

高等学校公共课计算机规划教材

C语言 程序设计教程

张敏霞 孙丽凤 主编

<http://www.phei.com.cn>



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

高等学校计算机专业系列教材

C语言 程序设计教程

张德君 张朝晖 主编

清华大学出版社

TP312

2275

2007

高等学校公共课计算机规划教材

C 语言程序设计教程

张敏霞 孙丽凤 主 编

王秀鸾 副主编

罗 容 迟春梅 编

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

本书分为基础篇、提高篇和实验篇。基础篇的主要内容包括程序设计和 C 语言基础知识, 以及顺序、选择和循环结构程序设计, 使读者初步建立起利用 C 语言进行简单程序设计的思想, 学会进行简单的程序设计; 提高篇的主要内容包括函数、编译预处理、数组等构造型数据类型、指针类型及对文件的操作, 使读者学习并体会 C 语言模块化的编程思想及对数组、指针类型的应用, 学会使用构造型数据类型和指针类型处理问题, 学会对文件进行操作; 实验篇共设计了 10 个实验。

本书在编写时兼顾了全国计算机等级考试的要求。书中例题丰富, 注重实用, 且均在 Turbo C 环境下调试通过。各章都配有丰富的习题。本书程序源代码、配套课后习题指导和教学用多媒体电子课件, 请通过华信教育资源网免费索取。

本书可作为高等学校本科、高职高专软件专业及相关专业程序设计的入门教材, 也可作为全国计算机等级考试的辅导教材, 还可供广大程序设计初学者自学使用。

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

C 语言程序设计教程 / 张敏霞, 孙丽凤主编. —北京: 电子工业出版社, 2007.3

(高等学校公共课计算机规划教材)

ISBN 978-7-121-03920-1

I. C… II. ①张… ②孙… III. C 语言—程序设计—高等学校—教材 IV. TP312

中国版本图书馆 CIP 数据核字 (2007) 第 025170 号

责任编辑: 王羽佳

印 刷: 北京季蜂印刷有限公司

装 订: 三河市鹏成印业有限公司

出版发行: 电子工业出版社

北京市海淀区万寿路 173 信箱 邮编: 100036

开 本: 787×1092 1/16 印张: 22.25 字数: 569.6 千字

印 次: 2007 年 3 月第 1 次印刷

印 数: 5000 册 定价: 29.80 元

凡所购买电子工业出版社图书有缺损问题, 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) 68279077; 邮购电话: (010) 88254888。

质量投诉请发邮件至 zltz@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线: (010) 88258888。

前 言

计算机程序设计基础是高等学校各专业开设的一门必修的计算机基础课程,课程的重点在于培养学生的程序设计思想和程序设计能力,以适应当今社会对人才的需求。C语言由于其自身简洁、紧凑和灵活的特点,以及具备其他高级语言所不具备的低级语言的特性,而使得它成为一种在计算机软件设计和计算机程序设计教学中备受欢迎的程序设计语言。

本书是作者多年教学经验和应用C语言体会的结晶,根据教育部高教司非计算机专业计算机教学指导分委员会提出的高等学校计算机基础课程教学基本要求,在广泛参考有关资料的基础上编写而成。

内容分为基础篇、提高篇和实验篇。

基础篇主要包括程序设计和C语言基础知识,以及顺序、选择和循环结构程序设计,使读者初步建立起利用C语言进行简单程序设计的思想,学会进行简单程序设计。

提高篇主要包括函数、编译预处理、数组等构造型数据类型、指针类型及对文件的操作,使读者理解并体会C语言模块化的编程思想及对数组、指针类型的应用,学会使用构造型数据类型和指针类型处理问题,学会对文件进行操作。

实验篇共设计了10个实验,采用循序渐进的方式引导读者掌握C语言程序设计的特点,详细的上机实践练习,便于读者深入理解语法和培养程序设计能力。在实验篇的最后还设计了若干综合性实验项目以开拓读者的思路,激发读者的学习兴趣。

同时本书在编写时兼顾了全国计算机等级考试的要求。

本书在结构组织上合乎学习逻辑,内容循序渐进,每个知识点的介绍都以引起读者的学习热情和兴趣为出发点,以提高读者的程序设计思想和能力为目标,既注重理论知识,又突出实用性。书中例题丰富,注重实用,且均在Turbo C环境下调试通过。各章均配有丰富的习题。

本书可作为高等学校本科、高职高专软件专业及相关专业程序设计的入门教材,也可作为全国计算机等级考试的辅导教材,还可供广大程序设计初学者自学使用。

本书具有以下特点:

- ① 突出算法理解,重视实际操作。
- ② 加强对学生程序设计思想和实际编程能力的培养,以适应信息社会对人才的需求。
- ③ 提供多样式的学习环境。本书同时提供丰富的教学资源,包括课程学习网站(<http://jx.qtech.edu.cn/dxjsj>)、程序源代码、配套教学用多媒体电子课件及习题指导与参考答案。(免费索取程序源代码、配套教学用多媒体电子课件及习题指导与参考答案,请登录华信教育资源网<http://www.huaxin.edu.cn>或<http://www.hxedu.com.cn>。)
- ④ 注重可读性。本书的编写小组由具有丰富的教学经验,多年来一直从事计算机基础教育的一线资深教师组成,教材内容组织合理,语言使用规范,符合教学规律。

全书由张敏霞和孙丽凤两位老师统稿，其中第 2、4、5、8、9 章由张敏霞老师编写，第 1、3、6、7、10、11、12 章由孙丽凤老师编写，实验篇由王秀鸾老师编写，电子课件由罗容老师和迟春梅老师制作。本书的编写得到了众位专家和领导的关心和支持，在此一并表示诚挚的感谢。

由于作者水平有限，书中存在误漏之处在所难免，敬请广大读者批评指正。

作 者

2007 年 1 月

目 录

基 础 篇

第1章 程序设计及C语言概况	2
1.1 程序设计的基本概念	2
1.1.1 程序和程序设计语言	2
1.1.2 程序设计	3
1.2 算法	4
1.2.1 算法及算法的特性	4
1.2.2 算法的描述工具	4
1.3 结构化程序设计方法	5
1.4 C语言的初步知识	7
1.4.1 C语言的起源与发展	7
1.4.2 C语言的特点	8
1.4.3 C语言程序的构成	9
1.4.4 C语言程序的开发过程	12
本章小结	13
习题1	13
第2章 C语言基础	15
2.1 数据类型	15
2.2 标识符、常量和变量	16
2.2.1 字符集	16
2.2.2 标识符	16
2.2.3 常量	17
2.2.4 变量	18
2.3 基本类型数据	18
2.3.1 整型数据	18
2.3.2 实型数据	20
2.3.3 字符型数据	21
2.3.4 字符串常量	25
2.3.5 变量赋初值	26
2.4 运算符及表达式	26
2.4.1 算术运算符	27
2.4.2 赋值运算符	30
2.4.3 自增、自减运算符	34

2.4.4 逗号运算符	36
2.4.5 sizeof 运算符	37
本章小结	37
习题 2	38
第 3 章 顺序结构	41
3.1 C 语言程序的语句	41
3.1.1 说明语句	41
3.1.2 执行语句	41
3.2 数据的输入和输出	43
3.2.1 字符输入函数 getchar()	43
3.2.2 字符输出函数 putchar()	43
3.2.3 格式输出函数 printf()	45
3.2.4 格式输入函数 scanf()	49
3.3 程序举例	52
本章小结	54
习题 3	54
第 4 章 选择结构	62
4.1 关系运算	62
4.1.1 关系运算符	62
4.1.2 关系表达式	63
4.2 逻辑运算	64
4.2.1 逻辑运算符	64
4.2.2 逻辑运算的值	65
4.2.3 逻辑表达式	65
4.3 if 语句	67
4.3.1 if 语句的 3 种形式	67
4.3.2 if 语句的嵌套	73
4.3.3 条件运算符和条件表达式	74
4.4 switch 语句	74
4.5 程序举例	77
本章小结	80
习题 4	81
第 5 章 循环结构	86
5.1 while 语句	86
5.2 do-while 语句	89
5.3 for 语句	91
5.4 转移语句	95
5.4.1 goto 语句	96
5.4.2 break 语句	97

5.4.3 continue 语句	97
5.5 程序举例	98
本章小结	103
习题 5	104

提 高 篇

第 6 章 函数	114
6.1 标准函数	114
6.2 函数的定义和返回值	115
6.2.1 函数的定义	115
6.2.2 函数的返回值	116
6.3 函数的调用和函数的声明	117
6.3.1 函数的调用	117
6.3.2 函数的声明	119
6.4 函数的参数	120
6.5 函数的嵌套调用和递归调用	121
6.5.1 函数的嵌套调用	121
6.5.2 函数的递归调用	123
6.6 变量的存储类别	126
6.6.1 局部变量及其存储类别	127
6.6.2 全局变量及其存储类别	130
6.6.3 函数的作用域和存储类别	135
6.7 程序举例	135
本章小结	139
习题 6	140
第 7 章 编译预处理	147
7.1 宏定义	147
7.1.1 不带参数的宏定义	147
7.1.2 带参数的宏定义	149
7.2 文件包含	152
7.3 条件编译	153
本章小结	153
习题 7	154
第 8 章 数组	157
8.1 一维数组	157
8.1.1 一维数组的定义	157
8.1.2 一维数组的初始化	158
8.1.3 一维数组元素的引用	159
8.1.4 一维数组的应用	160

8.2 二维数组	165
8.2.1 二维数组的定义	165
8.2.2 二维数组的初始化	166
8.2.3 二维数组元素的引用	166
8.2.4 二维数组应用举例	167
8.3 字符数组	171
8.3.1 字符数组的定义	171
8.3.2 字符串与字符数组	172
8.3.3 字符数组的初始化	172
8.3.4 字符数组的输入/输出	173
8.3.5 字符串(字符数组)处理函数	175
8.3.6 字符数组的应用	179
8.4 数组元素作为函数实参	181
本章小结	184
习题 8	184
第 9 章 指针	191
9.1 指针变量	191
9.1.1 变量的指针和指针变量	191
9.1.2 指针变量的定义和初始化	192
9.1.3 指针变量的引用	193
9.1.4 指针变量作为函数参数	197
9.2 数组指针变量	198
9.2.1 数组指针变量的定义和引用	199
9.2.2 数组名和数组指针变量作为函数参数	203
9.2.3 二维数组的指针	207
9.2.4 指针与字符串	211
9.3 指针与函数	217
9.3.1 指向函数指针变量的定义和赋值	217
9.3.2 指向函数的指针变量作为函数参数	219
9.3.3 指针函数	220
9.4 指针数组和二级指针变量	223
9.4.1 指针数组	223
9.4.2 main()函数的形参	225
9.4.3 二级指针变量	226
本章小结	227
习题 9	229
第 10 章 结构体与共用体	240
10.1 自定义类型标识符	240
10.2 结构体的定义与引用	241

10.2.1 结构体类型的定义	242
10.2.2 结构体类型变量、数组和指针变量的定义	243
10.2.3 结构体类型变量、数组和指针变量的初始化	246
10.2.4 结构体类型变量、数组和指针变量的引用	247
10.2.5 函数之间结构体类型数据的传递	252
10.2.6 用指针处理链表	254
10.3 共用体的定义与引用	260
10.3.1 共用体类型的定义	261
10.3.2 共用体类型变量的定义	261
10.3.3 共用体变量的引用	262
10.4 枚举类型	264
本章小结	266
习题 10	267
第 11 章 位运算	274
11.1 位运算符	274
11.2 位运算符的功能	274
11.2.1 按位与运算 (&)	274
11.2.2 按位或运算 ()	275
11.2.3 按位异或运算 (^)	275
11.2.4 按位取反运算 (~)	275
11.2.5 按位左移 (<<)	276
11.2.6 按位右移 (>>)	276
11.2.7 复合位赋值运算符	276
本章小结	277
习题 11	277
第 12 章 文件	279
12.1 C 语言文件	279
12.2 文件指针	280
12.3 打开、关闭和检测文件	281
12.3.1 打开文件 (fopen() 函数)	281
12.3.2 关闭文件 (fclose() 函数)	282
12.3.3 检测文件	283
12.4 文件的输入/输出	284
12.4.1 fputc() 函数和 fgetc() 函数	284
12.4.2 fgets() 函数和 fputs() 函数	286
12.4.3 fprintf() 函数和 fscanf() 函数	287
12.4.4 fread() 函数和 fwrite() 函数	289
12.5 文件的定位	291
12.5.1 rewind() 函数	291

12.5.2 fseek()函数和随机读/写	292
12.5.3 ftell()函数	293
本章小结	293
习题 12	293

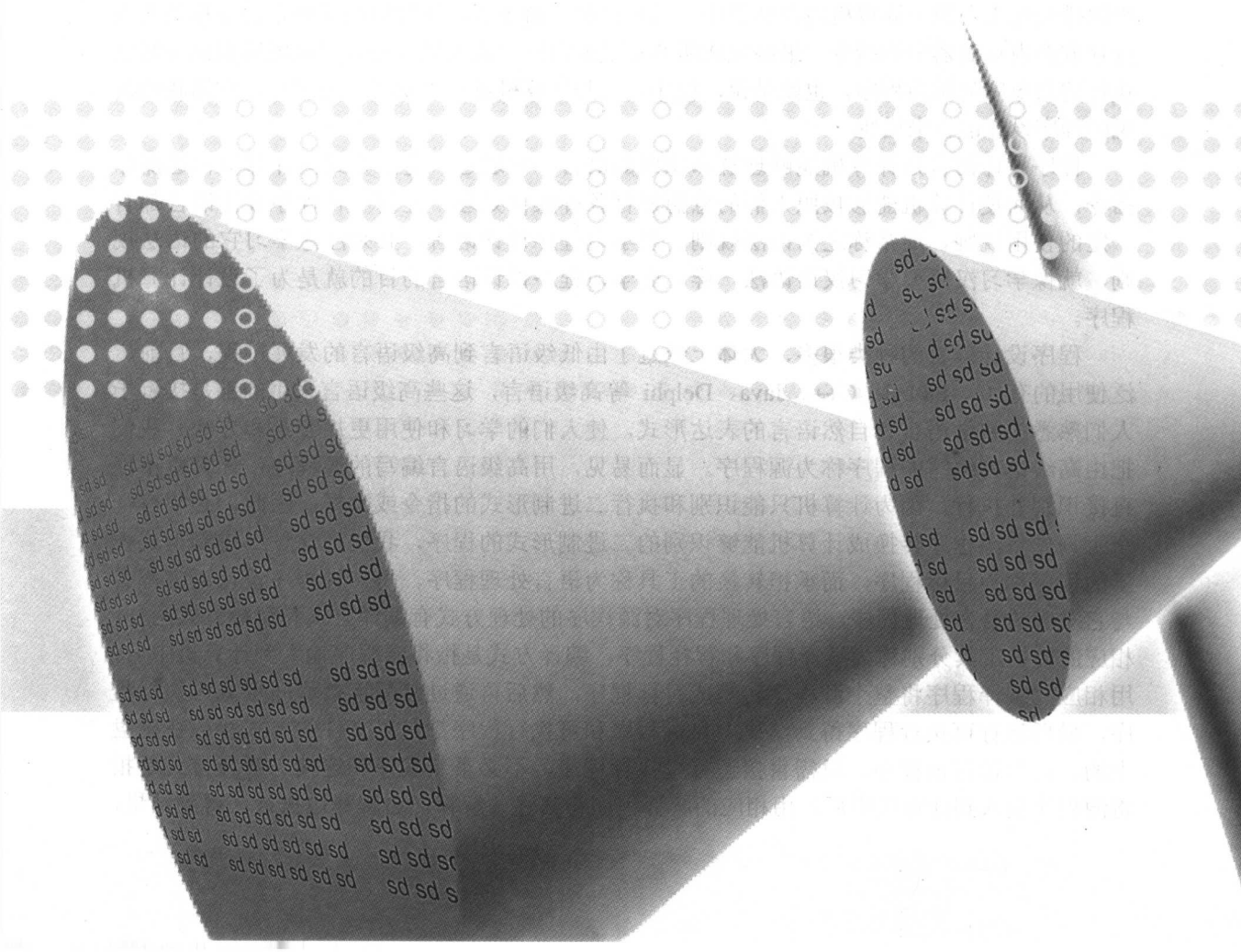
实 验 篇

实验 1 Turbo C2.0 软件开发环境	298
实验 2 顺序结构程序设计	301
实验 3 选择结构程序设计	304
实验 4 循环结构程序设计	306
实验 5 函数	309
实验 6 数组	311
实验 7 指针	313
实验 8 结构体和共用体	315
实验 9 文件	316
实验 10 综合编程	317

附 录

附录 A C 语言中的关键字	330
附录 B 运算符的优先级和结合性	331
附录 C 标准函数	332
附录 D ASCII 字符编码表	335
附录 E 程序调试中常见错误信息一览	336

基础篇



第1章 程序设计及C语言概况

本章主要介绍程序设计的基本概念、算法描述及结构化程序设计方法、C语言的起源与发展、C语言的特点、简单的C语言程序和Turbo C环境下C语言程序的开发过程。通过本章的学习，读者可对程序设计的基本概念和C语言的概貌有一个大致的了解。

1.1 程序设计的基本概念

1.1.1 程序和程序设计语言

什么是程序？怎样设计程序？这往往是计算机语言初学者首先遇到的问题。有人以为计算机是“万能”的，只要把任务告诉计算机，计算机就会自动完成一切，并给出正确结果。这其实是一种误解，要让计算机按照人的意志来完成某项任务，首先要制订该项任务的解决方案，再将其分解成计算机能够识别并可以执行的基本操作指令，把这些指令按一定的规则组织排列起来存放于计算机内存储器中，当给出执行命令后，计算机按照规定的流程依次执行存放在内存储器中的指令，最终完成所要实现的目标。人们把这种计算机能够识别并可以执行的指令序列称为程序。也就是说，程序是人与计算机进行“交流”的工具，它用我们常说的程序设计语言来描述。

程序设计语言是计算机能够理解和识别的语言。它通过一定的方式向计算机传送操作指令，从而使计算机能够按照人们的意愿进行各种操作处理。任何一种程序设计语言都有一定的使用规则，通常称之为语法规则。要学习程序设计语言，必须注意学习它的语法规则，就像学习汉语要学习汉语语法一样。而学习程序设计语言的目的就是为了设计计算机程序。

程序设计语言的种类很多，大体上经过了由低级语言到高级语言的发展过程，目前广泛使用的有C、Pascal、C++、Java、Delphi等高级语言，这些高级语言采用的都是接近于人们熟悉的数学语言和自然语言的表达形式，使人们的学习和使用更加容易和方便。我们把由高级语言编写的程序称为源程序。显而易见，用高级语言编写的源程序，计算机不能直接识别并执行，因为计算机只能识别和执行二进制形式的指令或数据，因此，必须有一个工具先将源程序转换成计算机能够识别的二进制形式的程序，我们把这种二进制形式表示的程序称为目标程序，而承担转换的工具称为语言处理程序。每一种程序设计语言都有与它对应的语言处理程序，语言处理程序对源程序的处理方式有编译方式和解释方式两种，相应的转换工具分别称为编译程序和解释程序。编译方式是指将源程序输入到计算机中后，用相应的编译程序将整个源程序转换成目标程序，然后再通过装配连接程序形成可执行程序，最后运行可执行程序得到结果。目标程序和可执行程序都是以文件的方式存放在磁盘上的，再次运行该程序，只需直接运行可执行程序，不必重新编译和连接。解释方式是指将源程序输入到计算机中后，用相应的解释程序将其逐条解释，边解释边执行，得到结果，

而不保存解释后的机器代码，下次运行该程序时还要重新解释执行。采用编译方式，程序的运行速度快，效率高。因此，在目前常用的高级语言中除 BASIC 语言采用解释方式外，大部分高级语言都采用编译方式。

1.1.2 程序设计

对于初学者来说，往往把程序设计简单地理解为只是编写一个程序，这是不全面的。程序设计是指利用计算机解决问题的全过程，它包含多方面的内容，而编写程序只是其中的一部分。利用计算机解决实际问题，通常先要对问题的性质与要求进行深入分析并确定求解问题的数学模型或方法，然后考虑数据的组织方式和算法，并用某一种程序设计语言编写程序，最后调试程序，使之运行后能产生预期的结果，这个过程称为程序设计。程序设计的基本目标是实现算法和对初始数据进行处理，从而完成对问题的求解。

有些初学者，在没有把所要解决的问题分析清楚之前就急于编写程序，结果编程思路紊乱，很难得到预期的效果。因此，为了用计算机解决一个实际问题，作为设计人员，从拿到任务到得出正确结果，往往要经过以下6个设计阶段。

① 分析问题。即分析问题需求，也就是弄清楚该问题有哪些已知数据，程序运行需要什么输入什么数据，需要输出什么结果，需要进行哪些处理，等等。

② 确定处理方案。如果是数学问题，就要根据该问题的数学解法，考虑所用到的数学公式或相关函数；如果是工程问题，就要先建立该问题的数学模型，把工程问题转化成数学问题，以使用计算机解决。对同一个问题可以用不同的方案来处理，不同的方案决定了不同的处理步骤，效率也有所不同。

③ 确定操作步骤。根据选定的处理方案，具体列出让计算机如何进行操作的步骤。这种规定的操作步骤称为算法，而这些操作步骤之间的执行顺序就是控制结构。通常使用流程图来描述算法，把算法思想表达清楚，比较简单的问题可直接进入编写程序阶段。

④ 根据操作步骤编写源程序。用计算机语言编写的操作步骤就是计算机程序。

⑤ 运行调试程序。将计算机程序输入计算机中，经过编译、连接和运行，如果程序是正确的，应该能得到预期的结果。如果得不到正确的结果，应检查程序是否有错误，改正后再试运行，直到得出正确的结果为止。

⑥ 整理输出结果，写出相关文档。

图 1.1 所示为程序设计的一般过程。

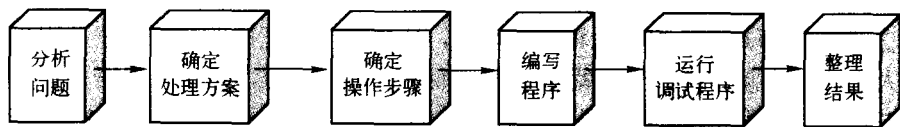


图 1.1 程序设计的一般过程

图 1.1 中，前两个步骤类似于人们解决问题的一般过程，即分析问题，然后确定一种处理方案。后 4 步则是程序设计的环节，其中最关键的是第③步“确定操作步骤”，或称算法设计。只要算法是正确的，编写程序就不会太困难。对于一个具体问题，一个编程者应该具备设计算法和正确使用已有算法的能力。

1.2 算 法

1.2.1 算法及算法的特性

算法是对具体问题求解步骤的一种描述。做任何事情都必须事先想好进行的步骤，然后按步骤进行，才能避免产生错乱。对同一个问题，可以有不同的解题方法和步骤。有的方法需要的步骤很少，而有些方法需要的步骤则较多。一般来说，希望采用过程简单明了和思路清晰正确的方法。因此，为了有效地解题，一个算法应具有下列 5 个特性。

① 有穷性。一个算法必须在执行有限个操作步骤之后结束，且每一步都可以在有限时间内完成。

② 确定性。算法中的每一条指令必须有确切的含义，不会产生二义性。

③ 可行性。算法中描述的操作在计算机上都是可以实现的。

④ 输入。一个算法应有零个或多个输入。

⑤ 输出。一个算法应有 1 个或多个输出。

1.2.2 算法的描述工具

为了使算法表达得更清晰，更容易实现算法的编写，在程序设计时通常使用专门的算法表达工具对算法进行描述。对于复杂的问题，可以先用算法表达工具对算法进行描述，再进行编程，算法的最终实现应该是计算机程序。算法的评价标准涉及很多方面，但正确性和清晰易懂性永远是一个好算法的基本条件。

计算机算法的表达工具通常有以下 4 种。

1. 用自然语言表示算法

用自然语言表示算法就是把算法的各个步骤用人们所熟悉的自然语言依次表示出来。但要注意，用自然语言所表示的每一个操作步骤必须是计算机能够实现的。

【例 1.1】 求两个整数 m 与 n 的和，用自然语言求解该问题的步骤如下。

步骤 1：输入整数 n 和 m 。

步骤 2：求和 $\text{sum}=m+n$ 。

步骤 3：输出两数之和 sum 。

以上 3 个步骤都是在计算机上可以实现的，所以是一个正确的算法描述。用自然语言表示算法，人们比较容易理解，但书写较烦琐，而且在某些场合，由于自然语言含义的不确切性，容易引起歧义，造成误解；另外对比较复杂的问题，用自然语言又难以表达准确。因此，较少采用自然语言表示复杂算法。

2. 用流程图表示算法

用流程图表示算法就是用一些大家共识的专用图形符号和带有箭头的流程线来表示算法。用图形符号表示算法必须要有一组统一规定、含义确定的专用符号。表 1.1 所示为传统流程图所用的基本符号。图 1.2 所示为例 1.1 中求两个整数 m 与 n 之和的传统流程图算法。

可见，用流程图来表示算法，直观形象，绘制简单方便，流程清晰，各种操作一目了然，而且不会产生“二义性”。缺点是，占用面积大，由于使用流程线指出各框的执行顺序，且对流程线的方向没有任何限制，可以随意转向，往往使人弄不清楚流程的思路。

针对上述问题，1973 年由美国计算机科学家 I.Nassi 和 B.Shneiderman 提出了结构化程序设计流程图，又称 N-S 流程图。N-S 流程图保留了传统流程图可以形象直观地表示算法的优点，去掉了流程线，即不允许流程任意转移，只能从上到下顺序进行，算法的每一步都用一个矩形框来描述，把这些矩形框按执行的次序连接起来就是一个完整的算法。这种流程图规定了几种基本结构作为构造算法的基本单元，将在下一节中结合 3 种基本结构来介绍。

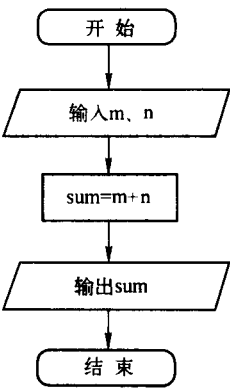


图 1.2 例 1.1 算法

表 1.1 传统流程图所用的基本符号表

图 形 符 号	符 号 名 称	说 明
	起始、终止框	表示算法的开始或结束
	输入、输出框	框中标明输入、输出的内容
	处理框	框中标明进行什么处理
	判断框	框中标明判定条件，并在框外标明判定后的两种结果的流向
	流程线	表示从某一框到另一框的流向
	连接点	表示算法流向出口或入口连接点

3. 用伪代码表示算法

所谓伪代码是指用介于自然语言和程序设计语言之间的一种代码来描述算法。它的表示形式比较自由灵活，而且由于与程序设计语言比较接近，因此可以比较容易地转换成计算机程序。伪代码无统一的语法，只要自己或别人能看懂就行。

4. 用程序设计语言表示算法

用自然语言和用图形符号所表示的算法有一个共同的缺点就是计算机都不能识别和执行。只有用计算机能理解和执行的程序设计语言把算法表示出来，然后把程序输入到计算机并执行，计算机才能按照预定的算法去解决问题。

1.3 结构化程序设计方法

结构化程序设计是指，为使程序具有一个合理的结构以保证程序正确性而规定的一套如何进行程序设计的原则。结构化程序设计的原则是：采用自顶向下、逐步求精的方法；程序结构模块化，每个模块只有一个入口和一个出口；使用 3 种基本控制结构描述程序流程。其中，模块化是结构化程序设计的重要原则。所谓模块化就是把一个大型的程序按照功能分解为若干相对独立的、较小的子程序（即模块），并把这些模块按层次关系进行组织。按照结构化程序设计的原则，一个程序只能由顺序结构、选择结构和循环结构这 3 种