

最新计算机等级考试教程

【二级】

新大纲，新考点，新法宝
——助您顺利过关，取证

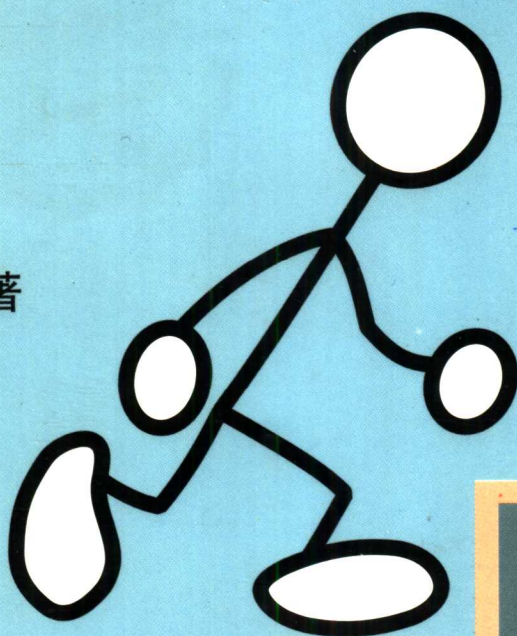
PASCAL

语言

程序设计



李光琳 等 编著



机械工业出版社
China Machine Press

最新计算机等级考试教程

(二级)

PASCAL 语言程序设计

李光琳 等编著



机械工业出版社
China Machine Press

本书是根据教育部考试中心 1998 年新颁布的《全国计算机等级考试二级考试大纲》中“PASCAL 语言程序设计考试内容”编写的,针对大纲的要求,系统地讲解了考生应该掌握的内容。

全书以 PASCAL 语言为工具,介绍了结构化程序设计的基本思想和方法。帮助读者准确理解有关重要概念,掌握规范化的程序设计技术,培养良好的程序设计风格和习惯;并通过各种类型的大量模拟试题,立足根本,引导读者掌握审题技术,揭示解题思路,着力帮助读者提高解题能力。

因此,本书可作为准备参加计算机等级考试人员的培训和自学教材,也可作为非计算机专业 PASCAL 程序设计教学用书。

图书在版编目(CIP)数据

PASCAL 语言程序设计/李光琳编著. —北京:机械工业出版社,1999.9

(最新计算机等级考试教程:二级)

ISBN 7-111-07299-5

I. P… II. 李… 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 20830 号

出 版 人:马九荣(北京市百万庄大街 22 号 邮政编码 100037)

责任编辑:郭东青

北京牛山世兴印刷厂印刷·新华书店北京发行所发行

1999 年 9 月第 1 版第 1 次印刷

787mm×1092mm^{1/16}·25.5 印张

印数:0 001—5 000 册

定价:38.00 元

凡购本书如有缺页、倒页、脱页,由本社发行部调换

最新计算机等级考试教程编委会

主任委员:史济民

副主任委员:李光琳 周启海

秘书长:白晓毅

副秘书长:王 松

委 员:	于中华	冉蜀阳	揭金良	廖 果
	黄桂钦	李 薇	董 恒	高元鹏
	赵进延	李自力	杨祥茂	丁 庆
	陈 康	文 进	张 维	黄晓榕

序 言

计算机技术是近 20 年来发展最迅猛、应用最广泛的现代科学技术之一,也是 20 世纪信息技术产业的关键核心技术,21 世纪知识经济时代的重要基础技术。在当今社会中,人人都应当掌握这一基本的生存技能与必备的文化素质。因此,在全民中普及推广和开发应用计算机技术,对国家的生存和发展,无疑具有不可低估的社会经济价值和历史进步意义。

当今在中国开展的“全国计算机等级考试”,对推动全社会学习计算机技术具有不容轻视的影响。计算机技术的开发与应用,说到底,无非是在普及计算机基础知识和基本操作的前提下,进行计算机程序的设计与运用。因此,“如何使学生轻松、愉快地掌握计算机基础知识和基本操作以及学好计算机程序设计;怎样让教师愉快、轻松地教好计算机基础知识和计算机程序设计”,自然地成为“全国计算机等级考试”教材要实现的重要目标,是值得探索与实践的重大课题。

为此,我们根据教育部考试中心 1998 年新颁布的“全国计算机等级考试大纲”,组织富有教学经验的计算机基础课教师编写了这一套系列教材,作为全国计算机等级考试的教学(包括自学)用书。本系列教材深入浅出地讲解了大纲规定的一、二、三级的应知应会内容;同时,编者通过各种类型的模拟试题,立足根本,引导读者掌握审题技术,揭示解题思路,着力于帮助读者逐步提高解题能力和应试技巧。希望读者在学习这些教材时,不要死记硬背,多从理解上下功夫,以达到融会贯通。我们衷心希望这些教材能成为广大考生和计算机爱好者的良师益友,在教学中收到事半功倍的效果。

史济民

1999 年 2 月

前 言

PASCAL 语言,是瑞士 N·Wirth 教授在 1971 年提出来的。它的命名是为了纪念法国数学家、物理学家、哲学家 PASCAL。

PASCAL 语言是第一个非常成功地融合了结构化程序设计技术的现代程序设计语言,也是学习结构化程序设计方法的良好工具。因此,它是程序设计语言发展史上的一个里程碑,对现代程序设计语言的发展,具有深刻影响。后来出现的 C、ADA、JAVA 等语言,大致都以 PASCAL 为基础或蓝本。

编者长期从事 PASCAL 程序设计、数据结构等课程的教学和教材研究,并广泛参与全国和四川省的计算机等级考试、软件专业技术资格和水平考试方面的业务活动。本书是多年来实践经验的产物之一。

本书根据教育部考试中心 1998 年新颁布的《全国计算机等级考试二级考试大纲》中“PASCAL 语言程序设计考试内容”的要求,全面、系统地讲解了考生应该掌握的内容。

在编写的指导思想,力图引导读者把牢固掌握 PASCAL 程序设计的基础知识、基本理论与学会比较系统、完整的审题思考方法和解题基本技能有机地结合起来,立足根本,帮助读者提高解题能力和应试技巧。

全书的内容和结构,从目录上就可大致看出来,不再赘述。只有一点需要说明,为了方便在校生和自修人员、初学者和希望进一步复习、深造等各层次、多方面读者的需要,我们把“教材”和“复习、考试指导”两项内容统一在一起,配合更紧凑密切,采用一本加“半本”的新模式,不再分成两本书。因此,在全书的每一节,都遴选了类型多样、针对性强、富有启发性、难易程度适中、解题思路清晰、透彻并且数量较多的例题;每一章的最后一节,又安排了足够数量的模拟试题。但需要指出,并非要求所有读者都完成所有的模拟试题。每位读者,可根据自己的学习要求和时间、精力、基础等方面的实际情况,选作其中的一部分就可以了。

总之,本书所追求的目标是:逼真、丰富的试题,切实、新颖的解题指导 $1+0.5>2$ 的使用效果。但是以上这些想法,毕竟只是编者的良好愿望,实效到底如何,还得由读者评说。因此编者殷切希望得到同行专家和广大读者的批评意见和反馈信息。

本书第 1~4 章和第 8~11 章,分别由伍良富、向孟光、何明儒等编写,李光琳编写了其余各章并负责全书的统稿。丛书编委会的成员,对本书的编写提供了指导性的意见。其中的一部分成员,以及张平治、文德生、廖春忠、黄正东等同志,为本书提供了数量不等的例题和模拟试题。我们都殷切希望得到同行专家和广大读者对本书的批评意见和反馈信息。

编者

1999 年 7 月

目 录

序言	
前言	
第 1 章 PASCAL 程序的构成	1
1.1 PASCAL 源程序的结构	1
1.2 PASCAL 的字符和符号	5
1.2.1 PASCAL 语言的字符集	6
1.2.2 PASCAL 语言的符号	6
1.2.3 PASCAL 语言的特定符号	7
1.3 PASCAL 的数	8
1.3.1 整数	8
1.3.2 实数有两种表示格式	9
1.4 标识符	11
1.5 字符串	13
1.6 注解	15
1.7 模拟试题	16
1.7.1 单项选择题	16
1.7.2 多项选择题	16
1.7.3 填空题	17
1.7.4 问答题	17
第 2 章 简单数据类型、简单语句和简单程序设计	18
2.1 PASCAL 的数据类型	18
2.2 标准数据类型	19
2.2.1 整数类型	19
2.2.2 实数类型	21
2.2.3 字符类型	22
2.2.4 布尔类型	24
2.3 常量和常量说明	27
2.3.1 常量说明区的格式	27
2.3.2 一种常用的语法表示方法	28
2.3.3 常量说明的意义	29
2.4 变量和变量说明	31
2.4.1 变量说明区的格式	31
2.4.2 变量说明的意义	31
2.5 标准函数	33
2.5.1 标准函数的意义	33
2.5.2 PASCAL 标准函数及其分类	33
2.5.3 标准函数的自变量和函数值的类型	34
2.5.4 顺序类型	34
2.6 表达式	37
2.7 PASCAL 的语句及其分类	40
2.8 赋值语句	41
2.8.1 赋值语句的格式	41
2.8.2 赋值语句的语义	41
2.8.3 赋值语句的执行步骤	41
2.8.4 赋值相容性	42
2.9 输入与读语句	46
2.9.1 赋值语句功能的局限性	46
2.9.2 读语句的格式	46
2.9.3 读语句的语义(功能)	46
2.10 输出与写语句	53
2.10.1 写语句的格式	53
2.10.2 写语句的功能	53
2.10.3 写行语句的功能	56
2.11 顺序结构和简单的源程序	59
2.11.1 简单的源程序的意义	59
2.11.2 简单的源程序举例	59
2.12 模拟试题	64
2.12.1 单项选择题	64
2.12.2 多项选择题	68
2.12.3 填空题	69
2.12.4 判断(是非)题	69
2.12.5 编程题	70
第 3 章 条件语句与选择结构	71
3.1 PASCAL 的控制结构	71
3.2 分层缩进、对齐的书写格式	72
3.3 IF(如果)语句和空语句	74
3.3.1 引进如果语句的必要性	74
3.3.2 IF 语句的语法	74
3.3.3 语义	74
3.3.4 流程图(框图)	76
3.3.5 简单语句与构造语句	77
3.3.6 多重 IF 语句(IF 语句	

的嵌套).....	77	4.4.1 REPEAT 语句的语法	125
3.3.7 空语句和复合语句	78	4.4.2 REPEAT 语句的语义	125
3.4 CASE(多情况)语句	83	4.4.3 三种重复性语句的比较.....	128
3.4.1 引进 CASE 语句的必要性	83	4.5 模拟试题.....	131
3.4.2 推广工作从三方面进行	84	4.5.1 多项选择题.....	131
3.4.3 CASE 语句的语法	85	4.5.2 单项选择题.....	131
3.4.4 CASE 语句的语义	85	4.5.3 填空题.....	139
3.4.5 带“否则项”的 CASE 语句	89	4.5.4 判断题.....	140
3.5 GOTO(转移)语句和标号说明	90	4.5.5 改错题.....	142
3.5.1 引例	90	4.5.6 程序填空题.....	143
3.5.2 GOTO 语句的语义(功能)	91	4.5.7 编程题.....	144
3.5.3 语法	91	第 5 章 算法和结构化程序设计的	
3.5.4 对引例中程序的分析	91	初步知识	146
3.5.5 语法对 GOTO 语句使用		5.1 算法.....	146
范围的限制	92	5.2 结构化程序和结构化程序设计.....	151
3.5.6 情况标号与语句标号二者		5.3 良好的程序设计风格和习惯.....	154
的区别	92	5.3.1 在设计阶段注意合理分解.....	154
3.5.7 GOTO 语句以空语句作		5.3.2 编码阶段.....	155
为程序的转移目标	93	5.4 一个实例.....	158
3.5.8 无标号语句和带标号语句	94	5.5 几个思考题.....	161
3.5.9 慎重使用 GOTO 语句	94	第 6 章 子界(域)类型和数组类型	162
3.6 模拟试题	96	6.1 子界(域)类型.....	162
3.6.1 单项选择题	96	6.1.1 数据类型的分类.....	162
3.6.2 填空题.....	101	6.1.2 PASCAL 程序的类型说	
3.6.3 判断题.....	102	明区.....	163
3.6.4 改错题.....	103	6.1.3 子界(域)类型说明的语法.....	163
3.6.5 问答题.....	104	6.1.4 子界类型的语义.....	164
3.6.6 程序填空题.....	105	6.1.5 类型说明与变量说明合并	
3.6.7 编程题.....	105	进行.....	164
第 4 章 重复性语句与循环结构	107	6.1.6 引入子界类型的主要优点.....	164
4.1 FOR(计数循环)语句	107	6.2 数组类型.....	166
4.1.1 FOR 语句的语法	107	6.2.1 引入数组的必要性.....	166
4.1.2 FOR 语句的语义	108	6.2.2 数组类型说明的语法.....	167
4.1.3 算法的自然语言描述方式.....	114	6.2.3 数组类型说明的语义.....	167
4.2 多重循环语句.....	116	6.2.4 再谈顺序类型的性质.....	174
4.2.1 多重循环语句的意义.....	116	6.2.5 引入数组类型的好处.....	176
4.2.2 多重循环的特性.....	117	6.2.6 简单数据类型的特征,简	
4.3 WHILE(当循环)语句	119	单类型与构造类型的联系	
4.3.1 引入当循环的必要性.....	119	与区别.....	176
4.3.2 WHILE 语句的语法	120	6.3 使用数组和循环结构的一些	
4.3.3 WHILE 语句的语义	120	例子.....	186
4.3.4 条件循环与计数循环.....	121	6.4 模拟试题.....	198
4.4 REPEAT(直到循环)语句	125		

6.4.1 单项选择题·····	198	7.6.3 递归程序分析、阅读示例·····	242
6.4.2 填空题·····	203	7.6.4 超前引用·····	244
6.4.3 改错题(请分别改正下列 各程序段中的错误)·····	204	7.7 模拟试题·····	246
6.4.4 程序填空题·····	205	7.7.1 单项选择题·····	246
6.4.5 编程题·····	206	7.7.2 多项选择题·····	252
第7章 过程(函数)与模块结构和 层次结构·····	208	7.7.3 判断题·····	253
7.1 无参过程,程序与它的主程序、 子程序·····	208	7.7.4 填空题·····	253
7.1.1 子目录法·····	208	7.7.5 程序填空题·····	255
7.1.2 无参过程·····	208	7.7.6 编程题·····	256
7.1.3 程序与其主程序、子程序·····	210	第8章 紧缩数组与串变量·····	257
7.2 有参过程与数值参数和变量 参数·····	211	8.1 数据类型之间的关系·····	257
7.2.1 有参过程·····	211	8.1.1 数据类型的重要性·····	257
7.2.2 数值参数与变量参数的 区别·····	212	8.1.2 数据类型的概念·····	257
7.2.3 有参过程说明的语法·····	215	8.1.3 数据类型的分类·····	258
7.2.4 过程语句的语法·····	219	8.1.4 数据类型之间的关系及其 在程序设计中的应用·····	258
7.2.5 过程体与程序体·····	219	8.1.5 引入众多数据类型的好处·····	261
7.2.6 过程结构与程序结构的 区别·····	219	8.2 紧缩数组与串变量的关系·····	262
7.2.7 什么情况下使用过程·····	220	8.2.1 紧缩数组·····	262
7.3 函数说明与函数调用·····	220	8.2.2 串变量·····	265
7.3.1 函数的意义·····	220	8.3 模拟试题·····	272
7.3.2 函数说明的语法·····	220	8.3.1 单项选择题·····	272
7.3.3 函数的调用·····	220	8.3.2 程序填空题·····	273
7.3.4 函数与过程的主要区别·····	220	第9章 枚举类型·····	275
7.3.5 标准函数、标准过程与用 户自定义函数和用户自 定义过程·····	225	9.1 枚举类型的语法·····	275
7.4 标识符和标号的作用域(辖域)·····	226	9.2 枚举类型的语义·····	276
7.4.1 标识符在程序中的两种 出现·····	226	9.3 模拟试题·····	280
7.4.2 几个概念·····	226	9.3.1 单项选择题·····	280
7.4.3 作用域概念对程序设计的 意义·····	228	9.3.2 填空题·····	282
7.4.4 实例·····	229	9.3.3 判断题·····	283
7.5 非局部量与子程序的副作用·····	234	9.3.4 程序填空题·····	283
7.6 递归·····	237	9.3.5 编程题·····	285
7.6.1 递归过程(函数)的定义·····	237	第10章 记录类型·····	286
7.6.2 递归函数(过程)设计举例·····	237	10.1 记录类型的语义·····	286
		10.2 记录类型的语法·····	286
		10.3 记录数组·····	288
		10.4 开域语句·····	289
		10.5 带变体的记录·····	293
		10.6 模拟试题·····	297
		10.6.1 单项选择题·····	297
		10.6.2 填空题·····	299
		10.6.3 程序填空题·····	300

10.6.4 编程题	300	12.6.4 改错题	355
第 11 章 集合类型	302	12.6.5 程序填空题	355
11.1 集合类型的语义	302	12.6.6 编程题	357
11.2 集合类型的语法	302	第 13 章 指针类型与动态数据	
11.3 集合类型的运算	304	结构	359
11.4 集合的输入、输出	307	13.1 指针类型	359
11.5 再谈集合类型的基类型	308	13.1.1 静态变量满足不了程序设计	
11.6 集合类型的相容性	310	领域某些方面的需要	359
11.7 集合类型的优、缺点	311	13.1.2 指针的意义	360
11.8 模拟试题	313	13.1.3 指针类型和指针变量	362
11.8.1 单项选择题	313	13.1.4 注意区分指针变量与指	
11.8.2 填空题	315	针的目标变量	364
11.8.3 程序填空题	317	13.2 动态变量的生成(建立)与废	
11.8.4 编程题	319	弃释放(撤消)	366
第 12 章 文件类型	320	13.2.1 标准过程 new	366
12.1 文件的概念	320	13.2.2 递归数据结构	367
12.2 FILE 类型文件	321	13.2.3 目标变量的性质和运算	368
12.2.1 语法	321	13.2.4 标准过程 dispose	370
12.2.2 文件的特性	323	13.3 指针和动态变量的应用	370
12.2.3 五个常用的名词	323	13.3.1 不含指针的动态变量	
12.2.4 文件的操作	325	应用举例	370
12.3 文本文件	331	13.3.2 链表的分类	374
12.3.1 文本文件的重要性	331	13.3.3 向前链表	374
12.3.2 文本文件的结构特点	331	13.3.4 双向环形链表举例	380
12.3.3 文本文件类型是 PASCAL		13.4 模拟试题	381
的标准类型	332	13.4.1 多项选择题	381
12.4 FILE 类型文件与 text 类型		13.4.2 单项选择题	382
文件的区别	337	13.4.3 填空题	386
12.5 TURBO PASCAL 的文件		13.4.4 判断题	387
类型	338	13.4.5 改错题	388
12.5.1 文件变量与实际文件		13.4.6 程序填空题	389
之间联系的建立与联		13.4.7 编程题	391
系的解除	338	附录 A PASCAL 关键字(保留	
12.5.2 随机(读、写)文件	344	关键字)	392
12.5.3 TURBO PASCAL 中的		附录 B PASCAL 保留字(标准	
文本文件	347	标识符)	393
12.6 模拟试题	350	附录 C PASCAL 运算符一览表	394
12.6.1 单项选择题	350	附录 D ASCII 码表	395
12.6.2 填空题	353		
12.6.3 判断题	354		

第 1 章 PASCAL 程序的构成

一套住房,从“自顶向下”的宏观角度观察,由卧室、客厅、厨房、餐厅、卫生间等“功能部件”,按照适用、美观的设计规则组成;从“自底向上”的微观角度观察,则又由钢筋、水泥、砖头、瓦片等建筑材料,按照坚固、节约的建筑规则组成。

对一个 PASCAL 源程序,我们也可以通过“自顶向下”的宏观和“自底向上”的微观两个角度去观察和认识掌握它。

1.1 PASCAL 源程序的结构

这一节,我们先从宏观角度观察,分析一个 PASCAL 源程序应该由哪些部分组成。

一本书,开头总有个封面,罗列出书名、作者、出版社之类的信息,以便对此书的主要特征,进行简明的概括描述。例如,当你看到《PASCAL 语言程序设计》的书名,你就应该明白,这本书的内容是讲解用 PASCAL 语言进行程序设计的有关知识,譬如,它不会去讲述化工知识,更不至于把它误认成是一本小说。一个 PASCAL 源程序,自然也应该有个类似的导引。

紧接在导引之后,就该进入程序的实质部分。一个程序,总要完成一定的数据处理任务。于是就涉及被处理的数据,以及对有关数据进行怎样的处理(操作)两个方面。以求 2 的平方根 y_1 为例,或更一般地,以求任意给定的正实数 x 的平主根 y_2 为例,我们知道,在数学中,前者可表示为

$$y_1 = \sqrt{2}$$

而后者,可表示为

$$y_2 = \sqrt{x}$$

其中的 2 或 x 是操作对象,而 $\sqrt{\quad}$ 是对操作对象所应进行的操作。我们所关心的“平方根的值”,是由上述操作对象和操作两个因素所决定的。

可见,一个程序的实质部分,应该分别对程序的操作对象(数据)的属性和对数据应进行的操作两个方面分别进行描述。而在实质部分之前加上导引部分,从功能角度看,一个程序的组成就应该如表 1-1 所示。

表 1-1

程序	首部:是描述一个程序主要特征的导引	
	程序体(程序的实质部分) <table><tr><td>说明部分:描述将要被操作的数据的属性</td></tr><tr><td>执行部分:描述对有关数据所应进行的操作</td></tr></table>	说明部分:描述将要被操作的数据的属性
说明部分:描述将要被操作的数据的属性		
执行部分:描述对有关数据所应进行的操作		

例 1.1.1 输出 $\sqrt{2}$ 的近似值的 PASCAL 程序。

```
PROGRAM squarerootoftwo(output);    {输出 2 的平方根}
BEGIN
    write(sqrt(2))
END.
```

例 1.1.2 输出任意给定的正实数 x 的平方根的近似值 y_2 的 PASCAL 程序。

```
PROGRAM squareroot (input,output);
{output square roor of x}
VAR
  x,y2:real;
```

```
BEGIN
  read(x);
  y2:=sqrt(x);
  write('y2=',y2){y2 是 x 的平方根}
END.
```

例 1.1.1 和例 1.1.2 中的程序虽然很简单,但它们都是结构完整的 PASCAL 程序。前者,通过上机编译和运行,就输出 2 的平方根的近似值;而后者经编译之后,在运行时,只要输入一个正实数,程序就会计算并输出此正实数平方根的近似值。

通过以上例子,我们来介绍 PASCAL 源程序的组成。

一个 PASCAL 源程序的组成格式为:

程序首部(又称程序头)

程序体(又称分程序) { 程序说明部分
程序执行部分(又称语句部分)

1. 程序首部

程序首部是程序的导引,它依次包含以下三个项目:

1) 程序起始符 PROGRAM。每个 PASCAL 程序,总是必须以 PROGRAM 开头,不能例外。
2) 程序名。例 1.1.1 中的 squarerootoftwo 和例 1.1.2 中的 squareroot 都是程序名。它是由程序员为程序所取的名称。在书写程序时,应该注意在程序起始符与程序名之间必须留适当的空格。

3) 程序参数表。程序参数表放在一对圆括号之中,跟在程序名后面,程序名与程序参数表之间的空格可有可无。圆括号里的每个参数,表示一个输入或输出的文件名,各参数之间用逗号隔开。参数的个数,根据每个程序的实际需要确定,譬如例 1.1.1 中只有一个 output,而例 1.1.2 中却有 input 和 output 两个参数。input 和 output 是系统预定义的文件名,input 通常指输入键盘,而 output 通常指终端显示器。由于例 1.1.1 程序中,不需要输入,因此程序参数表中不出现 input。对于其它程序参数,等后面的“文件”那一章中再叙述。

2. 程序首部结束符

用一个分号“;”表示程序首部结束。但这个分号已不是程序首部的成分。

注意 1 标准 PASCAL 与 PASCAL 的各种实现。

国际标准化组织 ISO(International Standards Organization)公布的“标准 PASCAL 语言”的版本,作为大家共同遵循的标准,称为“标准 PASCAL”。一个 PASCAL 程序,要它在某一计算机上运行,必须以这一计算机上安装有 PASCAL 编译系统为前提条件。而 PASCAL 的编译系统有若干种,每一种 PASCAL 编译系统,称为 PASCAL 的一种实现。目前,在微型机上实现的有 Turbo PASCAL、MS PASCAL、UCSD PASCAL、ProPASCAL 等多种。一方面,“标准 PASCAL”作为各种实现共同遵循的标准,大体上满足了“标准”的规定;但是另一方面,几乎所有的实现都对标准版本有所取舍。所谓取舍,是指扩充或减少了某些功能,放宽或加强了某些限制等。而且同一种实现的不同版本,仍然存在差异。当然随着技术的进步,总的说来是版本越高,功能越强,限制越少。

注意 2 Turbo PASCAL。

Turbo PASCAL 是美国 Borland 公司推出的 PASCAL 编译系统。由于它具有遵守标准

最好、快速有效、功能齐全、系统性能最优等显示特点,几乎成了微机 PASCAL 的规范,拥有最广泛的用户。

作为一本教材,本书以标准 PASCAL 为基础进行讲述,但在涉及到与标准不同的实现之处,我们以 Turbo PASCAL 为例。

注意 3 Turbo PASCAL 的汉化情况。

Turbo PASCAL 1.0 版自 1983 年问世之后,1984 年就被介绍到中国。1985 年起,江苏省计算技术研究所等单位相继推出了 Turbo PASCAL 3.01A 等版的汉化版,为我国用户提供了方便。但从 4.0 版开始的各版本,引入了多窗口、多色彩、下拉式“菜单”等技术,在风格上有了根本改观,给汉化工作带来很大困难,加之 Turbo PASCAL 版本更新速度极快,7.0 版也早已推到软件市场,汉化工作远远跟不上发展。好在 Windows 操作系统已经采用“内核汉化”技术,不难使 Turbo PASCAL 开发的程序显示并接受汉字。作为教学实习,建议选用 90 年的 6.0 版或更低的版本即可。因为 7.0 版加进了汇编语言的内容,对初学者而言,显得过于繁杂。

注意 4 现在回到对程序首部格式的讨论。按照标准 PASCAL 的规定,程序首部的三个项目必须依次出现,缺一不可。程序中所要用到的文件,也必须一一罗列在程序参数表中,缺一不可。但 Turbo PASCAL 则对程序参数放宽要求,即使程序中用到的文件,在程序参数表中可以列出,也可以不列出,甚至整个程序参数表可被省略不写(当然写出来也不算错)。当然,不论哪种版本,程序首部紧后的分号是不可少的。

3. 程序说明部分

程序说明部分的功能是用以说明程序中出现的标识符,也就是对程序中所用到的每个标识符,对其性质、范围等特性进行定义。程序说明部分,可以包含五个说明区,它们依次是:

- 1) 标号说明区,用“标号说明区起始符”LABEL 作开始标志。
- 2) 常量说明区,用“常量说明区起始符”CONST 作开始标志。
- 3) 类型说明区,用“类型说明区”起始符 TYPE 作开始标志。
- 4) 变量说明区,用“变量说明区”起始符 VAR 作开始标志。
- 5) 过程和函数说明区,用“过程说明起始符 PROCEDURE”或“函数说明起始符”FUNCTION 作开始标志。

注意 1 对标识符进行“说明”,也就是对标识符的性质、范围等特征进行“定义”。因此在教材的叙述中,往往把“说明”与“定义”等同使用,譬如“变量说明区”,也可以表述为“变量定义区”等。

注意 2 “部分”、“区”、“段”等名词,没有严格的定义或约定。因此,不同教材,对程序说明部分有“包含标号说明部分等五个部分”、“包含标号说明区等五个区”、“包含标号说明段等五个段”等不同表述。有少数教材还称“节”,但从上、下文都不难理解,这种不统一性不致于使读者造成学习上的困难。

注意 3 程序说明部分和说明区都是任选项。所谓“任选项”的意思是:程序中有实

际需要就出现,不需要就不出现。譬如例 1.1.1 中的程序,不需要任何说明,因此五个说明区(即整个说明部分)都缺省(不出现);而例 1.1.2 中的程序,因为要用到变量 x 和 $y2$,因此就在 VAR 打头的变量说明区中对变量 x 与 $y2$ 的特征进行定义,其它四个区都不需要出现。

注意 4 在程序中,每个说明区至多只能出现一次。譬如例 1.1.2 中的程序,用到两个变量 $x, y2$,只能放在一个变量说明区中进行说明。

注意 5 标准 PASCAL 规定,五个说明区出现的顺序不得改变(可以缺省其中的若干个)。譬如某个程序需要设置类型说明区和变量说明区,则必须类型说明区在前,变量说明区在后,顺序改变了就是语法错误。

但是某些 PASCAL 实现,则对标准的这一规定进行了修改。例如 Turbo PASCAL,直到 3.0 版,都按此标准实现,但在此之后的各版本,就取消了这一限制,而采用说明部分的“各个说明区顺序任意”。这对编程和上机带来某些方便,但却产生了大家很关心的一个问题:考试时该怎样答题?我们在此统一约定如下:如果题目没有特别提问某些版本,则指的就是标准 PASCAL。

4. 执行部分(又称语句部分)

由若干个语句所构成的语句序列(可以只有一个语句),用“语句括号”BEGIN 与 END “括起来”就构成执行部分。其语句序列中的各个语句,就是对数据所应进行的操作(处理)。各个语句,两两之间用分号“;”隔开,分号不是语句的组成部分,而作为语句分隔符。因为 BEGIN 和 END 都不是语句,所以 BEGIN 紧后面和 END 紧前面,都不需要分号。

5. 程序尾部

用一个点号“.”表示整个程序到此结束。

注意 1 程序首部和执行部分都是必选项,每一个程序都必不可少。

注意 2 说明部分和执行部分合在一起,统称程序体,也称为分程序,它就是一个程序的实体。

例 1.1.3 以例 1.1.2 中的程序为例,它的各部分划分如下所示:

```
PROGRAM squareroot (input,output);           程序首部(不包含;)
    {output square root of x}
VAR
    x,y2:real;                                }说明部分
BEGIN
    read (x);
    y2:=sqrt(x);
    write('y2=',y2)
    {y2 is square root of x}
END.                                           }程序体
                                           }执行部分(不包含.)
```

6. 复合语句

用语句括号 BEGIN 和 END“括起来”的一个语句序列,连同 BEGIN 和 END 所构成的

整体,在语法上称之为复合语句。因此,程序的执行部分,简单地就说就是一个复合语句。

7. 总结

综上所述,一个完整的 PASCAL 程序,严格地说,按语法规定,由程序首部、分号、程序体、点号所构成。但为避免叙述上的烦琐,通常就把分号归并入程序首部,点号归并入程序体;因此简单地说,一个完整的程序,由程序首部和程序体所构成。但是本书不采用这种不够严格的表述方式。

例 1.1.4 多项选择题(本题的备选答案中,有一个或多个正确答案,请选出其中所有的正确答案,并将其序号填写在括号中。多选、错选、漏选均不能得分)。

下列的四条叙述中,可分别组成一个完整的 PASCAL 程序的是()。

- A. 程序首部、分号、说明部分、执行部分、点号
- B. 程序首部、分号、程序体、点号
- C. 程序名、分号、说明部分、执行部分、点号
- D. 程序首部、分号、分程序

[解] A 与 B 均分别符合程序的组成格式;而程序名只是程序首部的一个必要组成部分,不能代替整个程序首部;因此 C 不合格;一个完整的程序,必须以点号结束标志,而 D 中无此标志。

所以答案为 A、B。

例 1.1.5 单项选择题(本题的备选答案中,有一个是正确的。请选出其中的正确答案,并将其序号填写在题末的括号中)。

在 PASCAL 源程序说明部分中,各说明区的排列顺序是()。

- A. 标号说明区、常量说明区、类型说明区、变量说明区、过程说明区、函数说明区
- B. 标号说明区、常量说明区、类型说明区、变量说明区、过程与函数说明区
- C. 标号说明区、常量说明区、类型与变量说明区、过程说明区、函数说明区
- D. 常量说明区、标号说明区、类型说明区、变量说明区、过程与函数说明区

[解] 答案为 B。

例 1.1.6 问答题(简明扼要地回答下列问题)

关于一个完整的 PASCAL 程序的构成,以下三种叙述,是否都是正确的?

- A. 由程序首部和程序体构成
- B. 由程序首部和分程序构成
- C. 由程序首部、说明部分和执行部分构成

[解] 程序体,又称为分程序,它由说明部分和执行部分组成。综上所述,以上 A、B、C 三种叙述,内容都是相同的。

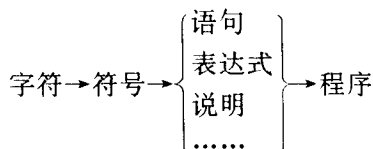
用其中的任何一种叙述以描述一个完整的程序的结构,只能说基本正确,但并不完全正确。不完整之处,表现在忽略了程序首部与程序体之间的分号和程序结束符点号。

1.2 PASCAL 的字符和符号

在汉语里,由字组成词,词组成句子,再由句子组成文章。即

字→词→句子→文章

相应地,在 PASCAL 语言里,有类似的关系



现在,我们又从与 1.1 节相反的微观角度,“自底向上”,按字符→符号→程序的线索去考察 PASCAL 程序的构成。下面的 1.2 节到 1.6 节的所有内容,就是进行这种考察。

一篇汉语的文章,是由汉语的各种词汇,按照汉语的语法规则组成。PASCAL 语言,自然也需要规定一套基本语法元素和一套严密的语法规则,以便按照语法规则将语法元素组织为语言的各种成分(如表达式、语句、说明等)和程序。PASCAL 的基本语法元素称为符号,而每个符号又由一个或若干个字符所组成。也就是说,字符是 PASCAL 语言的最小组成单位。

1.2.1 PASCAL 语言的字符集

在 PASCAL 语言中,要用到下列三类字符:

1) 字母(26 个) a b c d e f g h i j k l m n o p q r s t u v w x y z, PASCAL 语言中也可以使用大写字母,但除非在字符串(见 1.5 节)之内,否则大写与小写被认为是相同的字符。

2) 数字(10 个) 0 1 2 3 4 5 6 7 8 9

3) 其它(20 个) + - * / = < > . , ' : ; ^
() [] { } ~

这 56 个字符的全体,就是 PASCAL 字符集。但是由于不同的计算机系统,所能使用的字符要受到输入输出设备的限制,往往需要采取补救措施。例如有的计算机系统不具备方括号 '[' 和 ']', 就规定分别用 '(.' 和 '.)' 来代替;不具备花括号 '{' 和 '}', 就分别用 '(' 和 ')' 来代替;不具备 '^', 就用 ^ 或 @ 来代替,等等。

字符是构成程序的最小单位,它们的功能是组成符号。

1.2.2 PASCAL 语言的符号

什么叫程序? 对这个问题,我们在上一节中,从语义角度进行过说明。现在从语法角度来考察,一个 PASCAL 程序,是由具有一定含义的符号,按照 PASCAL 的语法规则组成的符号序列。因此,在 PASCAL 中,符号是具有一定含义的最小语法成分。语法成分,又称为语法元素或语法单位。

有的符号由一个字符组成(尤如汉语中的单音节词),更多的符号,则由一个以上的字符组成(尤如汉语中的多音节词)。

PASCAL 语言的符号又分为五种,如表 1-2 所示。

表 1-2

PASCAL 符号(五种)	特定符号
	数
	标识符
	字符串(简称串)
	注解(又称注释)

上列五种之中,特定符号的数量不多,用枚举法罗列介绍如下。其它四种符号,每种所包

含的数量很多,无法逐一枚举,只能使用语法(构成规则)进行介绍。

1.2.3 PASCAL 语言的特定符号

PASCAL 语言的特定符号(共 59 个),又分为非字特定符号和字特定符号两类,其结构及意义或用法,分别列举如下:

(1)非字特定符号(24 个,见表 1-3)

表 1-3

+	加	·	指针类型
-	减	.	点号
*	乘	,	分隔表中的项
/	除	'	撇号
<	小于	:	分隔变量名字与类型等等
<=	小于等于	;	分隔语句、说明和定义等等
=	等于	:=	赋值运算符
<>	不等于	..	隔开子域类型中的上、下界
>=	大于等于	(和)	用于参数表或表达式
>	大于	[和]	用于下标表或集合表达式
↑	用于文件缓冲变量、标识(动态)变量和	{ 和 }	用于注解

其中,由一个以上字符组成的符号,称为复合符号,如<=,<>,:=等都是复合符号。

(2)字特定符号(界限符 35 个,见表 1-4)

表 1-4

AND	逻辑与	MOD	取模运算符
ARRAY	数组类型定义符	NIL	指针常量
BEGIN	复合语句起始符	NOT	逻辑非
CASE	CASE 语句起始符	OF	用于数组、文件和集合类型的说明
CONST	常量说明区起始符	OR	逻辑非
DIV	整除运算符	PACKED	紧缩构造类型定义符
DO	用于 FOR 语句、WHILE 语句、WITH 语句	PROCEDURE	过程说明起始符
DOWNTO	用于 FOR 语句	PROGRAM	程序起始符
ELSE	用于 IF 语句	RECORD	记录类型定义符
END	复合语句结束符,记录类型结束符,CASE 语句结束符	REPEAT	REPEAT 语句起始符
FILE	文件类型定义符	SET	集合类型定义符
FOR	FOR 语句起始符	THEN	用于 IF 语句
FUNCTION	函数说明起始符	TO	用于 FOR 语句
GOTO	GOTO 语句起始符	TYPE	类型说明区起始符
IF	IF 语句起始符	UNTIL	用于 REPEAT 语句
IN	集合成员关系运算符	VAR	变量说明区起始符
LABEL	标号说明区起始符	WHILE	WHILE 语句起始符
		WITH	WITH 语句起始符