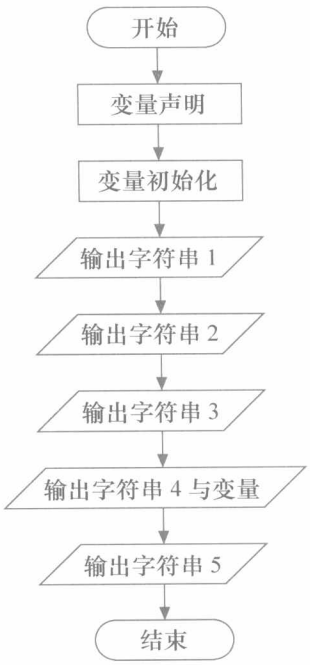


图 1-2 代码流程图

要输出上述结果，程序代码可以进行简化的内容不多。但在编写代码之前或阅读分析代码时，可以以一种比较简单而且清晰的方式描述程序执行的流转过程——流程图。图 1-2 所示的即为上述代码主体部分的描绘。它是从 main 后的“{”开始的。其中，“int nNumber1;”语句为变量的声明；“nNumber1 = 1;”语句为变量的初始化。

5 个 printf 及其所带内容为输出语句、输出字符串与变量值。这 5 条输出语句中有 4 条结构是一样的（你能找出来吗？）。它们在流程图中可以用一个条目描述，如“输出字符串 1，2，3…”。依照这样的描述，程序的实现也可进行精简。怎样精简？请读者自己思考。换句话说，同样的流程图，同样的结果，可以有不同的程序实现方法，这也从一个侧面反映了 C 语言的灵活性。

流程图中用到了 4 种图形符号，如图 1-3 所示。这些符号是美国 ANSI 系统流程图的标准符号（请初学者不要自创符号）。除了“控制流 / 程序流转方向”符号外，都需要在图形内部加文字说明。通常，在“程序的入口 / 出口”中填写“开始 / 结束”；在“加工 / 处理”中填写程序所完成功能的关键步骤，其描述粒度取决于对程序本身的分析粒度；“输入 / 输出”可以填写输入的内容、源数据的要求或输出的结果、数据要求等。流程图是每个程序员都应该掌握的程序或算法描述的手段之一。在程序设计或分析时，要用国际通用流程图符号来表达，以便能与其他人进行交流。

图 1-3 流程图组件

细看图 1-2 中描述的流程图，会发现一个有意思的现象：在此流程图的描述中没有一个 C 语言符号！那是不是说这个流程图可以不用 C 语言实现？回答是肯定的！程序设计和程序实现是两个关系紧密而又不同的阶段。当需解决较为复杂的某个问题时，不要忙着写代码，优秀的工程师通常先用流程图（或其他程序设计的工具）把将要实现的关键步骤描述出来，确认这样设计是正确的，没有任何逻辑上的问题后，再用程序实现。因为任何设计上的逻辑错误所导致的程序修改，通常工作量都是巨大的，工作过程都是艰苦的！所以程序设计是解决问题的关键！有了上述分工，程序设计者可以不必受语言的限制，充分发挥自己的智慧进行设计；而程序编写人员可选择自己最熟悉的

语言。如果C语言不熟，可以换其他语言，如Basic、Pascal等，以最快的速度完成任务。这就是使用流程图的一大好处。如果把程序设计人员当成是工程师，那么程序编写员只能算是一个普通工人。为了能做一个“白领”，在学好C语言的同时，也必须要学好使用标准的流程图来完成程序设计。当然必须说明一点：流程图不是程序设计描述的唯一手段。

2. 第一遍阅读程序

由流程图的分析可以知道程序的设计思路，而程序清单1-1中所包括的内容却不止这些。让我们一起快速阅读程序，从上至下分析程序中出现的每一条语句，给出具体语句的解释及所起的作用。

```
/* 一个简单的C程序实例 */                                    ← 注释
/*
    purpose: I have a dream
    author : Zhang Weigang                                    ← 注释
    created: 2008/06/30 21:18:08
*/
```

符号/*和*/之间包含程序的注释性内容，是为了帮助读者理解程序，对编译和运行程序不起作用。好的程序风格通常在程序前面写出该程序要完成的功能、程序作者等。

```
#include  <stdio.h>                                           ← 包含另一个文件
```

将文件stdio.h中的全部信息包含到本程序中，成为本程序的一部分。文件stdio.h是所有C语言编译包的一个标准部分，是和输入输出有关的头文件，该文件对接收由键盘输入的数据和向显示器输出运行结果提供支持。

```
#include  <stdlib.h>                                          ← 包含另一个文件
```

将文件stdlib.h包含到本程序中。

```
void main()                                                   ← 主函数名
```

C程序是由一个或多个函数组成的，函数是C程序的基本组成部分。上面的程序中包含一个名字为main的函数。圆括号表明main是一个函数的名字，void表明main()函数没有返回值，也没有传入参数。

```
{                                                             ← 函数体开始标志
```

左花括号表明组成函数语句的开始，而右花括号标志函数的结束。

```
int  nNumber1;                                                ← 变量声明语句
```

该语句定义1个变量，变量名为nNumber1，它是整数（int）类型，可以被后面的程序使用。

```
nNumber1 = 1;                                                 ← 赋值语句
```

表明将1赋值给变量nNumber1。

```
printf("Hello C language!\n");                                ← 函数调用语句
```

调用printf()函数在显示器上输出“Hello C language!”，并且使光标移到下一行的开始处。符号\n告诉计算机要另起一行，即将光标移到下一行的开始处。

```
printf("I have a dream that one day I
        will be skillful in C programming!\n ");              ← 一个函数调用语句
```

接下来的printf()语句表示在下一行的开始处显示“I have a dream that one day I will be skillful in C programming!”，输出之后使光标移到下一行的开始处。

```
printf("I have a dream that one day I will
be a famous computer scientist!\n");                          ← 一个函数调用语句
```

使用printf()语句在下一行的开始处显示“I have a dream that one day I will be a famous computer scientist!”，输出后光标移到下一行开始处。

```
printf("I have a dream that one day I will solve the Goldbach's conjecture problem and
""prove that %d + %d = %d by using C!\n", nNumber1, nNumber1, 2); ← 一个函数调用语句
```

调用printf函数在下一行的开始处输出“I have a dream that one day I will solve the Goldbach's conjecture problem and prove that 1 + 1 = 2 by using C!”，使用printf()把变量nNumber1、nNumber1和2的值内嵌在用引号引起来的字符串中进行输出，%d指示输出变量值的位置和形式。第一个%d的位置输出变量nNumber1的值，第二个%d的位置输出变量nNumber1的值，第三个%d的位置输出变量2的值。

```
    printf("I hava a dream today!\n");          ← 函数调用语句
```

最后使用printf函数输出“I hava a dream today!”，并使光标移到下一行开始处。

```
}                                               ← 函数体结束标志
```

函数体必须以右花括号作为结束标志。

3. 深入理解

通过上述阅读，你对上面的程序会有一个感性的理解和认知，本节将会带着你迈上C的又一个台阶。

图1-4所示的内容在程序清单1-1中出现过，在第一遍阅读中已经知道包含在/*……*/之间的文字是程序的注释。使用注释的目的是使人们（包括编程者本人）更容易理解程序的功能和结构，尤其针对一段难懂的代码可以用注释进行简要的说明。C语言注释可以放在任意的地方，甚至是和它所要解释的语句在同一行。一个较长的注释可以放在一行或者多行。注释内容不是程序代码的正式内容，都会被编译器忽略掉。另外，在C99（ANSI在1999年推出了新的标准C规范，即被称为C99）中增加了另一种风格的注释“//”，其被限制在一行里，例如，

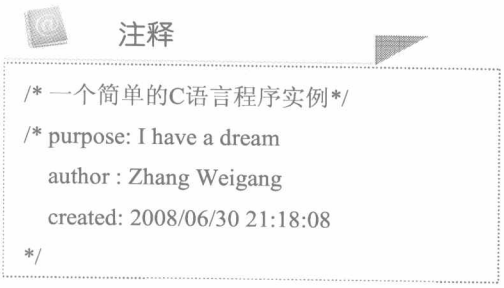


图1-4 /* */注释举例

```
    printf("This is an example.");          //将此行字符串内容打印到显示器屏幕上
```

在程序清单1-1中的int和nNumber1之间有个空格，被称为空格符，在这里是必须有的。编译时，它也会被忽略掉。类似的符号在C语言中还有制表符（Tab键产生的字符）、换行符（Enter键产生的字符），统称为空白符。在程序编译时的词法分析阶段，将程序正文分解为词法记号和空白符。空白符用于指示词法记号的开始和结束位置，除此之外不具有任何其他功能。读者可以尝试一下，在程序清单1-1中加10000个换行符、空格符或制表符，看看源码文件（dream.c）的大小与未加之前有何区别？编译后生成的dream.exe又有何不同？

#include 预处理命令和头文件

在一个程序中经常要使用到很多函数，但不是所有的函数都是由程序员自己编写的，如在程序清单1-1中所用的printf()函数，就是由C语言提供的，被称为库函数。C语言提供了很多这样的函数，并将这些函数进行分类，存放到不同的文件里。为了方便大家使用，并保护这些函数免遭破坏及侵权，将每一类函数定义文件拆成两个——函数说明文件和函数定义文件。函数说明文件以.h为扩展名，包括函数名称、参数及相关的使用说明等，是可以用常用编辑器打开的一类文本文件，由于其通常被放到程序的开始部分，故又称头文件，而函数的定义文件是二进制文件，通常不能被阅读。如图1-5所示，stdio.h包括了基本的输入/输出函数，stdlib.h包含了字符串转换、随机数、内存管理等常用函数。

```
#include <stdio.h>
#include <stdlib.h>
```

图1-5 预处理命令

那么在运行一个程序的时候，怎么先通知编译器该程序都使用了哪些库函数呢？在C语言中，是用以“#”号开头的预处理命令，如包含命令#include、宏定义命令#define等。这些命令都放在函数之外，源文件的开始处，称为预

处理部分。所谓预处理，是指在进行编译的第一遍扫描（词法扫描和语法分析）之前所作的工作。预处理是 C 语言的一个重要功能，由预处理程序负责完成。当对一个源文件进行编译时，系统将自动引用预处理程序对源程序中的预处理部分作处理，处理完毕后，再自动进入对源程序的编译。图 1-5 中用 include 指令将 stdio.h 和 stdlib.h 包含到程序当中，在编写程序时，就可以使用这两个文件中所提及的函数了。

C 语言提供了丰富的库函数，大大简化了程序设计的复杂度。可以说，在掌握了基本的 C 语言的语法知识后，驾驭 C 语言的能力完全取决于对库函数的熟悉程度和使用技巧。

main() 函数

程序清单 1-1 中有一个 main() 函数，例如，

```
void main()
{
    ...
}
```

在今后的学习当中，我们会发现，任何一个程序里都有且仅有一个 main() 函数。无论 main() 函数放在哪里，程序都是先从 main() 函数开始执行，完成对其他函数的调用后再返回 main() 函数，最后由 main() 函数结束整个程序。即正常的 C 语言的程序执行是起始于 main()，结束于 main()。这一点可在后续的跟踪调试章节观察到。

语句与分隔符

例如，代码

```
int  nNumber1;
nNumber1=1;
printf("Hello, C language!\n");
...
```

中出现了一个个由“；”结束的语句。事实上，C 程序的执行部分就是由语句组成的，这些语句后都是由“；”结束。程序的功能也是由若干执行语句实现的，可分为 5 类：表达式语句、函数调用语句、控制语句、复合语句、空语句。“；”在 C 语言中被称为分隔符，不表示任何实际的操作，仅用于构造程序。分隔符还包括“()”、“{}”、“,”、“：”。

函数是 C 的基本单位

我们的编程不是从 0 开始，而是站在“巨人”的肩膀上的。在一个庞大的工程设计当中，要用到很多优秀工程师编写的一段段功能完善并相对独立的代码，这一段段的代码被封装成为一个个函数，如前所述，它们被分类归放到不同的文件中。需要时，将它们“组装”进自己的程序中即可。程序清单 1-1 中用到的 printf() 就是一个写好的输出函数。

每个 C 程序都是由函数构成的。函数是 C 程序的基本组成单位，是独立完成某种功能的代码的集合，分为标准函数（或称库函数）和自定义函数。

1.1.3　C 程序结构

在经过对 dream.c 程序的详细阅读和相关知识扩展后，做适当的总结与抽象，就可以得到图 1-6 所示的 C 程序组成结构。C 程序依照从上到下的顺序结构组成，主要由程序功能注释块、#include 等预处理命令块、main 函数、函数体等构成。一般来说，C 程序由一个函数或多个函数组成，如前所述，其中有且仅有一个主函数 main()。函数由函数头和函数体构成。函数头的主体标识是函数名及圆括号 ()。() 中可能为空，也可能有一些参数。函数体则由若干语句构成。

简而言之，一个简单的标准 C 程序应该具有如图 1-7 所示的代码结构。

程序功能注释块
预处理命令
main()
{
函数体
}
其他函数…

图 1-6 C 程序结构

```
/* C 程序标准组成结构 /*
/*                                    ← 程序功能注释块
   purpose:
    author:
   created:
*/
                                       ← 预处理命令
#include <stdio.h>
#include <stdlib.h>

void main()  ← main 函数头
{
    declaration statement;
    initialization statement;          ← 函数体
    operation statement;
    output statement;
}
```

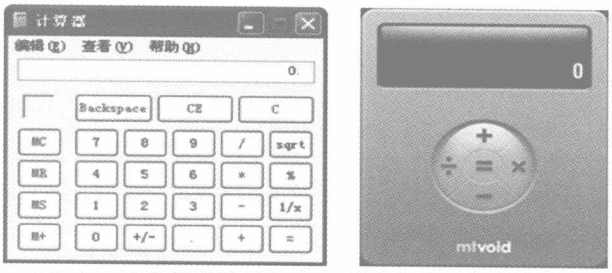
图 1-7 C 语言代码结构

1.2 简单程序扩展

计算性问题与我们的生活紧密相伴，各种各样的计算方法或工具也在根据实际的需要不断发展，功能由简单到复杂，如指算、结绳计算、算盘、对数计算器等，到如今的各种功能强大的电子计算机。本节将以计算器为例，对其进行功能分析，给出计算功能的解决方案和程序代码，并对计算性问题的 C 语言实现进行初步的探讨。

1.2.1 计算器基本功能

在软件开发过程中，首先要解决“做什么”的问题，这样编码才能做到有的放矢。我们的目标是设计并实现一个计算器，使之具有基本的数学运算功能。在设计一个程序时，首先应从整体和全局出发，再考虑局部的细节，这就是“自顶向下，逐步求精”的设计原则。根据这一原则，分析的路线是：计算器功能→组成计算器的主要构件功能→组成每个构件的函数功能。分析设计完成后，用 C 语言实现。



Windows XP 标准计算器　　手机上使用的简单计算器

图 1-8 简易计算器图形界面

图 1-8 展示了两个不同硬件、不同操作系统支持，具有不同显示风格的简易型计算器。比较这两款计算器，可以发现很多不同：右边手机上使用的简单计算器没有数字按键，没有正、负号，没有小数点，没有“sqrt”、“%”、“1/x”等功能，也没有外围字母标识的按键等。在 Windows XP 标准计算器中多出的这些按键是干什么用的呢？数字、小数点是用来输入操作数的，如果没有，是不可能完成计算的！手机上使用的计算器输入当然是在手机键盘上，Windows XP 标准计算器的操作数输入也有键盘对应，在图形上显示只是为了满足鼠标操作的需要。尽管计算器软件有上述多种形式，但通过使用和上述对比，可以总结出以下几点。

（1）能够输入操作数，通过点击数字键盘完成。

（2）能够指定所做运算方式，即 +、－、×、／、% 等，通过点击功能键完成。

（3）能计算出正确的结果，并在显示屏上显示。