

程序设计方法与计算机  
语言教学新体系教程

# 程序设计方法 与 PASCAL 语言

彭 澎 著

清华大学出版社

<http://www.tup.tsinghua.edu.cn>



(京)新登字 158 号

### 内容简介

本书是作者多年教学的总结,通过分析、加工、提炼而成。针对程序设计语言的特点,针对人的思维特点,以面向程序设计为中心,以程序设计方法为主线来介绍 PASCAL 语言的基于结构化程序设计思想的教科书。作者力求使读者通过学习本书不仅掌握 PASCAL 语言,更主要的是使读者建立和掌握结构化程序设计的思想和方法,为进一步学习其他计算机软件知识打下良好的基础。全书打破了传统的以语言为中心的以语法、词法和语法规则为主线的教学体系,使读者能在轻松自如状态下学好较为枯燥的计算机语言和掌握程序设计方法。

本书是一本以程序设计方法为主线来介绍 PASCAL 语言的教科书,全书有大量的附有详细程序设计方法分析的例题,并且对主要例题都有详细的一题多解的分析、说明。所以,本书适用于所有希望学习程序设计的读者,还特别适合教师作教学参考,自学考试学生自学使用。

**版权所有,翻印必究。**

**本书封面贴有清华大学出版社激光防伪标签,无标签者不得销售。**

### 图书在版编目(CIP)数据

程序设计方法与 PASCAL 语言/彭澎著. —北京:清华大学出版社,1998.10  
ISBN 7-302-03150-9

I. 程… I. 彭… III. PASCAL 语言-程序设计 IV. TP312

中国版本图书馆 CIP 数据核字(98)第 28226 号

出版者:清华大学出版社(北京清华大学校内,邮编:100084)

<http://www.tup.tsinghua.edu.cn>

责任编辑:徐培忠

印刷者:北京丰华印刷厂

发行者:新华书店总店北京科技发行所

开本:787×1092 1/16 印张:15 字数:351 千字

版次:1998年10月第1版 1998年10月第1次印刷

书号:ISBN 7-302-03150-9/TP·1680

印数:0001~5000

定价:24.00 元

## 前 言

程序设计方法一直是计算机科学家研究的重要课题,结构化程序设计方法的诞生,在程序设计方面具有划时代的意义。其典型代表 PASCAL 语言,是根据荷兰计算机科学家、TURING 奖获得者 E. W. Dijkstra 提出的“结构程序设计”思想,由瑞士苏黎世联邦工业大学沃斯教授于 1968 年提出的。它最早发表在 1971 年的《ETH》杂志上,是为了使程序具有合理结构,以保证和验证其正确性而规定的一套进行程序设计的方法。

PASCAL 语言是 70 年代以来最有影响和最重要的一种程序设计语言,它是从 ALGOL60 语言发展而来。它注重于语言的可靠性、易于验证性、概念的清晰性和实现的简化性。PASCAL 语言效能很强,适合于教学、管理、编写各种系统软件和进行科学计算,是目前世界上流行最广泛的一种程序设计语言。

PASCAL 语言既是一种结构化程序设计语言,又是一种系统程序设计语言,同时还是一种自编译语言。这种语言具有丰富的数据类型、简明的通用语句、清晰的程序结构;而且书写格式自由,编译紧凑,风格优美,受到人们的普遍欢迎。世界各国都用它进行程序设计教学,效果良好。人们通过学习 PASCAL 语言能较好地掌握结构化设计思想,有效地提高逻辑思维能力,它对培养高级软件人员起着积极的推进作用。

目前,采用结构化程序设计方法培养计算机软件设计人员,用结构化程序设计方法进行教学,形成以结构化程序设计方法为主线的培养软件人员体系,是当务之急。长期以来我们形成了以面向语言为中心的以语言的语法、词法及语言结构为核心的教学体系,而这个体系有许多缺陷,其最大的问题是学了语言仍不会设计程序,且感到学习枯燥乏味。故有人提出改革此体系。目前有许多介绍某种程序设计语言的书籍都以“程序设计”命名,以突出程序设计的主导思想。然而,这些书籍往往还是以介绍某种语言的词法、语法和语言结构为核心,没有突破旧的体系。

程序设计语言概括地说包括语言和设计两方面的内容。学习程序设计语言的关键、难点不在于语言的语法、词法、语言结构方面。因为,从程序设计的角度上看,计算机程序设计语言的词法、语法和语言结构可以说是属于最简单的一类,掌握它们相对来说是比较容易的。而学习程序设计语言的关键、难点是在设计上。这是因为:程序设计对人的逻辑思维和形象思维以及对问题的描述和表达的能力要求高并且严格。而人们通常所接受的训练主要是各个单项的训练。人们缺乏的就是程序设计所需要的综合能力。所以,程序设计的重点应该放在培养学习者逻辑思维、形象思维、分析和表达、描述问题和解决问题的综合能力上。学习程序设计语言的目的也在于设计,这一点同人们学习文字语言、会话语言的目的是一样的。另外,对于不同的程序设计语言,它们在词法、语法及语言结构上都有许多相似之处,用不同的程序设计语言编写程序的设计思想和方法具有共性。从这个角度上讲,学习程序设计语言的目的也不应该是以掌握某种语言的具体语法、词法和语言结构为重点。

以面向语言为中心的学习与教学体系的缺陷还表现在,讲授内容的主次不明确,对设计

方面的内容介绍较少,即使是介绍设计的内容也是从语言的角度来介绍,而不是从设计的角度来介绍。这就造成学习者学习缺乏主导思想,虽然某种具体语言的知识学到了,但不能把所学到的语言知识贯通起来,不能利用所学到的语言知识进行设计,这是与学习程序设计语言的目的相悖的。所以也难于入门,往往半途而废。对自学的学习者来说则更为困难。

本书是作者多年教学的总结,通过分析、加工、提炼而成。针对程序设计语言的特点,针对人的思维特点,以面向程序设计为中心、以程序设计方法为主线介绍 PASCAL 语言的基于结构化程序设计思想的教科书。作者力求使读者通过学习本书不仅掌握 PASCAL 语言,更主要的是使读者建立和掌握结构化程序设计的思想和方法,为进一步学习其他计算机软件知识打下良好的基础。全书打破了传统的以面向语言为中心的教学体系,使读者能在不知不觉,轻松的状态下学好和掌握程序设计。

本书是一本教材,全书有大量的附有详细程序设计方法分析的例题,并且对主要例题都有详细的一题多解的分析、说明。所以,本书适用于所有希望学习程序设计的读者,本书还特别适合教师作教学参考和自学考试学生自学程序设计使用。

在本书的写作出版过程中得到了清华大学侯炳辉教授、人民大学魏靖宇、陈禹教授、首都经济贸易大学盛定宇教授、原电子工业部高级工程师、北京软件协会副秘书长沈林兴先生的大力支持和帮助,在此对这些同志表示衷心的感谢。首都经济贸易大学信息管理系的王元同学和郭屹同学对书中列出的全部例题进行了验证,并提出了许多宝贵意见,同时两位同学还参与了本书出版过程中的排版和校对等工作,在此对这些同志表示衷心的感谢。

彭澎

1998 夏于首都经济贸易大学

# 目 录

<b>第 1 章 语言与程序</b> .....	1
1.1 语言 .....	1
1.2 语言、思维与程序 .....	3
<b>第 2 章 PASCAL 语言基础</b> .....	9
2.1 计算机语言概述 .....	9
2.2 PASCAL 语言的语法表示方法 .....	11
2.3 PASCAL 语言符号系统 .....	11
2.4 PASCAL 语言的数据类型 .....	14
2.5 常量及标准数据类型 .....	16
2.6 表格管理与 PASCAL 语言程序结构 .....	24
2.7 PASCAL 语言的语言要素 .....	27
2.8 表达式 .....	29
<b>第 3 章 PASCAL 简单语句及简单程序</b> .....	31
3.1 赋值与赋值语句 .....	31
3.2 读语句 .....	34
3.3 写语句 .....	37
3.4 简单程序 .....	39
<b>第 4 章 程序设计基础</b> .....	43
4.1 算法 .....	43
4.2 程序设计的一般方法 .....	47
4.3 结构化程序设计 .....	49
4.4 算法的描述 .....	50
4.5 实例分析 .....	52
<b>第 5 章 基本控制结构</b> .....	64
5.1 顺序结构 .....	64
5.2 选择结构 .....	67
5.3 循环结构 .....	77
5.4 算法结构与程序语言 .....	80
5.5 程序实例 .....	84
5.6 GOTO 结构语句 .....	87
<b>第 6 章 数据组织</b> .....	89
6.1 基本概念 .....	89
6.2 数据组织 .....	90
6.3 数据的逻辑结构 .....	91

<b>第 7 章 用户定义的简单数据类型</b> .....	95
7.1 类型定义 .....	95
7.2 枚举类型 .....	95
7.3 子界类型 .....	98
7.4 类型定义的补充说明 .....	99
<b>第 8 章 构造类型</b> .....	101
8.1 集合 .....	101
8.2 数组 .....	105
8.3 记录类型 .....	115
8.4 字符串 .....	123
<b>第 9 章 进一步的结构化程序设计方法</b> .....	130
9.1 系统与结构化 .....	130
9.2 过程 .....	132
9.3 变量与参数 .....	137
9.4 函数 .....	141
9.5 嵌套与递归 .....	143
9.6 标准过程与标准函数 .....	147
<b>第 10 章 数据文件</b> .....	149
10.1 文件的概念 .....	149
10.2 文件的说明 .....	150
10.3 用于用户定义文件的操作 .....	151
10.4 建立文件 .....	152
10.5 读文件 .....	154
10.6 文本文件 .....	155
10.7 顺序文件的更新 .....	157
10.8 应用举例 .....	157
<b>第 11 章 指针</b> .....	163
11.1 指针类型 .....	163
11.2 指针的应用 .....	173
<b>第 12 章 单元与应用</b> .....	188
12.1 单元 .....	188
12.2 单元的使用 .....	190
<b>第 13 章 实例分析</b> .....	191
13.1 分析与研究 .....	191
13.2 系统实施 .....	192
<b>附录 1 TURBO PASCAL 操作和编辑</b> .....	229
<b>附录 2 TURBO PASCAL 与标准 PASCAL 的主要区别</b> .....	232

# 第 1 章 语言与程序

随着电子计算机技术的普及与广泛的使用,随着人们掌握计算机科学与技术的平均水平的提高,计算机的使用者,已不满足于仅仅是对计算机进行简单的操作和使用,开始向利用计算机程序语言编程的方向发展。

程序设计简单的说,就是人们利用计算机能够识别和处理的语言编写“文章”,“文章”即所谓的程序,“编写”就是设计。程序是利用计算机某种语言规则,表达出要解决的问题的解决方法和步骤,需要解决的问题就是“任务”,任务是通过把程序交给计算机,计算机通过对程序的处理来完成的。由此可知,计算机语言是人与计算机进行交流的工具,计算机语言与人们现实生活中所用的文字语言、声音语言、哑语等本质上是一样的,其区别只是交流对象不同,语言结构、规则等不同。

计算机语言既然是一种“语言”,那么学习计算机语言就必须先掌握与语言有关的知识,并且把语言与计算机结合起来,真正了解了什么是计算机程序语言,在此基础上学习计算机语言知识。这是轻松、快速真正掌握计算机语言的非常有效的途径,这样可以避免出现学习计算机语言半途而废,学习计算机语言一知半解等现象。

初学计算机程序语言的人,在学习过程中会遇到许多难于理解的问题,通常解决的方法是通过“艰苦”的死记硬背加不完全理解的理解这一过程,逐步掌握计算机程序语言的。有很多人就是在无法通过这个过程的情况下终止学习的。出现这种情况的根本原因就是学习者没有从根本上弄清楚什么是语言,没有弄清楚什么是计算机程序语言,没有弄清楚语言的本质、特征等一些非常重要的概念。试想,如果在对现实生活中天天使用的语言的本质、结构及其基本特征都不清楚的情况下,怎么能顺利地学好、用好计算机语言呢!

## 1.1 语言

语言是社会发展的产物,它与社会、自然与思维有着密切的关系。掌握语言的一般知识是学习程序语言的基础。

### 1.1.1 语言的概念

语言是一套具有词法、语法规则的系统,不同的语言系统具有不同的具体内容及特点。例如:汉语、英语、法语、阿拉伯语及计算机程序语言(PASCAL 程序语言、C 程序语言、BASIC 语言)等,它们在词的构成、语法等方面的规则上都有自己的特点。

### 1.1.2 语言的特点

语言是人类相互间进行交流的工具。语言具有如下的特点:

(1) 社会性。社会性是指语言属于人们共有的交际工具,语言不属于某个人,或某个组

织。一个人只要掌握了某种语言,就能够在这种语言的环境下,与熟知此语言的对象进行交流,并在语言功能的范畴内实现使用语言者要达到的目的。如:掌握汉语的人之间使用汉语进行交流;掌握汉语的人与掌握英语的人之间进行交流,就需要选一种交流双方都能接受的语言,汉语、英语或汉英结合、手语等。但不论使用哪种语言,采用哪种方式(文字、声音、手势)都必须是交流双方认可和所识别的。人与计算机交流采用人为计算机设计的,被人和计算机共同接受的计算机语言进行,所以一个人只要掌握了某种计算机语言就能用这种语言和计算机讲话,专业术语就是编写程序。学习计算机语言的目的也在于此。

(2) 稳定性。稳定性是指用以表达语言的符号、词法、语法等是稳定、固定不变的,否则人类就无法掌握语言,其社会性也不存在了。没有稳定性,本书的出版也就无任何意义了。

(3) 语言交流的各个结构是有限的。所有语言系统的各个结构都是有限的,但是人们都可以用这些有限的结构编出无数的句子来。如:音乐语言。音乐的音符只有七个,音调也只有 A、B、C... 等几种,然而作曲家却能用它们写出无数支不同的曲子。计算机程序语言的结构与所有语言相比都是比较简单的,只有“顺序、选择、循环三种结构,语言的句型也不多,但人们用它却能编写出无数个能解决不同问题的程序。

### 1.1.3 语言系统

语言是一套具有语法、词法规则的系统,而这个系统是由什么构成的呢?是什么支撑着这个系统的呢?语言其实是一种符号系统。我本人认为:语言实际上是由一定量的符号,按一定规则排列起来构成词,这些所谓的词具有独立的意义;各种词按一定的规则组合起来,以达到扩充词意义的目的,这就是句子,组成句子的规则就是语法;句子之间再按一定规则排列、组合,以表现出比句子更广泛的意义,这就是语言系统。汉语如此,英语、法语、音乐语言、计算机程序语言,所有的语言都是如此。本书主要强调的是计算机语言。

### 1.1.4 符号

所谓符号就是被约定用来指代某种事物的标志。任何符号都具有两个特点:

- (1)符号和符号指代的事物之间没有必然的联系,它是人为约定的;
- (2)任何一种符号都必须具备形式和内容两个基本部分。

如汉语、英语、阿拉伯语各语言符号的形式是不一样的,各语言中的每个符号都有各自的内容。计算机语言系统中所包含的各种符号也都具备上述特点,它们也是本书将介绍的重要内容。

### 1.1.5 符号的表达

人为的符号是丰富多彩、多种多样的,如:视觉符号,包括文字、旗语、信号灯、手势等;听觉符号,包括乐曲、鼓声、号声、说话声等;触觉符号,包括:盲文...。计算机语言符号是多种符号为一体的综合性符号,处于不同阶段的计算机语言符号的表达、表现形式是不一样的。所以在学习程序语言过程中,学习者在学习程序语言的词法、语法规则的同时,需要了解程序语言的符号在计算机中的表达形式和计算机对语言中不同类型的符号(词)的管理方式,这是非常必要的。如果学习者只是学习程序语言外在的词法、语法等规则,而不了解程序语言的符号在计算机中的表达形式和计算机对语言中不同类型的符号(词)的管理方式,在学



习过程中就会遇到各种各样的难于理解的问题,编程也是知其然而不知其所以然,学习程序设计就会走许多弯路,甚至学习不下去,半途而废。

## 1.2 语言、思维与程序

学习程序设计需要思维,不仅需要逻辑思维,同时也需要形象思维。许多初学者认为程序设计只需要逻辑思维,这是非常错误的想法。

### 1.2.1 思维与语言

所谓思维,是人脑借助语言、表象或动作对客观现实的能动反映。思维分形象思维和逻辑思维两大类。形象思维主要依靠形象进行思维,根据形象在场与不在场进行的思维,形象思维又分直观动作思维和表象思维两种。逻辑思维主要是根据语言进行思维。

(1) 形象思维。又分为直观动作思维和表象思维。直观动作思维是指思维能直观思维对象,并通过思维者自身的动作去影响思维的对象。也就是说,这种思维的思维者与思维对象处于同一环境。现实生活中人们所作的每一件事情都离不开这种形式的思维。表象思维。它是指思维时通过想象对思维对象进行加工改造的思维。思维者与思维对象并不处于同一环境。

(2) 逻辑思维。逻辑思维是指运用概念、判断和推理的形式进行的思维。逻辑思维运用的是抽象理论,而不是具体的形象。逻辑思维是在语言的基础上进行的,人掌握了语言,也就掌握了思维的能力。因为人掌握的都是会话语言、文字语言,所以人们都具有以会话语言、文字语言进行思维的能力,而要利用计算机语言进行思维就必须在掌握会话语言、文字语言及语言概念、特征的基础上,把握人的自然语言与计算机语言的共性与个性,掌握计算机程序语言的思维方式,使人具有利用计算机程序语言思维、描述和解决问题的能力。

从上述对思维概念的介绍,可以把形象思维理解为对事物特殊性的分析、处理;把逻辑思维理解为对事物普遍性的一般规律的分析、处理。计算机程序语言,程序设计解决的中心问题就是如何处理好特殊性与普遍性之间的关系,如何通过事物的特殊性找出事物的普遍性,找出事物的一般规律,其次是在找出事物的一般规律的基础上利用计算机语言工具,用计算机语言来描述出所找出的规律,这就是程序设计或称之为编程,最后用具有不同特征的同类具体事物来验证用计算机语言来描述出的规律的正确性。

### 1.2.2 语言与程序

语言与程序是相互依存的,语言是程序的工具,语言对程序而言具有强制性的规范作用。任何程序都必须遵照一定的规则进行,否则就不能被理解和承认。语言与程序的区别表现在:

(1) 语言是稳固的,是进行交际的工具;程序是人们运用这个工具进行交际的过程和结果。

(2) 语言是社会共有的交际工具,社会因素是语言的本质因素;程序除了具有社会因素外,还具有个人的因素。

(3) 语言系统的各个结构成分是有限的;程序是无限的。

掌握语言不是人们学习语言的目的,学习语言的目的是运用语言编写程序。这同人们生活中讲话、写文章与语言的关系是一样的,学习语言的目的是为了使用语言,用语言实现人与人之间的交流,用语言表达出人的思维。

### 1.2.3 思维与程序

语言是思维的工具,思维是通过语言来表现、描述的。程序是思维的具体实现者。没有程序,思维就无法被体现出来。

### 1.2.4 语言、思维与程序

为了进一步说明语言、思维和程序之间的关系,为了使读者对语言、思维和程序有较深入的认识,请看实例。

假设有这样两组正整数:

(1) 50, 15, 24, 8, 7

(2) 1, 36, 26, 5, 30, 28, 100

要求读者做两件事:分别从小到大对这两组数排列和把这两组数合并起来从小到大排列,对人们来说这两个问题并不陌生,很快就会说出答案,答案分别为:

(1) 7, 8, 15, 24, 50

(2) 1, 5, 26, 28, 30, 36, 100

和

1, 5, 7, 8, 15, 24, 26, 28, 30, 36, 50, 100

类似的问题在人们生活中是经常出现的,人们解决起来也非常轻松。但是很少有人思考过这些问题是如何解决的,解决这样的问题有没有规律。同样是这个问题,下面换一个提法,读者看如何解决。

设有任意两组正整数,第一组数有五个数,它们分别为: $X_1, X_2, X_3, X_4, X_5$ ;第二组有7个数,它们分别为: $Y_1, Y_2, Y_3, Y_4, Y_5, Y_6, Y_7$ ,要求做与前面一样的事,即分别从小到大对这两组数排列和把这两组数合并起来从小到大排列,对于这样的问题又如何解决呢,这就是具体与抽象。它深刻的体现了什么是形象思维,什么是逻辑思维,形象思维与逻辑思维的关系。

对于上述 50, 15, 24, ..., 它们分别代表着某个具体的数值,这些具体的数值之间的关系一目了然,它们各具不同特性,所以在对这些具有具体数值的数进行处理时,处理者通过形象思维很容易就能找出每组数中所有数之间的关系,在此基础上加简单的逻辑思维,就把问题解决了,而对于像  $X_1, X_2, Y_1, Y_2, \dots$ , 虽然它们代表的也是数,但对解决问题的人来说,它们都是些抽象的、不确切和不固定的符号,它们代表了解决上述问题的一般性。用这些抽象的、不确切和不固定的符号解决上述问题,就要找出具有通用性的解决方法,也就是说,不论具体的数是什么,不论具体的数之间表现出的是什么关系和特征,用所找出的一般方法都能解决。为了实现对  $X_1, X_2, Y_1, Y_2, \dots$  抽象的、不确切和不固定的符号求上述解,就需要具有较高的逻辑思维能力,充分发挥自己的想象力。但在解决这类抽象问题过程中,不能离开形象思维,必须把形象思维与逻辑思维紧密地联系起来。并且还要把形象思维放在解决问题的首位。以本例为例:

为了实现对  $X_1, X_2, Y_1, Y_2, \dots$  抽象的、不确切和不固定的符号求上述解,首先是任意确定一组与  $X_1, X_2, X_3, X_4, X_5$  相对应的具有确定值的数(在数学中这被称作代数),如:

5, 15, 3, 7, 10

对这组数从小到大进行排列,同时确切地描述出排列这组数所采用的方法。比如:

第一步处理为:5 与 3 交换,即第一个数与第三个数交换,结果为

3, 15, 5, 7, 10

第二步处理为:15 与 5 交换,即第二个数与第三个数交换,结果为

3, 5, 15, 7, 10

第三步处理为:15 与 7 交换,即第三个数与第四个数交换,结果为

3, 5, 7, 15, 10

第四步处理为:15 与 10 交换,即第四个数与第五个数交换,结果为

3, 5, 7, 10, 15

对 5, 15, 3, 7, 10 从小到大排列结束。

同样按上述步骤对任意给定的另一组数进行排列,这个过程既是验证上述方法的正确性过程,又是发现问题,继续寻找解决问题的一般方法和规律的过程,如:

19, 4, 16, 2, 7

第一步处理为:19 与 16 交换,即第一个数与第三个数交换,结果为

16, 4, 19, 2, 7

第二步处理为:4 与 19 交换,即第二个数与第三个数交换,结果为

16, 19, 4, 2, 7

第三步处理为:4 与 2 交换,即第三个数与第四个数交换,结果为

16, 19, 2, 4, 7

第四步处理为:4 与 7 交换,即第四个数与第五个数交换,结果为

16, 19, 2, 7, 4

按对 5, 15, 3, 7, 10 这组数排列的方法对 19, 4, 16, 2, 7 从小到大排列,其结果并不是所期望的结果。其原因在于:

(1) 这两组数内部数字之间的关系是不一样的,即数字大小的顺序关系不一样。

5, 15, 3, 7, 10 和 19, 4, 16, 2, 7 这两组数原始的排列顺序如表 1. 2. 1 所示。

表 1. 2. 1

位 置	第一组		第二组	
	顺序	值	顺序	值
1	3	2	19	5
2	15	5	4	2
3	5	1	16	4
4	7	3	2	1
5	10	4	7	3

表 1.2.1 中清楚地反映了两组数内部各数的原始位置和按大小排列的先后顺序。作为上述处理方法显然只适合具有第一组数内部特征的数之间的从小到大的排列。所以上述处理方法不具有通用性。这种方法不能用于解决任意五个数之间从小到大的排列问题。

(2) 处理方法表达得不完整。上述处理方法只反映了处理过程的一小部分,对 5,15,3,7,10 这组数的处理,看似是经过四步处理完成的,而实际上这四步只是处理过程中的“数之间的交换操作”,而交换前的数与数的比较操作过程,比较操作的顺序与方法被忽略了。试想,为什么第一步处理选择 5 与 3 交换,第二步处理选择 15 与 5 交换,第三步处理选择 15 与 7 交换,第四步处理选择 15 与 10 交换,为什么通过这四步处理就认为对 5,15,3,7,10 这组数进行排列的过程结束了,为什么通过同样这四步对 19,4,16,2,7 这组数的处理就认为有错误,处理没有结束,处理结果失败呢?原因很简单,在确认某两个数进行交换前存在着对具体数进行相互间的“比较”操作,通过比较确认谁与谁进行交换,不断地进行这一过程,直到当比较到最后被认为数之间再没有可交换的了,就认为处理结束。这正是解决本问题的本质所在,也正是人们忽略的。前述的四步处理实际上反映的是对比较的结果进行的交换操作,比较在前,交换在后,交换是对比较结果进行的一种处理。

确切地说明比较过程和方法是非常重要的,只有这样才能反映出解决问题的方法、步骤,才能通过对所采用的方法、步骤进行分析,看它是否反映了解决此问题的一般规律。

找出具有通用性的解决问题的方法,仅仅是解决问题的第一步,第二步考虑的问题是用什么语言形式表达和描述出所找出的具有一般性的、反映问题本质的解决方法。没有描述语言,不能用某种语言对处理的方法进行表达,实际上就是没有找到解决问题的方法。反过来,用语言表达出解决问题的方法,并不一定是正确的,不一定是带有普遍性的和反映问题本质特征的处理方法。例如有如下两组数:

19,12,8,1 和 28,24,14,19,对这两组数分别进行从小到大排列,下面给出两种排列解法。

#### **排列解法(算法)1:**

##### **步骤:**

(1) 第一个数与第四个数比较,如果第四个数小于第一个数,则第四个数与第一个数交换,否则两个数的位置保持不变;

(2) 第二个数与第三个数比较,如果第三个数小于第二个数,则第三个数与第二个数交换,否则两个数的位置保持不变。

#### **排列解法(算法)2:**

##### **步骤:**

(1) 第一个数与第三个数比较,如果第三个数小于第一个数,则第三个数与第一个数交换,否则两个数的位置保持不变;

(2) 第二个数与第四个数比较,如果第四个数小于第二个数,则第四个数与第二个数交换,否则两个数的位置保持不变;

(3) 第三个数与第四个数比较,如果第四个数小于第三个数,则第四个数与第三个数交换,否则两个数的位置保持不变。

利用不同的解法,对这两组数进行排列,其结果如表 1.2.2 所示。

表 1.2.2

组数 \ 算法 \ 结果	19,12,8,1	28,24,14,19
1	1,8,12,19	19,14,24,28
2	8,1,12,19	14,19,24,28

表 1.2.2 表明上述两种算法,都不具有通用性,下面再给出一个解法(算法):

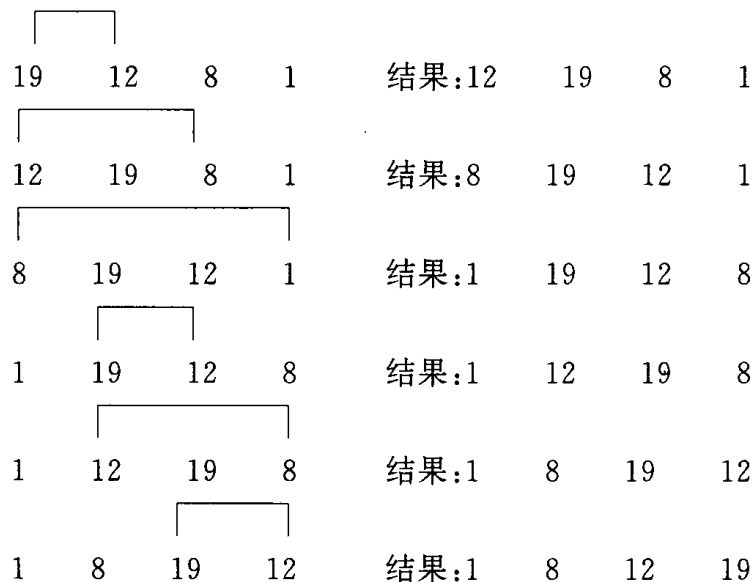
**步骤:**

(1) 第一个数与其后面的所有数按排列位置顺序,顺序连续进行比较,当遇到被比较的数小于第一个数时,两个数就交换位置,使第一个数总是成为已被比较过的数之中最小的,直到第一个数与最后一个数比较,包括交换之后,第一个数就成为一组数中最小的一个;

(2) 按此方法,以第二个数为基准,与其后面的所有数相比较、交换,方法同第(1)步骤一样,这样第二个数就成为一组数中次小的一个数;

(3) 以此类推,直到倒数第二个数为止。

仍用上例写的 19,12,8,1 这组数排列过程如下:



上述算法具有一般性,对任意个数从小到大的排列都可以采用这种方法。

对于算法,前面采用的都是文字描述、文字表达,这是人们容易理解和容易接受的对算法的表示方法。但是如果把这种算法移植到计算机上,使计算机接受这种算法,并使计算机能根据这个算法解决人们交给它的一组数的从小到大的排列问题,人们就需要学习计算机语言,并根据计算机语言规则,按解决排列问题算法编写出解决问题的程序。

由此可以看出,语言是基础,是表达算法的工具,掌握语言的目的是编写程序,程序是算法的具体表达。算法来源于思维,是思维的结果,是语言对思维的表达。

**思考题：**

1. 什么是语言,语言的本质是什么?
2. 计算机语言与人们日常生活中语言之间的相同点和不同点是什么?
3. 学习语言的目的是什么?
4. 什么是思维,思维包括几种方式,各种方式的思维特征是什么?
5. 如何理解语言与程序之间的关系。

## 第 2 章 PASCAL 语言基础

### 2.1 计算机语言概述

计算机是信息处理的有力工具,但目前的计算机尚不能直接理解人类的自然语言,人们要指挥计算机运行,不得不使用特定的语言与之打交道,这个用来和计算机打交道的,人类针对计算机的特点设计的,为使计算机能够识别,并据此按人的意志为人类服务的语言就是计算机语言。使用计算机时,事先要为待处理的问题编排出确定的工作步骤,把预定的方案用指定的一种计算机语言表示出来,即编写程序,当程序被输入到计算机后,在计算机软、硬件系统支持下,计算机按程序要求和规定自动地进行工作,以实现为使用者提供服务的目的。这种用计算机系统所能接受的语言来编写程序的过程称为程序设计,而其语言为程序设计语言。

#### 2.1.1 计算机语言分类

程序设计语言,按其发展可分为机器语言、汇编语言、高级语言。

##### (1) 机器语言

计算机能够直接识别与执行的基本操作,如数据传送,加、减、乘、除等是计算机硬件在设计和制造时确定的。由于计算机的基本物理器件是具有两种状态的器件,所以计算机的基本操作都是由二进制代码表示的,该编码称为机器指令,每一条指令规定机器完成一定的动作,每台计算机在设计时,对于能执行什么指令,执行多少条指令,怎样表示这些指令都有自己的规定。一台机器所能执行的所有指令的集合叫作机器的指令系统,每一种系列的计算机都有它们各自的指令系统,不同系列的计算机指令形式与功能是不同的。

用机器语言编写程序,计算机可以直接识别,占用内存少,执行效率高,但其存在两个严重缺点:一是机器语言难懂、难记,程序直观性差,易于出错,又不便修改,编程工作十分繁琐;二是机器语言不能移植,没有通用性,用某种系列机的机器语言编写的程序不能在其他系列机上运行,限制了计算机的应用推广。

##### (2) 汇编语言

为了克服使用机器语言的困难,在 50 年代初出现了汇编语言,它与机器语言相比前进了一大步,便于理解和记忆,但它仍是面向机器的语言。汇编语言中的语句与机器指令之间基本上是一一对应的,它在格式与内容上均类似于机器语言,各种类型的计算机有自己特有的汇编语言。因此,使用汇编语言需要足够的专业训练,熟悉具体机器的指令系统。

##### (3) 高级语言

人们希望有独立于机器、接近于常用的数学表达式和自然语言的计算机语言。从 50 年代末、60 年代初开始,相继提出了多种高级程序设计语言,目前已经设计和实现了数百种不

同的程序设计语言。所谓高级语言主要是指通用性,用户不必对计算机的指令系统有深入地了解就可以用它来编写程序。用同一种高级语言编写的程序在不同型号的计算机上运行时,一般只需作某些微小的改动即可,也就是说具有较强的可移植性。如: BASIC, DBASE, FORTRAN, PASCAL 等。

由于高级语言的上述特点,使用高级语言的优越性突出表现在提高程序人员的劳动生产率,它比低级语言更易于编写、查错、验证、阅读和修改。程序员只需要正确地掌握词汇、语言,合理地表达语义就可编制程序,而无需关心语言实现的细节。

### 2.1.2 高级语言的特点

高级程序设计语言是一种面向自然语言的一类计算机语言,其特点如下:

- ① 接近于数学和自然语言,比较直观;
- ② 易读,易写,易于交流、出版和存档;
- ③ 有利于防止程序出错和便于验证其正确性,发现错误易于纠正;
- ④ 编程代价低而效率高;
- ⑤ 独立于机器,具有标准版本;
- ⑥ 同一程序可在不同的机器上执行,用户无需针对具体机器重新编程。

### 2.1.3 程序语言的本质

程序语言是一种记号系统。它是由词法、语法和语义几个方面定义的。任何语言程序都是一定符号集上的字符串的有限序列,它是描述一定数据的处理过程。

### 2.1.4 词法和语法

#### (1) 词法

任何语言都具有一个语言所能接受的符号集,即符号表。词法规则规定了符号表中哪样的字符串是一个单词符号。词法具有如下特点:

- ① 一个程序语言只使用一个有限字符集作为符号表;
- ② 单词符号是语言中具有独立意义的最基本结构。

语言中的单词符号主要包括:

常数、标识符、保留字、运算符和分界符等。标识符与保留字的概念将在本章第3节中介绍。

#### (2) 语法

语法是构成程序的规则。语法规则规定了如何从单词符号组成更大的结构,语法具有如下特点:

- ① 语法比单词符号具有更丰富的意义;
- ② 程序书写必须服从语言的语法格式;
- ③ 不同语言的语法规则是不一样的;
- ④ 由于同一种语言有可能存在多种版本,所以不同版本的同一种语言的语法不完全一样。

语法规则主要包括如下几个方面:



表达式、语句、分程序、过程和函数等。

## 2.1.5 程序的基本功能

程序的基本功能只有两个：描述数据和对数据进行处理。请看实例：

```
program ex1(input,output);  
var s,x,y :integer;  
begin  
    read(x,y);  
    s:=x+y;  
    writeln('x+y=',s)  
end.
```

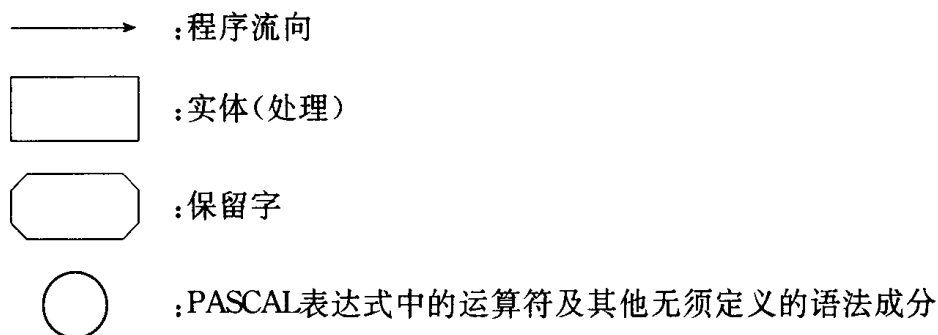
这是一个完整的程序，程序的第二行是对数据进行描述的部分，说明程序中有  $x, y, s$  三个变量，并且  $x, y, s$  是整数类型的变量。对数据的描述是为后面程序对数据处理所做的必要的准备工作。

`begin` 与 `end.` 包含的部分是对数据进行处理的部分，处理的对象就是程序第二行所说明过的变量。如果程序中出现了没有被说明的变量，程序在处理这个变量时，就会因为这个变量在前面未被系统注册而找不到和不能确定它的具体类型，程序出错，程序无法正常执行。

## 2.2 PASCAL 语言的语法表示方法

计算机程序设计语言，既然是一种语言，它就同任何一种自然语言一样，也有自己的语法规则。不同的计算机语言其语法规则是不同的。利用图示的方法描述它们，能够帮助读者更好地理解 and 掌握语言中各语法的功能及使用方法。

PASCAL 语言的语法图包含四种基本符号，符号及符号功能如下：



## 2.3 PASCAL 语言符号系统

任何一种程序设计语言，都需要建立一套自己的基本符号，这就是语言本身具有的、被语言接受的，并能反映出语言本身某些特征的符号系统。这些基本符号按照语法规则构成