



普通高等教育“十一五”国家级规划教材

普通高等教育“十三五”数字化规划教材

C语言 程序设计教程

第4版

主 编 杨路明



- 精美的PPT电子教案
- 详细的习题答案和程序源代码
- 专属的APP提供海量视频、学习文案
- 在线课堂，签到、编程、问答一体化



北京邮电大学出版社
www.buptpress.com



普通高等教育“十一五”国家级规划教材

普通高等教育“十三五”数字化规划教材

C 语言程序设计教程

(第 4 版)

主 编 杨路明

副主编 谭长庚 沙 莎 严 晖



广益教育“九斗”
APP 操作说明

北京邮电大学出版社

· 北京 ·

内 容 简 介

C语言在当今软件开发领域有着广泛的应用。本书全面地介绍了C语言的基本概念,C语言的数据类型、语句及结构特点,系统地讲述了C语言程序设计的基本方法和技巧。

全书选材先进,内容丰富,讲述力求理论联系实际、深入浅出、循序渐进,注重培养读者的程序设计能力,培养良好的程序设计风格和习惯。书中还给出了全国计算机等级考试二级C语言程序设计涉及的考点分析,以及试题解析等内容,供参加二级C语言等级考试的读者学习参考。

本书可作为高等院校计算机专业或其他专业的计算机程序设计教学用书,也可作为从事计算机应用的科技人员的参考书、培训教材。为了配合本书的学习,作者还编写了与本书配套的《C语言程序设计上机指导及习题选解》(第4版),可供学习时参考使用。

图书在版编目(CIP)数据

C语言程序设计教程/杨路明主编. —4版. —北京:北京邮电大学出版社,2018.11

ISBN 978-7-5635-5604-5

I. ①C… II. ①杨… III. ①C语言—程序设计—教材 IV. ①TP312.8

中国版本图书馆CIP数据核字(2018)第224768号

书 名	C语言程序设计教程(第4版)
主 编	杨路明
责任编辑	向 蕾
出版发行	北京邮电大学出版社
社 址	北京市海淀区西土城路10号(100876)
电话传真	010-82333010 62282185(发行部) 010-82333009 62283578(传真)
网 址	www.buptpress3.com
电子信箱	ctrd@buptpress.com
经 销	各地新华书店
印 刷	北京泽宇印刷有限公司
开 本	787 mm×1 092 mm 1/16
印 张	18.5
字 数	474千字
版 次	2018年11月第4版 2018年11月第1次印刷

ISBN 978-7-5635-5604-5

定价:45.00元

如有质量问题请与发行部联系

版权所有 侵权必究

前 言

C语言概念简洁,数据类型丰富,表达能力强,运算符多而灵活,控制流和数据结构新颖,程序结构性和可读性好,能有利于培养良好的编程习惯,易于体现结构化程序设计思想。它既具有高级语言程序设计的特点,又具有汇编语言的功能;既能有效地进行算法描述,又能对硬件直接进行操作;既适合于编写应用程序,又适合于开发系统软件。它是当今世界上应用广泛、最具影响的程序设计语言之一。C语言本身还具有语言紧凑整齐、设计精巧、编辑方便、编译与目标代码运行效率高、操作简便、使用灵活等许多鲜明的特点。特别是它扩充的图形、窗口等功能,以及高效的集成开发环境,赢得了广大用户的喜爱,得到了广泛的应用。

本书全面介绍了C语言的概念、特性和结构化程序设计方法。全书共有12章,第1章介绍C语言程序设计的基本知识;第2章介绍了C语言的基本数据类型、常量和变量以及表达式表示;第3、第4、第5章介绍了C语言进行结构化程序设计的基本方法,包括结构化程序的顺序结构、选择结构、循环结构及其设计方法;第6章介绍了函数与编译预处理;第7、第8章对C语言的数组、指针作了充分阐述;第9章对结构体、共用体作了较详细的介绍;第10章对C语言文件操作作了详细的阐述;第11章介绍了C语言的综合编程应用,着重介绍使用C语言的编程规范与实际应用技巧;第12章以电子版的形式展示,对全国计算机等级考试二级C语言程序设计的主要考点进行分析,并结合真题作了讲解。本书在电子版资源上附加了全国计算机等级考试C语言真题试卷及参考答案,供相关读者学习参考。

本书是作者在多年从事C语言及计算机专业相关课程的教学实践中,在多次编写讲义、教材的基础上编写而成的。本书内容充实,循序渐进,选材上注重系统性、先进性、实用性。全书精选了大量例题和实用案例,且已上机通过,可直接引用。

为配合读者学习本书,作者另外编写了一本《C语言程序设计上机指导及习题选解》(第4版),作为本书的配套参考书。本书中的习题及习题解答均安排在其中,供读者复习和检查学习效果。

本书由杨路明担任主编,由谭长庚、沙莎、严晖担任副主编。蒋外文编写第1、第2章;沙莎编写第3章到第6章;谭长庚编写第7、第8章;刘卫国编写第9章;罗三定编写第11章;严晖编写第12章;李香林编写、整理附录;杨路明编写第10章,并负责全书的统稿,参与本书编写与讨论的还有曹菊英。另外,马双武、向蕾构思并设计了全书在线课程教学资源的结构与配置,赵梅、苏国强编辑了教学资源的内容,柯炎坤参与了微课的录制及后期处理,苏文刚审查了全书配套在线课程的教学资源,王奥提供了版式和装帧设计方案。此外,部分章节习题后的综合案例由北京信息科技大学计算机学院C语言课程组提供,在此一并感谢。

本书修订过程中,得到了多所高等院校教师及企业应用开发技术人员的大力支持和帮助,在此,向他们表示衷心的感谢。由于作者水平有限,书中难免存在疏漏与不妥之处,敬请读者批评指正。

作 者

2018年08月于中南大学

CONTENTS

目 录

第 1 章 C 语言程序设计概述 /1

- 1.1 程序与程序设计语言 /1
- 1.2 算法及其描述 /3
- 1.3 C 语言的发展及特点 /6
- 1.4 C 语言程序的基本结构 /7
- 1.5 C 语言字符集、标识符与关键字 /9
- 1.6 C 语言程序的开发环境 /10

本章小结 /16

习题 1 /16

第 2 章 C 语言的基本数据类型与表达式 /18

- 2.1 C 语言的基本数据类型 /18
- 2.2 常量与变量 /21
- 2.3 运算符与表达式 /25
- 2.4 数据类型转换 /30

本章小结 /33

习题 2 /33

第 3 章 顺序结构程序设计 /36

- 3.1 C 语言的基本语句 /36
- 3.2 数据输入与输出 /39
- 3.3 程序举例 /48

本章小结 /52

习题 3 /53

第 4 章 选择结构程序设计 /55

- 4.1 关系运算符与关系表达式 /55
- 4.2 逻辑运算符与逻辑表达式 /56
- 4.3 if 语句 /58
- 4.4 switch 语句 /63
- 4.5 结构嵌套程序举例 /66

本章小结 /70

习题 4 /70

第 5 章 循环结构程序设计 /73

- 5.1 while 语句 /73
- 5.2 do...while 语句 /74
- 5.3 for 语句 /75
- 5.4 break、continue 和 goto 语句 /79
- 5.5 循环的嵌套 /81
- 5.6 复合结构程序举例 /83

本章小结 /88

习题 5 /88

第 6 章 函数与编译预处理 /92

- 6.1 模块化程序设计与函数 /92
- 6.2 函数的定义与调用 /96
- 6.3 函数的递归调用 /103
- 6.4 变量作用域与存储方式 /108
- 6.5 编译预处理 /116
- 6.6 函数设计举例 /123

本章小结 /126

习题 6 /126

第 7 章 数组 /130

- 7.1 一维数组 /130
- 7.2 二维数组 /136
- 7.3 数组的应用 /143
- 7.4 字符数组和字符串 /147
- 7.5 数组作为函数的参数 /155
- 7.6 程序举例 /160
- 本章小结 /165
- 习题 7 /165

第 8 章 指针 /169

- 8.1 指针与指针变量 /169
- 8.2 指针与函数 /177
- 8.3 指针与数组 /184
- 8.4 指针与字符串 /191
- 8.5 指针数组与命令行参数 /194
- 8.6 程序举例 /198
- 本章小结 /203
- 习题 8 /204

第 9 章 结构体数据类型与链表 /208

- 9.1 结构体类型的定义 /208
- 9.2 结构体类型变量 /209
- 9.3 结构体类型数组 /215

- 9.4 结构体类型指针 /217
- 9.5 结构体与函数 /220
- 9.6 链表 /223
- 9.7 共用体与枚举类型 /234
- 9.8 程序举例 /240
- 本章小结 /245
- 习题 9 /246

第 10 章 文件 /252

- 10.1 文件概述 /252
- 10.2 标准文件操作 /255
- 10.3 非标准文件操作 /267
- 本章小结 /271
- 习题 10 /271

第 11 章 C 语言综合应用程序示例 /275

- 11.1 多模块编程 /275
- 11.2 绘图和图像处理应用 /277
- 本章小结 /286
- 习题 11 /286

第 12 章 等级考试考点分析与真题讲解 /288

参考文献 /289

第1章

C语言程序设计概述



本章导读

电子计算机自从20世纪40年代诞生以来,无论在硬件还是在软件方面,都有了极大的发展,在计算机应用的各个领域也都取得了丰硕的成果。

计算机本身是无声息、无生命的机器,要使计算机能够运行起来,为人类完成各种各样的工作,就必须让它执行相应的程序。这些程序都是依靠程序设计语言编制出来的。

在众多的程序设计语言中,C语言有其独特之处。它作为一种高级程序设计语言,具备很强的方便性、灵活性和通用性;同时,它还向程序员提供了直接操作计算机硬件的功能,具备低级语言的特点,适合各种类型的软件开发。因此,C语言是深受软件工作者欢迎的程序设计语言。

本章主要从程序设计的角度,介绍有关程序设计的基本概念,结合C语言的发展,描述C语言程序的基本结构、算法以及C语言程序的开发环境等内容。



C语言在计算机专业中的地位

1.1 程序与程序设计语言

1.1.1 程序的基本概念

为了让计算机按人的意图处理事务,人们必须预先设计好完成各种任务的程序,并预先将它们存放在存储器中。

所谓程序,实际上是用计算机语言描述的某一问题的解决步骤,是符合一定语法规则的符号序列。人们借助计算机能够处理的语言,告诉计算机要处理什么(处理哪些数据)以及如何处理(按什么步骤来处理),这便是程序设计。通过在计算机上运行程序,向计算机发出一系列指令,便可按人们的要求解决特定问题。

解决某一个问题所编写的程序并不是唯一的,不同的用户所开发的程序也往往不相同。不同的程序有不同的效率,这涉及程序的优化,程序所采用的数据结构及算法等多方面因素的综合。

1.1.2 程序设计语言

完成程序设计,自然离不开程序设计语言。不同的问题,可以用不同的程序设计语言来解决,但解决问题的难易程度会各不相同。了解程序设计语言的发展过程,有助于读者加深对程序设计语言的认识,能更好地利用程序设计语言来解决有关问题。



程序与设计语言

当今程序设计语言发展非常迅速,新的程序设计语言层出不穷,其功能越来越强大。但不管现代程序设计语言的功能如何增强,程序设计语言的种类怎样增多,从其发展历史以及功能情况看,大致可分成如下几个阶段。

1. 机器语言

由于受物理器件和实现难易的影响,直到目前,计算机还只能存放和识别由 0 和 1 组成的序列所表示的数据和指令。所谓机器语言,就是指该机器能够识别的指令的集合,即指令系统。在机器语言中,每条指令都用 0 和 1 组成的序列来表示。例如,以下是某计算机的两条机器指令。

加法指令:1 0 0 0 0 0 0 0

减法指令:1 0 0 1 0 0 0 0

不同类型的计算机,使用的机器语言不相同。

用机器语言编写的程序,计算机可以直接识别和执行,执行效率高。但机器语言的指令不直观,难认、难记、难理解、易错。编写机器语言程序时,要求程序员必须相当熟悉计算机结构,且较烦琐,写出的程序不能通用。因而,目前很少用机器语言编程。

2. 汇编语言

20 世纪 50 年代中期,为了减轻人们使用机器语言编程的负担,开始采用一些“助记符号”来表示机器语言中的机器指令,这样便形成了汇编语言。助记符一般都是表示一个操作的英文字母的缩写,与机器语言相比便于识别和记忆。例如,上例中的两条指令用汇编语言描述如下:

```
ADD A,B
```

```
SUB A,B
```

不过,计算机不能直接识别和执行用汇编语言编写的程序,它必须经过一个叫汇编程序的系统软件翻译成机器语言程序后才能执行。通常称前者为源程序,后者为目标程序。

汇编语言指令和机器语言指令之间具有一一对应的关系。因而,不同的计算机其汇编语言也不尽相同,并且程序编写时仍需要对计算机内部结构比较熟悉,依然比较烦琐。但相对于机器语言而言,汇编语言要好多了。所以在实际中,如果程序运行时间要求比较严格,程序与硬件操作联系紧密,人们还是常用汇编语言编写有关程序来解决这些问题。

3. 算法语言

汇编语言和机器语言是面向机器的,它们仍同属于低级语言的范畴。人们在使用它设计程序时,仍要求对机器比较熟悉。为了克服低级语言的这一缺点,使人们将程序设计的精力集中在解决问题的算法上,便出现了面向算法过程的程序设计语言,称之为算法语言。例如,FORTRAN 语言、ALGOL 语言、Pascal 语言、C 语言等。这类语言接近自然语言的形式,因而可以较大地降低编程的难度,提高编程的效率和质量,且设计的程序也更容易理解。

算法语言的一条语句相当于多条汇编语言指令或机器语言指令,表达能力强。在使用算法语言时,程序设计人员不需要熟悉计算机的指令系统,可以把精力集中在研究问题的求解方法步骤(过程)上。因此,算法语言也称为面向过程的程序设计语言。同时,算法语言不依赖于机器。为某种类型的计算机编写的算法语言程序,可以很方便地移植到其他类型的计算机上运行。相对于与机器有关的低级语言而言,算法语言也就称为高级语言。

当然计算机也不能直接执行算法语言程序。如同汇编语言程序一样,算法语言程序(源程序)也必须先经过编译程序(或解释程序)编译成机器语言程序(目标程序)后,才能由计算机执行。

4. 面向任务的程序设计语言

利用算法语言求解一个复杂的问题,必须先要分析解决问题的过程,描述问题如何求解,然后才能用算法语言进行程序设计来实现。面向任务的程序设计语言是非过程化的语言,也就是说,不需要知道问题是如何求解的,只需描述需求解的问题是什么,然后便可用程序设计语言来实现。

数据库操纵语言便是一种面向任务的程序设计语言。例如,设在某数据库应用系统中,有一个学生情况表 SS,若要从表中查找学生的信息,可以使用数据库查询语言(SQL)。采用 SELECT 语句便可完成这个任务,SELECT 语句描述如下:

```
SELECT SSNO,SSNAME,SSAGE,SSSEX FROM SS
```

该语句从 SS 表中查找到学生的 SSNO、SSNAME、SSAGE、SSSEX 等方面的信息。至于 SELECT 语句是如何进行查询的,这个过程用户就不必了解了。

面向任务的程序设计语言可以提高应用程序的开发速度和质量,也可使应用程序能容易迅速地修改。同时,非计算机专业人员也能很方便地使用面向任务的程序设计语言开发自己的程序。

这类语言在应用系统的开发中使用较为广泛,尤其是一些管理信息系统应用程序的开发。

5. 面向对象的程序设计语言

面向对象程序设计语言在 20 世纪 90 年代开始流行。现在,面向对象的程序设计语言已成为程序设计的主流语言之一。由 C 语言发展出来的 C++ 就是一种非常优秀的面向对象的程序设计语言。

面向对象的方法是一种分析方法、设计方法和思维方法的综合。面向对象方法学的出发点和所追求的基本目标是使人们分析、设计和实现一个系统的方法尽可能接近人们认识一个系统的方法。

在面向对象编程中,程序被看作是相互协作的对象集合,每个对象都是某个类的实例,所有的类构成一个通过继承关系相联系的层次结构。面向对象的程序设计语言具有对象生成功能、消息传递机制以及类和继承机制。

对象是对客观的事物的抽象,面向对象的编程,就是针对客观的事物设计程序。因此,面向对象的编程是非常直观的。面向对象的程序设计方法比面向过程的程序设计方法更清晰,更易适合开发大型复杂的软件。

综上所述,每一种语言都有它的优势和劣势,对于不同的问题,要根据实际情况来选择,以便更高效更优质地解决相关的问题。



算法概述

1.2 算法及其描述

在程序设计中,不可避免地需要涉及算法。有人这样说过:“计算机科学就是研究算法的科学”,足见算法在程序设计中的重要性了。下面从算法的概念和描述方法两方面对算法的问题进行讨论。

1.2.1 算法的概念

著名的瑞士计算机科学家、Pascal 语言发明者 N·沃思教授提出了程序定义的著名公式：

程序 = 算法 + 数据结构

这个公式的重要性在于它说明了程序与算法的关系。同时,也说明了数据结构的选择也是十分重要的。对于程序而言,算法与数据结构是统一的关系。

通常认为,算法是对特定问题求解步骤的一种描述,是由一套规则组成的一个过程,是指令的有限序列,其中每一条指令表示一个或多个操作。

算法应当具备以下几个方面特点。

- 有穷性:一个算法必须保证执行有穷步之后结束。
- 确定性:算法的每一个步骤必须具有确切的定义。
- 输入项:应对算法给出初始量。
- 输出项:算法具有一个或多个输出。
- 可行性:算法中的每个步骤都应当能有效地进行,不能进行不被允许的操作,如“计算 $X/0$ ”。

算法必须能在有限时间内完成,且对相同的输入有相同的输出。在程序设计语言中,与算法密切相关的便是语句,包括与程序执行处理有关的“功能语句”(如输入语句、输出语句、赋值语句、调用语句等)和与程序执行流程有关的语句(如条件语句、循环语句等)。对程序设计而言,算法的确定也就是如何合理安排这些语句以完成人们要求的特定功能。

1.2.2 算法的描述方法

从上面的分析说明可知,算法是描述某一问题求解的有限步骤,而且必须有结果输出。设计一个算法,或者描述一个算法,最终是由程序设计语言来实现的。但算法与程序设计又是有区别的,主要是一个由粗到细的过程。算法是考虑实现某一个问题的求解方法和步骤,是解决问题的框架流程;而程序设计则是根据这一求解的框架流程进行语言细化,实现这一问题求解的具体过程。

一般可以使用下面几种类型的工具描述算法。

1. 自然语言

自然语言即是人们日常进行交流的语言,如英语、汉语等。自然语言用来描述算法,分析算法,作为用户相互之间进行交流,是一种较好的工具。但是用自然语言描述算法直接在计算机上进行处理,目前还存在许多困难,这里包括有诸如语音语义识别等方面的问题。

2. 专用工具

要对某一个算法进行描述,可以借助于有关图形工具或代码符号。常用的工具有流程图、N-S 图等。

20 世纪五六十年代兴起的流程图几乎成为了程序设计及算法描述的必须步骤。

关于流程图,人们已经提出了多种描述算法的流程图。这种方法的特点是用一些图框表示各种类型的操作,用线表示这些操作的执行顺序,如图 1.1 所示为我国国家标准 GB 1526—89 中推荐的一套流程图标准化符号的一部分。

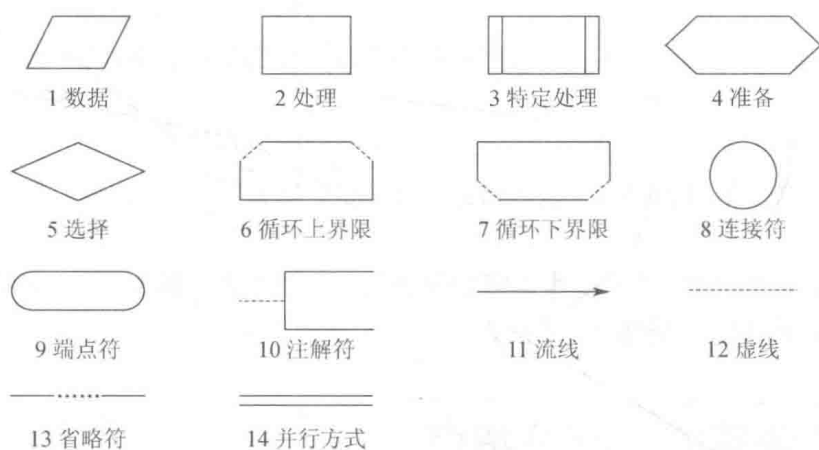


图 1.1 流程图标准化符号

数据 平行四边形表示数据,其中可注明数据名称、来源、用途或其他文字说明。

处理 矩形表示各种处理功能。矩形内可注明处理名称或其简要功能。

特定处理 带有双竖边线的矩形。矩形内可注明特定处理名称或简要功能,表示已命名的处理。该处理表示在另外地方已得到详细说明的一个操作或一组操作。

判断 菱形表示判断。菱形内可注明判断的条件。它只有一个入口,但可以有若干个可供选择的出口。

循环界限 循环界限表示循环的上界和下界,中间是要循环执行的处理内容,称为循环体。循环界限由去上角的矩形(表示上界限)和去下角的矩形(表示下界限)构成。

端点 扁圆形表示转向外部环境或外部环境转入的端点符。例如,程序流程的起始点。

注解 注解是程序的编写者向阅读者提供的说明。它用虚线连接到被注解的符号或符号组上。

3. 常用的算法

在程序设计时,常常要通过一些特定的算法来求解。现在比较常用的算法有下列几种。

(1) 迭代法

一般的一元五次或更高次的方程,以及几乎所有的超越方程、微分方程的数值解,这类问题的求解都无法用解析方法通过求根公式来求解,人们只能用数值方法求其近似解。用事先估计的一个根的初始值 X_0 ,通过迭代算式 $X_{K+1} = G(X_K)$ 求出另一个近似值 X_1 ,再由 X_1 求出 X_2 ,从而获得一个求解的序列 $\{X_0, X_1, X_2, \dots, X_n, \dots\}$ 来逼近方程 $f(x) = 0$ 的根。这种求解的过程称为迭代。

(2) 枚举法

枚举法的基本思想是首先依据题目部分条件确定答案的大致范围,然后逐一验证所有可能的情况,这种算法也称穷举法。

(3) 递归法

递归是指一个过程直接或间接地调用它自身,递归过程必须有一个递归终止条件。

如:

$$n! = \begin{cases} 1 & (n = 0) \\ n(n-1)! & (n > 0) \end{cases}$$

(4) 递推法

算法从递推初始条件出发,应用递推公式对问题进行求解。例如,Fibonacci 数列存在递推关系:

$$F(1)=1, F(2)=2, F(n)=F(n-1)+F(n-2), (n>2)$$

若需求第 30 项的值,则依据公式,从初始条件 $F(1)=1, F(2)=1$ 出发,可逐步求出 $F(3), F(4), \dots$,直到求出 $F(30)$ 。

除此之外,还有回溯法(一种基本策略)、分治法(问题分解成较小部分,求解再组合)等其他的常见算法,在此不一一阐述。

1.3 C 语言的发展及特点

1.3.1 C 语言的发展概况

C 语言是当今社会应用很广泛,受到众多用户欢迎的一种计算机算法语言。它既可适合于作为系统软件描述语言,也可用来开发应用软件。

C 语言的出现是与 UNIX 操作系统紧密联系在一起,C 语言本身也有一个发展过程,目前仍然处于发展和完善过程之中。

从历史发展来看,C 语言起源于 1968 年发表的 CPL(combined programming language)语言,它的许多重要思想来自于 Martin Richards 在 1969 年研制的 BCPL 语言,以及以 BCPL 语言为基础的由 Ken Thompson 在 1970 年研制成的 B 语言。K. Thompson 用 B 语言写了第一个 UNIX 操作系统,用在 PDP-7 计算机上。1972 年 D. M. Ritchie 在 B 的基础上研制了 C 语言,并用 C 语言写成了第一个在 PDP-11 计算机上实现的 UNIX 操作系统。1977 年出现了独立于机器的 C 语言编译文本《可移植 C 语言编译程序》,从而大大简化了把 C 语言编译程序移植到新环境所需做的工作,这本身也就使 UNIX 操作系统迅速地在众多的机器上实现。如 VAX,AT&T 等计算机系统都相继开发了 UNIX。随着 UNIX 的日益广泛使用,C 语言也迅速得到推广。

1983 年美国国家标准化协会(ANSI)根据 C 语言问世以来的各种版本,对 C 语言的发展和扩充制定了新的标准,称为 ANSI C。1987 年 ANSI 又公布了新标准——87ANSI C。

目前在微型计算机上使用的 C 语言有多种版本,这些不同的版本,基本部分是相同的,但在有关规定上又略有差异。

1.3.2 C 语言的特点

C 语言是一种面向结构的语言。事实证明,C 语言是一种极具生命力的语言,它的特点是多方面的,一般可归纳如下。

① C 语言具有结构语言的特点,程序之间很容易实现段的共享。它具有结构化的流程控制语句(如 if...else 语句,while 语句,do...while 语句,switch 语句,for 语句),支持若干种循环结构,允许编程者采用缩进书写形式编程,因此,用 C 语言设计出的程序层次结构清晰。

② C 语言的主要结构成分为函数,函数可以在程序中被定义完成独立的任务,独立地编译

成代码,以实现程序的模块化。

③C语言运算符丰富,运算符包含的范围很广泛。它把赋值、括号、强制类型转换都当作运算符处理。灵活地使用各种运算符可以实现在其他的高级语言中难以实现的运算。

④C语言数据类型丰富。数据类型有整型、实型、字符型、数组型、指针型、结构体型、共用体型等。能用来实现各种复杂的数据结构(如链表、树、栈等)的运算。尤其是C语言的指针型数据的运算,更是灵活、多样。

⑤C语言允许直接访问物理地址,即可直接对硬件进行操作,实现汇编语言的大部分功能。由于C语言这一特点,使得它成为编制系统软件的基本语言(UNIX的绝大部分就是由C语言写成的)。

⑥C语言语法限制不太严格,程序设计自由度大。这样使C语言能够减少对程序员的束缚。“限制”与“灵活”是一对矛盾。限制严格,就易失去灵活性;而强调灵活,就必然放松限制。从这个角度来看,使用C语言编程,要求编程者对程序设计技巧要更加熟练一些。

⑦用C语言编程,生成的目标代码质量高,程序执行效率高。同时用C语言写的程序可移植性好。

C语言优点很多,但是它也存在一些缺点,如运算优先级太多,数值运算能力方面不像其他高级语言那样强,语法定义不严格等。尽管C语言目前还存在一些不足之处,但由于它目标代码质量高、使用灵活、数据类型丰富、可移植性好而得到广泛的普及和迅速的发展,成为一个受到广大用户欢迎的实用的高级程序设计语言,一种在系统软件开发、科学计算、自动控制等各个领域被广泛应用的程序设计语言。

1.4 C语言程序的基本结构



初识C语言

在学习C语言的具体语法之前,先通过一个简单的C语言程序示例,初步了解C语言程序的基本结构。

例 1.1 已知3个整型数8、12、6,按公式 $s=a+b \times c$ 计算,并显示结果。

```
#include <stdio.h>           /* 标准输入输出头文件 */
void main( )
{
    int a,b,c,s;              /* 定义4个整型变量 */
    a=8;b=12;c=6;            /* 变量赋值 */
    s=a+b*c;                  /* 算术运算并赋值 */
    printf("s=%d\n",s);      /* 输出结果 */
}
```



代码演示

从例1.1可以看出:

(1)C语言程序由函数构成

每一个C语言程序都必须具有一个main()函数作为程序的主控函数,称为主函数。main()函数是C语言编译系统使用的专用名字。main()后面由花括号对“{}”括起来的部分是程序的主体,程序从main()函数的第一个语句开始执行。

本例中main()函数内的第一行是变量说明语句,定义了函数内所需使用的4个变量a、

b 、 c 和 s ，变量类型由关键字 `int` 定义为整数型。三个赋值语句格式书写非常自由，可以放在一行上，也可以分成单独的行写，分隔符采用分号“;”。每个语句结束符也是采用分号“;”。

`printf` 是 C 语言提供的按指定格式进行标准输出的函数。

(2) 注释

注释符“`/*`”与“`*/`”之间的内容构成 C 语言程序的注释部分。“`/*`”与“`*/`”之间的内容可以是一行，也可以是多行，即多行注释。程序中可以有单行注释，单行注释用符号“`//`”开始，“`//`”后面为被注释的内容。

无论是单行注释还是多行注释，注释部分均不参与程序的编译和执行，只是起说明作用，增加程序的可读性。

(3) 用预处理命令 `#include` 可以包含有关文件的信息

C 语言提供了多个头文件，分类包含了各类标准函数的原型说明，需要用到某些标准库函数时，只需将对应的头文件用 `#include` 语句包含在程序的首部，就可直接使用了，头文件的扩展名一般为“.h”。

(4) 大小写字母在 C 语言中是有区别的

如“Main”、“MAIN”、“main”和“maiN”在 C 语言中是不同的。C 语言所理解的程序主函数名称为“main”，这点应引起读者注意。

(5) 用户自编的函数

除 `main()` 函数和标准库函数外，用户自己也可以编写函数，应用程序一般由多个函数组成，这些函数指定实际要做的工作。

一个较完整的 C 语言程序大致包括：

- ① 头文件（一组 `#include < * . h >` 语句，也称包含文件）；
- ② 用户函数说明部分；
- ③ 全局变量定义；
- ④ 主函数；
- ⑤ 若干子函数。

在主函数和子函数中一般又包含了局部变量定义、若干个库函数、控制流程语句、用户函数的调用语句等。若设 $f_1() \sim f_n()$ 代表用户定义的函数，则 C 语言程序的一般形式可表达如下：

```
头文件 #include... 语句
全局变量说明
void main() {
    局部变量
    程序段
}
f1() {
    局部变量
    程序段
}
f2() {
    局部变量
    程序段
}
```

```

}
.....
fn() {
    局部变量
    程序段
}

```



字符集、标识符与关键字

1.5 C语言字符集、标识符与关键字

1.5.1 C语言字符集

任何一个计算机系统所能使用的字符都是固定、有限的,它要受硬件设备的限制。要使用某种计算机语言来编写程序,就必须使用符合该语言规定的,并且计算机系统能够使用的字符。

C语言和其他语言一样,它的基本字符集包括有英文字母、阿拉伯数字以及其他一些符号,具体归纳如下。

- 英文字母:大小写各 26 个,共计 52 个。
- 阿拉伯数字:0~9,共计 10 个。
- 下划线: _。
- 其他特殊符号:主要指运算符,运算符通常由一至两个特殊符号组成。特殊符号集如下。

+	-	*	/	%	++	--	<
>	=	>=	<=	==	!=		&
!		&&	^	~	《	》	()
[]	{ }	\	?	:	.	,	;

1.5.2 标识符与关键字

1. 标识符

C语言的标识符用来表示函数、类型及变量的名称,它是字母、下划线和数字的排列,但必须是字母或下划线开头。标识符中大小写字母的含义不同。

在C语言中,标识符可以是一个或多个字符,其有效长度为1~32个字符,它可以用来作为变量名、函数名、标号及用户定义对象名。读者应注意编程时所用系统对标识符长度的规定,有的系统对标识符的使用往往有较多的限制。对于初学者来讲,建议在程序中一般标识符不要取太长。

下面给出一些合法与不合法命名的标识符。

合法标识符: _22A, lea_1, avg3, day, ABCde43xyw8

不合法标识符: M. J. YORK, \$ _238, # xy, a * b, 8Tea

2. 关键字

关键字是一种语言中规定具有特定含义的标识符。关键字不能作为变量或函数名来使用,用户只能根据系统的规定使用它们。根据 ANSI 标准,C 语言可使用以下 32 个关键字:

auto	break	case	char	const	continue	default	do
double	else	enum	extern	float	for	goto	if
int	long	register	return	short	signed	sizeof	static
struct	switch	typedef	union	unsigned	void	volatile	while



程序开发基本过程

1.6 C 语言程序的开发环境

编写好的 C 语言程序要经过编辑(输入)、编译和连接后才能形成可执行的程序。如图 1.2 所示,表示了 C 语言程序设计上机开发过程。



图 1.2 C 语言程序上机开发过程

目前,程序员大多使用集成开发工具来开发 C 语言程序,适合 C 语言的集成开发工具有多种,常用的有 Turbo C、Microsoft C、Visual C++、Borland C++、Dev C++、C++ Builder 等,它们各具特色,分别适合于不同的操作系统环境。Visual C++、Borland C++、Dev C++、C++ Builder 既适合开发 C++ 语言程序,也适合开发 C 语言程序。

Visual C++ 简称 VC++, 它包含有编辑器、调试器和编译器。VC++ 有 6.0/2005/2008/2010/2013 等多个版本。

VC++、VB、C# 等语言的集成开发工具又被集成在 Visual Studio(简称 VS)集成开发平台中,VS 也有 2003/2005/2008/2010/2012/2013/2015 等不同的版本。它非常灵活,集成了多种功能,可帮助程序员按照需要,实施有效的开发实践。实际开发中程序员往往乐于使用 VS 集成开发环境。

本书主要介绍在 VS 2010 环境下的程序开发,书中大部分程序都在 VS 2010 环境下调试通过。

1.6.1 开发环境 Visual Studio 2010 简介

VS 2010 工作于 Windows 环境,双击桌面上的 Microsoft Visual Studio 2010 图标,进入 VS 2010 集成环境起始页,如图 1.3 所示。

该软件窗体中包括标题栏、菜单栏、工具栏、工作区窗口和程序编辑窗口等。

(1) 标题栏

VS 2010 主窗口的第一行就是标题栏。标题栏的左边显示当前文件的文件名以及版本信息,默认文件名是“起始页”。



图 1.3 VS 2010 集成环境起始页

(2) 菜单栏

标题栏下面依次是菜单栏和工具栏。各菜单的功能如下。

文件 用来创建、打开、保存项目文件,也是必须要做的第一步。

编辑 用来编辑文件。

视图 用来查看代码,显示其他窗口,打开/关闭工具栏。

项目 用来添加类、添加新项和设置启动项(应当注意的是该项只有在新建项目或打开项目时才会出现)。

调试 用来设置项目的各项配置,编译、创建和执行应用程序、调试程序。

工具 用来对工具栏、菜单以及集成开发环境进行定制。

窗口 用来新建/拆分窗口和窗口布局。

帮助 给出相关的帮助。

(3) 工具栏

工具栏和菜单栏的作用是一样的,只不过工具栏是把菜单栏中经常用到的编辑功能选出来并用图形表示,方便我们操作。VS 2010 有多个工具栏,有标准工具栏和文本编辑器工具栏。如图 1.3 所示的工具栏就是一个标准工具栏。

(4) 工作区

界面中的左窗口为“解决方案资源管理器窗口”。解决方案资源管理器包含外部依赖项、头文件、源文件和资源文件。

(5) 输出窗口

主界面最下侧的窗口为输出窗口,显示程序运行状态,其作用是在编译、链接时显示编译、链接信息。