



计算机基础教育系列教材

PASCAL 语言程序设计教程

朱 武 姚合生 孙家启 编著

安徽大学出版社

附赠(CIP)目录附录并图

Pascal 语言是通用的程序设计语言。它具有结构性好、数据类型丰富、表达能力强、书写格式规范、程序风格优异等一系列优点，特别适合作为高等院校计算机专业及非计算机专业的高等学校中得以普遍推广，且经久不衰。

本书共分十章，全面介绍了 Pascal 语言的语句、程序结构、数据类型及运算符、数组、指针、文件、过程设计方法与技巧的训练。附录中关于 Turbo Pascal 6.0 的主要内容及其集成开发使用的介绍。

PASCAL 语言程序设计教程

Pascal 语言及其程序设计的较好教材，可作为普通高等院校计算机专业及非计算机专业的大学生使用或工程技术人员的自学参考。

朱 武 姚合生 孙家启 编著

编委会名单



主任：孙家启

委员：王忠仁 王志雁

石 竹 冯崇岭

朱学勤 齐学梅

责任编辑：何主合

任：孙家启

员：王忠仁 王志雁

石 竹 冯崇岭

朱学勤 齐学梅

朱学勤 责任编辑：何主合

责任编辑：何主合

安徽大学出版社

员：王忠仁 王志雁 王永

石 竹 冯崇岭 孙家

朱学勤 齐学梅 吴国

责任编辑：何主合 张四平 陈桂

图书在版编目(CIP)数据

计算机等级(水平)考试系列教材

PASCAL 语言程序设计/朱武,姚合生,孙家启编著.
合肥:安徽大学出版社,1999.11
计算机等级(水平)考试系列教材
ISBN 7-81052-292-2

I. P… II. ①朱…②姚…③孙… III. PASCAL 语言-程序设计-水平考试-教材 IV. TP312

中国版本图书馆 CIP 数据核字(1999)第 51536 号

PASCAL 语言程序设计教程

朱 武 姚合生 孙家启 编著

出版发行 安徽大学出版社
(合肥市肥西路3号 邮编 230039)
联系电话 总编室 0551-5107719
发行部 0551-5107784
责任编辑 李 虹
封面设计 孟献辉
经 销 新华书店

印 刷 合肥朝阳印刷责任有限公司
开 本 787×1092 1/16
印 张 16.5
字 数 400 千
版 次 1999 年 11 月第 1 版
印 次 2000 年 11 月第 2 次印刷
印 数 4000

ISBN 7-81052-292-2/TP·27

定价 21.80 元

内容简介

Pascal 语言是通用的程序设计语言。它具有结构性好、数据类型丰富、表达能力强、书写格式规范、程序风格优异等一系列优点,特别适合作为讲授程序设计的教学语言,因之在高等学校中得以普遍推广,且经久不衰。

本书共分十章,全面介绍了 Pascal 语言的语句、程序结构、数据类型及编程方法,注重程序设计方法与技巧的训练。附录中关于 Turbo Pascal 6.0 的主要内容及其集成环境使用的介绍,颇具实用价值。

本书体系合理,概念清晰,通俗易懂,循序渐进,内容丰富,实用性强。是学习和掌握 Pascal 语言及其程序设计的较好教材,可作为普通高等学校的教学用书,也可供职大、电大、夜大学生使用或工程技术人员、自学者参考。

编委会名单

主任:孙家启

委员:王忠仁 王志雁 王永国 方潜生 尹荣章

石竹 冯崇岭 孙家启 仲红 朱武

朱学勤 齐学梅 吴国凤 李雪 李宁辉

何明 张国平 陈桂林 欧阳为民 孟浩

郑尚志 周鸣争 周恒忠 姚合生 梅申信

赵林玲 聂会星 黄毅 程承士 谢荣传

蔡之让 潘瑜

(按姓氏笔划为序)

秘书长:聂会星

编者

1999年8月

编写说明

图书在版编目(CIP)数据

计 为了支持计算机基础教育改革与建设,促进计算机基础课程教学与水平考试向纵深发展,我们按照计算机文化基础教育、技术基础教育和应用基础教育三个层次,组织编写了计算机基础教育系列教材。这套教材囊括了计算机文化基础、高级语言(QBasic, Visual Basic, C, Visual C++, PASCAL, FORTRAN77, FORTRAN90, FoxPro2.5b For Windows, Visual FoxPro6.0等)程序设计、软件技术基础、微型计算机原理、计算机网络、微型机组装与维护、CAI课件制作及应用等方面内容,涵盖计算机水平考试的一、二、四级(全国等级考试的一、二、三级),因而具有广泛的适应性。这套教材所具有的突出特点是:紧扣计算机基础教育大纲(即计算机水平考试大纲),兼具普通教材与考试辅导材料的双重功能;立意创新,内容简练,大量针对性极强的习题和典型例题分析是其他教材所少见;编写人员都是教学、科研第一线有着丰富教学与实践经验的教师,他们深谙相关知识点的张弛取舍。我们还聘请了三位知名专家担任高级顾问,以确保本系列教材的编写质量。

本系列教材的先期版本现已问世,第一辑各册已于1999年底全部出齐。由于计算机技术的发展比人们想像的还要快,所以本系列教材又增加了不少新内容,我们今后还将不断调整教材内容、平台和版本,使之与届时发展相适应,以便教材以更新更好的面目呈现在读者面前。

本系列教材编写目的明确,它特别适合于作为普通高校非计算机专业的本、专科教学用教材或成教、夜大、函大计算机专业的教材,也可供各地计算机水平考试考点使用,还可供广大计算机自学者、工程技术人员参考。

编 委 员

编 委 员

编写委员会
2000年5月

编 委 员

编 委 员

编 委 员

编 委 员

编 委 员

编 委 员

编 委 员

发行部 0551-5107794

责任编辑 李 虹

封面设计 蓝 雁

经 销 新华书店

编 委 员

第 1 次 1999 年 11 月第 1 版

(中成版)第 2 次印刷

印 数 1000

编 委 员

ISBN 7-81032-292-2/TP·27

定价: 21.80 元

如有影响阅读的质量问题,请与出版社发行部联系调换

前 言

Pascal 语言是通用计算机程序设计语言。它具有基本结构简单、数据类型丰富、表达能力强等诸多优点,用它编写的程序结构合理、实现效率高、可移植性好,因而被广泛地应用于多种系统软件和应用软件的设计之中。纵使在程序设计语言群星荟萃、各领风骚的今天,Pascal 语言仍然在计算机科学中占有重要的一席之地。

Pascal 语言又因为语言规范、逻辑严密、结构紧凑,特别适合于教学,所以又被视作良好的程序设计教学语言。许多初学者也正是从这里起步,然后再继续前进的。现今的计算机系统,无论是大、中、小型机还是微型机,都配备了 Pascal 语言运行环境。这些又为人们学习 Pascal 语言创造了十分便利的条件。

本书全面介绍了 Pascal 语言的基本成分、各种语句、程序结构、数据类型及程序设计方法。旨在帮助读者掌握 Pascal 语言的基本内容及初步程序设计的方法。兼顾全国高等学校计算机考试的需要,书中特意安排了一些典型例题分析和习题,以期对应试者有所帮助。附录中关于 Turbo Pascal 6.0 的主要内容及其集成环境的使用的介绍,为其他教材所不多见,对上机实践具有指导意义和实用价值。

全书共分十章及附录。第一章至第四章由孙家启编写,第五章、第六章和第十章由朱武编写,第七章至第九章、附录由姚合生编写。编写过程中,得到了本系列教材高级顾问们的悉心指导,参阅了国内外有关著作,安徽大学出版社为本书出版付诸了极大努力,编者在此一并表示衷心的感谢。

由于水平有限,加之时间仓促,错误与疏漏之处难免,恳请读者和专家斧正。

编 者

1999 年 8 月

目 录

第一章 Pascal 语言与程序设计概论	(1)
1.1 Pascal 语言的特点	(1)
1.2 Pascal 语言源程序结构	(1)
1.3 Pascal 语言的符号	(4)
1.4 程序设计与算法	(6)
1.4.1 程序设计	(6)
1.4.2 算法的概念	(6)
1.4.3 算法的特性	(7)
1.4.4 算法的表示	(8)
1.5 结构化程序设计	(10)
1.6 Pascal 语言程序设计步骤	(11)
1.7 典型例题分析	(12)
小结	(13)
习题一	(14)
第二章 顺序结构程序设计	(16)
2.1 数据类型	(16)
2.2 标准数据类型	(16)
2.2.1 整数类型	(16)
2.2.2 实数类型	(18)
2.2.3 字符类型	(20)
2.2.4 布尔类型	(21)
2.3 常量和变量	(22)
2.3.1 常量	(22)
2.3.2 常量定义	(23)
2.3.3 变量	(24)
2.3.4 变量说明	(24)
2.4 表达式与赋值语句	(25)
2.4.1 表达式	(25)
2.4.2 赋值语句	(27)
2.5 输入与输出语句	(28)
2.5.1 输入语句(读语句)	(29)
2.5.2 输出语句(写语句)	(31)
2.6 典型例题分析	(34)
小结	(38)
习题二	(39)

第三章 选择结构程序设计	(43)
3.1 if 语句	(43)
3.1.1 单分支 if 语句	(43)
3.1.2 双分支 if 语句	(44)
3.1.3 复合语句	(45)
3.1.4 if 语句的嵌套	(47)
3.2 Case 语句	(50)
3.3 典型例题分析	(54)
小结	(55)
习题三	(56)
第四章 循环结构程序设计	(59)
4.1 while 循环语句	(59)
4.2 repeat 循环语句	(62)
4.3 for 循环语句	(64)
4.4 嵌套循环	(68)
4.5 goto 语句	(71)
4.6 典型例题分析	(73)
小结	(77)
习题四	(78)
第五章 过程和函数	(84)
5.1 过程的定义和调用	(84)
5.1.1 过程定义	(85)
5.1.2 过程语句	(86)
5.2 函数的定义和调用	(86)
5.2.1 函数定义	(86)
5.2.2 函数引用	(87)
5.3 变量的作用域	(88)
5.4 数值参数和变量参数	(91)
5.4.1 数值形参和变量形参的定义	(91)
5.4.2 数值形参和变量形参的使用	(91)
5.5 过程的嵌套、递归及向前引用	(93)
5.5.1 嵌套	(93)
5.5.2 递归	(95)
5.5.3 向前引用	(96)
5.6 过程参数和函数参数	(98)
5.6.1 过程参数	(98)
5.6.2 函数参数	(100)
5.7 典型例题分析	(101)
小结	(108)

习题五	(108)
第六章 枚举类型和子界类型	(113)
6.1 类型定义	(113)
6.2 枚举类型	(114)
6.3 子界类型	(117)
6.4 典型例题分析	(119)
小结	(121)
习题六	(122)
第七章 数组类型	(124)
7.1 结构数据类型	(124)
7.2 数组的概念	(124)
7.2.1 数组说明	(124)
7.2.2 数组操作要点	(125)
7.3 一维数组	(126)
7.3.1 一维数组说明	(126)
7.3.2 一维数组应用举例	(126)
7.4 二维和多维数组	(130)
7.4.1 多维数组说明	(130)
7.4.2 二维数组应用举例	(131)
7.5 紧缩数组和字符串	(136)
7.5.1 紧缩数组的概念	(136)
7.5.2 字符串与字符数组	(137)
7.5.3 其他基类型的紧缩数组	(140)
7.6 典型例题分析	(140)
小结	(146)
习题七	(146)
第八章 集合和记录	(151)
8.1 集合	(151)
8.1.1 集合类型	(151)
8.1.2 集合的运算	(152)
8.1.3 集合变量的输入输出	(154)
8.1.4 集合类型应用举例	(154)
8.2 记录	(157)
8.2.1 记录类型	(157)
8.2.2 记录数组及其应用	(161)
8.2.3 记录作为过程和函数的参数	(162)
8.2.4 开域语句	(163)
8.3 带变体的记录	(167)
8.4 类型间的关系	(171)

8.4.1 变量的类型同一	(171)
8.4.2 类型相容	(172)
8.4.3 赋值相容	(173)
8.5 典型例题分析	(175)
小结	(177)
习题八	(178)
第九章 文件	(183)
9.1 文件的概念	(183)
9.1.1 文件概述	(183)
9.1.2 文件说明	(183)
9.1.3 文件指针与文件缓冲器变量	(184)
9.2 文件的建立和使用	(185)
9.2.1 文件的建立	(185)
9.2.2 文件的使用	(187)
9.2.3 文件的其他操作	(188)
9.3 文本文件	(190)
9.3.1 文本文件的特点	(190)
9.3.2 文本文件的操作	(191)
9.4 标准文件 input 和 Output	(195)
9.5 用文件缓冲器变量操作文件简介	(195)
9.6 典型例题分析	(196)
小结	(199)
习题九	(200)
第十章 指针类型与动态数据结构	(203)
10.1 指针类型	(203)
10.1.1 指针类型概述	(203)
10.1.2 动态变量的建立	(204)
10.1.3 动态变量的赋值和指针运算	(205)
10.1.4 动态变量的撤销	(206)
10.2 链表	(207)
10.2.1 链表的建立与遍历	(207)
10.2.2 结点的删除	(210)
10.2.3 结点的插入	(211)
10.3 树与二叉树	(212)
10.3.1 二叉树	(213)
10.3.2 二叉树的遍历	(213)
10.3.3 二叉树的建立	(214)
10.4 典型例题分析	(215)
小结	(221)

习题十.....	(221)
附录.....	(225)
附录 A ASCII 码表.....	(225)
附录 B Pascal 的保留字、标识符和运算符	(226)
B.1 保留字共 35 个	(226)
B.2 标准标识符共 40 个	(226)
B.3 运算符	(226)
附录 C Pascal 语言语法图	(227)
附录 D Turbo Pascal 上机及内容简介	(232)
D.1 Pascal 语言上机步骤	(232)
D.2 Turbo Pascal6.0 软件安装	(232)
D.3 Turbo Pascal6.0 集成环境及其应用	(233)
D.3.1 集成环境的启动	(233)
D.3.2 常用编辑键	(233)
D.3.3 菜单的使用	(234)
D.3.4 编译和运行 Turbo Pascal 程序	(234)
D.3.5 其他子菜单功能简介	(235)
D.4 Turbo Pascal 部分扩充功能简介	(237)
D.4.1 标准数据类型	(237)
D.4.2 变量类型强制转换	(237)
D.4.3 语句	(238)
D.4.4 过程与函数	(238)
D.4.5 数组	(240)
D.4.6 文件	(244)

同的数据对象、复杂的算法,得到高质量的程序。

(3)能适用于数值运算和非数值运算。有些语言(如 Fortran)适用于数值运算,而有的语言(如 Cobol)则适用于管理领域。Pascal 的功能较强,能应用于各种领域。

(4)程序的书写格式自由。它不像 Fortran 和 Cobol 那样对程序的书写格式有严格的规定。Pascal 程序允许一行写多个语句,一个语句可以写在多行上,这样可使写出的程序格式优美、便于阅读。

(5)用 Pascal 语言描述算法简明、清晰、便于接受,适宜初学者学习,可以为进一步学习其他语言打下良好的基础。

由于以上特点,目前国内外大多数高等院校都把 Pascal 语言作为高级程序设计语言的教学内容。这能给学生以严格而良好的程序设计的基本训练。

1.2 Pascal 源程序结构

Pascal 程序是由字母、数字和特殊符号按照它的一套语法规则组成的。为了使初学者从一开始对 Pascal 程序的结构建立一个整体概念,先举一个简单的 Pascal 源程序例子:

第一章 Pascal 语言与程序设计概论

1.1 Pascal 语言的特点

Pascal 语言属于编译型的高级语言,于 1968 年由瑞士苏黎世联邦工业大学的 N. 沃思(N. Wirth)教授设计完成,并于 1971 年正式发表。命名为 Pascal 是为了纪念法国数学家 B. Pascal,他是世界上第一台机械式加法器的创造者。国际标准化组织(ISO)在对 Pascal 语言作了一些修改后,将它定为 ISO 标准 Pascal 语言。Pascal 的各种实现系统与 ISO 标准 Pascal 相比,一般都作了适当的修改和补充,并且和标准 Pascal 兼容。这样,按照标准 Pascal 写出的程序,在不同的实现系统上一般都能通过。所以本书以介绍标准 Pascal 为主。

Pascal 语言是在 ALGOL-60 的基础上发展起来的。它是一种结构化的程序设计语言。使用 Pascal 语言可以编写系统软件,也可以用于编写各种应用软件。它的功能强、编译程序简单,已成为最流行的高级语言之一。

Pascal 语言有以下几个主要特点:

(1)它是结构化的语言。Pascal 语言遵循结构化程序设计准则,提供了直接实现 3 种基本结构的语句,可以方便的写出结构化的程序,Pascal 语言强调的是程序的可靠性、易于验证性、结构的清晰性和实现的简化性。

(2)有丰富的数据类型。Pascal 语言提供了整型、实型、字符型、布尔型、枚举型、子界型,以及由以上类型数据构成的数组、集合、记录和文件类型。此外还提供了指针类型。Pascal 语言所提供的丰富的数据结构和它的结构化性质,使得 Pascal 可以用来描述各种不同的数据对象、复杂的算法、得到高质量的程序。

(3)能适用于数值运算和非数值运算。有些语言(如 Fortran)适用于数值运算,而有的语言(如 Cobol)则适用于管理领域。Pascal 的功能较强,能应用于各种领域。

(4)程序的书写格式自由。它不像 Fortran 和 Cobol 那样对程序的书写格式有严格的规定。Pascal 程序允许一行写多个语句,一个语句可以写在多行上,这样可使写出的程序格式优美、便于阅读。

(5)用 Pascal 语言描述算法简明、清晰、便于接受,适宜初学者学习,可以为进一步学习其他语言打下良好的基础。

由于以上特点,目前国内外大多数高等院校都把 Pascal 语言选作高级程序设计语言的教学内容。这能给学生以严格而良好的程序设计的基本训练。

1.2 Pascal 源程序结构

Pascal 程序是由字母、数字和特殊符号按照它的一套语法规则组成的。为了使初学者从一开始对 Pascal 程序的结构建立一个整体概念,先举一个简单的 Pascal 源程序例子:

Program area(input,output);——程序首部

```

const
    pi= 3.14159;
var
    s,r:real;
begin {calculate s}
    write('input r:');
    readln(r);
    s:= pi * sqr(r);
    writeln('s=',s);
end.

```

程序体

语句
部分

说明
部分

这是一个计算半径为 r 的圆面积的 Pascal 语言源程序。先从键盘输入半径 r ，经计算，由显示器输出面积 s 的值。程序中用大括号“{}”括起来的部分是用来为程序或程序中某些部分作注释用的，注释的内容可以根据需要书写，它给编程人员阅读程序提供方便，而对程序的执行不起任何作用。在程序的任何位置都可以插入一个或若干个注释行。

从上面简单程序中可以看出：一个 Pascal 源程序是由程序首部、分号、程序体、句号(.)组成。如图 1-1 所示：

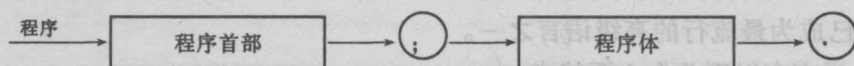


图 1-1

这样的图称为语法图。Pascal 语言的语法规则是用语法图来描述的。每一个语法图描述一个语法规则。图中最左边的文字“程序”是需要说明的对象；圆框内是 Pascal 程序中实际书写的符号；矩形框内是程序单元的名称，需要进一步说明。如上图，程序首部和程序体需要进一步说明；连线和箭头表示语法单元的单向连接关系，沿箭头方向把框内指出的内容顺序写下来，就是一个正确的程序。

注意：语法图只是规定了程序中各部分应书写的内容和顺序，而不是规定程序执行的步骤，不要将它和程序流程图相混淆。

1. 程序首部

程序首部是程序的开头部分，以保留字 program 为标志，任何一个 Pascal 程序都以此字开头。接着给出程序名，上例中程序名为 area。在圆括号内列出的 input 和 output 是程序参数，input 是系统预定义的输入文件，output 是系统预定义的输出文件，在微型机的标准运行环境中，分别表示键盘和显示器。程序首部的语法图如图 1-2 所示。



图 1-2

“标识符”的概念在下一节介绍，文件名和程序参数都属于“标识符”。图中的端圆框和图 1-1 中的圆框是同一类型，只是根据框内要求书写的内容多少而画成圆框或端圆框。对此语法图可以作如下理解：首先是保留字 program；接着是标识符；再下去有两个路径，一个

是直通下去,该语法部分结束;另一个是到左括号“(”,接着是标识符;再下去又有两个路径,一个是直接到右括号“)”,结束该部分;另一个是到逗号“,”,然后又到标识符,循环下去,直到右括号“)”,程序首部结束。由此可见,程序首部可以有圆括号括起来的参数表,其中可以有一个或用逗号分隔的多个参数;也可以没有用圆括号括起来的参数表。即下面的几种写法都是允许的:

```
program a
program b(output)
program c(input,output)
```

对以后用到的其他语法图可按同样的方法理解。

2. 程序体

程序体包括两部分:说明部分和语句部分。其语法图如图 1-3 所示。



图 1-3

可以看出,程序体可以不包括说明部分。

(1)说明部分

程序的说明部分用来描述程序中用到的数据的属性。上例中第二、三行是常量说明部分,以保留字 const 作标志,定义 pi 是常量 3.14159;第四、五行是变量说明部分,以保留字 var 作标志,定义 s,r 为实型(real)变量。

Pascal 语言规定,凡是程序中用到的标识符(标准标识符除外)都必须在程序说明部分进行定义(或称“说明”)。也就是说,不允许使用未给定义的标识符。程序说明部分又分为以下 5 个部分:(1)标号说明部分;(2)常量说明部分;(3)类型说明部分;(4)变量说明部分;(5)过程与函数说明部分。这 5 部分在说明部分中必须按上述顺序出现。说明部分语法图如图 1-4 所示。

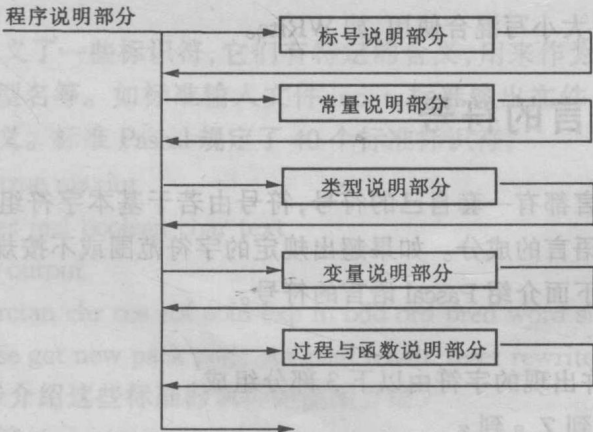


图 1-4

(2)语句部分

它的作用是通知计算机执行指定的操作。如果一个程序中不写语句部分,在程序运行

时计算机什么工作也不做。语句部分是 Pascal 程序的核心部分。
语句部分以保留字“begin”开始,以“end”结束。中间包括一系列语句,语句之间用分号分隔。每个语句执行一定的操作,完成一定的功能。语句部分语法图如图 1-5 所示。

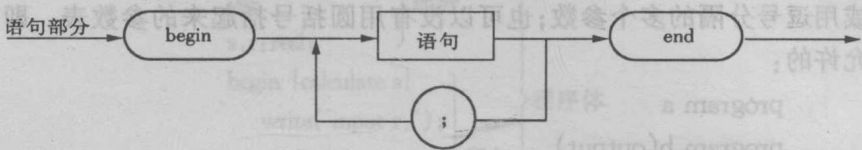


图 1-5

编程时应注意如下几点:

- (1) “begin”和“end”不是语句,它们只作为语句部分的起止标志。
- (2) 在最后一个语句与“end”之间可以不加分号,这是因为“end”不是语句。
- (3) 在语句部分的“end”后必须加句号“.”,表示整个程序结束。

如上例中“begin”和“end”括起来的语句部分包括 4 个语句。其功能示意如下:

begin

输出提示字符串‘input:’;
输入半径的值,存入变量 r 中;
计算面积值,存入变量 s 中;
输出字符串‘s=’和面积 s 的值

end.

整个程序以“end.”结束。

程序的说明部分是描述数据属性的,而语句部分则是描述对数据进行的操作。这两部分是相互联系,缺一不可的。

Pascal 程序的书写方法比较自由。允许一行中写几个语句,也允许一个语句分几行写。但是,应注意使自己写出的程序结构清晰、易读性好。

此外,程序中除了用引号括起来的字符串以外,大写和小写是等价的。例如 WRITE 等价于 write。甚至可以大小写混合使用,如 WRite。

1.3 Pascal 语言的符号

每种程序设计语言都有一套自己的符号,符号由若干基本字符组成。用这些符号按一定的语法规则构成该语言的成分。如果超出规定的字符范围或不按规则书写,都视为非法,编译系统不能识别。下面介绍 Pascal 语言的符号。

1. 基本字符

Pascal 程序中允许出现的字符由以下 3 部分组成:

- (1) 英文字母: A 到 Z, a 到 z
- (2) 数字: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9
- (3) 特殊字符: + - * / ^ = < > () [] { } ' ; , . : 空格

在字符串里还可以出现实现系统能表示的任意字符,例如 ! " # \$ % 等。

2. 保留字

Pascal 规定了一批英文单词,赋予它们特定的含义,不允许被再定义,因此不能用作下节介绍的用户自定义标识符。Pascal 中一共有 35 个保留字,它们是:

and	array	begin	case	const
div	do	downto	else	end
file	for	function	goto	if
in	tabel	mod	nil	not
of	or	packed	procedure	program
record	repeat	set	then	to
type	until	var	while	with

前面已经介绍了作为程序标志的保留字 program,以及作为语句部分起止标志的保留字 begin 和 end。其他保留字的具体含义将在以后各章中逐步介绍。

3. 标识符

标识符是用来表示程序、过程、函数、常量、变量、类型和文件等名称的符号。如 1.2 节所举程序例子的 area(程序名)、pi(常量名)、s,r(变量名)都属于标识符。

Pascal 规定,标识符必须以字母开头,后面可以是字母和数字的任意组合。对标识符长度(所包含的字符个数)的限制视不同的 Pascal 系统而有所差异。标准 Pascal 系统规定最长为 8 个字符,如果标识符长度超过 8 个字符,只识别前 8 个字符。例如, AverageOfClass 和 AverageOfSchool 被视为是同一标识符。标识符语法图如图 1-6 所示。

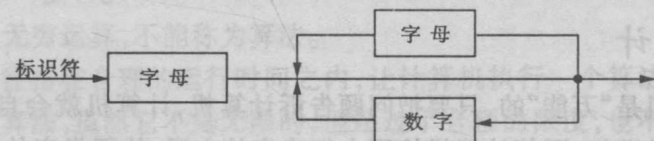


图 1-6

标识符分为标准标识符和用户定义的标识符两种。

(1) 标准标识符

Pascal 语言预先定义了一些标识符,它们有特定的含义,用来作为标准的函数名、过程名、常量名、文件名、类型名等。如标准输入文件 input、标准输出文件 output。在程序中可直接使用,不必事先定义。标准 Pascal 规定了 40 个标准标识符:

标准常量名: false true maxint

标准类型名: integer real boolean char text

标准文件名: input output

标准函数名: abs arctan chr cos eof eoln exp ln odd ord pred word sin sqr sqrt succ trunc

标准过程名: dispose get new pack page put read readln treset rewrite unpack write writeln

在以后各章将逐步介绍这些标准标识符的使用方法。

(2) 用户定义标识符

用户定义的标识符是用户按标识符定义规则自己定义的。例如定义一个变量名或常量名。在定义标识符时应注意“见名知义”的原则,即用能说明某种含义的英文单词来作为名字。用户定义的标识符必须事先定义或说明,才能在程序中使用。

应用标识符时应注意如下几点:

- ①保留字不允许作用户定义的标识符使用,否则按出错处理。
- ②用户定义的标识符最好不要选用标准标识符。但是语法允许将标准标识符作为一般用户定义的标识符使用。例如 `sin` 本来是正弦函数名,如果在程序的说明部分将 `sin` 定义为常量:

```
const sin = 10;
```

则在程序中 `sin` 不代表正弦函数,而代表常数 10。这是标准标识符与保留字的不同之处。尽管标准标识符可以被赋予新的含义,但这样做容易引起混乱,应该尽量避免,以保持程序清晰易读。

下列用户定义各标识符是合法的:

```
name sum year grade num1
```

下列用户定义各标识符是不合法的:

`ab x` 在标识符中不能有空格;

`begin` 保留字不能标识符;

`5x` 标识符不能以数字开头;

`ex%` 标识符不能有既非字母又非数字的字符。

1.4 程序设计与算法

1.4.1 程序设计

有人认为计算机是“万能”的,只要把问题告诉计算机,计算机就会自动完成一切,给出结果。其实这是一种误解,要使计算机按照人们确定的方案,执行指定的操作步骤,必须先编写相应的让计算机执行的程序。程序设计是指从拿到任务分析问题开始,到确定算法、编写程序、上机调试、直到分析结果、整理资料的全过程。如图 1-7 所示。

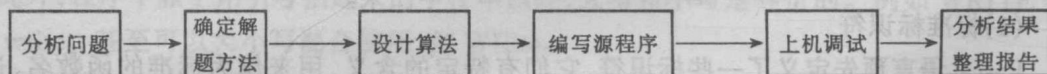


图 1-7

在学习程序设计时,既要掌握所采用的某一种计算机语言,如 Pascal 语言,更要掌握解题的方法和步骤,即设计算法,这是程序设计的关键。语言只是一个工具,只懂得语言的规则并不保证能编制出有效的、高质量的程序。如果算法是正确的,将它转换为任何一种高级语言程序并不困难。程序设计人员水平的高低在于他们能否设计出好的算法。

1.4.2 算法的概念

人们希望计算机按自己的愿望进行某项计算或者数据处理任务。但怎样才能使计算机按照一定的步骤进行工作,最后达到预定的目的呢?在多数计算机系统中,要求使用者不仅要先告诉计算机“做什么”,还要告诉计算机“怎么做”。例如,让计算机解一个一元二次方程:

$$ax^2 + bx + c = 0$$

计算机是解不出来的,必须确定用什么方法去解这个方程,以及一步一步怎么解。人们做任