

内 容 简 介

本书内容主要包括数据类型、运算符和表达式、输入输出、流程控制、函数与程序结构,其中对悦语言的表达式、类型转换、变量的存储类型 and 指针等重点和难点均作了明确而详细的介绍。本书内容严格遵循悦语言标准,结构紧凑、概念清楚准确、叙述流畅、重点突出、清晰明了,本书例题典型丰富,不仅给出了经过调试运行的源程序,而且有对算法思路的分析及算法步骤,每章末附有重点小结及习题,便于广大读者进行自学、复习和练习。

本书既可作为高等教育计算机专业本科悦语言程序设计课程的教材,也可作为成人高等教育相应专业的本科和自学考试相应专业本、专科学生的教学参考书,还可供有关专业的工程技术人员参考。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有,侵权必究。

图书在版编目(CIP)数据

悦语言程序设计 秦友淑 曹化工编著—北京:电子工业出版社, 2004.10

高等教育教材

陈昇苑 魏建刚 魏建刚

I ①悦.. Ⅱ ①秦...②曹... Ⅲ ①悦语言—程序设计—成人教育. ②高等教育—教材 Ⅳ ①计算机

中国版本图书馆CIP数据核字(2004)第194928号

责任编辑:束传政 摇朱怀永

印刷者:北京东光印刷厂

出版发行:电子工业出版社 地址:北京 朝阳区 惠新东街 10 号

北京市海淀区万寿路 19 号信箱 邮编 100036

经销:各地新华书店

开印:2004 年 10 月 1 日 印张:13.5 摇字数:300 千字

版:2004 年 1 月第 1 版 2004 年 1 月第 1 次印刷

印:2004 年 1 月第 1 次印刷 定价:19.80 元

凡购买电子工业出版社的图书,如有缺损问题,请向购买书店调换。若书店售缺,请与本社发行部联系。

联系电话:(010) 5189 5189

总摇摇序

成人高等教育是高等教育的重要组成部分。江泽民同志在中国共产党第十五次全国代表大会报告中指出：“发展教育和科学，是文化建设的基础工程。培养与现代化要求相适应的数以亿计高素质的劳动者和数以千万计的专门人才，发挥我国巨大人力资源的优势，关系到世纪社会主义事业的全局。”要“积极发展各种形式的职业教育和成人教育，稳步发展高等教育。”我国成人高等教育发展的历史已经证明，大力发展成人高等教育，对于培训有理想、有道德、有文化、有纪律的社会主义精神文明和物质文明的建设，加快实现高等教育大众化，已经并正在发挥着巨大的作用。

华中科技大学（原华中理工大学）是我国较早开展成人高等教育的高等院校之一。四十多年来，华中科技大学的成人教育始终坚持社会主义办学方向，坚持高标准严要求，努力探索成人教育发展规律、积累了较丰富的教学与管理经验，教育质量稳步提高，管理制度不断完善，培养了数万名成人高等教育毕业生，赢得了广泛的社会赞誉。

为更好地适应成人高等教育发展的需要，保证成人高等教育的教学质量，在电子工业出版社的大力支持下，我院组织编写了这套成人高等教育系列教材，主要以本科层次（含专科起点本科和高中起点本科）部分专业课为主。编写过程中遵循了以下原则。

（员）贯彻教育必须为社会主义现代化建设服务，必须与生产劳动相结合，培养德、智、体等方面全面发展的社会主义事业的建设者和接班人的教育方针。

（圆）充分体现成人高等教育的特点，强调以应用为目的，以“必需”、“够用”、“适用”为度，使学生重点掌握基本概念、基本原理和基本分析计算方法。同时，为学生可持续发展奠定一定的理论基础。

（猿）遵循成人教育的教学规律，注重理论联系实际，注重能力和素质培养，使于学生自学。

（源）教学内容要尽可能反映本学科的新理论、新思想、新方法、新技术、新规范和新成果。

在编写风格上，这套系列教材力求做到立论严谨、概念准确、叙述流畅、简明清晰、重点突出、例题丰富、章有小结、便于自学。

参加编写本套教材的各课程主编和编者均为长期从事成人教育教学工作且经验丰富的教师。这套教材的出版，既是他们多年来不懈追求的写照，也是他们长期辛勤耕耘汗水和心血的结晶，愿这套教材像一朵初绽的小花，在成人高等教育教材的百花园中越来越鲜艳！希望有更多的人关注它，关心它，关爱它，为它的成长培土、施肥、浇水！

华中科技大学成人教育学院

前摇摇言

悦语言是一种优秀的通用程序设计语言，也是目前全世界广泛使用的面向对象的程序设计语言悦语的基础语言。它具有功能强、简洁、高效、灵活、实用及可移植性好等特点，是当今世界上的主流程序设计语言之一。近十多年来，悦语言在我国的实际程序设计应用中也占据统治地位。

悦语言标准的制定标志着悦语言发展的成熟，标准悦语言综合了悦语言发展过程中各种版本对传统悦语言功能的许多重要扩充。标准悦语言对传统悦语言最重要的修改是加强了对类型的处理，目前几乎所有厂商推出的悦编译程序都是依据标准悦语言来实现的。

《悦语言程序设计》是计算机专业的基础课程之一，同时运用悦语言进行程序设计也是从事计算机专业的科技人员的一门专业技能。为满足我国高等教育普及和不断发展的需要，作者在多年程序设计语言课程教学（悦语言程序设计、面向对象的程序设计、~~编译语言~~程序设计）和编译实现（悦编译程序和~~编译~~编译程序等）的经验和体会的基础上，同时参阅了国内外许多关于悦语言的著作编著了本教材。本书严格遵循悦语言标准，系统地、全面地、深入浅出地阐述了悦语言的基本概念、语法规则和用悦语言进行程序设计的方法和技巧，内容主要包括数据类型、运算符和表达式、输入输出、流程控制、函数与程序结构，其中对悦语言的表达式、类型转换、变量的存储类型和指针等重点和难点均作了明确而详细的介绍。有些高级应用（例如二叉树）以及字段结构和输入输出的低层接口，因在一般应用中不是必不可少的基本内容，故未列入本教材，对此感兴趣的读者可参阅参考文献

本书结构紧凑、概念准确、叙述流畅、重点突出、例题丰富、实用性强。每章末附有重点小结及习题，习题一般在正文中可以找到类似的例题，或者是对例题程序的修改、扩充或应用，目的是便于自学和模仿，从而巩固基本知识，提高实际编程能力。

本书既可作为高等教育计算机专业本科悦语言程序设计课程的教材，也可作为成人高等教育相应专业的本科和自学考试本、专科的教学参考书，还可供有关专业的工程技术人员参考。

本书正文中所有的程序实例均在~~栽~~悦~~圃~~上通过运行（附录~~阅~~给出了在~~栽~~悦~~圃~~集成环境下运行悦程序的方法）。为了突出重点和节省篇幅，书中有的实例仅给出函数而未列出完整的程序，对于给出完整程序的实例一般都有输出结果。有的实例侧重于说明语法和用法，有的则为了介绍算法或说明应用，讲课时，教师可根据需要选讲部分实例，其余留给学生自己阅读或上机练习。

悦语言比较复杂，如果用严格的、形式化的方法一次给出完整的、准确的表示形式，则既不利于教学，也不利于初学者理解和掌握。因此对于复杂的语法（例如悦说明），本书采用循序渐进、逐步深入的方法，将一般形式和文字说明相结合来阐述悦语言的概念和规则。附录~~月~~给出了悦语言严格的语法。

学习和掌握悦语言最有效的方法是实践。实践包括：阅读教材中给出的例题程序或函数，理解程序所要完成的任务（即程序功能），从中学习编程的方法和技巧；模仿编写功能类似的程序（做习题）；自己独立设计和编写完成指定任务的程序（结合实际应用）。写程序时必须严格按语法规则一丝不苟地写，而且只有通过上机运行程序才能加深对概念和规则

的理解，才能真正掌握程序设计的方法和技术。

全书的编著由秦友淑完成，审校及部分程序的编制与调试由曹化工完成。感谢华中科技大学计算机学院领导和成人教育学院领导对此书编写工作给予的关心和支持，感谢电子工业出版社的有关同志为此书的编辑和出版所付出的辛勤劳动。

由于作者水平有限和时间仓促，书中错误和疏漏之处恳请广大读者批评指正。

编 者

1996年 9月

目 录

第 1 章 引 论	(1)
1.1 基础知识	(1)
1.2 计算机系统	(1)
1.3 算法及其表示	(1)
1.4 程序设计及程序设计语言	(1)
1.5 悦语言的发展过程及特点	(1)
1.6 悦语言的发展过程	(1)
1.7 悦语言的特点	(1)
1.8 悦程序的基本结构	(1)
1.9 悦语言的基本语法单位	(1)
1.10 标识符	(1)
1.11 关键字	(1)
1.12 分隔符	(1)
1.13 运行悦程序的一般步骤	(1)
1.14 小 结	(1)
1.15 习 题 一	(1)
第 2 章 基本数据类型和运算	(2)
2.1 基本数据类型	(2)
2.2 悦的数据类型分类	(2)
2.3 基本类型的名字及长度	(2)
2.4 常量和变量	(2)
2.5 常量的表示方法	(2)
2.6 符号常量	(2)
2.7 变量说明	(2)
2.8 运算符和表达式	(2)
2.9 基本概念	(2)
2.10 算术运算	(2)
2.11 关系运算	(2)
2.12 逻辑运算	(2)
2.13 自增和自减运算	(2)
2.14 位运算	(2)
2.15 赋值运算	(2)
2.16 条件运算	(2)
2.17 顺序求值运算	(2)
2.18 类型转换	(2)
2.19 类型转换的规则	(2)
2.20 类型转换的方法	(2)
2.21 枚举类型	(2)

摇摇缘用摇摇宏替换	(页码)
摇摇缘用摇摇文件嵌入	(页码)
摇小结	(页码)
摇习题五	(页码)
第 远章摇摇数组	(页码)
摇摇远用摇摇数组的说明、引用和初始化	(页码)
摇摇远用摇摇数组说明	(页码)
摇摇远用摇摇数组的引用	(页码)
摇摇远用摇摇数组初始化	(页码)
摇摇远用摇摇数组的运算	(页码)
摇摇远用摇摇数组的应用	(页码)
摇摇远用摇摇字符串处理函数	(页码)
摇小结	(页码)
摇习题六	(页码)
第 苑章摇摇指针	(页码)
摇摇苑用摇摇指针的概念与指针的使用	(页码)
摇摇苑用摇摇指针的概念	(页码)
摇摇苑用摇摇指针的使用	(页码)
摇摇苑用摇摇指针运算	(页码)
摇摇苑用摇摇指针参数	(页码)
摇摇苑用摇摇数组的指针表示	(页码)
摇摇苑用摇摇一维数组的指针表示	(页码)
摇摇苑用摇摇字符串的表示	(页码)
摇摇苑用摇摇多维数组的指针表示与数组的指针	(页码)
摇摇苑用摇摇指针数组	(页码)
摇摇苑用摇摇指针数组的说明及使用	(页码)
摇摇苑用摇摇指针的指针	(页码)
摇摇苑用摇摇函数的指针	(页码)
摇摇苑用摇摇指针函数	(页码)
摇摇苑用摇摇复杂说明	(页码)
摇小结	(页码)
摇习题七	(页码)
第 愿章摇摇结构与联合	(页码)
摇摇愿用摇摇结构的说明和引用	(页码)
摇摇愿用摇摇结构的说明	(页码)
摇摇愿用摇摇结构的引用	(页码)
摇摇愿用摇摇结构的指针	(页码)
摇摇愿用摇摇结构和函数	(页码)
摇摇愿用摇摇结构数组	(页码)
摇摇愿用摇摇结构和指针的应用	(页码)
摇摇愿用摇摇动态数据结构	(页码)

摇摇愿粤悦的动态存储分配函数	(愿粤)
摇摇愿粤链表	(愿粤)
摇摇愿粤联合	(愿粤)
摇摇愿粤赠粤说明	(愿粤)
摇摇小结	(愿粤)
摇摇习题八	(愿粤)
第 愿章 摇摇输入输出	(愿粤)
摇摇愿粤流式 赠粤	(愿粤)
摇摇愿粤标准输入输出函数	(愿粤)
摇摇愿粤文件的打开与关闭	(愿粤)
摇摇愿粤文件的读写操作	(愿粤)
摇摇愿粤文件的随机存取	(愿粤)
摇摇愿粤其他有关函数	(愿粤)
摇摇愿粤文件处理程序实例	(愿粤)
摇摇小结	(愿粤)
摇摇习题九	(愿粤)
附录 粤粤粤粤粤粤粤粤粤粤	(愿粤)
附录 粤粤粤粤粤粤粤粤粤粤	(愿粤)
附录 粤粤粤粤粤粤粤粤粤粤	(愿粤)
附录 粤粤粤粤粤粤粤粤粤粤	(愿粤)
参考文献	(愿粤)

第 1 章 引 论

1.1 基础知识

1.1.1 计算机系统

任何一个计算机系统，无论是大型机、中型机、小型机、微型机、个人计算机或计算机网络，均由硬件和软件两部分组成。硬件部分是构成计算机系统的物理设备，是整个计算机系统的物质基础；软件部分包括使用和管理计算机所需的各种程序、有关的数据和文档，软件是计算机系统的灵魂。

1. 硬件部分

计算机单机系统的硬件部分包括主机和外部设备。主机由中央处理单元（简称 CPU）和主存储器（简称内存）组成，外部设备包括输入设备、输出设备和外部存储设备。输入设备和输出设备通常简称为 I/O 设备。外部存储设备通常简称为外存，外存是内存的辅助存储设备。计算机系统的硬件组成如图 1.1 所示。

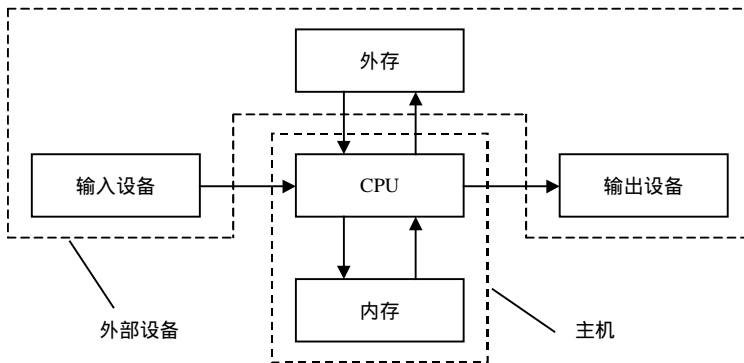


图 1.1 计算机系统的硬件组成

(1) CPU CPU 是计算机系统的核心，主要包括运算器和控制器两部分。运算器完成所有的算术运算和逻辑运算；控制器负责从内存取出指令和数据，并执行指令规定的动作以及将执行结果存入内存。

(2) 内存 内存用于存储正在执行的程序和被程序处理的数据。内存由若干连续的存储单元组成，存储单元是一些仅具有 0 和 1 两种状态的电子元件。因此，每个这样的电子元件能够表示二进制数的一位（称为一个 bit），多个相邻的 bit（二进制位）就能够表示由 0 和 1 数字串组成的代码（称为二进制代码）。内存中的程序和数据就是以这种二进制代码形式存储的。

存储单元的编号称为存储单元的地址(或内存地址)。8个相邻的二进制位称为一个字节,存储单元从0开始依次按字节编址。内存所包含的存储单元的总字节数称为内存的容量。CPU数据总线的bit数称为计算机的机器字长,例如:若机器字长为2字节则称为16位机,若机器字长为4字节则称为32位机。内存中存储的程序和数据称为存储单元的内容,从存储单元取出内容称为读内存,将程序或数据存入内存称为写内存。存取或读写内存统称为访问内存,访问内存必须由CPU通过存储单元的地址进行。

在高级语言程序中,存取内存中的数据一般是通过程序中的变量进行的,每个变量对应于一定数目的存储单元(以字节为单位),存储单元的内容就是变量的值。在C程序中,还可以通过存储单元的地址(即指针)访问内存中的数据。

作为一个程序员,必须了解内存的以下重要性质:

- 任何时刻计算机一经加电则存储单元中必有内容,但在未通过程序存入内容之前,这些内容仅是随机的,无意义的信息。
- 任何时刻计算机一旦切断电源则存储单元中的一切内容立即消失,将不复存在。
- 读内存操作永远不改变内存中的内容,即如果仅仅执行读内存的操作,无论读多少次,存储单元中的内容保持不变。
- 写内存操作要破坏存储单元中原来的内容。任何时候一旦执行了写内存操作,则存储单元中原来的内容立即被新写入的内容所代替,原来的内容不复存在(通常称为被冲掉或被覆盖)。

(3) 输入输出(I/O)设备 I/O设备用于人和计算机通讯。输入设备用于将程序、各种形态的数据或文档送入计算机内存,输出设备用于将存储在计算机内存中的各种信息送到外部设备上显示、打印或保存。

输入设备分为字符设备和图形设备两类。最普通的字符输入设备是终端键盘,最普通的图形输入设备是鼠标。输出设备主要包括终端显示器、打印机和绘图仪,打印机和终端显示器可以兼作字符输出设备和图形输出设备,绘图仪只能用于输出图形。其中,终端键盘和终端显示器通常被系统约定为标准输入设备和标准输出设备。

(4) 外存 外存既是信息的永久存储设备又起着输入输出设备的作用。主要的外部存储设备有磁盘驱动器、磁带机和光盘驱动器,内存的信息可以通过这些外部存储设备写到磁盘、磁带或光盘等存储媒体上。信息一经写入,只要存储媒体未被损坏或未重新写入新的信息,则信息将被永久保存(关掉电源以后信息也不会丢失);另一方面,保存在外存上的任何程序必须从外存读入内存才能被执行,任何数据必须从外存读入内存才能被处理。换言之,当需要执行外存上的程序或需要处理外存上的数据时,必须首先将它们从外存输入到内存。

外存容量比内存容量大得多,外存容量一般以GB($1\text{GB}=10^9\text{B}$)为度量单位,而内存容量一般以MB($1\text{MB}=10^6\text{B}$)为度量单位。所以,外存是内存的辅助存储设备,可用于存放大量的程序、数据和文档。

2. 软件部分

计算机系统的软件部分包括系统软件和应用软件。

(1) 系统软件

系统软件由计算机厂家及软件公司提供,是用于使用和管理计算机的各种程序及相关数据和文档的总称。系统软件主要包括操作系统、语言处理程序、各种实用程序和数据库管理

系统等。

- 操作系统 操作系统是最基本的核心系统软件，用于管理计算机系统的各种资源并控制各种程序的正常执行，例如 MS-DOS、Windows、UNIX 等。

- 语言处理程序 语言处理程序是将程序设计语言转变为计算机能够直接识别的机器语言的翻译程序，例如 IBM 汇编程序、Basic 解释程序、C 编译程序等。

- 实用程序 实用程序是为用户提供某种服务功能的程序，例如 DOS 环境下的文本编辑程序 Edit，Windows 环境下的图文排版程序 Word，计算机病毒杀毒软件等。

- DBMS DBMS（数据库管理系统）是能够集中存储和统一管理某应用领域用户的所有相关信息，并具有数据独立性、完整性、一致性和安全性的数据管理软件。

(2) 应用软件

应用软件是为使用计算机处理各种实际问题而开发的具有专门用途的程序，由于计算机的应用极为广泛，因而应用软件多种多样，例如各种 MIS 系统、CAD 系统和过程控制系统等。

1.1.2 算法及其表示

1. 算法的基本特性

人们通常把计算机为解决某一问题所需的方法与步骤称为算法。算法可分为两大类：一类是科学计算领域用于处理数值数据的算法，例如求定积分、解方程、求极限等；另一类是数据处理领域用于处理非数值数据的算法，例如分类排序、情报检索、绘图等。算法具有以下基本特性：

- 有穷性 一个算法包含有限个步骤，即算法经过有限步执行后必须终止。

- 确定性 算法的每一步所规定的动作不能有两种以上的理解，即算法每一步的动作不能有语义二义性。

- 有输入 一个算法有一个或多个输入，输入是执行算法时所需的信息，包括被算法处理的数据和执行的控制信息。


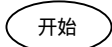
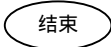
- 有输出 一个算法有一个或多个输出，输出是算法执行的结果。

- 有效性 算法每一步所规定的动作都能够执行。例如两个数相除，当除数为零时则不能执行除运算，即算法所规定的动作无法执行。

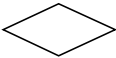
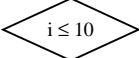
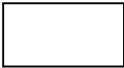
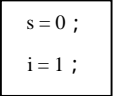

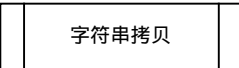

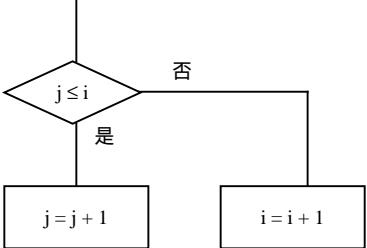

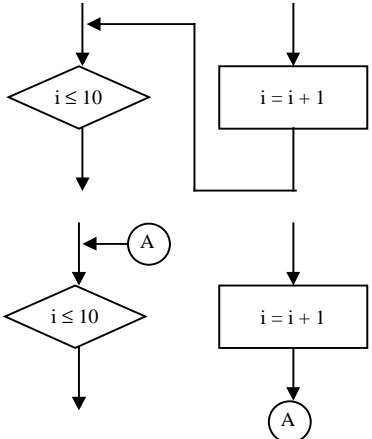
2. 算法的表示

将算法用一种适当的方式描述出来称为算法的表示。算法的表示有多种方式，本书采用了两种最基本和最常用的方式——自然语言和传统流程图来表示算法。传统流程图通常简称流程图或框图。流程图用图形符号（框和线）表示算法的每一步及各步之间的联系，流程图常用符号及其含义如表 1-1 所示。

表 1-1 流程图常用符号

符 号	符号名称	意 义	例
	起止框	表示算法的开始或结束	 

续表

符 号	符号名称	意 义	例
	判断框	表示判断选择：根据框中条件从两种可选动作中选一执行	
	处理框	表示按顺序执行的处理	
	调用框	表示调用函数	
	流程线	表示两个步骤相邻，且执行顺序由箭尾一方到箭头一方。对于自上而下和自左而右的顺序，箭头可省缺	
	连接点	连接点必须以相同的形式成对出现，用于表示一条流程线被断开后的两个端点	下面两种表示是等价的： 

3. 算法表示举例

问题：求 $s = \sum_{n=1}^{10} n^n$

下面分别用自然语言和流程图表示该级数求和的解题算法。

● 用自然语言表示

用变量 s 存放各项的累加和，置初值为 0；用变量 i 作项数计数器，置初值为 1。

如果 $i > 10$ 则计算 s ，否则转步骤 。

计算 s ：

- 1 用变量 j 作每一项的累乘次数计数器，置初值为 1；
 - 2 用变量 ai 存放第 i 项的累乘积，置初值为 1；
 - 3 如果 $j > i$ 则计算第 i 项，否则转步骤 -4；
- 计算第 i 项：

- 3-1 $a_i = a_i * i$;
- 3-2 $j = j + 1$, 转步骤 -3 ;
- 4 将第 i 项加到累加和中去 :
 - 4-1 $s = s + a_i$;
 - 4-2 $i = i + 1$, 转步骤 ;

输出 s , 结束。

● 用流程图表示

其具体流程图见图 1.2。

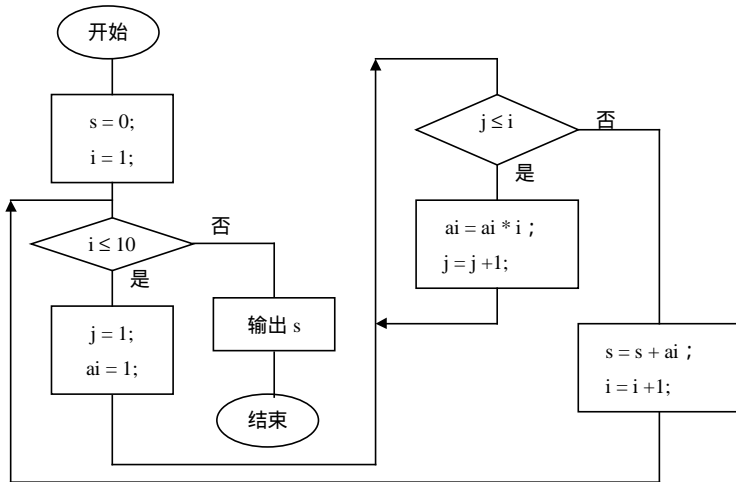


图 1.2 计算 $s = \sum_{n=1}^{10} n^n$ 的算法流程图

显然，用流程图表示算法比用自然语言表示算法更为简明、直观。

1.1.3 程序设计及程序设计语言

程序是用程序设计语言表示的计算机解题算法或计算机解题任务。程序设计是将解题任务转变成程序的过程，一般包括分析问题，确定算法（对复杂算法需画出流程图），用选定的程序设计语言编写源程序，最后上机调试、运行程序等基本步骤。

程序设计语言是计算机能够理解的、用于人和计算机通讯之间的语言，程序设计语言由低级到高级可分为三类：低级语言、高级语言和专用语言。

1. 低级语言

低级语言又分为机器语言、符号语言和汇编语言。机器语言用二进制代码表示机器指令和数据，机器语言程序能够直接被机器理解和执行，因而程序效率高；但编程繁琐，且不利于记忆和阅读，因而程序维护困难。符号语言用符号代替二进制代码表示机器指令，汇编语言进而用符号来表示指令和数据的内存地址。现在人们用低级语言编程通常指用汇编语言编程。

用汇编语言编写的程序称为汇编语言源程序（简称汇编语言程序）。汇编语言程序必须被转换为机器语言程序才能被计算机理解和执行，完成这种转换任务的系统软件称为汇编程序；

这种转换过程称为汇编。低级语言是面向机器的，用低级语言写的程序效率高，但没有可移植性，即不能从一个机器系统上移到另一个机器系统上运行。此外，用机器语言编写程序要求程序员必须懂得具体机器系统的硬件结构（指令系统）。

2. 高级语言

高级语言是类似于人类自然语言，英语和数学语言的程序设计语言，例如 C 语言和 Pascal 语言。用高级语言编写的程序称为高级语言源程序，简称**源程序**或程序。例如 C 源程序或 C 程序，Pascal 源程序或 Pascal 程序。

与汇编语言源程序类似，高级语言源程序也必须被转换成机器语言程序才能够被机器理解和执行，完成这种转换任务的系统软件称为**编译程序**。例如，C 编译程序、Pascal 编译程序。将高级语言源程序转换成机器语言程序的过程称为**编译**。

高级语言是面向解题过程的，语言本身与具体机器系统无关，因而用高级语言编写的应用程序可移植性好。编译程序是一种语言的具体实现，编译程序与具体机器系统有关，人们常将一个编译程序称为某种语言的一个版本。同一种语言的不同版本不完全相同，在使用一种具体的高级语言及其编译程序开发软件时，必须参考与编译程序配套的有关资料。本书阐述的内容遵从 C 标准，对于大多数 C 编译程序具有通用性。考虑到上机环境，尽量简单以便于学习和练习，书中所有例题均在 Turbo C 2.0 上运行通过。

3. 专用语言

专用语言是专门为某个应用领域而设计的计算机程序设计语言。例如，数据库系统中的数据查询语言 SQL、军事应用领域的 ADA 语言、商业应用领域的 COBOL 语言等。专用语言是面向问题的语言，它比高级语言更抽象，描述能力比高级语言更强。用专用语言编程不需要指出“如何做”，只需说明“做什么”。

1.2 C 语言的发展过程及特点

1.2.1 C 语言的发展过程

C 语言是迄今为止世界上应用最广泛的通用程序设计语言之一。C 语言的发展过程可粗略地分为诞生、发展和成熟三个阶段。

1. C 语言的诞生（1970 年~1973 年）

C 语言是由于编写 UNIX 操作系统的需要而诞生的，作者是美国 AT&T 公司贝尔实验室的 Dennis. M. Ritchie。C 语言的前生是 B 语言（1970 年，作者：贝尔实验室的 Ken Thompson），B 语言的前生是 BCPL 语言（1967 年，作者：英国剑桥大学的 Martin Richards）。

B 语言被用在 PDP_7 计算机上实现了第一个 UNIX 操作系统。1973 年至 1975 年 K.Tompson 和 Dennis. M. Ritchie 用 C 语言重写了 UNIX 操作系统，先后推出了 UNIX V5，UNIX V6。此时的 C 语言是附属于 UNIX 操作系统的。

2. C 语言的发展 (1973 年~1978 年)

1977 年 C 语言作者发表了不依赖于具体机器系统的 C 语言编译文本《可移植 C 语言编译程序》，1978 年又发表了对上述编译文本的补充《C 语言修订报告》，从而使 UNIX 操作系统推广到各种计算机上并推动了 UNIX 操作系统的不断发展：1978 年推出了 UNIX V7，1983 年推出了 UNIX System 。UNIX 操作系统的巨大成功和广泛使用展示了 C 语言的突出优点，从而又促进了 C 语言的迅速推广。同时，C 语言也伴随着 UNIX 操作系统的发展而不断发展。

1978 年 Brian W. Kernighan 和 Dennis. M. Ritchie 以 UNIX V7 中的 C 编译程序为基础写了经典著作《The C Programming Language》，这本书是以后介绍 C 语言的各种书籍的蓝本。由《可移植 C 语言编译程序》和《C 语言修订报告》定义的 C 语言被称为**传统 C**。1978 年以后，C 语言先后移植到各种计算机系统上，且不再依赖于 UNIX 操作系统而独立存在。

3. C 语言的成熟 (1988 年以后)

C 语言的不断发展产生了各种 C 语言版本，不同的 C 语言版本对传统 C 都有所扩充。1988 年，美国国家标准协会 (ANSI) 在综合各种 C 语言版本的基础上制定了 C 语言文本标准，称为 ANSI C 标准。ANSI C 实现了 C 语言的规范化和统一化，B. W. Kernighan 和 Dennis. M. Ritchie 按照 ANSI C 标准重写了《The C Programming Language》一书，于 1990 年正式发表了《**The C Programming Language Second Edition**》。1990 年国际标准化组织 (ISO) 公布了以 ANSI C 为基础制定的 C 语言的国际标准 ISO C，人们通常称之为**标准 C**。C 语言标准的制定标志着 C 语言的成熟，1988 年以后推出的各种 C 语言版本对标准 C 是兼容的。

1.2.2 C 语言的特点

C 语言之所以成为目前世界上使用最广泛的程序设计语言，并被选作适应近代软件工程需要而发展起来的面向对象的程序设计语言 C++ 的基础语言，是由于 C 语言的诸多突出优点所决定的。下面从语言本身和应用角度两个方面概括了 C 语言的主要特点。

C 语言的突出优点是表达能力强、语言简洁、使用灵活、效率高、可移植性好。

- 表达能力强 数据类型和运算符丰富；可以直接访问内存物理地址和硬件寄存器；能够表示直接由硬件实现的针对二进制位的运算。

- 语言简洁 只有 6 种基本语句 (见第 3 章)，且运算符、语句等语言成份的表示简明扼要，使 C 源程序显得精练。

- 使用灵活 C 语言是弱类型语言，基本类型的变量及其值在数据类型上不要求严格对应；表达式可以作为语句使用；用非零及整数零表示具有真假含义的逻辑值；数组元素和结构成员既可以用其名称访问也可以用指针访问，即同一数据可以有多种等价的表示形式。

- 效率高 C 语言在代码质量上几乎可以与汇编语言媲美，即 C 源程序经编译后生成的目标程序运行速度快且占用的存储空间小。

- 可移植性好 由于 C 语言的标准化，以及 C 程序的输入输出、内存管理等操作采用 C 库函数实现，不仅使得 C 编译程序很容易在不同的机器系统上实现，而且使得用户 C 程序可以不作修改或作少量修改就能在不同的机器系统上运行。

注意：C 语言使用灵活是 C 语言深受程序员喜爱的原因之一。但如果用法不正确，会使写出的程序不能得到正确的结果，或不能正常运行，甚至死机。用法上的错误不属于语法错，

因此编译是检查不出来的。初学者往往误认为没有语法错的程序就是正确的程序，编写正确、可靠的 C 程序要靠程序员正确使用 C 语言来保证。

例如最常见的用法错误有：写表达式时运算符的优先级与结合性同所要表示的逻辑不一致，或未注意表达式求值的顺序、表达式求值的副作用、后缀式自增自减的计算延迟、运算过程中的类型转换等对计算结果的影响；写循环语句时循环变量未赋初值，或用作循环条件的表达式的值永为真（非 0），或组成循环体的多个语句未写成复合语句；写格式输入输出语句时用于说明数据格式的转换字符与数据的类型不一致；特别是使用指针时没有向指针赋值之前就向指针所指对象赋值，或引用指针所指对象的值，或写地址表达式时不注意指针的类型等都是严重的用法错误。

1.3 C 程序的基本结构

一个计算机高级语言程序均由一个主程序和若干个（包括 0 个）子程序组成，程序的运行从主程序开始，子程序由主程序或其他子程序调用执行。在 C 语言中，主程序和子程序都称之为函数，规定主函数必须以 main 命名。因此，一个 C 程序必须由一个名为 main 的主函数和若干个（包括 0 个）子函数（简称函数）组成，程序的运行从 main 函数开始，其他函数由 main 函数或其他函数调用执行。

下面是几个具有代表性的简单 C 程序。

【例 1.1】 输出字符串 “ The c programming language ”，并在输出后换行。

程序 1.1

```
/* 1*/ /*example1-1.c*/  
/* 2*/ #include<stdio.h>  
/* 3*/  
/* 4*/ void main ( void )  
/* 5*/ {  
/* 6*/     printf ( "The c programming language\n" );  
/* 7*/ }
```

程序运行时输出下面一行信息，显示器屏幕光标停在下一行的行首：

```
The c programming language  
_
```

程序 1.1 仅由一个主函数（4~7 行）组成。

第 1 行：/*example1.1.c */是注释，/*是注释的开始符号，*/是注释的结束符号，注释符号之间的文字是注释内容。/*和*/必须成对出现且不能嵌套，例如/*.../*...*/...*/是非法的。注释是供程序员看的，编译对注释不作翻译。注释内容任意，例如 example1-1.c 为该程序的磁盘文件名，也可以是对程序功能、被处理数据或处理方法的说明。注释可以出现在程序中分隔符（见 1.4.3 节）可以出现的任何位置。

第 2 行：#include 是编译程序的预处理指令（见 5.7.2 节），它不是 C 语言的语句，指令末尾不能加分号。stdio.h 是 C 编译程序提供的系统头文件（或称为包含文件）之一，程序中凡是调用了标准输入输出函数则必须在调用之前写上#include<stdio.h>（**注意**，只有在 DOS 操作系统环境下可以缺省#include<stdio.h>）。预处理指令必须独占一行，一般写在一个源程序文