

# C语言程序 设计项目教程

宋海燕◎主编

 北京理工大学出版社  
BEIJING INSTITUTE OF TECHNOLOGY PRESS

# C 语言程序设计项目教程

主 编 宋海燕

副主编 陈瑞芳 哈里白

 **北京理工大学出版社**  
BEIJING INSTITUTE OF TECHNOLOGY PRESS

版权专有 侵权必究

---

图书在版编目 (CIP) 数据

C 语言程序设计项目教程 / 宋海燕主编. —北京: 北京理工大学出版社, 2016.8  
ISBN 978-7-5682-1746-0

I. ①C… II. ①宋… 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 (2016) 第 316360 号

---

---

出版发行 / 北京理工大学出版社有限责任公司

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)  
(010) 82562903 (教材售后服务热线)  
(010) 68948351 (其他图书服务热线)

网 址 / <http://www.bitpress.com.cn>

经 销 / 全国各地新华书店

印 刷 / 北京泽宇印刷有限公司

开 本 / 787 毫米×1092 毫米 1/16

印 张 / 10.5

字 数 / 247 千字

版 次 / 2016 年 8 月第 1 版 2016 年 8 月第 1 次印刷

定 价 / 42.00 元



责任编辑 / 钟 博

文案编辑 / 钟 博

责任校对 / 周瑞红

责任印制 / 李志强

---

图书出现印装质量问题, 请拨打售后服务热线, 本社负责调换

# 前 言

C 语言作为一门重要的软件开发入门课程，是学习程序设计的首选，其语法灵活、易学易用，深受广大程序设计人员的青睐。

本书以项目案例为载体呈现 C 语言的知识点和技能点，在项目选取上，精选贴近生活的趣味应用实例，力争做到复杂问题简单化，简单问题实用化，旨在培养学生的程序设计思想，训练学生编写和调试程序的能力，突出“以学生为中心”的教育理念。本书整合了数据结构的部分知识点，遴选了十二个项目，将每个项目分解成若干个学习任务，项目编写遵循“项目—知识链接—技能训练—实力检验—小结”的模式，深入浅出。“项目”选择依据由易到难、由浅入深的原则，“知识链接”详细介绍实现项目所需要的知识点，“技能训练”选取实用案例训练知识点的应用能力，“实力检验”选取相关的二级考试题检验知识点的理解程度，“小结”回顾该项目涉及的相关知识。

本书由宋海燕任主编，由陈瑞芳、哈里白任副主编。其中，项目一、二、三、四、五、六、七由宋海燕编写，项目八、九、十一由哈里白编写，项目十、十二由陈瑞芳编写。由于编者水平有限，书中难免有疏漏和不妥之处，恳请读者批评指正。

编 者

# 目 录

项目一 认识软件开发	1
任务 1.1 软件开发流程	1
任务 1.2 软件开发的相关课程	2
任务 1.3 认识 C 语言	6
技能训练	6
实力检验	6
项目二 自我介绍	8
任务 2.1 熟悉 DEV-C++集成环境	8
任务 2.2 熟悉 C 源程序的构成	11
任务 2.3 格式输出函数 printf 的使用	13
技能训练	15
实力检验	15
项目三 计算器	16
任务 3.1 程序流程图	17
任务 3.2 数据类型	18
任务 3.3 常量与变量	19
任务 3.4 运算符与表达式	21
任务 3.5 格式化输入函数 scanf 的使用	27
技能训练	28
实力检验	28
项目四 体型测试	30
任务 4.1 if 语句	32
任务 4.2 if...else 语句	33
任务 4.3 if...elseif...else 语句	34
任务 4.4 if 语句的嵌套	36
任务 4.5 switch 语句	37
任务 4.6 条件运算符	39
技能训练	40
实力检验	41
项目五 猜数游戏	44
任务 5.1 while 语句	46
任务 5.2 do...while 语句	48
任务 5.3 for 语句	49

---

任务 5.4	break	52
任务 5.5	continue	52
技能训练		53
实力检验		53
项目六	技能大赛评分	56
任务 6.1	一维数组	58
任务 6.2	二维数组	61
任务 6.3	排序方法	63
技能训练		68
实力检验		69
项目七	投票统计	72
任务 7.1	认识结构体	74
任务 7.2	结构体数组	77
任务 7.3	typedef 的应用	78
技能训练		79
实力检验		79
项目八	小学数学能力测试程序	82
任务 8.1	无参函数的定义及调用	87
任务 8.2	有参函数的定义及调用	88
任务 8.3	return 语句的使用	89
任务 8.4	数组名作为函数参数	90
任务 8.5	函数的嵌套调用	91
任务 8.6	函数的递归调用	92
任务 8.7	变量的作用域	94
任务 8.8	变量的存储	95
技能训练		96
实力检验		97
项目九	字符串转换	100
任务 9.1	指针的定义	102
任务 9.2	指针与数组	103
任务 9.3	字符串操作	104
任务 9.4	指针与字符串	108
任务 9.5	指针与结构体	109
任务 9.6	指针与函数	110
技能训练		112
实力检验		113
项目十	通信录	116
任务 10.1	初识单链表	123
任务 10.2	建立动态链表	125

---

任务 10.3 链表插入运算	128
任务 10.4 链表查找	129
任务 10.5 链表删除运算	131
技能训练	134
实力检验	134
<b>项目十一 学生信息录入查询系统</b>	<b>136</b>
任务 11.1 文件基础	139
任务 11.2 文件的打开与关闭	139
任务 11.3 文件读写操作	140
任务 11.4 文件的定位操作	143
技能训练	144
实力检验	145
<b>项目十二 二叉树</b>	<b>147</b>
任务 12.1 初识二叉树	150
任务 12.2 二叉树的遍历	153
技能训练	156
实力检验	156

# 项目一 认识软件开发



## 学习目标

- (1) 了解软件开发流程。
- (2) 了解软件开发的相关课程。
- (3) 了解 C 语言的发展过程及特点。



## 重点和难点

- (1) 软件开发过程。
- (2) 对软件工程、数据结构和算法的理解。

软件开发作为计算机应用的主流，其流程是怎么样的呢？涉及哪些相关课程呢？C 语言又是在怎样的情况下诞生的呢？

## 任务 1.1 软件开发流程

软件的开发使用从需求分析开始到软件报废处理，经历 10 个步骤，如图 1-1 所示。

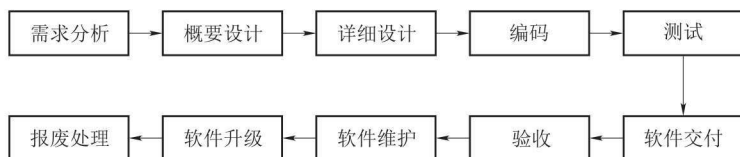


图 1-1 软件开发流程

(1) 需求分析：系统分析员向用户了解需求，然后用列出要开发的系统的大功能模块、每个大功能模块有哪些小功能模块，确定用户需要的界面，并与用户确认。

(2) 概要设计：其也称为系统设计。概要设计需要对软件系统的设计进行考虑，包括系统的基本处理流程、系统的组织结构、模块划分、功能分配、接口设计、运行设计、数据结构设计和出错处理设计等，为软件的详细设计提供基础。

(3) 详细设计：在详细设计中，描述实现具体模块所涉及的主要算法、数据结构、类的层次结构及调用关系，需要说明软件系统各个层次中的每一个程序（每个模块或子程序）的设计考虑，以便进行编码和测试。应当保证软件的需求完全分配给整个软件。详细设计应当足够详细，能够根据详细设计报告进行编码。

(4) 编码：开发者根据《软件系统详细设计报告》中对数据结构、算法分析和模块实现等方面的设计要求，开始具体的编写程序工作，分别实现各模块的功能，从而实现目标系统的功能、性能、接口、界面等方面的要求。在规范化的研发流程中，编码工作在整个项目流



程里的比例最多不会超过 1/2，通常为 1/3。

(5) 测试：测试编写好的系统及系统各个功能模块。软件测试有很多种：按照测试执行方，可以分为内部测试和外部测试；按照测试范围，可以分为模块测试和整体联调；按照测试条件，可以分为正常操作情况测试和异常情况测试；按照测试的输入范围，可以分为全覆盖测试和抽样测试。

(6) 软件交付：在软件测试证明软件达到要求后，软件开发者应向用户提交开发的目标安装程序、数据库的数据字典、《用户安装手册》、《用户使用指南》、需求报告、设计报告、测试报告等双方合同约定的产物。

(7) 验收：用户对开发方交付的产品进行合格验收。

(8) 软件维护：其包括软件数据库管理、用户跟踪培训、故障分析解决等维护工作。

(9) 软件升级：其包括需求调整分析、软件功能拓展、优化系统等工作。

(10) 报废处理：当软件不能适应业务发展或者新软件项目重新立项时，应对原有的软件进行报废处理。

## 任务 1.2 软件开发的相关课程

### 1. 软件工程

20 世纪 60 年代中后期，软件数量越来越多，规模越来越大，而软件的生产基本上是各自为战，缺乏科学规范的系统规划与测试、评估标准，这致使大批耗费巨资研制出来的软件系统，因含有错误而无法使用，造成了巨大损失。由于问题软件不断涌现，人们感觉软件越来越不可靠。这一切极大地震撼了计算机界，历史上称之为“软件危机”。一些软件开发人员认识到：大型程序的编制不同于小程序的编写，它应该是一项新的技术，应该像管理工程一样管理软件研制的全过程。于是，1968 年北大西洋公约组织在联邦德国的一次学术会议上，首次提出了“软件工程”的概念，从而形成一门新兴的学科——软件工程。软件工程是一门用工程化方法，指导计算机软件开发和维护的学科，它采用工程的概念、原理、技术和方法来开发、维护以及管理软件。软件工程是目前计算机学科中一个年轻并且充满活力的研究领域。

软件工程是一门用工程化方法，指导计算机软件开发和维护的学科，它采用工程的概念、原理、技术和方法来开发、维护以及管理软件。软件工程是目前计算机学科中一个年轻并且充满活力的研究领域。

软件工程包括 3 个要素，即方法、工具和过程。方法是完成软件工程项目的手段。它包括诸多方面，例如项目计划与估算、需求分析、系统总体结构设计、数据结构设计、算法设计、代码编写、软件测试与维护等。工具是人类在开发软件的活动中智力和体力的延伸和扩展，它提供了自动或半自动的软件支撑环境。过程贯穿于软件开发的各个环节，支持软件开发各个环节的控制和管理。

软件工程是一门工程性学科，目的是成功地建造一个大型软件系统，同时达到这样几个目标：较低的成本、较好的性能、完善的功能、较好的移植性、较低的维护费用、及时的交付以及较高的可靠性等。

## 2. 数据结构

数据结构作为一门独立的课程始于 1968 年,我国在 20 世纪 80 年代初出现该课程。早期的数据结构对课程的范围没有明确的规定,课程内容几乎和图论、树的理论是相同的。20 世纪 60—70 年代,随着大型程序的出现,软件业相对独立,计算机处理的数据大部分已由数值型向非数值型转化,结构程序设计逐步成为程序设计方法学的主要内容,人们已经认识到程序设计的实质就是对所确定的问题选择一种好的结构,从而设计一种好的算法。数据结构承担了这一重任,它将非数值数据归类,分类描述数据的逻辑结构、存储结构和一些基本运算。

数据结构是指相互之间存在一种或多种特定关系的数据元素的集合,是计算机存储和组织数据的方式。它可分为逻辑结构和存储结构两大类。

数据的逻辑结构是对数据元素之间逻辑关系的描述,是数据的组织形式。数据元素之间有 4 种基本逻辑结构:集合、线性结构、树形结构和图形结构。

(1) 集合。集合中任意两个数据元素间没有逻辑关系,组织形式松散,如图 1-2 所示。

(2) 线性结构。数据元素间构成一种顺序的线性关系,如图 1-3 所示。

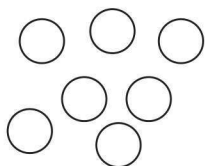


图 1-2 集合



图 1-3 线性结构

(3) 树形结构。数据结构具有分支、层次特性,数据元素间形成一种一对多的关系,如图 1-4 所示。

(4) 图形结构。其也称网状结构,是最复杂的一种数据结构,结构中数据元素之间存在多对多的关系,如图 1-5 所示。

数据的逻辑结构是独立于计算机的,与数据在计算机中的存储无关。但是,要使用计算机对数据进行处理,就必须将数据存储于计算机中,数据的逻辑结构在计算机中的表示称为数据的存储结构,也称为物理结构。数据的存储结构有 4 种:顺序存储结构、链式存储结构、索引存储结构和散列存储结构。

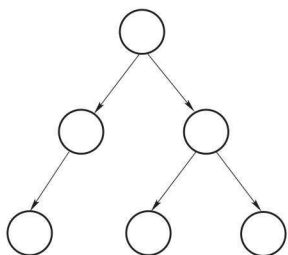


图 1-4 树形结构

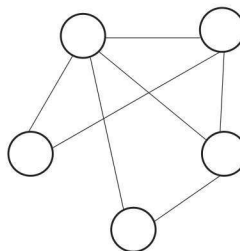


图 1-5 图形结构

(1) 顺序存储结构。顺序存储结构通常用于存储具有线性结构的数据。将逻辑上相邻的结点连续地存放在存储器中,使得逻辑上相邻的结点在物理位置上也是一定是相邻的。

(2) 链式存储结构。结点在存储器中随意存放,结点之间的物理关系与逻辑关系无关。

链式存储方式是给每个结点附加一个指针字段，一个结点的指针所指的是该结点的后继结点的存储地址。

(3) 索引存储结构。索引存储结构中的每个结点仅含一个数据元素，所有结点连续存放。此外，增设一个索引表，索引表中的索引号指示各结点的存储位置，根据结点的索引号可以确定该结点的存储地址。

(4) 散列存储结构。散列存储的思想是构造一个从集合  $K$  到存储区  $M$  的函数  $h$ ，该函数的定义域为  $K$ ，值域为  $M$ ， $K$  中的每个结点  $k_i$  在计算机中的存储地址由  $h(k_i)$  确定。

### 3. 算法

算法是对特定问题求解步骤的一种描述，提供了解决问题的思想、方法和步骤。人们在生产活动和日常生活中经常会用到算法。例如，购物的“算法”是：确定购买商品，确定购物商场，到达商场，购买商品，返回。

一个算法应具有以下 5 个特性：

(1) 有穷性。一个算法应包含有限的操作步骤，在执行有穷步后结束。对于包含循环结构的算法应避免出现死循环，否则算法就会无限制地被执行下去。

(2) 确定性。算法中的每一个步骤必须有确切的含义，不会产生二义性，并且在任何条件下，算法只有唯一的一条执行路径，对于相同的输入只能得到相同的结果。

(3) 有效性。算法的每一个步骤都应当能够有效地执行，并得到确定的结果。例如，若  $b=0$ ，那么  $a/b$  就不能有效执行。

(4) 有零个或多个输入。输入是指在执行算法时需要从外界取得的必要的信息，如果一个算法在程序中已经给出了原始数据，那么此算法可以没有输入。

(5) 有一个或多个输出。算法的目的是求解问题，没有输出的算法是没有意义的。

描述一个算法的方法有多种，常用的有自然语言、类语言、流程图、N-S 图等。一个算法的优劣直接关系到程序执行的效率，在评价一个算法的优劣时，通常用到的指标是空间复杂度和时间复杂度。

### 4. 程序及程序设计语言

计算机系统如何才能有条不紊地工作呢？计算机系统的工作是由事先设计好的程序来控制的。人们根据自己的需要，把需要计算机做的工作编制成计算机程序，并把程序输入计算机，然后发出运行程序的命令使程序执行。程序开始执行后计算机的控制器从程序的第一条指令开始，逐条地读取指令进行翻译，然后按指令的规定和要求指挥整个计算机系统工作，直到程序结束，完成人们交给计算机完成的工作。

程序是为解决特定问题或实现特定目标而用计算机语言编写的命令的集合。著名的计算机学家尼古拉斯·沃斯 (Niklaus Wirth) 对程序定义为

$$\text{程序} = \text{算法} + \text{数据结构}$$

这是著名的 Wirth 公式。该公式说明，程序是由两部分组成的——算法和数据结构。其中，算法是对特定问题求解步骤的描述，数据结构是对数据的描述。

程序设计包括原始数据、计算方法和输出结果三个部分。

(1) 原始数据。多数程序都需要原始数据，所以在程序设计之初，一般都要确定数据类型和数据的初始值。

(2) 计算方法。计算方法是程序设计的核心和关键，用来处理原始数据、解决特定的问题。

(3) 输出结果。程序都应该有结果，只是表达方式不同，有的需要在屏幕上显示，有的需要从打印机上输出，有的只是控制计算机的运行。

语言是交流的工具，自然语言是人与人之间交流的工具，而程序设计语言是人与计算机交流的工具。语言的基础是一组记号和一组规则。根据规则由记号构成的记号串的总体就是语言。程序设计语言是用于书写计算机程序的语言。在程序设计语言中，这些记号串就是程序。程序设计语言有两个方面的因素，即语法和语义。语法表示构成语言的各个记号之间的组合规律，但不涉及这些记号的特定含义。语义表示程序的含义。自 20 世纪 60 年代以来，世界上公布的程序设计语言已有上千种之多，但是只有很小一部分得到了广泛的应用。从发展历程来看，程序设计语言可以分为 4 代：第一代语言（机器语言）、第二代语言（汇编语言）、第三代语言（高级语言）、第四代语言（面向对象语言）。

#### 1) 机器语言

机器语言是由一串串“0”和“1”组成的指令序列，如 CPU 指令序列中的“加指令”是 10000000，“减指令”是 10010000。由此可见，使用机器语言编写程序是一件非常烦琐的事情，特别是在程序出错需要修改时，工作就更加困难，而且这样编写的程序不便于记忆、阅读和书写，还容易出错。由于每台计算机的指令系统各不相同，所以，在一台计算机上执行的程序，要想在另一台计算机上执行，必须重新编写，其可移植性差。但由于使用的是针对特定型号计算机的语言，故而运算效率是所有语言中最高的。

#### 2) 汇编语言

为了克服机器语言难度大、难掌握和易出错的缺点，用与代码指令实际含义相近的英文缩写词、字母和数字等符号取代指令代码。例如，用“ADD”代表加法，用“MOV”代表数据传递等，这种程序设计语言称为汇编语言。汇编语言也是面向机器的语言，使用起来比较烦琐，通用性差。但是，用汇编语言编写的程序，其目标程序所占用的内存空间少，运行速度快，有着高级语言不可替代的作用。

#### 3) 高级语言

不论是机器语言还是汇编语言，它们都是面向硬件操作的，低级语言对机器的过分依赖，要求使用者必须对硬件结构及其工作原理都十分熟悉，这对非计算机专业的人员来说是很难的，不利于计算机的应用和推广。计算机的发展促使人们设计出与自然语言相接近、通用易学的计算机语言，即高级语言。高级语言的特点是在一定程度上与机器无关、易学、易维护、面向对象的语言，如 Pascal 语言、Basic 语言、C 语言等都属于高级语言。

#### 4) 面向对象语言

从 20 世纪 80 年代开始，在软件设计思想上，又产生了一次革命，其结果就是面向对象的程序设计。面向对象的程序设计可以更直接地描述客观世界存在的事物（即对象）及事物之间的相互关系。面向对象技术强调的基本原则是直接对客观事物本身进行抽象，并在此基础上进行软件开发，将人类的思维方式与表达方式直接应用在软件设计中。当前比较流行的 C++、C#、Java 等都属于面向对象的程序设计语言。

## 任务 1.3 认识 C 语言

早期的操作系统等系统软件主要是用汇编语言编写的，如 UNIX 操作系统。由于汇编语言依赖计算机硬件，程序的可读性和可移植性都比较差。为了提高可读性和可移植性，最好改用高级语言，但一般高级语言难以实现汇编语言的某些功能，而汇编语言可以直接对硬件进行操作，例如，对内存地址的操作、位（bit）操作等。人们设想能否找到一种既具有一般高级语言的特性，又具有低级语言的特性的语言，集它们的优点于一身。于是，C 语言就在这种情况下应运而生了，之后成为国际上广泛流行的计算机高级语言。它适合作为系统描述语言，既可用于写系统软件，也可用于写应用软件。

C 语言的发展颇为有趣，它的原型是 ALGOL 60 语言（也称为“A 语言”）。1963 年，剑桥大学将 ALGOL 60 语言发展成为 CPL（Combined Programming Language）语言。1967 年，剑桥大学的 Martin Richards 对 CPL 语言进行了简化，于是产生了 BCPL 语言。1970 年，美国贝尔实验室的 Ken Thompson 对 BCPL 进行了修改，并为它起了一个有趣的名字“B 语言”，意思是将 CPL 语言煮干，提炼出它的精华，并且用 B 语言写了第一个 UNIX 操作系统。1973 年，B 语言也被人“煮”了一下，美国贝尔实验室的 D.M.RITCHIE 在 B 语言的基础上最终设计出了一种新的语言，他取 BCPL 的第二个字母作为这种语言的名字，这就是 C 语言。为了使 UNIX 操作系统推广，1977 年 Dennis M.Ritchie 发表了不依赖具体机器系统的 C 语言编译文本《可移植的 C 语言编译程序》。1978 年 Brian W.Kernighan 和 Dennis M.Ritchie 出版了名著《The C Programming Language》，从而使 C 语言成为目前世界上最广泛流行的高级程序设计语言。1988 年，随着微型计算机的日益普及，出现了许多 C 语言版本，由于没有统一的标准，这些 C 语言之间出现了一些不一致的地方，为了改变这种情况，美国国家标准研究所（ANSI）为 C 语言制定了一套 ANSI 标准，其成为现行的 C 语言标准。

C 语言作为一种结构化语言，层次清晰，便于按模块化方式组织程序，易于调试和维护。其表现能力和处理能力极强。它不仅具有丰富的运算符和数据类型，便于实现各类复杂的数据结构，还可以直接访问内存的物理地址，进行位（bit）一级的操作。此外，C 语言还具有效率高、可移植性强等特点，因此被广泛地移植到各类型的计算机上，从而形成了多种版本的 C 语言。目前最流行的 C 语言有以下几种：Microsoft C 或称 MS C；Borland Turbo C 或称 Turbo C；AT&T C。这些 C 语言版本不仅实现了 ANSI C 标准，而且在此基础上各自作了一些扩充，使之更加方便、完美。

## 技能训练

阅读相关软件开发书籍，写一篇关于软件开发的论文。

## 实力检验

1. 软件开发有哪些步骤？
2. 软件开发相关的几门课程在软件开发中起到了什么作用？

3. 什么是 C 语言？
4. 软件工程的三要素是\_\_\_\_\_、\_\_\_\_\_和\_\_\_\_\_。
5. 数据结构主要研究数据的\_\_\_\_\_、\_\_\_\_\_和\_\_\_\_\_和\_\_\_\_\_数据的基本运算三个方面。
6. 数据结构包括数据的\_\_\_\_\_结构和\_\_\_\_\_结构。
7. 数据的逻辑结构包括\_\_\_\_\_、\_\_\_\_\_、\_\_\_\_\_和\_\_\_\_\_。
8. 数据的存储结构包括\_\_\_\_\_、\_\_\_\_\_、\_\_\_\_\_和\_\_\_\_\_。
9. 评价一个算法优劣的指标是\_\_\_\_\_和\_\_\_\_\_。
10. 算法的五个特性是\_\_\_\_\_、\_\_\_\_\_、\_\_\_\_\_、输入和输出。

## 小 结

本项目介绍了软件开发流程和与软件开发相关的软件工程、数据结构、算法、程序及程序设计语言等课程，通过学习这些相关知识，引导学生了解软件开发，熟悉软件开发流程，为进一步学习 C 语言奠定基础。

# 项目二 自我介绍



## 学习目标

- (1) 熟悉 DEV-C++集成环境。
- (2) 掌握 C 语言源程序的构成。
- (3) 了解 C 语言源程序的书写规范。
- (4) 掌握 printf 函数的使用。



## 重点和难点

- (1) C 语言源程序的构成及主程序框架。
- (2) printf 函数的使用方法。

新同学见面，向大家介绍一下自己吧，该如何向屏幕上输出自我介绍信息呢？

### 【项目源码】

```
#include <stdio.h>
main()
{
    printf("各位同学大家好，欢迎大家来到软件工程系！我是大家的 C 语言老师小颖，希望和大家共同度过美好的一学期!\n");
    system("pause");
}
```

### 【运行结果】

项目二的运行结果如图 2-1 所示。

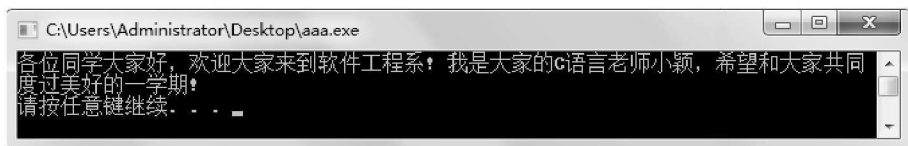


图 2-1 项目二的运行结果

## 任务 2.1 熟悉 DEV-C++集成环境

(1) DEV-C++的安装。

- ① 双击“DEV-C++ 5”，出现介绍界面，如图 2-2 所示。

② 选择显示语言，如图 2-3 所示。



图 2-2 DEV-C++的安装界面



图 2-3 选择显示语言

③ 安装向导，点击“下一步”按钮，如图 2-4 所示。

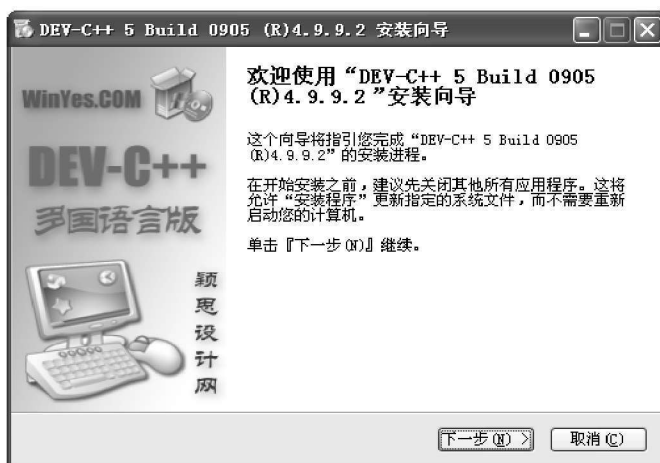


图 2-4 安装向导

④ 选择组件，如图 2-5 所示。



图 2-5 选择组件

⑤ 选择安装位置，如图 2-6 所示。



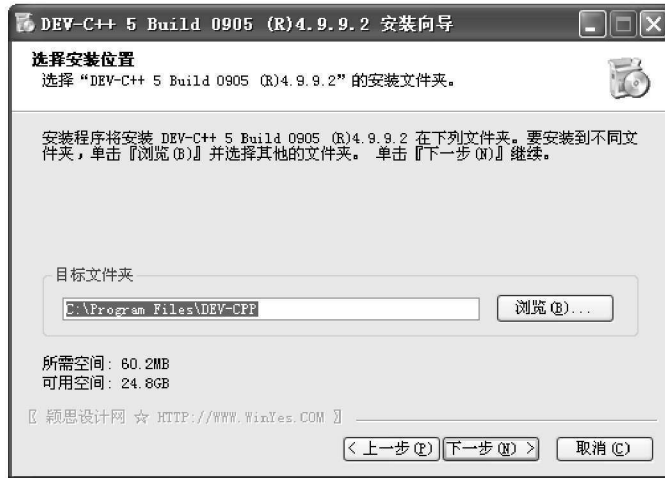


图 2-6 安装位置

⑥ 选择“开始菜单”选项，点击“安装”按钮，如图 2-7 所示。

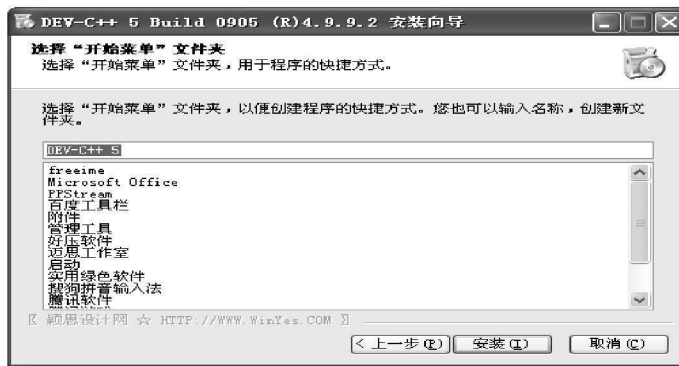


图 2-7 选择“开始菜单”文件夹

⑦ 提示安装完成，点击“完成”按钮，完成安装过程，如图 2-8 所示。

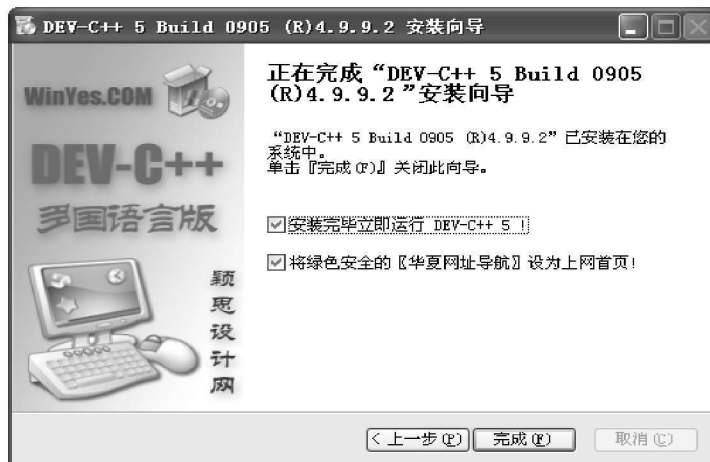


图 2-8 安装完成