

21 世纪高等职业教育计算机技术规划教材

程序设计基础——C 语言篇

王桂芝 主编

连卫民 何 樱 副主编

人民邮电出版社

图书在版编目 (CIP) 数据

程序设计基础 . C 语言篇 / 王桂芝主编 . —北京 : 人民邮电出版社 , 2005.10

21 世纪高等职业教育计算机技术规划教材

ISBN 7-115-14028-6

. 程... . 王... . 程序设计 - 高等学校 : 技术学校 - 教材 C 语言 - 程序设计 -
高等学校 : 技术学校 - 教材 . TP311 TP312

中国版本图书馆 CIP 数据核字 (2005) 第 107908 号

内 容 提 要

本书主要介绍 C 语言的基本概念和语法结构 , 并通过大量实例介绍基于 C 语言的程序设计的基本概念、方法和技巧。内容包括 : 程序设计概述 , C 语言的基本知识 , 顺序结构的程序设计 , 选择结构的程序设计 , 循环结构的程序设计 , 模块化的程序设计 , 数组的程序设计 , 指针的程序设计 , 编译预处理与记录类型 , 文件的程序设计 , 图形的程序设计。在这些内容的讲解过程中 , 坚持语言为程序设计服务的原则 , 讲究实用性。

本书内容新颖、实例丰富 , 强调程序设计的方法和技巧 , 注重培养程序设计的思维方式和提高程序的开发能力。编者力求概念准确、编排合理、循序渐进、通俗易懂。本书的读者对象定位为程序设计的初学者 , 阅读本书不需具备任何编程基础。

本书为各类高等职业院校计算机专业和非计算机专业的教材 , 也可供计算机爱好者自学 , 同时可作为从事软件开发、程序设计及计算机教学人员的参考书。

21 世纪高等职业教育计算机技术规划教材

程序设计基础——C 语言篇

-
- ◆ 主 编 王桂芝
 - 副 主 编 连卫民 何 樱
 - 责任编辑 赵慧君
 - ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号
 - 邮编 100061 电子函件 315@ptpress.com.cn
 - 网址 <http://www.ptpress.com.cn>
 - 北京艺辉印刷有限公司印刷
 - 新华书店总店北京发行所经销
 - ◆ 开本 : 787×1092 1/16
 - 印张 : 15.5
 - 字数 : 371 千字 2005 年 10 月第 1 版
 - 印数 : 1 - 3 000 册 2005 年 10 月北京第 1 次印刷

ISBN 7-115-14028-6/TP · 4982

定价 : 22.00 元

读者服务热线 : (010)67170985 印装质量热线 : (010)67129223

编者的话

随着计算机的广泛应用，各行业对计算机的应用要求也在不断提高，少量的商业软件远远不能很好地解决本部门、本行业的实际要求。于是，许多单位需要研发自己的应用软件；另一方面，随着应用软件的大量使用，需要许多人员进行软件维护工作。因此，对计算机应用人员的要求已不再停留在操作员的层次，而是要求他们具有一定的程序设计技术，以解决本行业、本单位更为复杂和专业的问题。

为适应时代要求，提高计算机应用人员的程序设计能力，就应该从程序设计语言入手、从实际应用出发，循序渐近、逐步培养。在众多的程序设计语言中，C 语言具有功能丰富、表达力强、使用灵活、程序高效、应用面广等特点，它既具有高级语言的形式，又具有低级语言的功能。因此，C 语言已被广大计算机专业人士和程序爱好者所钟情。另外，C 语言作为 C++ 的一个子集，可以为进一步学习 C++ 和 VC++ 打下良好的基础。C/C++ 语言几乎具备了现代程序设计语言的所有语言成分，一旦掌握了 C/C++ 语言，就可以较为轻松地学习其他任何一种程序设计语言。现在很多学校把 C 语言作为程序设计的入门语言。

目前有关 C 语言的书籍很多，但大多书中对 C 语言的一些语法细节太过强调，从而使学习者感觉 C 语言太琐碎，难以掌握。本书根据计算机技术的最新发展，采用“以用促学”的编写原则，对其内容及表现方式作了精心安排，具有如下特点。

1. 起点低，要求少。

本书的读者对象定位为程序设计的初学者，阅读本书不需具备任何编程基础。在本书的第 1 章介绍了与程序设计有关的计算机基础知识。因此，即使没有任何计算机知识，也能很好地使用本书。

2. 语言严谨，通俗易懂。

C 语言语法灵活，编程方便，但容易给读者的理解造成偏差。本书采用既通俗又严谨的语言风格，使初学者能够建立起正确的概念，准确掌握 C 语言本身的语法特点。

3. 详略得当，组织合理。

本书本着“够用”的原则，主要讲解 C 语言最基本、最常用的内容，避开一些使用频率很低且比较烦琐的内容。在内容组织上，注意分散难点，有意减轻学习疲劳。

4. 突出“程序设计”这一主题。

坚持语言为程序设计服务的原则。在介绍 C 语言基本知识的过程中，通过大量的实例，引导读者掌握程序设计的基本思路和方法。

本书由连卫民、王桂芝制定编写大纲，王桂芝负责定稿工作。全书共 11 章：第 1 章和附录内容由连卫民编写；第 2、3、6 章由何樱编写；第 4、5 章由李丹编写；第 7、9 章由孙凌编写，第 8 章由田源编写；第 10 章由王桂芝编写；第 11 章由谢莉（黄委会水文局信息中心）编写。

在本书的编写过程中，我们参阅了大量的资料，在此对所有的编著者表示衷心的感谢。由于编者水平有限，时间仓促，书中难免存在不当之处，恳请有关专家和广大读者批评指正。

编者
2005 年 8 月

目 录

第 1 章 程序设计概述	1
1.1 程序设计的基本概念	1
1.1.1 计算机系统的组成	1
1.1.2 软件	2
1.1.3 程序设计	4
1.2 程序设计的过程	4
1.2.1 程序设计的因素	4
1.2.2 程序设计的步骤	5
1.2.3 程序设计的注意事项	6
1.3 算法的描述工具	7
1.3.1 采用自然语言描述	7
1.3.2 采用图示方法描述	7
1.3.3 采用计算机语言描述	8
1.3.4 程序评判的标准	9
1.3.5 结构化程序设计方法简介	10
1.4 C 程序调试	10
1.4.1 C 语言概述	10
1.4.2 程序调试的步骤	11
1.4.3 C 程序的调试环境及上机操作实现	12
本章小结	16
练习题	16
第 2 章 C 语言的基本知识	17
2.1 数据类型	17
2.1.1 数据类型概述	17
2.1.2 基本的数据类型	18
2.2 数据的表示	21
2.2.1 常量的表示	21
2.2.2 变量的表示	21
2.3 数据的运算	26
2.3.1 算术运算	27
2.3.2 赋值运算	28
2.3.3 自增运算和逗号表达式	29

2.3.4 各类数值型数据间的混合运算	32
2.3.5 关系运算和逻辑运算	32
2.4 语句的表示	35
2.4.1 语句的分类	35
2.4.2 语句的特点	36
2.5 程序的组成	36
2.5.1 C 程序的组成	36
2.5.2 C 程序的设计过程	37
本章小结	37
练习题	38
第 3 章 顺序结构的程序设计	41
3.1 顺序结构程序的概念	41
3.1.1 顺序结构的概念	41
3.1.2 顺序结构程序解决的问题	41
3.2 顺序结构采用的语句	42
3.2.1 printf 函数	42
3.2.2 putchar 函数	46
3.2.3 scanf 函数	47
3.2.4 getchar 函数	50
3.3 顺序结构程序设计举例	50
本章小结	53
练习题	53
第 4 章 选择结构的程序设计	56
4.1 选择结构程序的概念	56
4.1.1 选择结构的概念	56
4.1.2 选择结构程序解决的问题	56
4.2 选择结构采用的语句	57
4.2.1 条件语句	57
4.2.2 多路分支语句	61
4.3 选择结构程序设计实例	63
4.3.1 二路分支程序设计	63
4.3.2 多路分支程序设计	64
4.3.3 复杂条件的程序设计	66
4.3.4 嵌套条件的程序设计	67
本章小结	69
练习题	69
第 5 章 循环结构的程序设计	73
5.1 循环结构程序的概念	73

5.1.1 循环结构的概念	73
5.1.2 循环结构程序解决的问题	73
5.2 循环结构采用的语句	74
5.2.1 while 语句	74
5.2.2 do-while 语句	75
5.2.3 for 语句	76
5.3 循环结构程序设计实例	79
5.3.1 当型循环的程序设计	79
5.3.2 直到型循环的程序设计	80
5.3.3 记数循环的程序设计	81
5.3.4 双层循环的程序设计	82
5.3.5 多层循环的程序设计	83
本章小结	84
练习题	85
第 6 章 模块化的程序设计	88
6.1 模块化结构程序的概念	88
6.1.1 模块化结构的概念	88
6.1.2 函数的概念	88
6.1.3 模块化结构程序解决的问题	89
6.2 模块化结构程序的实现	89
6.2.1 函数的定义和调用	89
6.2.2 函数的嵌套调用和递归调用	95
6.3 变量的存储属性	98
6.3.1 内部变量	99
6.3.2 外部变量	102
6.3.3 静态变量	105
6.4 模块化结构程序设计实例	106
本章小结	110
练习题	110
第 7 章 数组的程序设计	116
7.1 数组程序的概念	116
7.1.1 数组的基本概念	116
7.1.2 数组程序解决的问题	116
7.2 数组的定义与使用	117
7.2.1 一维数组的定义与使用	117
7.2.2 二维数组的定义与使用	120
7.2.3 字符数组的定义与使用	122
7.3 数组程序设计实例	126

7.3.1 一维数组的程序设计	126
7.3.2 二维数组的程序设计	128
7.3.3 字符数组的程序设计	130
本章小结	132
练习题	133
第 8 章 指针的程序设计	137
8.1 指针的基本概念	137
8.1.1 指针的定义	137
8.1.2 引入指针的原因	138
8.2 指向变量的指针变量	139
8.2.1 指向变量的指针变量的定义	139
8.2.2 对指针变量的处理	139
8.3 指向数组的指针变量	141
8.3.1 指向一维数组的指针变量	141
8.3.2 指向二维数组的指针变量	144
8.3.3 指向字符串的指针变量	148
8.4 有关的函数的指针变量	150
8.4.1 指针变量作为函数参数	150
8.4.2 指向函数的指针变量的定义	152
8.4.3 指向函数的指针变量的处理	153
8.4.4 返回指针值的函数	153
8.5 指针数组和指向指针的指针	154
8.5.1 指针数组	154
8.5.2 指向指针的指针	157
8.6 指针程序设计实例	158
8.6.1 变量的指针	158
8.6.2 数组的指针	158
8.6.3 函数的指针	161
本章小结	163
练习题	164
第 9 章 编译预处理与记录类型	166
9.1 编译预处理的使用	166
9.1.1 编译预处理的作用及基本特点	166
9.1.2 文件包含	166
9.1.3 宏定义	167
9.1.4 条件编译	171
9.2 构造的数据类型	172
9.2.1 构造的数据类型	172

9.2.2 用户自定义类型的一般方法	173
9.3 记录类型的定义与使用	174
9.3.1 记录类型的定义	174
9.3.2 记录型变量的定义与使用	175
9.3.3 记录型数组的定义与使用	178
9.3.4 记录型指针的定义与使用	179
9.4 记录类型的程序设计实例	181
9.4.1 单链表中结点的动态生成和释放	182
9.4.2 单链表的建立与输出	183
9.4.3 单链表的插入与删除	184
本章小结	185
练习题	186
第 10 章 文件的程序设计	190
10.1 文件的基本概念	190
10.1.1 文件的定义	190
10.1.2 缓冲文件系统	191
10.1.3 文件的处理过程	192
10.1.4 文件程序解决的问题	193
10.2 文件处理采用的函数	193
10.2.1 打开函数	193
10.2.2 读取函数	195
10.2.3 写入函数	195
10.3 文件程序设计实例	195
10.3.1 文件的建立	195
10.3.2 文件的查看	197
10.3.3 文件的修改	199
10.3.4 文件的复制	201
本章小结	203
练习题	204
第 11 章 图形程序设计	208
11.1 图形程序设计的基本概念	208
11.2 文本方式下的字符图形程序设计	210
11.2.1 一个字符图形程序实例	210
11.2.2 字符图形程序设计常用的函数	210
11.2.3 字符图形程序设计实例	212
11.3 图形方式下的图形程序设计	213
11.3.1 一个图形程序实例	214
11.3.2 图形程序设计常用的函数	214

11.3.3 图形程序设计实例	216
本章小结	221
练习题	221
附录 1 常用字符的 ASC II 码表	223
附录 2 C 的库函数	225
附录 3 编译错误信息	229
主要参考书目	238

第 1 章

程序设计概述

本章教学目标：

- 熟悉程序设计的基本概念和程序设计的基本过程；
- 熟悉算法的描述和 C 程序的运行过程。

主要知识点：

- 程序设计的基本概念；
- 程序设计的过程；
- 算法的描述工具；
- C 程序调试。

1.1 程序设计的基本概念

程序设计的基本概念是学习计算机软件的基础。本节主要介绍计算机系统的组成、软件、程序设计的概念。

1.1.1 计算机系统的组成

根据冯·诺依曼设计计算机的思想，计算机系统由计算机硬件和计算机软件两部分组成。

1. 计算机硬件

计算机硬件是指组成计算机系统的设备或机器，是“看得见，摸得着”的物理部件，它是组成计算机系统的基础。计算机硬件一般包括中央处理器（CPU）、内存储器、外存储器、输入设备和输出设备。中央处理器包括运算器和控制器。CPU 与内存储器合称为主机，外存储器、输入设备和输出设备合称为外部设备。计算机硬件的组成可以用下列公式表示，它们之间的关系如图 1-1 所示。

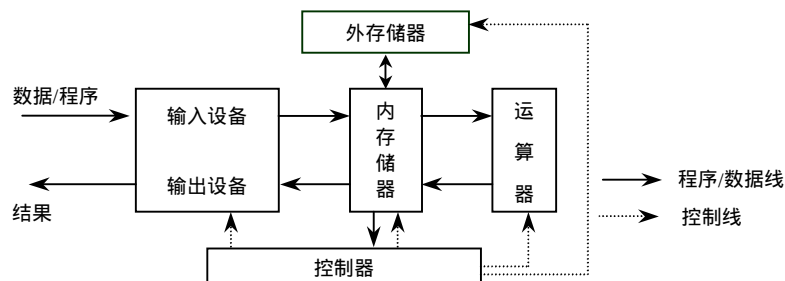


图 1-1 计算机硬件之间的关系

计算机硬件=主机（CPU+内存）+外设（外存储器+输入设备+输出设备）

2. 计算机软件

计算机软件是指组成计算机系统的程序、数据和文档。程序是指令的有序集合；数据是信息在计算机中的表示，是计算机处理的对象；文档是各种说明文本，是软件操作的辅助性资源。计算机的所有工作都必须在软件的控制下才能进行，没有软件的计算机称为“裸机”，是任何工作都不能做的。计算机软件的组成可以用下列公式表示。

计算机软件=系统软件（OS+DBMS+编译+服务程序）+应用软件（源程序+应用软件包）

计算机硬件和计算机软件在计算机系统中是相辅相成、缺一不可的，它们共同组成了计算机系统，如图 1-2 所示。计算机硬件是计算机的躯体和基础，计算机软件是计算机的头脑和灵魂，没有软件的计算机和缺少硬件的计算机都不能成为完整的计算机系统。二者相互推动，共同促进计算机的发展。

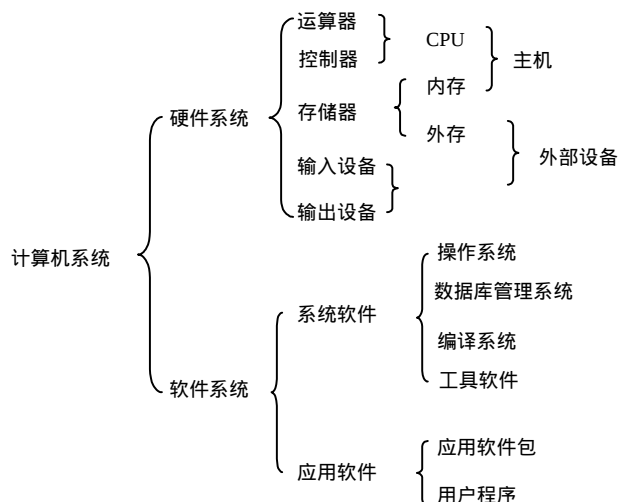


图 1-2 计算机系统的组成

1.1.2 软件

软件指程序及程序运行所需的数据以及与程序有关的文档资料的总称。根据软件的作用可以把软件分为系统软件和应用软件。系统软件是支持和管理计算机硬件的软件，是服务于硬件的，它创立的是一个平台；应用软件是完成用户某项要求的软件，是服务于特定用户的，它满足某一个应用领域。软件的作用如图 1-3 所示。

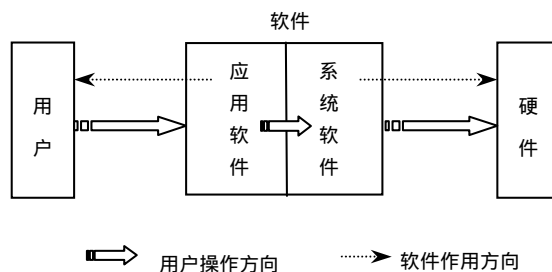


图 1-3 软件的作用

系统软件包括操作系统、数据库管理系统、计算机编译语言和各种系统服务性程序。应用软件包括计算机源程序和应用软件包。所有这些软件，操作系统是基础，是其他软件的平台。没有操作系统，其他软件就无法工作。

从软件的构成上，可以用以下公式描述软件：

$$\text{软件} = \text{程序} + \text{数据} + \text{文档资料}$$

1. 程序

通俗地讲，程序是让计算机解决问题的方法步骤。从专业的角度讲，程序是指令的有序集合，即程序=算法+数据结构。算法就是解决问题的方法。数据结构就是要计算机处理的数据在计算机中的存储和描述，即为数据的表示范围、数据间的关系和数据所能参与的运算。

也可以说程序是符号+规则，用规定的符号和符合规则的一组有意义的有序的指令集合称之为程序。为了解决同一个问题而编写的若干个程序的集合叫做程序集。在程序集中，首先被执行的程序叫主程序，在主程序执行期间被调用的程序称之为子程序。在程序定义中的“有意义”强调了程序要解决问题的目的性。

要让计算机按人们的意图做某项工作，必须使计算机能接受人们向它发出的各种命令和信息。但目前的大部分计算机还不能直接懂得人类的自然语言，只能接受用特定的信息代码编写成的一条条指令。计算机能识别并执行的信息符号代码的指令集合称为计算机语言，根据其发展，把它分为机器语言、汇编语言和高级语言三类。

(1) 机器语言。机器语言是指用二进制代码直接表达指令的语言。机器语言程序可让计算机识别并执行，但由于每条指令都是由一连串的0和1组成，不好记忆，容易出错，不同机型的指令代码不同，一般的计算机用户很少使用它设计程序。

(2) 汇编语言。汇编语言是指用一些指令助记符（指令功能的英文词缩写）来表示机器语言每条指令的一种语言。如加法指令 ADD，减法指令 SUB，传送数据指令 MOV 等。汇编语言的指令要比机器语言易于记忆理解，不易出错。但这种语言没有从根本上减少指令，且烦琐，不易移植，功能弱，一般用户很少用这种语言编程序。用汇编语言编写的程序不能被计算机直接执行，必须经过翻译，转换成机器语言程序后，才能执行。

(3) 高级语言。高级语言是比较接近人类自然语言和数学描述的语言。高级语言一条命令的功能可以代替几条、几十条汇编语言的指令功能。因此，高级语言更简便、易学、易记，通用性强，兼容性好，便于移植。高级语言是目前广泛使用的程序设计语言，常用的有 BASIC、Fortran、Pascal、C 语言等。

高级语言虽然好学易用，但用高级语言编写的源程序，实际上是不能直接在计算机内运行的，还必须经过翻译转换成计算机能识别的机器语言程序，即目标程序，才可运行。

高级语言通常采用编译程序和解释程序两种翻译方式。如图 1-4 所示。

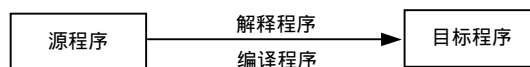


图 1-4 源程序的两种翻译方式

解释程序是边翻译边运行，即从源程序中取一条指令，转换一次执行一次，直到结束。

编译程序是翻译完后再运行，即将源程序的每条指令全部转换成机器语言表达的程序（目标程序）再运行。每一种高级语言，都有一套与之相对应的翻译程序。

2. 数据

数据有广义和狭义之分。广义的数据概念是指对客观事物的描述，即用语言、文字、声

音、图形和图像等手段对客观事物的形状、颜色、质地、运行状态等特性的描述。也就是用数字表示的是数据,用其他符号表示的也是数据。如 268 是数据,“我是一个学生”也是数据,一张照片、一段录音、一段电影都是数据。

在讲狭义的数据之前,先介绍一个概念——信息。信息是有用的数据或加工后的数据。所有的描述都是数据,但不一定是信息。今天对我来说数据有用,它就是信息;明天对我来说数据无用,就不是信息了。所以,信息具有一定的时效性,相对性。

从狭义上来讲,也就是是否能被计算机接收和处理,若能被计算机接收和处理的信息就是计算机数据。在计算机内部,通常数据和信息不加区分,数据就是信息,信息就是数据。在计算机中往往把数据分成不同的类型,对数量描述的能进行算术运算的数据是数值型数据,如 268。对事物性质描述的是字符型数据,如“我是一个学生”。

3. 文档资料

文档资料是由应用软件和系统软件产生的信息的集合,一般以文件的形式保存在磁盘上。在程序设计中,所处理的文档资料大多是数据文件。

1.1.3 程序设计

程序设计就是利用某种计算机语言,按照一定的算法,对特定的数据进行处理。程序设计的最终结果是用某种语言描述的计算机程序。

在程序设计的学习过程中,必须培养计算机解决问题的能力,计算机处理问题的意识。俗话说“三句话不离本行”,就是指这个人的专业意识很强,那么用计算机处理问题的意识是什么呢?一般要考虑如下 4 个问题:

第一,必须考虑数据在计算机中怎样描述,即存储类型;

第二,如何把数据输入到计算机的内存中;

第三,如何把已知数据经过一定的处理,得到所需要的结果;

第四,如何把结果输出来。

其处理过程如图 1-5 所示。首先要把计算机之外的信息输入到计算机内,然后由已知数据根据一定的算法进行处理,得到结果数据,最后把计算机结果输出到显示器或打印机上。

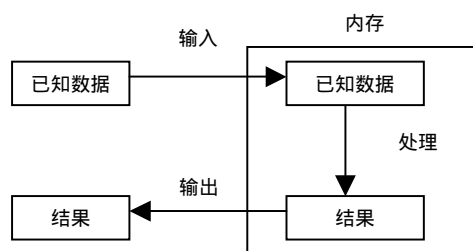


图 1-5 计算机处理问题的过程

1.2 程序设计的过程

1.2.1 程序设计的因素

程序设计就是设计程序,而程序=算法+数据结构,算法必须与计算机处理问题的步骤相一致。所以,程序设计的主要因素就是考虑算法和设计数据结构。

1. 算法

算法是为了解决某个特定问题而采取的确定且有限的步骤。所以,算法的要素主要是操作

和控制结构。即一个算法由一些操作组成，这些操作又是按一定的控制结构所规定的次序执行的，计算机可以执行的基本操作是以指令的形式向人们提供的，指令是计算机所完成的最基本的功能。

计算机的基本操作有算术运算、关系运算、逻辑运算、数据传送即赋值运算。算法的功能不仅取决于所选用的操作，而且还决定于控制结构。算法的控制结构有顺序、选择和循环三种。算法的控制结构可以用图示的方法形象地表示出来。这一点将在下一节讲解。

一个算法有以下3个特性：

(1) 有穷性。一个算法通过若干步骤即可实现预定目标，算法的每一步都可以在合理的时间内完成。

(2) 确定性。算法中的每一条指令都必须有确切的含义，同样的步骤、同样的条件结果也是相同的。

(3) 输入和输出。

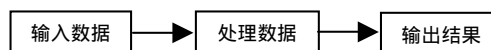


图 1-6 计算机的工作步骤

计算机的工作可以分为3步，如图1-6所示。

首先，把要处理的数据输入到计算机中；然后，根据处理要求对输入的数据进行处理，得到处理结果；最后，把结果显示或打印出来。

计算机的工作步骤具有不可逆的方向性。再简单、再复杂的计算机也概莫能外。也就是说，一个算法再简单，也要有输入、处理、输出3个步骤。

2. 数据结构

数据结构是相互之间存在一种或多种特定关系的数据元素的集合。数据结构的基本类型有4种。定义中的“关系”描述的是数据元素之间的逻辑关系，称为逻辑结构。数据结构在计算机中的表示（映象）称为数据的物理结构，有顺序结构（如数组）和随机结构（如链表）两种。数据结构要解决的3个问题是：数据的表示、数据元素间的关系以及在这种关系上的运算。

对计算机语言来说，数据结构体现为数据类型。数据类型是一个值的集合和定义在这个值集上的一组操作的总称，它是信息特性的抽象。

从硬件上看，引入数据类型的目的是为了了解释计算机内存中信息含义的一种手段；从用户的角度看，引入数据类型实现了信息的隐蔽，即将一切用户不必了解的细节都封装在类型中。

3. 算法与数据结构的关系

算法与数据结构是人们用计算机解题时所作的两种抽象。算法是从计算机的操作的角度对解题过程的抽象；数据结构是从如何组织被处理对象的角度进行抽象。这两种抽象相互依赖，互为补充，有效地降低了问题的复杂性。

1.2.2 程序设计的步骤

程序实际上是问题的实际处理过程在计算机中的体现。利用程序来解决实际问题，需要经过分析问题和解决问题两大步。

1. 分析问题

首先必须分析实际的处理过程，确定已知的数据、结果数据以及如何由已知获得结果。在这里重点考虑的是数据在计算机中的描述形式和处理过程。数据在计算机中的描述形式就

是在计算机中数据采用什么样的数据类型存储。选择数据类型的依据是数据表示的范围和参与的运算。数据的处理就是指数据如何输入，如何处理，如何输出。数据处理是程序中的主要算法，即由已知数据求结果数据的过程。

分析的结果，可以用图示的方法来表示。即用各种易于人们理解和接受的方式来表示数据的处理过程。请参考下一节内容。

2. 解决问题

解决问题就是把分析的结果用计算机语言描述出来，并在计算机上调试运行。对于 C 语言来说就是用 C 语言的符号和规则来描述，即编写 C 语言程序，在 C 语言的编辑和编译环境下调试和运行程序的过程。这里重点是掌握 C 语言的符号和规则，熟悉编辑和编译的环境。要熟悉和掌握 C 语言的符号和规则，重要的方法是通过大量的程序编写和上机调试。

解决问题的结果，就是要产生一个能够正确解决实际问题的可以运行的计算机程序。这个程序可以是源程序，也可以是编译后的可执行程序。

1.2.3 程序设计的注意事项

在程序设计的过程中，始终要注意这么几个问题：数据在计算机内存中如何存储，数据如何输入，数据如何处理，数据如何输出。

1. 数据存储

数据在计算机内存中的存储是通过变量来表示的，这就要重点考虑变量的数据类型。在计算机语言中，数据类型确定了这种数据的存储长度、表示范围和所允许的运算。如在 C 语言中，整型数据存储长度是 2 个字节，数据的表示范围是 $-32768 \sim 32767$ ，可以进行算术运算、关系运算和逻辑运算。

数据类型一般有基本的数据类型和构造的数据类型之分。基本的数据类型就是计算机语言已经定义的数据类型，即标准的数据类型，用户可以直接使用的数据类型。构造的数据类型是需要用户在使用之前先定义的数据类型。

2. 数据输入

数据输入就是指变量如何获得数据，即获得数据的途径。变量获得数据的途径有：从程序中获得，从键盘上获得，从文件里获得。

(1) 从程序中获得。即把要处理的数据通过指令传给变量。在 C 语言中，是通过给变量赋初值的形式来实现的。如：

```
int x=12;      /* 在定义变量 x 为整型数据的时候，给 x 赋值 12 */
```

(2) 从键盘上获得。即在编程时不考虑变量的具体值，在程序运行时，再从键盘上给变量赋值。用于临时数据或即时数据的输入。

(3) 从文件里获得。即把处理的数据事先存放在文件里，在编程时通过对文件的操作把数据传给变量。文件是指存放在外存上的信息集合，用于需要重复处理和长久存放的数据。

3. 数据处理

数据处理是指由已知数据求未知数据的过程。重点是考虑算法，即处理语句和控制结构。常用的算法有：数据的组织、筛选与排序，递推与递归，链表与文件的处理。

4. 数据输出

数据输出是指把处理的结果显示、打印或存储到文件里。

(1) 输出到显示器

输出到显示器是指把处理的结果按照一定的格式输出到显示器上。重点在于控制输出的格式：每个数据的输出宽度，一行输出几个数据，输出多少行。这也是初学编程的主要输出方式，需要重点掌握。

(2) 输出到打印机

输出到打印机是指把处理的结果按照一定的格式输出到打印机上。重点也在于控制输出的格式：每个数据的输出宽度，一行输出几个数据，输出多少行。

(3) 输出到文件

输出到文件是指把处理的结果按照一定的格式输出到外存的文件里。重点在于文件的类型，即是文本文件，还是二进制文件。

1.3 算法的描述工具

1.3.1 采用自然语言描述

采用自然语言对算法的描述称为伪代码。所谓伪代码是指用自然语言和指令语句（不求绝对正确的语句）结合起来描述算法的一种方式。

这种方式比画流程图省时省力，转换为程序容易，但是清晰度层次性不如流程图。伪代码描述算法没有统一规定，写出来只要自己或别人能看懂就行。如已知圆的半径求圆的面积，算法可以描述为：

- (1) 输入半径 r ；
- (2) 计算面积： $s=3.14*r^2$ ；
- (3) 输入面积： s_0 。

1.3.2 采用图示方法描述

图示的描述方法通常有两种：传统流程图和 NS 流程图。

1. 传统流程图

传统流程图是描述算法的一种常用工具，是由如图 1-7 中所示的几种基本框和方向线组成的。



图 1-7 传统流程图基本框

由这些框和方向线组成的流程图来表示算法，允许任意转向，形象直观，简单方便。但是，这种流程图描述复杂的算法时，所占篇幅较多，勾画费时费力，且描述复杂问题时不易阅读。

2. N-S 流程图

N-S 流程图是由美国学者 I.Nassi 和 B.Shneiderman 于 1973 年提出来用于描述算法的一种