



耿祥义 张跃平◎编著

程序设计

教学做 一体化教程



清华大学出版社



耿祥义 张跃平◎编著

程序设计

教学做一体化教程

清华大学出版社
北京

内 容 简 介

C 语言不仅是计算机学科的一门基础语言,而且它高效、灵活,特别适合用来编写操作硬件设备的程序,使得 C 语言在嵌入式领域有着广泛的应用。

本教材采用教、学、做一体化模式,以核心知识、能力目标、任务驱动和实践环节为单元组织本教材的体系结构。每章都由 4 个模块:核心知识、能力目标、任务驱动和实践环节所构成。在语法上严格遵守 ANSI C 标准,在程序设计思想方面强调模块化思想,在克服难点方面注重结构清晰地安排内容、循序渐进地展开知识,特别强调知识点的能力目标,通过合理的任务驱动和实践环节提高程序设计能力和综合运用知识的能力。全书分为 13 章,分别讲解了初识 C 程序,基本数据类型,运算符与表达式,分支与开关语句,循环语句,函数的结构与调用,数组,指针,指针与数组,指针与函数,处理字符串,结构体、共用体与枚举,读写文件。

本书不仅适合作为高等院校理工类学生学习 C 程序设计的教材,而且特别适合作为教、学、做一体化的教材。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

C 程序设计教学做一体化教程/耿祥义,张跃平编著. —北京:清华大学出版社,2013.4

ISBN 978-7-302-31191-1

I. ①C… II. ①耿… ②张… 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(2013)第 002531 号

责任编辑:田在儒

封面设计:李 丹

责任校对:袁 芳

责任印制:杨 艳

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>, 010-62795764

印 装 者:北京密云胶印厂

经 销:全国新华书店

开 本:185mm×260mm 印 张:21

字 数:507 千字

版 次:2013 年 4 月第 1 版

印 次:2013 年 4 月第 1 次印刷

印 数:1~5000

定 价:39.00 元

产品编号:045586-01

前言 FOREWORD

本教材是作者多年讲授 C 语言的结晶,按照教、学、做一体化模式,以核心知识、能力目标、任务驱动和实践环节为单元组织本教材的体系结构。核心知识体现最重要和实用的知识,是教师需要重点讲解的内容;能力目标提出学习核心知识后应具备的编程能力;任务驱动给出了教师和学生共同来完成任务;实践环节给出了需要学生独立完成的实践活动。在语法上严格遵守 ANSI C 标准,在程序设计思想方面强调模块化思想,在克服难点方面注重结构清晰地安排内容、循序渐进地展开知识,通过合理的任务驱动提高程序设计的能力和综合运用知识的能力。全书分为 13 章,分别讲解了初识 C 程序,基本数据类型,运算符与表达式,分支与开关语句,循环语句,函数的结构与调用,数组,指针,指针与数组,指针与函数,处理字符串,结构体、共用体与枚举,读写文件。

每章的核心知识模块强调在编程中最重要和实用的知识点,其中的简单示例起着帮助读者理解和掌握核心知识的作用;能力目标模块强调使用核心知识进行编程的能力;任务驱动模块起着训练编程能力的作用,其中的任务小结主要总结任务中涉及的重要技巧、注意事项以及扩展知识,通过该模块的训练,读者有能力完成后续的实践环节。第 1 章主要讲解开发 C 程序的基本步骤,读者可以迅速地开发出第一个简单的 C 程序,并充分认识到 C 语言的重要地位和应用领域。第 2 章是基本数据类型,在核心知识和任务的安排方面特别注重训练初学者应当掌握和理解的重要基础知识以及知识点的能力目标。第 3 章是 C 语言的运算符与表达式,在任务安排上注重结合实际问题训练读者熟练地计算各种表达式,识别各种语句和它们的作用。第 4 章和第 5 章分别是分支语句和循环语句,为了实现能力目标特别注重选择有启发的例子和任务,以此训练读者使用所学语句解决实际问题的能力。第 6 章是本书的重点内容之一,讲述了函数的结构与调用,讲解上注重强调 ANSI C 标准,强调使用函数进行模块化设计的思想。第 7 章是数组,强调数组在解决许多实际问题中的重要性,特别安排了多个任务训练数组有关的排序算法。第 8 章、第 9 章以及第 10 章是 C 语言最重要的内容——指针,注重由简到难、逐步展开,便于读者学习和掌握这部分内容,所选择的任务都非常有利于读者理解指针以及训练在程序设计中使用指针解决问题的能力。由于字符串是 C 程序设计中经常需要处理的数据,因此在第 11 章专门讲解了和字符串有关的知识点,在任务安排上注重综合运用所学知识来解决实际问题。第 12 章讲解结构体、

共用体与枚举,特别注意训练如何使用指针访问结构体中的数据。第13章讲解如何使用库函数读写文件,结合一些有应用价值的任务,训练在编程中对文件的读写能力。

本教材特别注重引导学生参与课堂教学活动,适合高等院校相关专业作为教、学、做一体化的教材。

本教材的示例和任务模块的源程序以及电子教案可以在清华大学出版社网站上免费下载,以供读者和教学使用。

编 者

2013年4月

作者简介

耿祥义,1995年中国科学技术大学博士毕业,获理学博士学位。1997年从中山大学博士后流动站出站,现任大连交通大学教授。已编写出版《Java 2 实用教程》、《Java 面向对象程序设计》、《Java 设计模式》、《JSP 程序设计》、《XML 程序设计》、《C 程序设计实用教程》、《Java 课程设计》等10余部教材。

张跃平,现任大连交通大学副教授。已编写和参编出版《Visual FoxPro 课程设计》、《Java 2 实用教程》、《JSP 实用教程》、《C 程序设计实用教程》、《Java 课程设计》等教材。

目 录

CONTENTS

第 1 章 初识 C 程序	1
1.1 开发环境	1
1.2 简单的 C 程序	3
小结	9
习题 1	9
第 2 章 基本数据类型	10
2.1 整型常量与变量	10
2.2 浮点型常量与变量	17
2.3 字符常量与变量	21
2.4 输入、输出函数	26
2.5 符号常量与 const 常量	31
小结	34
习题 2	34
第 3 章 运算符与表达式	36
3.1 算术运算符与赋值运算符	36
3.2 自增、自减运算符	40
3.3 关系与逻辑运算符	43
3.4 类型转换运算符	47
3.5 位运算符	49
小结	52
习题 3	52
第 4 章 分支与开关语句	55
4.1 单条件、单分支语句	55
4.2 单条件、双分支语句	59
4.3 多条件、多分支语句	63
4.4 开关语句	67

4.5 复合语句的嵌套	70
小结	73
习题 4	73
第 5 章 循环语句	76
5.1 while 循环语句	76
5.2 do...while 循环语句	80
5.3 猜测数字	82
5.4 for 循环语句	85
5.5 记忆测试	90
5.6 continue 与 break 语句	92
小结	94
习题 5	94
第 6 章 函数的结构与调用	99
6.1 C 程序与函数	99
6.2 函数的类型与 return 语句	105
6.3 参数传值	108
6.4 非主函数之间的调用	112
6.5 void 型函数	116
6.6 函数的递归调用	119
6.7 局部变量与全局变量	122
6.8 变量的存储方式	125
6.9 使用库函数	128
小结	130
习题 6	130
第 7 章 数组	134
7.1 一维数组	134
7.2 数组名做参数	139
7.3 数组排序	143
7.4 二维数组	149
小结	152
习题 7	152
第 8 章 指针	156
8.1 指针变量	156
8.2 指针的自增、自减运算	161
8.3 malloc 函数与内存	164

8.4 指针类型的参数	167
8.5 指针与函数之间的交互	171
小结	174
习题 8	174
第 9 章 指针与数组	177
9.1 指向数组元素的指针	177
9.2 指针与下标运算	182
9.3 calloc 内存分配函数	185
9.4 指针数组	187
9.5 指向行的指针与二维数组	190
小结	193
习题 9	194
第 10 章 指针与函数	197
10.1 指向函数的指针变量	197
10.2 指向函数的指针做参数	200
10.3 返回地址的函数	202
小结	205
习题 10	205
第 11 章 处理字符串	207
11.1 char 型数组与字符串	207
11.2 指针与字符串	212
11.3 puts 函数与 gets 函数	215
11.4 检索字符串	219
11.5 字符串转换为数字	221
11.6 排序字符串	223
小结	226
习题 11	226
第 12 章 结构体、共用体与枚举	229
12.1 结构体类型与结构体变量	229
12.2 指针与结构体变量	232
12.3 结构体数组	234
12.4 结构体与函数	237
12.5 共用体	239
12.6 枚举类型	242
小结	245

习题 12	245
第 13 章 读写文件	248
13.1 按文本读取文件.....	248
13.2 写文本文件.....	253
13.3 读写二进制文件.....	256
13.4 随机读写.....	260
小结.....	263
习题 13	263
附录 A 实践参考答案	266
附录 B 标准 ASCII 表	300
附录 C 运算符表	301
附录 D math 与 string 库函数	302
附录 E 习题解答	304

初识 C 程序

主要内容

- 开发环境
- 简单的 C 程序

贝尔实验室是人们熟悉的著名实验室之一,C 语言就诞生于该实验室,发明人是该实验室的 Dennis Ritchie。1972 年,Dennis Ritchie 和同事 Ken Thompson 为了编写 UNIX 操作系统设计了一种新的实用语言,并使用该语言完成了 UNIX 操作系统的设计。Dennis Ritchie 在设计该语言时借鉴了同事 Ken Thompson 开发命名的 B 语言,由于 B 语言的名字来自当时的 BCPL(Basic Combined Programing Language)语言的第一个字母,为此,Dennis Ritchie 将其发明的新语言命名为 C 语言(取 BCPL 语言的第二个字母)。1978 年 Dennis Ritchie 出版了第一部 C 语言著作《The C Programming Language》。1983 年,美国国家标准化组织(ANSI)根据 Dennis Ritchie 的著作确定出了第一个 C 语言标准:C89,习惯地将 C89 称为 ANSI C。

C 语言高效、灵活,被广泛应用于科学计算,可以在许多软件开发中都可看到 C 语言的身影。C 语言是非常适合用于编写操作硬件设备的编程语言,特别是随着信息时代的不断进步,嵌入式系统的开发越来越流行,例如,人们使用 C 语言为汽车、飞行器、机器人、照相机、冰箱、DVD 播放机和其他许多现代化设备中的微处理器编写程序,以便控制这些设备,简而言之,嵌入式领域更加注重代码的效率,促成 C 语言在嵌入式领域独领风骚。

C 语言也是进一步学习其他高级语言,如 C++、Java 和 C#(读作 see-sharp),以及相关的课程,如数据结构、操作系统、微机原理等的基础语言。

1.1 开发环境

1.1.1 核心知识

学习任何一门编程语言都需要选择一种针对该语言的开发工具。开发工具的核心任务之一就是要将人们按照该语言编写的代码(称为源文件)转变成计算机能够识别执行的指令(称为机器指令)。

本书将采用 VC++ 6.0 开发环境来开发 C 程序,其主要理由是:VC++ 6.0 不仅是适合开发 C++ 程序的开发工具,同时也适合用来开发 C 程序,而且国内有关 C 语言的二级认证考试已经开始使用 VC++ 6.0,因此使用 VC++ 6.0 有利于学生更好地熟悉认证考试环境。

1.1.2 能力目标

安装、配置 VC++ 6.0,能启动 VC++ 6.0。

1.1.3 任务驱动

1. 任务的主要内容

- 安装 VC++ 6.0。
- 配置 Include files。
- 配置 Library files。

2. 任务的模板

按照下列步骤完成有关操作。

(1) 安装 VC++ 6.0

将 VC++ 6.0 安装到某个目录,如 D:\VC 6.0。选择“开始”→“程序”→“VC++ 6.0”命令启动 VC++ 6.0,出现如图 1.1 所示的开发界面。

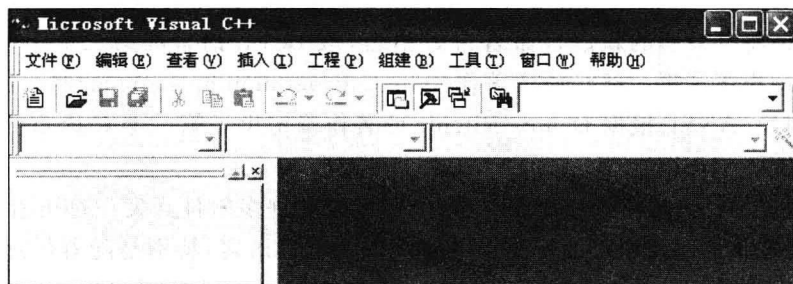


图 1.1 VC++ 6.0 的开发界面

(2) 配置 Include files

配置 Include files(如果安装时已经由系统自动配置过,可不必要重新配置)。

在图 1.1 所示的开发界面上选择“工具”菜单中的“选项”命令,弹出如图 1.2 所示的“选项”对话框。

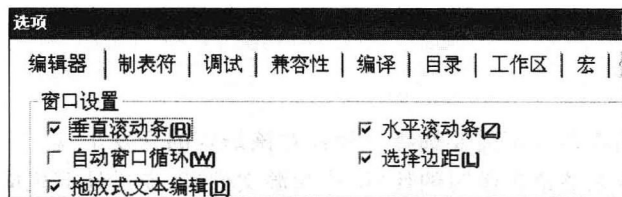


图 1.2 “选项”对话框

在图 1.2 所示的“选项”对话框中打开“目录”选项卡,然后将目录列表中的 Include files 选项对应的“路径”更改为:

```
D: \VC6.0\VC98\INCLUDE  
D: \VC6.0\VC98\MFC\INCLUDE  
D: \VC6.0\VC98\ATL\INCLUDE
```

(3) 配置 Library files

配置 Library files,如果安装时已经由系统自动配置过,可不必重新配置。在图 1.2 所示的“选项”对话框中打开“目录”选项卡,然后将目录列表中的 Library files 选项对应的“路径”更改为:

```
D: \VC6.0\VC98\LIB  
D: \VC6.0\VC98\MFC\LIB
```

3. 任务小结或知识扩展

要注意学会如何配置 Include files 和 Library files,其目的是为了能使用 VC++ 6.0 中的库函数。另外,尽管学习 C 语言可以选择任意一个成熟的开发工具,但 VC++ 6.0 是目前比较流行的开发工具。

1.1.4 实践环节

除了可以选择“开始”→“程序”→“VC++ 6.0”命令启动 VC++ 6.0 外,也可以执行 MSDEV.EXE 来启动 VC++ 6.0,双击 D:\VC6.0\COMMON\MSDEV98\BIN 目录中的 MSDEV.EXE 文件启动 VC++ 6.0。

1.2 简单的 C 程序

1.2.1 核心知识

无论 C 程序的规模大小如何,开发一个 C 程序须经过如下 4 个基本步骤。

1. 编写源代码文件

编写源代码文件,也简称为编写源文件。所谓源文件就是按照 C 语言的语法规则,使用文本编辑器编写的扩展名为.c 的文本文件,例如 first.c、second.c 等,也就是说 C 程序的源代码存放在扩展名为.c 的文本文件中。

2. 编译

计算机不能直接识别源代码文件,因此必须把源代码文件转化为计算机能够识别的机器指令。编译器将检查源代码文件中是否有语法错误;如果有语法错误,将提示有关错误;如果没有语法错误,编译器就会将源代码文件转化为一个二进制文件,该二进制文件被称为源代码文件的目标文件。目标文件的名字与源代码文件的名字相同,但扩展名为.obj。

3. 链接

目标文件是供链接器使用的文件,也就是说目标文件中含有待确定的链接信息,链接器必须把这些信息替换成真正的链接代码,形成一个完整的可执行的代码。当进行链接操作时,链接器还需要将程序使用的库函数链接进来(如 printf 函数就是库函数),并向可执行文件加入操作系统的启动代码。没有启动代码操作系统就无法运行可执行文件,该启动代码相当于程序和操作系统之间的接口。

4. 运行

产生可执行文件后,就可以把该文件交给操作系统去执行。

1.2.2 能力目标

能编写一个简单的 C 程序的源文件,并编译该源文件、链接目标文件得到可执行文件,然后运行可执行文件,即运行程序。

1.2.3 任务驱动

1. 任务的主要内容

编写一个简单程序,该程序输出两行文字:“很高兴学习 C 语言”和“We are students”。

2. 任务的模板

按照下列步骤完成有关操作。

- ① 创建工程。
- ② 向工程中添加源文件。
- ③ 编写源文件。
- ④ 编译。
- ⑤ 链接。
- ⑥ 运行。

(1) 创建名字是 myproject 的工程

VC++ 使用一个工程对应一个 C 程序,也就是说,在 VC++ 环境中,通过创建一个工程来创建一个 C 程序。

创建一个工程的步骤如下。

① 在 VC++ 开发界面上单击“文件”菜单,选择其中的“新建”菜单项,将弹出“新建”对话框,在该对话框中选择“工程”选项卡。

② 在当前对话框的左侧的选项列表中选中 Win32 Console Application(注意,不可以选择 Win32 Application)。

③ 在当前对话框的右侧的“位置”文本框中输入存放工程的位置,例如: D:\ok。

④ 在当前对话框中的“工程名称”文本框中输入工程的名称,例如: myproject。

⑤ 在弹出的“选择工程类型”对话框中选择“空工程(An empty project)”。

⑥ 在工作空间界面(VC++ 开发界面的左侧)的下方选择 FileView 视图。

(2) 向工程中添加名字是 first.c 的源文件

向工程中添加源文件的步骤如下。

- ① 在 VC++ 开发界面上单击“文件”菜单,选择其中的“新建”菜单项。
- ② 在弹出的“新建”对话框中选择“文件”选项卡。
- ③ 在当前对话框左侧的选项列表中选中 C++ Source File。
- ④ 在右侧的“文件名称”文本框中输入源文件的名称,如 first.c(必须带扩展名.c)。
- ⑤ 将“添加到工程”复选框选中,即把源文件添加到工程中。

(3) 编写源文件 first.c

在程序代码编辑区(VC++ 开发环境提供的一个文本编辑器)输入如下内容的源代码。

```
#include<stdio.h>
int main() {
    printf("很高兴学习 C 语言\n");           //输出"很高兴学习 C 语言",并换行
    printf("We are students\n");             //输出"We are students",并换行
    getchar();
    return 0;
}
```

(4) 编译源文件 first.c

在 VC++ 开发界面上单击“组建”(某些 VC++ 6.0 版本须单击“编译”)菜单,选择其中的“编译”菜单项对源文件进行编译,如果源文件没有错误,将产生目标文件;如果有错误,编译器将提示有关错误。生成的目标文件的名称是 first.obj(因为源文件的名称是 first.c)。

(5) 链接目标文件和库函数

在 VC++ 开发界面上单击“组件”菜单,选择其中的“构件(链接)”菜单项对目标文件以及所用的库函数进行链接,生成可执行文件。VC++ 6.0 将链接操作称为链接工程,生成的可执行文件的名称是 myproject.exe(因为工程的名称是 myproject)。

(6) 运行可执行文件

在 VC++ 开发界面上单击“组建”菜单,选择其中的“运行”菜单项运行可执行文件,即运行 myproject 工程。运行效果如图 1.3 所示。

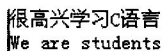


图 1.3 一个简单的程序

3. 任务小结或知识扩展

(1) main 函数

C 程序的基本结构就是函数,有时 C 程序可以只有一个函数,比如只有 main 函数(称为主函数)。除 main 函数外,C 程序还可以有多个其他函数,main 函数可以调用这些函数,比如调用 printf 库函数。现在只需知道:一个 C 程序必须有且仅有一个 main 函数,操作系统从 main 函数开始执行 C 程序。按照 ANSI C 标准,main 函数的格式如下:

```
int main(){
    ...
    return 0;
}
```

main 前面的 int 称为它的类型(要求 main 使用 return 返回一个整数值,通常返回 0 即可),main 后面的一对大括号表明这是一个函数,紧接着的一对大括号以及所包含的内容是它的函数体。

(2) printf 函数

printf 函数是库函数,其作用是可以输出数据到终端设备,比如输出一行文本,输出的文本中如果包含换行转义符: \n,那么 printf 在输出文本时将\n 输出为换行。为了使用 printf 库函数,需要在源文件中使用 #include 命令包含 stdio.h 函数库。

(3) 大括号的风格

C 程序的许多地方都涉及使用一对大括号,比如 main 函数的函数体。“行尾”风格是左大括号在上一行的行尾,而右大括号独占一行。“独行”风格是左、右大括号各自独占一行。下列是行尾风格:

```
int main() {  
  
}
```

(4) 注释

编译器忽略注释内容,注释的目的是有利于代码的维护和阅读,因此给代码增加注释是一个良好的编程习惯。C 编译器支持两种格式的注释:单行注释和多行注释。

单行注释使用“//”表示单行注释的开始,即该行中从“//”开始的后续内容为注释。多行注释的使用“/* */”表示注释的开始,以“*/”表示注释结束,例如:

```
/* 以下是 main 方法,操作系统首先执行该方法 */  
int main() {  
    printf("你好");      //输出你好  
    return 0;  
}
```

(5) 关于编译

C 语言是面向过程语言,简单地说,编程人员只需按照 C 语言的语法编写解决问题的过程,C 语言使用函数封装这样的过程(在第 6 章还会详细讲述)。过程语言的源文件的一个特点是更接近人的“自然语言”,例如,C 语言源程序中的一个函数:

```
int max(int a,int b){  
    if(a>b)  
        return a;  
    else  
        return b  
}
```

该函数负责计算两个整数的最大值。使用面向过程语言,人们只需按照自己的意图来编写各个函数,语言的语法更接近人们的自然语言,所以习惯上也称过程语言是高级语言。但是,无论哪种高级语言编写的源文件,计算机都不能直接执行,因为计算机只能直接识别、执行机器指令。因此,必须把源文件转换成机器指令,然后计算机去执行相应的机器指令。所谓机器指令就是可以被计算机平台直接识别、执行的一种由 0、1 组成的序列代码。每种计算机都会形成自己独特的机器指令,例如,某种计算机可能用 8 位序列代码 1000 1111 表示一次加法操作,以 1010 0000 表示一次减法操作,而另一种计算机可能用 8 位序列代码 1010 1010 表示一次加法操作,以 1001 0011 表示一次减法操作。因此必须把源代码转化为

计算机能够识别的机器指令。

编译器将检查源文件中的代码是否有语法错误,如果有语法错误,将提示有关错误;如果没有语法错误,编译器就会将源文件转化为一个二进制文件,该二进制文件被称作源文件的目标文件。目标文件的名字与源文件的名字相同,但扩展名为 .obj(某些编译器得到的目标文件的扩展名为 .O)。目标文件是供链接器使用的文件(不是可执行文件),也就是说目标文件中含有待确定的链接信息,链接器必须把这些信息替换成真正的链接代码,形成一个完整的可执行的代码。

1.2.4 实践环节

1. 控制回行(\n)

将任务驱动中的程序的代码: `printf("很高兴学习 C 语言\n");`,更改为: `printf("很高兴\n学习 C 语言\n");`,观察程序输出结果会是怎样?

2. 看看自己的工程

如果工程 myproject 建立在 D:\ok 目录中,那么在该目录下的 Debug 子目录中可以看到名字为 myproject.exe 的可执行文件,运行该文件(双击该文件)也称运行工程。

3. 多个源文件的工程

一个 C 程序(工程)中可以有多个源文件,这些源文件包含 C 程序所需要的函数。

(1) 建立名字为 jiafa 的空工程

在 VC++ 开发界面上单击“文件”菜单,选择其中的“新建”菜单项,将弹出“新建”对话框,在该对话框中选择“工程”选项卡,输入工程名 jiafa,如图 1.4 所示。要特别注意在当前对话框左侧的选项列表中选中 Win32 Console Application 选项,不可以选择 Win32 Application)。



选中 Win32 Console Application

图 1.4 输入工程名

(2) 向工程添加 add.c 源文件

在 VC++ 开发界面上单击“文件”菜单,选择其中的“新建”菜单项,在弹出的“新建”对话框中选择“文件”选项卡,并在当前对话框左侧的选项列表中选中“C++ Source File”,在右侧的“文件名称”文本框中输入源文件的名称: add.c(注意必须带扩展名.c),并将“添加到工程”复选框选中,即将 add.c 添加到 jiafa 工程中,然后在 add.c 的编辑区输入如图 1.5

所示的内容。

(3) 向工程添加 main.c 源文件

在 VC++ 开发界面上单击“文件”菜单,选择其中的“新建”菜单项,在弹出的“新建”对话框中选择“文件”选项卡,并在当前对话框左侧的选项列表中选中“C++ Source File”,在右侧的“文件名称”文本框中输入源文件的名称: main.c(注意必须带扩展名.c),并将“添加到工程”复选框选中,即将 main.c 添加到 jiafa 工程中,然后在 main.c 的编辑区输入如图 1.6 所示的内容。

```
int add(int ,int);
int add(int a,int b) {
    return a+b;
}
```

图 1.5 编辑源文件 add.c

```
/* main.c */
#include <stdio.h>
int add(int,int); //主函数要使用的add函数的原型
int main() {
    int result=0;
    result=add(12,67);
    printf("%d\n",result);
    result=add(-212,30);
    printf("%d\n",result);
}
```

图 1.6 编辑源文件 main.c

(4) 分别编译 add.c 和 main.c

让 add.c 的编辑区处于活动状态,然后编译 add.c;让 main.c 的编辑区处于活动状态,然后编译 main.c。如果没有语法错误,将产生名字为 add.obj 和 main.obj 的目标文件;如果有错误,编译器将提示有关错误。

(5) 链接

单击“组建”菜单,选择其中的“链接”菜单项对目标文件和所用的库函数进行链接,生成可执行文件。

(6) 运行

链接成功后,在 VC++ 开发界面上单击“组建”菜单,选择其中的“运行”菜单项运行可执行文件(也叫运行工程),效果如图 1.7 所示。

创建具有多个源文件的工程的实践步骤如图 1.8 所示。

79
182

图 1.7 程序运行效果

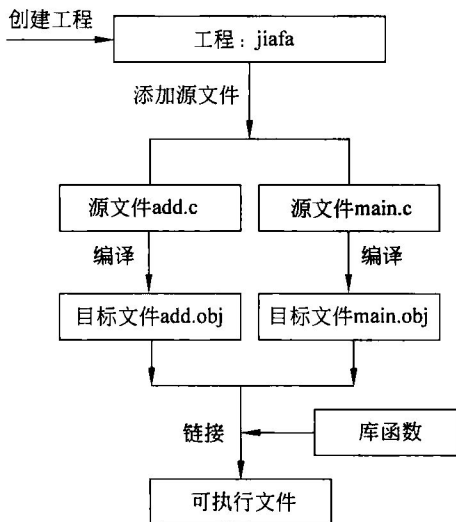


图 1.8 有多个源文件的工程