



高职高专“十一五”规划教材·计算机类

C语言程序设计

陈显刚 主编

C YUYANCHENGXUSHEJI



 北京理工大学出版社

BEIJING INSTITUTE OF TECHNOLOGY PRESS

高职高专“十一五”规划教材·计算机类

C 语言程序设计

陈显刚 冯艳茹 主编
张洪明 胡洪都 副主编



北京理工大学出版社

BEIJING INSTITUTE OF TECHNOLOGY PRESS

内 容 简 介

本书系统地讲述了C语言、结构化的软件设计方法、集成开发环境及VC++ 6.0的应用。

全书共分13章,以“学生成绩管理”案例贯穿始终,结合近百个以函数为框架的实用示例程序,突出了结构化程序设计。分布在各章的“做中学”、“跟我学”、“学以用”、“案例分析”、“常见编程错误”、“实训”、“习题”等特色部分非常具有实际指导意义。本书易于实践操作的内容,不仅使刚接触C语言的初学者得到乐趣,还可以让有经验的程序员得到启发。

本书是为高职高专院校计算机相关专业编写的教材,也可以作为C语言程序设计的自学教材和各类培训教材。

版权专有 侵权必究

图书在版编目(CIP)数据

C语言程序设计/陈显刚,冯艳茹主编. —北京:北京理工大学出版社, 2007.2

高职高专“十一五”规划教材.计算机类

ISBN 978-7-5640-0968-7

I.C… II.①陈… ②冯… 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(2007)第011809号

出版发行/北京理工大学出版社

社 址/北京市海淀区中关村南大街5号

邮 编/100081

电 话/(010)68914775(总编室) 68944990(批销中心) 68911084(读者服务部)

网 址/<http://www.bitpress.com.cn>

经 销/全国各地新华书店

印 刷/北京市业和印务有限公司

开 本/787毫米×1092毫米 1/16

印 张/20.25

字 数/391千字

版 次/2007年2月第1版 2007年2月第1次印刷

责任校对/张 宏

定 价/29.80元

责任印制/母长新

图书出现印装质量问题,本社负责调换

前 言

C 语言具有灵活、高效、可移植等优点，是软件开发中最常用的计算机语言。它既有高级语言的优点，又具有低级语言的诸多特点；既可以编写系统软件，又可以编写应用软件。近年来，面向对象编程语言 C++ 得到了很大发展。人们有一种误解，认为 C 语言没有前途了，这种理解是片面的。C 语言以其自身特色和优势，在工业控制、智能仪表、嵌入式系统、硬件驱动、系统底层开发、中间件等领域应用非常广泛。坚定学习好 C 语言的信心，相信会有很好的前景。

本书编写团队由有经验的技术开发人员和具有多年教学经验的人员组成，他们合力为高职高专计算机相关专业量身打造了一本实用性、适用性非常强的教材。

本书的主要特色是在内容安排上易于“自学”、结构安排上易于“操作”。

本书以数据处理、内存使用为主线，以程序设计方法为核心，以一个读者普遍熟悉的“学生成绩管理系统”案例贯穿始终。在内容设计上，体现“在什么地方应用，应用格式是什么，如何应用”三个方面。C 语言中函数是本质、指针是精华，对于这两部分编者精心设计，不是把函数、指针放到后面介绍，而是提到前面。从第 4 章开始介绍函数基础，自本章后，所有的程序都用函数实现，并在第 7 章中做了深入说明。从第 3 章开始介绍指针的基本概念，贯穿全书，且与内存知识一同介绍，并在第 12 章中做了总结性的深入讲解。全书体现了“程序是学出来的而不是教出来的”的思路。

本书有别于其他教材的还有，增加了软件开发方法的介绍，不是仅以 C 语言为主，而是把软件开发和 C 语言有机地结合在一起，共同介绍。通过本书的学习，使读者能够掌握软件设计和开发的方法，用 C 语言解决实际问题。

本书以 VC++ 6.0 为编辑、编译环境，全书代码均在 VC++ 6.0 和 TC 中调试通过。

本书由陈显刚、冯艳茹担任主编负责组织、筹划工作，并亲自执笔。由张洪明、胡洪都担任副主编。参加本书编写的人员还有孙凌凌、刘铁英、刘冰洁、董晶等。

由于编者水平有限，教材中难免有差错、疏漏之处，敬请同行专家及广大读者批评指正。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

第1章 软件开发方法	1	3.5.3 程序设计举例	26
第2章 C语言概述	5	3.6 顺序结构程序设计案例分析	27
2.1 运行一个简单的C语言程序	5	3.7 常见编程错误	29
2.1.1 创建工程	5	3.8 实 训	31
2.1.2 新建源文件	6	3.8.1 实训 1	31
2.1.3 编辑源文件	7	3.8.2 实训 2	32
2.1.4 运行程序	7	3.8.3 实训 3	33
2.2 C语言程序构成要素	8	3.8.4 实训 4	34
2.2.1 注释	8	习 题	35
2.2.2 文件包含指令#include	8	第4章 数据处理的基本单位：函数	36
2.2.3 main函数	8	4.1 概述	36
2.2.4 语句	9	4.1.1 函数概念及构成要素	36
2.2.5 保留字和标识符	9	4.1.2 用函数解决实际问题的 方法和步骤	37
2.2.6 C语言程序结构	10	4.2 库函数	37
2.3 实 训	10	4.2.1 常用数学库函数	37
习 题	11	4.2.2 几个常用的库函数	39
第3章 C语言的数据	12	4.2.3 程序设计举例	40
3.1 概 述	12	4.3 函数定义、调用	41
3.1.1 数据及数据处理	12	4.3.1 函数定义格式	41
3.1.2 数据处理的主要步骤	12	4.3.2 函数在程序中的布置	42
3.2 数据类型	13	4.3.3 函数的调用	43
3.3 数据的输出	14	4.3.4 程序设计举例	44
3.3.1 输出函数	14	4.4 没有参数的函数	47
3.3.2 程序设计举例	15	4.4.1 没有参数没有返回值的函数	47
3.4 数据的输入、存储	16	4.4.2 没有参数有返回值的函数	48
3.4.1 输入、存储概念	16	4.4.3 程序设计举例	49
3.4.2 变量	16	4.5 变量的使用范围	50
3.4.3 为变量提供值	19	4.5.1 局部变量和全局变量	50
3.4.4 常量与符号常量	22	4.5.2 静态变量	51
3.4.5 程序设计举例	23	4.5.3 程序设计举例	53
3.5 数据的算术运算	25	4.6 案例分析	54
3.5.1 算术运算符	25	4.7 常见编程错误	58
3.5.2 算术表达式	25	4.8 实 训	60

4.8.1 实训 1	60	步骤和方法	93
4.8.2 实训 2	61	6.2 for语句	94
4.8.3 实训 3	62	6.2.1 for语句格式	94
4.8.4 实训 4	64	6.2.2 程序设计举例	97
习 题	64	6.3 while语句	98
第 5 章 选择结构	66	6.3.1 while格式	98
5.1 概述	66	6.3.2 程序设计举例	99
5.1.1 概念	66	6.4 do...while语句	101
5.1.2 解决选择问题的基本 步骤和方法	67	6.4.1 do...while语句格式	101
5.2 条件的描述	68	6.4.2 程序设计举例	102
5.2.1 关系表达式	68	6.5 for、while和do...while三种 语句比较	103
5.2.2 逻辑表达式	70	6.5.1 for、while和do...while 三种语句回顾	103
5.2.3 常见运算符的优先级别	71	6.5.2 几种特殊循环	104
5.2.4 程序设计举例	72	6.6 循环中断控制	106
5.3 if语句	73	6.6.1 break语句	106
5.3.1 if语句格式	73	6.6.2 continue语句	107
5.3.2 程序设计举例	74	6.6.3 程序设计举例	108
5.4 if...else语句	75	6.7 循环嵌套	110
5.4.1 if...else语句格式	75	6.7.1 循环嵌套格式	110
5.4.2 if-else-if形式	76	6.7.2 程序设计举例	111
5.5 switch语句	78	6.8 案例分析	112
5.5.1 switch语句格式	78	6.9 常见编程错误	115
5.5.2 程序设计举例	79	6.10 实 训	117
5.6 案例分析	81	6.10.1 实训 1	117
5.7 常见编程错误	83	6.10.2 实训 2	118
5.8 实 训	85	6.10.3 实训 3	119
5.8.1 实训 1	85	6.10.4 实训 4	121
5.8.2 实训 2	86	6.10.5 实训 5	122
5.8.3 实训 3	87	6.10.6 实训 6	123
5.8.4 实训 4	88	习 题	123
5.8.5 实训 5	89	第 7 章 模块化程序设计	125
习 题	90	7.1 概述	125
第 6 章 循环结构	91	7.1.1 模块化编程概念	125
6.1 概述	91	7.1.2 模块化编程的基本 方法和步骤	125
6.1.1 概念	91	7.2 用函数实现模块化编程技术	129
6.1.2 解决循环问题的基本			

7.2.1 函数的嵌套调用.....	130	9.2.3 程序设计举例.....	181
7.2.2 指针变量作为形式参数.....	134	9.3 字符串与一维字符型数组.....	182
7.2.3 函数返回方法的比较.....	136	9.3.1 字符串与字符串结束标志.....	182
7.2.4 程序设计举例.....	138	9.3.2 用字符型数组输入、输出字符串.....	183
7.3 案例分析.....	140	9.3.3 程序设计举例.....	184
7.4 常见编程错误.....	143	9.4 常用字符串处理函数.....	187
7.5 实训.....	146	9.4.1 常用字符串处理函数.....	187
7.5.1 实训 1.....	146	9.4.2 程序设计举例.....	189
7.5.2 实训 2.....	147	9.5 字符串数组.....	190
7.5.3 实训 3.....	148	9.5.1 字符串数组作为二维字符型数组的输入和输出.....	191
7.5.4 实训 4.....	149	9.5.2 程序设计举例.....	193
习题.....	150	9.6 案例分析.....	195
第 8 章 数值型数组.....	151	9.7 常见编程错误.....	197
8.1 概述.....	151	9.8 实训.....	200
8.1.1 概念.....	151	9.8.1 实训 1.....	200
8.1.2 解决数组问题的基本方法和步骤.....	152	9.8.2 实训 2.....	200
8.2 一维数组.....	152	9.8.3 实训 3.....	202
8.2.1 一维数组的定义.....	152	9.8.4 实训 4.....	203
8.2.2 一维数组的输入和输出.....	154	9.8.5 实训 5.....	204
8.2.3 程序设计举例.....	157	习题.....	206
8.3 二维数组.....	162	第 10 章 结构体.....	208
8.3.1 二维数组的定义.....	162	10.1 概述.....	208
8.3.2 二维数组的输入和输出.....	163	10.2 定义结构体类型、结构体类型变量的使用.....	209
8.3.3 程序设计举例.....	165	10.2.1 结构体数据类型的定义.....	209
8.4 案例分析.....	168	10.2.2 结构体变量的引用.....	210
8.5 常见编程错误.....	170	10.2.3 结构体变量的初始化、输出.....	211
8.6 实训.....	172	10.2.4 程序设计举例.....	213
8.6.1 实训 1.....	172	10.3 结构体数组.....	214
8.6.2 实训 2.....	173	10.3.1 定义结构体数组.....	214
8.6.3 实训 3.....	174	10.3.2 程序设计举例.....	216
习题.....	175	10.4 案例分析.....	218
第 9 章 字符型数组与字符串.....	176	10.5 常见编程错误.....	221
9.1 概述.....	176	10.6 实训.....	222
9.2 字符型数组.....	178	10.6.1 实训 1.....	222
9.2.1 一维字符型数组.....	178		
9.2.2 二维字符型数组.....	179		

10.6.2 实训 2	223	12.2.3 程序设计举例	259
10.6.3 实训 3	224	12.3 函数的返回值及参数	262
习 题	225	12.3.1 返回指针值的函数	262
第 11 章 文件	226	12.3.2 函数指针和指向函数 的指针变量	262
11.1 概述	226	12.3.3 程序设计举例	265
11.1.1 文件概念	226	12.4 动态内存的申请	266
11.1.2 文件操作的基本方法 和步骤	226	12.4.1 动态分配内存的函数	266
11.2 文件的打开和关闭	227	12.4.2 使用指针处理动态链表	269
11.2.1 文件指针	227	12.4.3 程序设计举例	270
11.2.2 文件的打开	227	12.5 常见编程错误	273
11.2.3 文件的关闭	229	12.6 实 训	275
11.3 文件的读写函数	230	12.6.1 实训 1	275
11.3.1 fgetc、fputc 函数	231	12.6.2 实训 2	275
11.3.2 读写字符串 fputs、 fgets 函数	233	12.6.3 实训 3	276
11.3.3 数据块读写函数 fread 和 fwrite 函数	234	12.6.4 实训 4	276
11.3.4 程序设计举例	236	习 题	276
11.4 随机读写文件	238	第 13 章 课程设计：学生成绩 管理系统	278
11.4.1 rewind 函数和 fseek 函数	238	13.1 概述	278
11.4.2 程序设计举例	241	13.1.1 学生成绩管理系统的背景 ...	278
11.5 案例分析	242	13.1.2 系统流程概要	279
11.6 常见编程错误	248	13.2 明确问题	280
11.7 实 训	250	13.3 分 析	281
11.7.1 实训 1	280	13.4 设计算法	281
11.7.2 实训 2	251	13.4.1 概要设计	281
11.7.3 实训 3	253	13.4.2 详细设计	284
11.7.4 实训 4	254	13.5 实现	285
习 题	254	13.6 测试	312
第 12 章 C 语言高级话题：指针	255	13.7 设计说明	312
12.1 数据类型深入说明	255	13.7.1 综合性强	312
12.2 指针回顾	256	13.7.2 分析方法	313
12.2.1 运用指针的几个问题	256	13.7.3 测试	313
12.2.2 指针数组	258	13.7.4 待完善的问题	313
		课程设计题目	313

第 1 章 软件开发方法

数据处理是计算机主要的功能，解决数据处理问题，需要采用面向数据流的软件开发方法。由于 C 语言为结构化的程序语言，它适用于自顶向下的软件开发方法，自顶向下的方法产生清晰、易读懂的程序代码，使用程序易于维护，并能在编码之前就明白程序整体结构和操作，适合解决数据处理领域的实际问题。该方法采用自顶向下、逐步细化的设计方法，主要有 6 个步骤：确定问题，分析、设计算法，实现，测试，维护。本书的案例分析和课程设计都采用该方法。

1. 确定问题

确定问题要求我们清晰而明确地确定要解决什么问题。正确理解问题的需求，排除不重要的方面，把注意力集中在问题的本质上。

2. 分析

分析问题是确定问题的输入数据、输出数据以及它们之间的联系。如果第 1 步和第 2 步没有正确完成，则有可能解决了错误的问题。往往在陈述问题中通过找到相关的“短语”来确定问题的输入和输出，举个例子说明：

已知购买苹果的总数量和单价，要求计算和显示消费总额。

从陈述的短语中确定包含的信息：

问题的输入：

购买苹果的“总数量”(千克)

每千克苹果的“单价”(元/千克)

问题的输出：

苹果的“消费总额”(元)

明确了问题的输入和输出以后，需要找到它们之间的联系。也就是从现有的输入数据，如何实现预期的输出结果，这种联系或者是公式、或者是方案。本例中是通过公式(苹果的消费总额=总数量×单价)来实现输入和输出联系的。

3. 设计

设计就是设计算法，然后验证算法是否能够解决已确定的问题。设计算法往往是问题求解中最难的环节，一般采用自顶向下、逐步细化的设计原则和方法。先从原始问题开始，按照功能逐步分解为若干个子问题，子问题再分解为若干个下一级的子问题，依此类推，直到不能再分解为止。每个要解决的问题一般都包括以下几个子问题：

(1) 输入数据。

(2) 运算。

(3) 输出结果。

确定子问题后，可以通过算法精化过程，将“运算”的步骤分解成一系列更详细的步骤。最后用语言或图表等描述出来，本书采用的是自然语言描述。

4. 实现

实现就是用计算机语言实现算法。在这个环节中，把描述算法的每个步骤都转变为某种程序设计语言的一条或多条语句，最后把算法转为计算机程序。

5. 测试

程序的测试就是对已完成的程序进行测试，以验证是否满足用户的要求。测试中要用不同的输入数据多次运行程序，看程序是否完成预期的功能，若有错误必须修改程序，直到正确为止。

6. 维护

程序的维护是软件开发的最后一个环节，维护一个程序往往需要很长时间。维护的目的是更正程序设计中的漏洞、升级和更新程序。

下面通过一个案例来说明软件开发的方法和步骤。

【例 1.1】磅是西方常使用的重量单位，而千克是东方习惯使用的重量单位，那么如何将磅转换为千克(公斤)?

(1) 明确问题。

要解决的问题是：将磅转换成千克。

(2) 分析。

问题的输入是以磅为单位的重量，问题的输出是以千克为单位的重量。为了编写该程序，我们需要知道磅转换成千克的公式：1 磅等于 0.4536 千克。

下面列出了数据需求和相关的公式。

1) 数据需求。

问题的输入

weight 以磅为单位的重量

问题的输出。

kgn 以千克为单位的重量

2) 相关的公式。

1 磅=0.4536 千克

(3) 设计。

下一步是构造解决问题的算法。本问题由3个子问题构成。

S1 输入以磅为单位的重量。

S2 将磅转换为千克。

S3 输出以千克为单位的重量。

分析每个步骤是否需要进一步精化。S1步是输入数据，S3步是输出数据，不需要精化。

S2步某些细节需要进一步考虑：

精化 S2 步

S2.1 以千克为单位的重量等于 0.4536 乘上以磅为单位的重量。

下面列出精化后的完整的算法。

S1 获取以磅为单位的重量。

S2 将磅转换为千克。

S2.1 以千克为单位的重量等于 0.4536 乘上以磅为单位的重量。

S3 显示以千克为单位的重量。

设计算法后，对该算法进行检查，看这个算法是否正确。如果 S1 步得到的重量是 100.0 磅，则 S2.1 步将把它转换为 0.4536×100.0 ，即 45.36 千克。S3 步将显示这个正确的结果。经过检验算法是正确的。

(4) 实现。

我们用 C 语言程序实现该算法。将算法的每个步骤转变为一条或多条 C 语言语句。

下面给出源程序：

```
/*pl_1.c*/
/* 磅和千克转换*/
#include <stdio.h>
int main()                                /*C 语言基本结构*/
{
    #define KG_PER_POUND 0.4536          /*1 磅为 0.4536 千克*/
    double weight,kgn;
    printf("请输入磅数: ");
    scanf("%lf",&weight);
    kgn=KG_PER_POUND * weight;
    printf("相对应的千克数为:%f",kgn);
    return 0;
}
```

(5) 测试。

测试两组值。

第一组：

输入：100.0 ✓

输出：45.36

第二组：

输入：1 ✓

输出：0.453 6

第 2 章 C 语言概述

C 语言的特点是简洁、紧凑、方便、灵活，是比较理想的结构化语言。本章重点介绍 C 语言程序的运行步骤、构成要素和结构。

2.1 运行一个简单的 C 语言程序

我们以【例 1.1】中的源码为例，说明在 VC++ 6.0 中如何运行一个 C 语言程序。主要步骤有创建工程、新建源文件、编辑源文件、运行程序等。

2.1.1 创建工程

使用 VC++ 6.0 创建工程的具体操作步骤如下：

- (1) 启动“Microsoft Visual C++ 6.0”，如图 2.1 所示。
- (2) 打开“文件”菜单，选择“新建”命令，弹出如图 2.2 所示的“新建”对话框，选择“工程”选项卡，选择“Win32 ConSOLE Application”项目，在“C 位置”处选择或是新建文件夹，在“工程”处键入工程名。

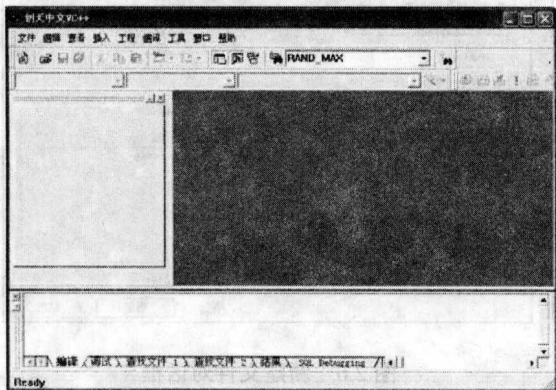


图 2.1 VC++ 6.0 界面

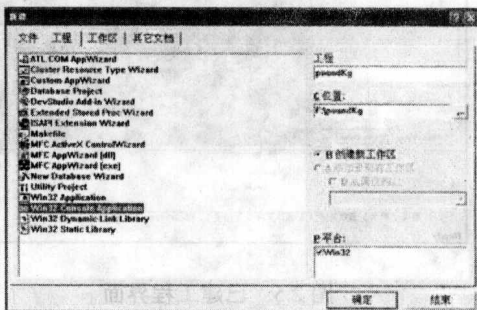


图 2.2 新建工程对话框

- (3) 单击“确定”按钮，弹出如图 2.3 所示的“Win32 Console Application-Step1 of 1”对话框，采用默认选项，单击“完成”按钮。

- (4) 这时将弹出如图 2.4 所示的“新建工程信息”对话框，这里显示新建工程的类型、目录等信息。

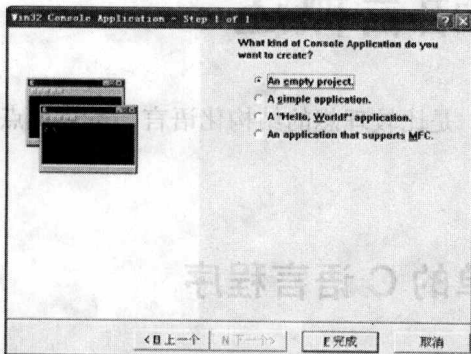


图 2.3 工程向导对话框

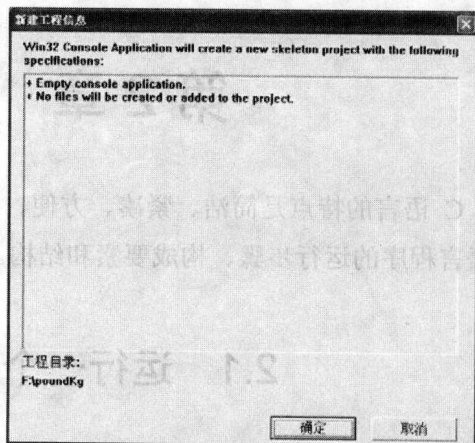


图 2.4 新建工程信息对话框

(5) 单击“确定”按钮，弹出如图 2.5 所示的界面。这时已新建一个工程，工程名为 poundKg，位置在 F:\poundKg 下。

2.1.2 新建源文件

新建源文件的具体操作步骤如下：

(1) 在图 2.5 中打开“文件”菜单，选择“新建”命令，弹出如图 2.6 所示的“新建”对话框。

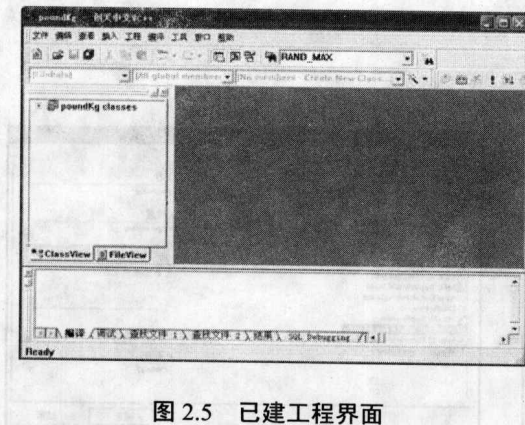


图 2.5 已建工程界面

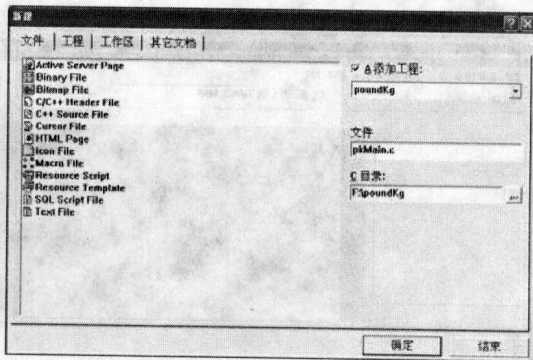


图 2.6 新建文件对话框

(2) 在该对话框的“文件”选项卡中选择“C++ Source File”文件类型，在标签为“文件”的文本框中输入文件名 pkMain.c，其他各参数不变。这里特别强调的是：文件的扩展名一定是.c，不能是其他的扩展名。

(3) 单击“确定”按钮，弹出如图 2.7 所示的源程序编辑界面。

2.1.3 编辑源文件

在图 2.7 所示的源程序编辑界面中输入【例 1.1】中的源程序。

2.1.4 运行程序

运行程序的具体操作步骤如下：

(1) 在图 2.7 中打开“编译”菜单，如图 2.8 所示。

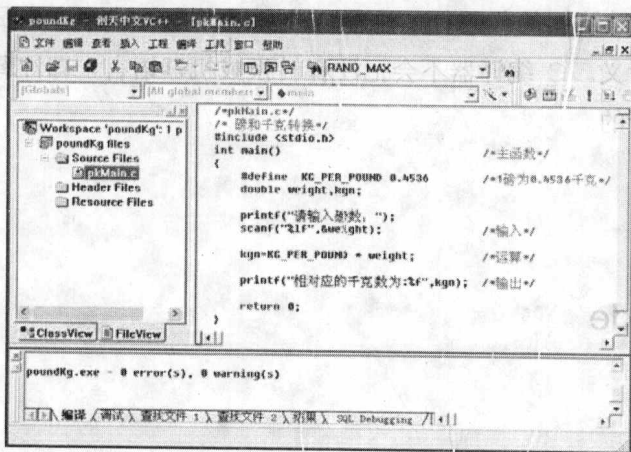


图 2.7 源程序编辑界面

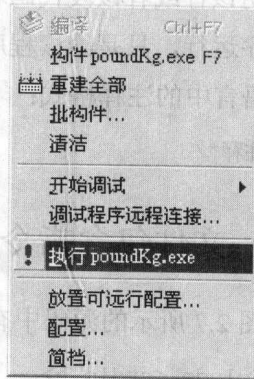


图 2.8 编译菜单

(2) 在该菜单中选择“执行 poundKg.exe”命令，或者单击工具条中的  按钮，运行程序，弹出消息提示对话框，如图 2.9 所示。

(3) 在该对话框中单击“是”按钮，弹出如图 2.10 所示的显示运行结果对话框，即显示程序运行结果，按任意键返回到图 2.7 所示的编辑界面。

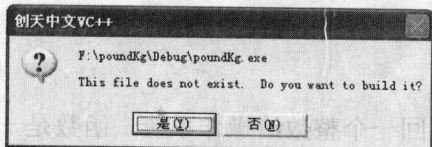


图 2.9 是否构建可执行程序消息框

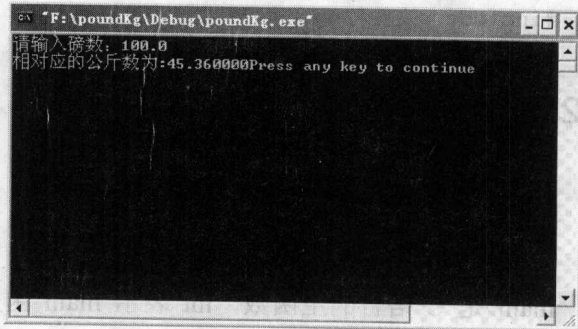


图 2.10 显示结果对话框

2.2 C 语言程序构成要素

2.2.1 注释

在图 2.7 所示的源码中：

```
/*weight.c*/  
/* 磅和千克转换*/
```

以“/*”符号开头并以“*/”符号结尾的代码行是注释，注释就是向阅读程序人员解释、说明该行或者该段代码表示的含义。C 编译器不会把注释转换成机器语言，即注释不参与程序运行，只是给读程序的人看的。

C 语言中的注释格式：

```
/*注释*/
```

2.2.2 文件包含指令#include

在图 2.7 所示的源码中有：

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

以“#include”开头的代码行表示文件包含处理，所谓文件包含处理是指一个源文件可以将另一个源文件的全部内容包含进来。被包含的文件称为头部文件，常以.h 为后缀，如 stdio.h。本行的作用是包含程序中用到的函数 printf 和 scanf，这两个函数是在 stdio.h 文件中定义的。

文件包含的一般格式为：

```
#include <文件名>
```

2.2.3 main 函数

在图 2.7 所示的源码中有：

```
int main()
```

main 是 C 语言的主函数，int 表示 main 函数返回一个整数给操作系统。函数是一组执行指定操作的相关语句。用一对大括号把相关语句括在一起。无论一个程序中有多少个函数，但必须有且只有一个 main 函数。C 语言程序总是从 main 函数体的第 1 行函数开始执行，即 main 函数是程序运行的入口。

main 函数的一般格式为:

```
int main()
{
    函数体
    return 0;
}
```

2.2.4 语句

在图 2.7 所示的源码中, 程序都是以行为单位的, 一行 C 语言程序的代码就是一条语句, 一个函数由多条语句行构成, 每条语句最后都有一个“;”标识。在一个函数中语句分为声明语句和执行语句。

1. 声明语句

```
#define KG_PER_POUND 0.4536
double weight, kgn;
```

第 1 行表示声明一个名为 KG_PER_POUND、值为 0.4536 的符号常量; 第 2 行表示声明两个变量 weight, kgn, 类型是 double 类型(双精度型), 关于数据类型的详细信息将在第 3 章中介绍。

2. 执行语句

在图 2.7 所示的源码中有:

```
printf("请输入磅数: ");
scanf("%lf",&weight);
kgn=KG_PER_POUND * weight;
printf("相对应的千克数为:%f",kgn);
```

这四行语句的作用是提示从键盘中输入磅数; 存入变量 weight 中; 计算结果存入变量 kgn 中; 最后输出结果。可执行语句是在程序执行时发出的某种动作。可执行语句由操作和操作的内容组成, 相当于汉语中的动宾结构。

2.2.5 保留字和标识符

1. 保留字

在图 2.7 所示的源码中有 include、define、double、return 等保留字, 所谓保留字就是 C 语言中已有的具有特殊含义的字符符号。这些保留字不能用于其他目的。