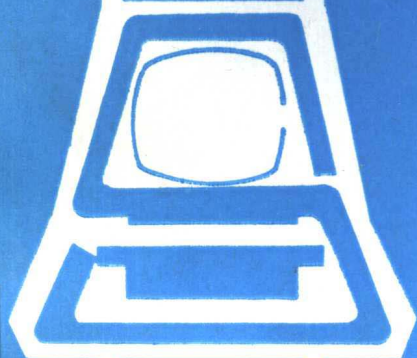


高等学校教材

PASCAL 程序设计

胡正国 主编



西北工业大学出版社

高等学校教材

PASCAL 程序设计

胡正国 主编

胡正国 陈锦纓 编
舒忠正 李立新

西北工业大学出版社

1989年5月 西安

内 容 简 介

PASCAL语言是一种十分流行的程序设计语言,它具有结构清晰、容易学习、使用方便等优点。目前,国内外许多学校都将它作为第一语言进行程序设计语言教学。

本书较详细地介绍了 PASCAL 语言的数据类型,各种语句以及它的程序结构。另外,本书还通过大量例子着重介绍了逐步求精的程序设计方法。

本书可作为计算机专业或其它理工科专业 PASCAL 语言课程的教材或教学参考书,也可供广大计算机工作者参考。

高等学校教材

PASCAL 程序设计

主 编 胡正国

责任编辑 王俊轩

责任校对 樊 力

西北工业大学出版社出版

(西安市友谊西路 127 号)

陕西省新华书店发行

西北工业大学出版社印刷厂印装

ISBN 7-5612-0125-7/TP·26(课)

开本 787×1092 毫米 1/16 12.75 印张 308 千字

1989 年 5 月第 1 版 1989 年 5 月第 1 次印刷

印数 1—4000 册 定价: 2.58 元

前 言

PASCAL 语言是由瑞士的 N·Wirth 教授于 1971 年正式提出的一种很受欢迎的程序设计语言。这种语言的特点是,语法比较严格,而且结构清晰、容易学习、容易掌握。目前,不仅大多数计算机都配置有这种语言,使它得以广泛地应用,而且从教学观点来看,它也是一种很受推崇的语言,国内外不少学校都将它作为第一语言进行程序设计语言的教学。

这本教材是在我们多年进行这门课程的教学以及原有讲义的基础上编写而成的。在这本教材中,我们除了介绍 PASCAL 语言的各种语法规则,并通过较大的例子介绍如何使用这种语言编写程序以外,还着重介绍了逐步求精的程序设计方法。希望通过这门课程的学习,使读者不仅能掌握 PASCAL 语言的基本编程方法,而且能形成一种较好的程序设计风格。

在这本教材的正文中,我们仅介绍了 PASCAL 语言的标准文本。但是,在实际使用的一些 PASCAL 文本(例如,MS-PASCAL,UCSD PASCAL)中,都对标准 PASCAL 文本作了许多扩充和限制。为了便于读者使用,在附录 C 中我们对常用的几种 PASCAL 文本中的扩充及限制的内容作了较为详细的介绍。

这本教材是列入航空工业部 1988~1990 年教材选题计划的部委规划教材。它可以作为计算机学科各专业 PASCAL 程序设计课程的教材,也可以作为其它理工科专业 PASCAL 语言教学的教材或教学参考书。讲授学时约 50~60 学时。

这本教材由西北工业大学计算机科学与技术系胡正国主编。参加编写工作的有,西北工业大学计算机科学与技术系胡正国(第一、五、六章)、陈锦纓(第七、八、九章,第十章部分内容及附录 C),南京航空学院计算机科学与工程系舒忠正(第三、四章)李立新(第二章,第十章部分内容及附录 B)。如果我们的合作能为提高 PASCAL 程序设计课程的质量,促进这门课程的改革作出一些微薄的贡献,我们将感到十分欣慰。

北京理工大学管理工程系梁遗金副教授在百忙中审阅了全书的内容,并提出了宝贵的意见。另外,南京航空学院王建东同志参加了部分章节的讨论,也提出了有益的建议。在此我们一并表示感谢。

由于编者水平所限,不妥之处在所难免,恳望读者提出宝贵意见,我们将不胜感激。

编 者

1988 年 2 月于西安

目 录

第一章 引言	1
§ 1.1 计算机系统与程序设计语言	1
§ 1.2 结构程序设计	3
§ 1.3 PASCAL 语言简介	5
第二章 PASCAL 字符集和基本数据类型	8
§ 2.1 PASCAL 字符集及标识符	8
§ 2.2 常量、变量和表达式	10
§ 2.3 标准类型	13
§ 2.4 枚举类型和子界类型	17
§ 2.5 标准函数	20
习题	22
第三章 简单语句和选择语句	23
§ 3.1 赋值语句	23
§ 3.2 读语句和写语句	24
§ 3.3 复合语句	29
§ 3.4 IF语句	30
§ 3.5 CASE语句	33
习题	37
第四章 重复语句和转向语句	40
§ 4.1 WHILE语句.....	40
§ 4.2 REPEAT 语句.....	44
§ 4.3 FOR语句	49
§ 4.4 GOTO语句	53
习题	55
第五章 过程、函数及程序结构	58
§ 5.1 过程	58
§ 5.2 函数	63
§ 5.3 变量和参数	65
§ 5.4 过程和函数的递归结构	73
习题	77

第六章 逐步求精的程序设计方法	80
§ 6.1 逐步求精方法的基本思想	80
§ 6.2 计算器模拟程序	82
第七章 结构数据类型	88
§ 7.1 数组类型	88
§ 7.2 集合类型	96
§ 7.3 记录类型	100
习题	110
第八章 文件类型	112
§ 8.1 顺序文件	112
§ 8.2 正文文件	117
§ 8.3 文件更新技术	122
§ 8.4 文件应用举例	127
习题	133
第九章 动态数据结构	135
§ 9.1 静态和动态数据结构	135
§ 9.2 指针类型	136
§ 9.3 链表	139
§ 9.4 二叉树	154
习题	156
第十章 程序实例	158
附录 A 可显示 ASCII 码表	174
附录 B PASCAL 语言语法图	175
附录 C 几种常用 PASCAL 系统的主要扩充和限制	181
参考文献	198

第一章 引言

§ 1.1 计算机系统与程序设计语言

世界上第一台电子计算机 ENIAC 是 1946 年在美国问世的。40 多年来，特别是近 15 年以来，计算机技术和其它新兴学科一样发展是十分迅速的。目前，计算机已经深入到国民经济的各个部门，发挥着十分显著的作用。

总的来看，计算机系统是由两大部分组成的，即硬件和软件。硬件通常又称裸机，它是指计算机的物理实体。软件是指为使计算机能正常运行或完成一些特定功能的程序系统，例如，操作系统、编译程序、数据库系统等。对一个计算机系统来说，这两个部分都是不可缺少的重要组成部分。硬件是软件及用户程序运行的舞台，软件是硬件能正常工作的保障。

更具体地说，计算机的硬件是由中央处理机 (CPU)、内存贮器和外部设备 (包括外存贮器和输入、输出设备) 等部分组成的。它们之间的相互连系如图 1.1 所示。

计算机硬件的发展经历了四个阶段，即通常说的第一代计算机 (电子管计算机)，第二代计算机 (晶体管计算机)，第三代计算机 (中、小规模集成电路计算机) 及第四代计算机 (大规模集成电路计算机)。随着计算机硬件的迅速发展，程序设计方法

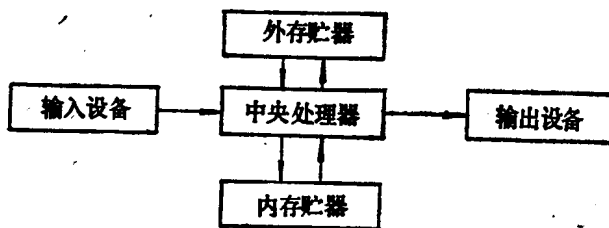


图 1.1 硬件示意图

也发生了很大的变化。对第一代和有些第二代计算机来说，由于其运算速度比较慢 (每秒 1 千次~1 万次运算)，内存容量比较小 (1K 到几个 K，所谓 1 K 是指 2^{10} ，即 1024 个单元)，因而大多数采用机器指令或汇编语言来编写计算程序。例如，有一台假想的计算机，它的指令系统是

操 作	指 令 形 式	说 明
+	0 0 A	$(S) + (A) \Rightarrow S$
-	0 1 A	$(S) - (A) \Rightarrow S$
×	0 2 A	$(S) \times (A) \Rightarrow S$
÷	0 3 A	$(S) \div (A) \Rightarrow S$
存数($\rightarrow$)	0 8 A	$(S) \Rightarrow A$
取数($\leftarrow$)	0 9 A	$(A) \Rightarrow S$
...	...	...
停机(Ω)	IF 0000	

其中，A表示某一内存地址，S表示累加器，(A)、(S)分别表示A及S中的内容。有这样一个指令系统，计算公式

$$P = \pi r^2 + \frac{1}{2}ab$$

可用下列指令序列来完成

```

←      π
×      r
×      r
→      α  (α 是存放中间结果的单元)
←      α
×      b
+      2
+      α
→      β  (β 是存放结果的单元)
Ω      0

```

然后，再进行地址分配，例如

0010:π, 0011:r, 0012:α, 0013:b, 0014:2, 0015:α, 0016:β

并假定指令从 0100 开始存放。这样，进一步将真地址代入以后即可得到下面的能为计算机直接接受的程序（假定地址和操作码按 16 进制编码）

```

0100:09 0010
0101:02 0011
0102:02 0011
0103:08 0015
0104:09 0012
0105:02 0013
0106:03 0014
0107:00 0015
0108:08 0016
0109:1F 0000

```

将这一程序输入计算机，运行后即可得到计算结果。这就是一个用机器语言，即上述的指令系统编写出的程序。

汇编语言和机器语言的主要区别在于，操作码及地址是用符号表示的，例如，+ 用 add，- 用 sub 表示等。这样，就省去了人工进行存贮分配和代入真地址等工作，也就是将这一部分工作交给计算机由汇编程序去完成。

从上面的这个简单例子可以看出，用机器指令及汇编语言编写程序是十分麻烦的。随着计算机运算速度的提高和计算机应用范围的扩大，它们远远适应不了客观现实的要求，于是，各种高级语言相继产生。高级语言的特点是简单易学，使用方便，从而为计算机的推广和使用开创了广阔的前景。例如，上面的计算公式用 PASCAL 语言可以表示为

$$P := 3.14159 * r * r + 0.5 * a * b$$

它和一般的代数公式是十分类似的。

自从1954年第一个比较完整的高级语言FORTRAN问世以来,相继出现了许多各具特点的高级语言。其中,比较常用的有BASIC语言(BASIC语言),ALGOL语言(ALGOL语言),COBOL语言(商用语言),LISP语言(表处理语言)及PASCAL语言(结构程序设计语言)等。

用高级语言编写的程序通常称为源程序。源程序虽然编写方便,但是计算机是不能直接执行的。为了执行它,计算机必需通过一定的软件(例如,解释程序、编译程序及连接程序等)对它进行加工,以生成计算机能直接执行的代码。一般地说,高级语言的加工过程可以分为解释执行和编译两种形式,而且大多数语言采取后一种形式。这种形式的加工过程,即由源程序到可执行代码的处理过程如图1.2所示。具体的操作过程可参阅有关计算机的使用说明。

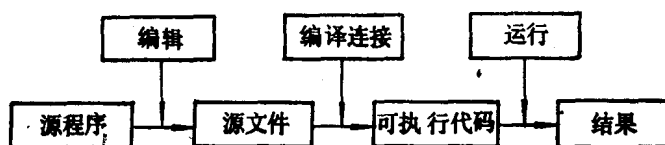


图 1.2 程序运行过程示意图

§ 1.2 结构程序设计*

前面已经提到,程序设计语言的发展经历了从机器语言、汇编语言到高级语言的几个阶段,它反映了程序设计方法的发展过程。我们也可以从另外一个角度上来看待程序设计方法的发展过程。

对第一代计算机和有些第二代计算机来说,由于计算机的运算速度较慢,内存容量较小,所以大多数采用机器语言和汇编语言编写程序,同时,在编写程序时必须十分注意节省内存单元和减少指令或语句的执行条数。在这种情况下,编写程序是一种技巧性很强的工作,也可以说是一种手工艺式的劳动。随着计算机速度的不断提高,内存及外存容量的不断扩大,目前,不仅大多数计算机都采用高级语言来编写计算程序,而且计算机的一些系统软件和应用软件也可以用某些高级语言(例如,PASCAL语言)来编写。这时,对一个程序来说,运行时间和占用存贮空间的大小已经不是编写者考虑的主要问题。主要问题已逐渐转化为希望编写的程序结构比较清晰,容易阅读,容易维护。同时,也希望编写程序的方法能规范化、工程化。程序设计方法的这一发展过程可以概括为:由手工艺式的设计方法到工程化的设计方法。

这一发展过程是方法论上的一个飞跃。为什么会产生这一飞跃呢?这是由计算机技术发展过程中的“软件危机”引起的。所谓软件危机是指:在本世纪70年代初,随着计算机硬件技术的发展和计算机应用范围的不断扩大,需要研制一些大的软件系统,例如,操作系统、程序库等,而一个大的系统研制的周期是较长的,工作量也很大(常常需要几百到几千人年),同时,由于原有的程序设计方法的局限性,设计出的软件系统往往隐藏着许多错

* 对于初次接触程序设计语言的学生,这一节可放在第六章进行讲授。

误，这就给软件的维护和使用带来很大的困难。一方面，客观上需要研制大量的软件，另一方面，按照原有的方法研制软件周期长，可靠性差，维护困难。这就是“危机”之所在。

1960年，Niklaus Wirth首先提出“GOTO语句是有害的”，向传统的程序设计方法提出了挑战，从而引起了人们对程序设计方法讨论的日益重视。许多著名的计算机科学家参加了这种讨论。结构程序设计这一新的课题也正是在这种广泛而深入的讨论中逐渐产生和形成的。

什么是“结构程序设计”呢？虽然在各种文献中，人们从不同的角度给出了不少的定义，但到目前为止，尚没有一个很严格的，又能为大家所普遍接受的定义。为了使大家对这一课题有一个概括的了解，我们给出以下的定义。

结构程序设计是一种进行程序设计的原则和方法，按照这种原则和方法设计出的程序的特点是：结构清晰，容易阅读，容易修改，容易验证。

按照结构程序设计的要求设计出的程序设计语言称为结构程序设计语言。

利用结构程序设计语言，或者说按照结构程序设计的思想编制出的程序称为结构化程序，或者好结构的程序。

更具体地讲，什么样的程序可以称为结构化程序呢？一般地，可以说结构化程序是由以下七种结构组成的。

1. 序列结构 序列结构如图 1.3 所示。

它的含义是顺序执行动作 A 和 B。其中 A，B 可以是一个语句（甚至空语句），也可以是这里介绍的七种结构中的任一种结构。这一点对下面的几种结构也是适用的。

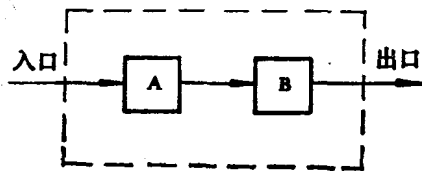


图 1.3 序列结构

2. 选择结构

选择结构有三种，最常用的两种选择结构（也称条件结构）如图 1.4 所示。更一般的选择结构（也称分情形结构）如图 1.5 所示。第一种条件结构的含义是，若条件 P 成立，则执行动作 A，否则什么也不作，第二种条件结构的含义是，若条件 P 成立，执行动作 A，否则执行动作 B。分情形结构的含义是，考查算式 E 的值（假设它的值为整数），若 $E = i$ ，执行相应的动作 $A_i (i = 1, 2, \dots, n)$ ，若 $E < 1$ 或 $E > n$ ，则执行动作 A_{n+1} 。

显然，图 1.4(a) 是图 1.4(b) 的特殊情况（这时 B 是空语句），而且图 1.5 所示的结构可以通过嵌套使用图 1.4 的结构表示出来。也就是说，这三种结构不都是组成程序所必需的结构。但是，为了更灵活地编写程序，我们仍然将它们看作用于组成程序的不同的结构。

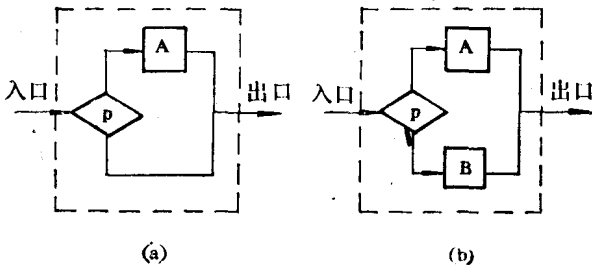


图 1.4 条件结构

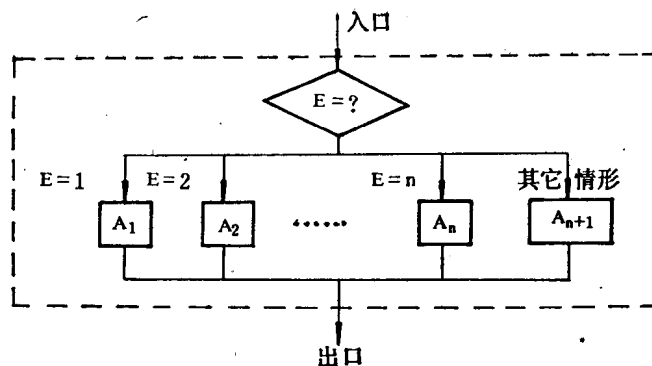


图 1.5 分情形结构

3. 重复结构

重复结构也有三种，它们分别如图 1.6(a)、(b)、(c) 所示。图 1.6(a) 所示的结构含义是，当条件 P 成立时，重复执行动作 A，直至条件 P 不成立时为止。其它两个结构的含义和它是类似的，这里不再赘述。和选择结构类似，虽然前两种重复结构可以看作是第三种重复结构的特殊情况，但是为了构造程序方便起见，仍然将它们看作是用于组成程序的不同的结构。在这三种重复结构中，常用的是前两种形式，和这两种结构对应，PASCAL 语言有相应的两种重复语句（参看第四章）。

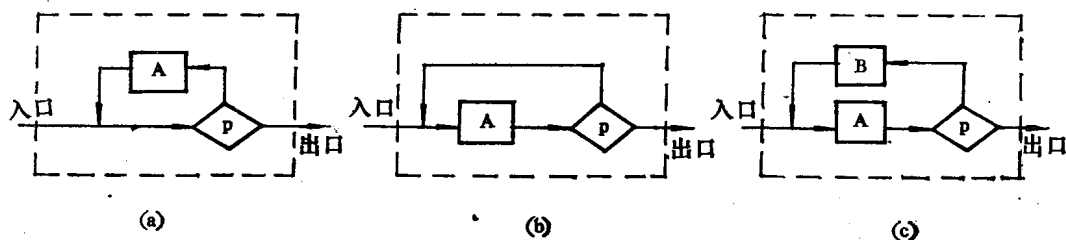


图 1.6 重复结构

分析以上七种结构我们发现，虽然它们形式不同，但却有一个共同的特点，即每一种结构都是一个入口，一个出口。这是结构化程序的最突出的特点。因而，可以说“一个入口，一个出口”是结构程序设计的一个基本原则。

为了设计出结构化的程序，改变传统的程序设计方法，除了遵循“一个入口，一个出口”的原则外，还必须有一些比较科学的程序设计方法。其中，最常用的一种方法是逐步求精的程序设计方法。在第六章我们将通过一些例子介绍这种方法，而且全书将采用这种方法来编写一些较为完整的程序。也希望读者能逐步学习和运用这种方法。

§ 1.3 PASCAL 语言简介

PASCAL 语言是由瑞士的 N. Wirth 教授于 1971 年提出的一种很受欢迎的程序设计语言。目前，不仅大多数计算机都配置有这种语言，使它得以广泛的应用，而且从计算机教育观点来看，它也是一种很受推崇的语言，不少学校都将它作为第一语言进行计算机语言教

学。PASCAL 语言之所以受到广泛的欢迎，我们感到，主要是由于它有以下一些特点。

1. 它是在 Algol-60 基础上发展的一种嵌套式结构的语言，语法比较严格而且简明，容易学习，容易掌握。

2. 它是一种结构程序设计语言，用它编写出的程序结构清晰，容易阅读，容易修改；同时，用它进行教学可以使用户得到较好的程序设计方法的训练。

3. 它是一种系统程序设计语言，或者说可以用它来编写系统程序和应用程序。最典型的一个例子是 PASCAL 语言的编译程序可以用 PASCAL 语言来编写。因此，PASCAL 语言应用是比较广泛的，它不仅可以用于编写一些数值计算的程序，而且更多的是用于非数值应用的问题。PASCAL 语言之所以具有这一特点，是由于它具有较为丰富的数据结构和控制结构，这些我们将在以后章节中逐渐予以介绍。

下面我们举一个比较简单的，但又是完整的 PASCAL 程序例子，为了叙述方便起见，我们将程序按行进行编号。

例 1.1 求边长为 x_1 , x_2 , x_3 的三角形面积的 PASCAL 程序。

```
1. PROGRAM ex11 (input, output);
2. VAR x1, x2, x3, p, s: real;
3. BEGIN
4.   read(x1, x2, x3);
5.   p:=(x1+x2+x3)/2;
6.   s:=sqrt(p*(p-x1)*(p-x2)*(p-x3));
7.   write('s=', s)
8. END.
```

这个程序中，第 1 行称为程序首部，它用于标志一个完整的程序的开始，并给程序命名（这个程序的名字为 *ex11*）；第 2 行是程序的变量说明部分，这里说明程序中用到的变量 x_1 , x_2 , x_3 , p , s 均为取实数值的变量；第 3 行到第 8 行是程序体，这是程序的主体部分，其中，第 3 行的 BEGIN 和第 8 行的 END 分别表示程序体的开始和结束，而且 END 后面应有符号“.”表示程序的结束。这一程序的程序体是由 4 条语句组成的。它们的作用分别是：读入三角形的边长（语句 4）；按海伦公式计算三角形的面积（语句 5, 6）；输出面积 s 。另外，每个语句的后面有分号，表示该语句的结束，而 END 前面的一个语句（语句 7）后面的分号可以省略。

按照我们前面介绍的观点来看，这个程序的程序体是由 4 条语句组成的序列结构。运行这个程序时，从输入设备上输入边长 x_1 , x_2 , x_3 ，即可在输出设备上得到三角形的面积 s 。

上面给出的是一个完整的程序，但是它是不完善的。我们知道，三个实数应该满足一定的条件（即任何两条边的和大于第三边）才能作为三角形的边长。这样，当读入的三个数不满足这一条件时，程序运行将产生错误。为了克服这一问题，我们可以引入一个逻辑型的量 p_1 （当以上条件满足时它成立，否则它不成立），并用它作为判断条件组成一个条件结构。程序如下（关于 p_1 的表达形式和条件语句的格式将在后面有关章节中详细介绍）：

```
1. PROGRAM ex12(input, output);
2. VAR x1, x2, x3, p, s: real;
3.   p1: boolean;
4. BEGIN
```

```

5.   read(x1, x2, x3);
6.   p1:=(x1+x2>x3)AND(x2+x3>x1)AND(x3+x1>x2);
7.   IF p1
8.     THEN BEGIN
9.         p:=(x1+x2+x3)/2;
10.        s:=sqrt(p*(p-x1)*(p-x2)*(p-x3));
11.        write('s=', s)
12.    END
13.   ELSE write('They Can not be triangulated')
14.   END.

```

这一程序的第 2、3 行为变量说明部分；第 4 行到第 14 行为程序体。需要特别指出的是，这一程序体虽然有 9 行，但是它是由三条语句组成的序列结构。其中，第 7 行到第 13 行是一个对应于第二种条件结构的条件语句，由第 8 行的 BEGIN 和第 12 行的 END 所限定的这一部分语句是这一条件结构的子结构，它又是由三条语句组成的序列结构。这就是 PASCAL 语言中最常用的嵌套形式，由于 PASCAL 语言采用了这种嵌套结构的形式，使得编写出的程序结构比较清晰，容易阅读，容易理解。更一般地说，任何一个程序都可以用序列、选择和重复三种结构表示出来，这样得到的程序就是一个好结构的程序。

从上面这个简单例子可以看出，PASCAL 程序的一般结构是：

```

程序首部；
说明部分；
程序体

```

其中，说明部分除了上面用到的变量说明以外，还有常量说明、类型说明、函数说明和过程说明等，有关内容我们将逐渐予以介绍。程序体一般地说就是由序列、选择和重复结构组成的语句序列。

上面我们对 PASCAL 语言的面貌作了一个十分简单的介绍，下面我们将逐步介绍它的一些具体内容。我们知道，任何一种程序设计语言总是由数据类型、控制结构和程序结构三个主要内容构成的。而控制结构又是通过语句，特别是一些控制语句实现的。因而，下面我们将分别介绍 PASCAL 语言的数据类型（第二章，第七章，第八章和第九章），各种语句（第三、四章）和程序结构（第五章）。另外，为了训练良好的程序设计风格和阅读、理解程序的能力，还将介绍逐步求精的程序设计方法和一些应用实例（第六、十章）。

第二章 PASCAL 字符集和基本数据类型

本章主要由三个部分组成。第一部分介绍 PASCAL 语言的字符集；第二部分介绍表达式的概念；最后介绍 PASCAL 语言的一些基本数据类型，即整型、实型、字符型、布尔型、枚举类型和子界类型。

§ 2.1 PASCAL 字符集及标识符

在 PASCAL 语言中，用以组成单词、语句以至整个程序的字符的全体称为 PASCAL 字符集。它们是 ASCII 代码字符集的子集（ASCII 代码字符集可参看本书附录 A）。

PASCAL 字符集包括下面三类字符：

· 数字：0，1，2，3，4，5，6，7，8，9。这些数字字符可以用来组成数（整数和实数），也可以用来组成下面将要介绍的标识符。

· 字母：a，b，c，…，z（或 A，B，C，…，Z）。在一般情况下，字母的大小写形式是不分的。但是，为了醒目起见，程序中的某些单词用大写字母表示，如 BEGIN，END，…等，某些单词用小写字母来表示，如 x1，x2，write，…等。

· 特殊符号（主要是一些运算符和分隔符）：+，-，*，/，<，=，>，（，），[，]，{，}，:，;，,，↑等。

有了以上字符即可组成 PASCAL 语言的单词，其中最基本的单词是标识符。它是用来给常量、变量、过程、函数等起名字用的。例如，在上一章给出的例子中，变量 x1，x2，x3 等就是用标识符表示的。一般地说，标识符是用以字母打头的字母数字字符串表示的。

为了比较严格地定义 PASCAL 语言的一些语法规则，通常采用语法图的形式。例如，标识符的语法图如图 2.1 所示。在这一语法图中，方框内的语法成分是需要用另外的语法图定义的（例如，字母和数字需要

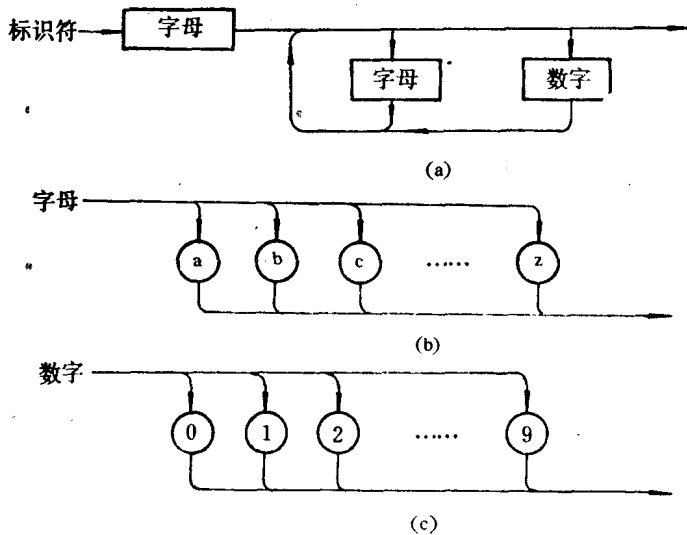


图 2.1 标识符的语法图

由图 2.1(b) 和 (c) 两个语法图定义)，圆形框以及后面将要用到的两端为圆弧的框内是 PASCAL 语言的保留字和专用符号，线段表示语法图中可以经过的途径，箭头表示该途径

的方向。

从图 2.1 可以看出, 标识符的第一个字符必须是字母, 后面可以不跟任何字符, 也可以跟任意多个字母或数字, 但不能包含任何其它符号。例如, 下面的标识符是合法的:

a, bookfile, BEGIN, x13

而下面的标识符是不合法的:

12P, IBM-PC, xy', \$30

另外, 从语法图来看, 对标识符的长度 (即标识符中包含字符的个数) 是没有限制的。但是在具体的计算机系统中, 对标识符的长度有自己的约定。通常仅取前面 8 个字符 (也有取 6 个字符的) 作为有效的字符, 例如, 下面两个标识符被认为是相同的。

bookfile 1, bookfile 2

在 PASCAL 语言中, 标识符按其性质可分为三种类型:

1. 保留字

保留字在 PASCAL 语言中有特定的含义, 用户不能作其他用途。为醒目起见, 在本书中保留字用大写字母表示。PASCAL 语言的保留字有:

AND	ARRAY	BEGIN	CASE	CONST
DIV	DO	DOWNT0	ELSE	END
FILE	FOR	FUNCTION	GOTO	IF
IN	LABEL	MOD	NIL	NOT
OF	OR	PACKED	PROCEDURE	PROGRAM
RECORD	REPEAT	SET	THEN	TO
TYPE	UNTIL	VAR	WHILE	WITH

2. 标准标识符

标准标识符在 PASCAL 语言中有特定的含义, 用户也可以根据自己的需要而重新定义。但是一般应尽量避免给标准标识符重新定义, 以免引起混乱。本书中标准标识符用小写字母表示。在 PASCAL 语言中, 标准标识符有:

abs	arctan	boolean	char	chr
cos	dispose	eof	eoln	exp
false	get	integer	input	ln
maxint	new	odd	ord	output
pack	page	pred	put	read
readln	real	reset	rewrite	round
sin	sqr	sqrt	succ	text
true	trunc	unpack	write	writeln

关于以上所列的保留字和标准标识符的含义和用法将在以后陆续介绍。

3. 用户自定义标识符

为了编写程序的需要, 用户可以自己定义一些标识符, 来给常量、变量、过程、函数等

起名用。用户自定义标识符通常用小写字母表示，例如，

loop, x12, pi, ...。

§ 2.2 常量、变量和表达式

一、常量

在 PASCAL 语言中凡是其值不变的量都称为常量。常量又可分为直接量和用户自定义常量两类。

所谓直接量是指在程序中直接引用的一些常数。在 PASCAL 语言中经常用到的常数有下面四种形式：

1. 整型常数

整型常数用带有正、负号的数字串表示，正号可以省略。例如，下面是一些正确的整型常数

12, -13, +378, 0

而下面几个数不是整型常数

3.1, -0.12, 21.000

整型数在计算机中是以定点数的方式存放的，它是一个精确的值。在不同的计算机系统中它的范围是不一样的。如在 IBM-PC/XT 微机系统中，整数是以16位二进制码存放的，并且用原码表示，其最高位表示数的符号。因此，对它来说整数表示的范围为：

$$|N| \leq 2^{15} - 1 = 32767$$

即它所能表示的最大整数为32767，最小整数为-32767。又如，在有些计算机系统中，整数用24位二进制数的原码形式存放。因此，对它来说整数的范围为 $-(2^{23}-1) \sim (2^{23}-1)$ 。为了使用方便，在 PASCAL 系统中，用标准标识符 maxint 表示最大整数，-maxint 表示最小整数。

2. 实型常数

实型常数有两种形式，一种是通常的小数点形式，例如

0.81, -2.56, +3.1416, 0.0

另一种形式是科学表示法，它是由一个小数形式的实数或一整数后跟指数部分组成的，而指数部分是用E后跟一整数表示的，例如

0.35E+2 (表示 0.35×10^2)

-7.2E-02 (表示 -7.2×10^{-2})

25E7 (表示 25×10^7)

不符合上述规定的数被认为是错误的，例如，下面一些写法是不允许的：

2. , .31 (小数点前后没有数字)

5.2E0.5, 7E-1.3 (E后面不是整数)

E-3 , .78E+03 (E前面没有数字或是错误数字)

和整型数一样,对具体的计算机系统来说,实型数也是有一定范围的。例如,对 IBM-PC/XT 微机来说,实数可用 32 位表示,其中阶码占 8 位,尾数(包括符号位)占 24 位,因而实数范围为

$$1\text{E}-38\sim 1\text{E}+38$$

若超出此范围则产生下溢(按 0 对待)和上溢。

3. 字符型常数

在一些非数值应用中经常会用到字符型常数以及字符串常数,它们用由单引号括起来的字符或字符串表示。例如:

'a', '0', '+', '*'
'abcd', 'this is a book'

对不同的 PASCAL 文本,在字符和字符串常量中允许出现的字符可以有所不同,但一般来说,按 ASCII 编码的可显示字符都可以使用。另外,如果在字符串中出现有单引号时,应将该单引号加倍以便和表示字符串的单引号区分。例如:

'ab"cd' 表示字符串 ab'cd

4. 布尔型常数

布尔型常数也称逻辑型常数,它只有两个,即 true(真)和 false(假)。

上面我们介绍了 PASCAL 语言中可以直接使用的几种常数,除它们以外,为了编写和调试程序方便,还允许用户在程序中自己定义一些常量。即可以给一些常数事先规定名子,然后在程序中将这名字作为常量使用。这些名字通常称为常量标识符,它们在常量说明部分予以定义。例如

```
CONST  
  pi=3.1416;  
  n=100;  
  k=-n;  
  t=true;  
  f=false;
```

这一常量说明定义了 5 个常量标识符 pi, n, k, t 和 f。常量说明的语法图如图 2.2 所示。其中,常数可以是前面介绍的四种常数中的任一种,它们的语法图不再一一列出了。从这一语法图可以看出,常量标识符既可以用常数定义也可以用已知的常量标识符定义。

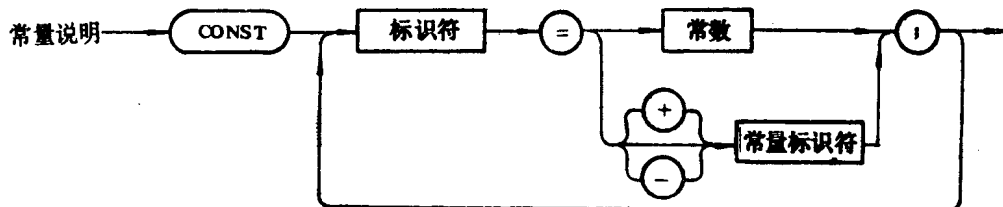


图 2.2 常量说明的语法图