

PASCAL语言应用丛书之二

IBM-PC PASCAL 及其综合应用

何诚 罗晓奔 刘金玉 编著

同济大学出版社

TP317
13
• PASCAL 语言应用丛书之二

BM-PCPASCAL 及其综合应用

何 诚 罗晓奔 刘金玉 编著

同济大学出版社

内 容 提 要

本书是 PASCAL 程序语言应用丛书之一。它提供了一组用 PASCAL 语言编写的综合类应用程序,包括事务处理、文件处理、数学计算和机电工程计算等。全部程序都在 IBM-PC 机上得到验证。本书力求通过所示实例,介绍 PASCAL 语言的结构程序设计思想和该语言的 I/O 技术、文件技术、建立数据类型技术以及用户自定义过程与函数的技术。

责任编辑 胡兆民

封面设计 陈益平

IBM-PC PASCAL 及其综合应用

何 诚 罗晓奔 刘金玉 编著

同济大学出版社出版

(上海四平路1239号)

新华书店上海发行所发行

同济大学印刷厂印刷

开本: 850×1168 1/32 印张: 9.125字数 265 千字

1991年1月第1次印刷 1991年1月第1版

印数 1—4600 定价 4.40 元

ISBN 7-5608 0724-0/TP·60

丛 书 前 言

PASCAL 程序设计语言是根据结构化程序设计的原则研制出来的，它以高度结构化的程序设计语言 ALGOL-60 为模型演变发展而成，并且具有更丰富灵活的数据结构，简洁明了的语法规则，清晰的模块化结构，以及优美的程序设计风格和较高的运行效率。自从70年代瑞士的 N. Wirth 教授发表 PASCAL 语言以来，引起了各国计算机学者和工业界的广泛重视。

目前，PASCAL 语言不仅因为它特别适用于对各种数据结构和算法进行描述而受到高等学校计算机工作者的普遍接受，从而成为计算机教育中的第一教学语言。而且，由于它的各种优越特点和功能的不断扩充，也使它成为一种标准化的工业语言。它的小巧、简洁、精致和易学性也使之成为和 BASIC 语言一样的最流行、最普及的程序设计语言之一。实践证明 PASCAL 语言是富有生命力的。

为了普及和推广 PASCAL 程序设计语言，我们编写了“PASCAL 语言应用丛书”。丛书之一“PASCAL 题解分析和上机指南”已于1988年6月由同济大学出版社出版。这是一本配合“PASCAL 程序设计语言”（李启炎、宋秋杰编著）一书的辅助读物。由丛书之二开始，我们将陆续推出一系列介绍 PASCAL 语言在各个领域内应用的丛书，其中内容涉及到 PASCAL 语言在事务处理、工程计算、计算机图形学、数据库管理系统、…等各个方面的应用。

K. Jensen 和 N. Wirth 在“PASCAL USER MANUAL AND REPORT”一书中定义了标准 PASCAL 的内容。但在以后的各种实现中已对标准 PASCAL 进行了大量的扩充。这对用户在各个应用领域中使用 PASCAL 语言带来了极大的方便，也

使 PASCAL 语言更具生命力。在我们的应用丛书中所涉及到的有目前广泛应用、也比较成熟的 MS-PASCAL 和 UCSD-PASCAL。这二种版本的 PASCAL 语言都已在广泛应用的 IBM-PC 微型计算机上安装实现, 在其它微机, 如 APPLE、AST-256 以及各种 386 微型机上也都能运行工作。另外, PASCAL 语言版本的后起之秀 TURBO-PASCAL 本丛书也有专册介绍。PASCAL 语言的几种实现虽有不同, 但其基本要素是一致的, 程序间的移植并不困难。

在国内, PASCAL 语言业已作为第一教学语言被各高等院校所采纳。同时, 在各种应用领域 PASCAL 语言也已经得到广泛的应用。例如, 已出现了使用 PASCAL 语言研制的系统软件, 包括操作系统, 编译程序等; 以 PASCAL 语言作为宿主语言的数据库管理系统, 图形处理系统, 图象处理系统, 汉字处理系统, 实验数据处理软件, 工程计算、分析、设计软件, 人工智能机器人软件系统, …, 等等。我们希望这套丛书的出版能对国内计算机应用水平的提高起到“抛砖引玉”的作用。

本丛书各个分册的编排既照顾到应用领域的不同, 又考虑了 PASCAL 的不同实现。相互间既有一定的联系, 又是自成一体的独立篇章, 读者可根据需要和自己接触到的领域选用。书中不再系统介绍 PASCAL 语言本身, 而着重于应用实例的算法描述, 算法实现的过程和技巧的介绍, 并给出完整的 PASCAL 源程序和应用示例。对于大型应用实例注意按结构化程序设计的思想。“自顶而下”的方法逐步求精, 使读者能具体领略到 PASCAL 语言优美的程序风格。同时, 书中还注意结合应用实例将各种 PASCAL 不同实现的特点和扩充功能介绍给读者。所有应用实例的源程序都经过上机考核和实例运行, 读者可以在各自的应用领域内参考引用。

丛书分为〈IBM-PC PASCAL 及其综合应用〉; 〈IBM-PC PASCAL 的工程应用〉; 〈PASCAL 与计算机图形学〉; 〈TURBO PASCAL 及其应用〉; 〈UCSD PASCAL 及其应用〉〈PASCAL

与数据库管理系统》等册。它们都由具有丰富经验的，多年从事 PASCAL 语言教学工作与有关科研工作的专家教授撰写。丛书由同济大学李启炎主编。

这套丛书的主要对象为各大专院校师生，广大从事计算机应用的工程技术人员。希望它能受到广大读者的欢迎，更希望读者对书中的不妥之处批评指正。

序

本书系 PASCAL 语言应用丛书之二，主要内容为介绍目前广泛应用的 IBM-PC PASCAL 及其综合应用。IBM-PC PASCAL 编译器是由美国 Microsoft 软件公司开发的 MS-PASCAL，因为它在 IBM-PC 微型计算机上实现并随之而被日益广泛地使用，本书中称其为 IBM-PC PASCAL。

书中内容分为五章。第一章为 IBM-PC PASCAL 简介，比较详细地介绍了 IBM-PC PASCAL 的基本内容和扩充功能；第二章介绍了 IBM-PC PASCAL 的几个简单的综合计算实例，内容包括电工、机械、方差计算及曲线拟合等；第三章为文件处理，其内容有文本文件列表、合并；字符串的查找与替换、字符计数、字计数及交叉引用；以及二进制文件的转储等。第四章为事务处理，有银行存贷分析；投资分析；生产成本和生产方案分析以及经济预测等。

书中对各个应用的算法作了描述，对程序功能和程序中所使用到的各种常量、变量、过程和函数作了说明，使读者能很方便地阅读附于其后的应用实例源程序。这些源程序都经过实例运行，书中给出了运行的结果和说明。

程序设计是一门实践性很强的学问，只有通过不断的实践，编程水平才会不断提高。希望广大读者在自身的实践中能从本书有所得益，更希望广大读者能对书中存在的问题批评指正。

目 录

丛书前言

序

第一章 IBM-PC PASCAL简介

1.1 编程步骤.....	2
1.2 基本词汇.....	4
1.3 元语言.....	7
1.4 常 量.....	10
1.5 数据类型.....	12
1.6 变 量.....	22
1.7 表达式.....	25
1.8 语 句.....	28
1.9 过程和函数.....	34
1.10 可用的过程和函数.....	38
1.11 文件系统.....	47
1.12 编译对象.....	52

第二章 综合计算

2.1 电工计算.....	58
2.2 机械工程计算.....	84
2.3 方差计算与曲线拟合.....	130

第三章 文件处理

3.1 文本文件列表.....	154
3.2 文本文件合并.....	158
3.3 字符串查找.....	162
3.4 字符串替换.....	166
3.5 文本文件排序.....	172
3.6 字符计数.....	179

3.7	字计数	183
3.8	交叉引用表	192
3.9	文件转储	200
3.10	重复字符串的删除	206

第四章 事务处理

4.1	定期存款	215
4.2	分期存款	220
4.3	分期付款	227
4.4	投资回收	235
4.5	产品成本分析	239
4.6	生产方案比较	245
4.7	项目折旧	253
4.8	业务预测	259
4.9	电话号码簿管理	267

第一章 IBM-PC PASCAL简介

PASCAL 语言具有小巧严谨的语法规则，非常易于实现，特别是在微型计算机上实现。微型计算机自70年代推出以来越来越广泛地应用于各个领域。其中尤以 IBM-PC 机最为流行。目前，国内一般的企事业单位都在使用 IBM-PC 机或与其相兼容的微型计算机。国产长城、东海、浪潮等系列微机就是 IBM-PC 的兼容机种。随着 IBM-PC 机的普及应用，PC-PASCAL 的用户也越来越多，用它编制的应用程序和各种软件系统也日渐增多。本书中所有的应用程序都是选用 IBM-PC PASCAL 编写的。如前言中所述，由于各种 PASCAL 的实现中其基本要素是一致的，所以，要将这类程序移植于其他机种并不困难。

IBM-PC PASCAL 是美国 Microsoft 软件公司研制的 MS-PASCAL 在 IBM-PC 机上的实现。它扩充了标准 PASCAL 的功能，使之既符合国际标准化组织 ISO 的标准，又具有很多满足用户愿望的、灵活而方便的扩充功能。例如，高级数组结构，直接存取模式文件，以及分块编译的能力。这些功能弥补了标准 PASCAL 语言的不足，给大、中型软件的开发提供了必要的支撑环境。加上 PASCAL 语言本身的结构化特性，很适宜于表达算法的层次性数据结构，PASCAL 源程序的清晰结构简明易懂、便于调试验证等，使 PC-PASCAL 成为最受用户欢迎的程序设计语言之一。编著者曾经使用 PC-PASCAL 开发了一些大型应用软件，确实证明了它是一种易于掌握、功能较强、使用方便的程序设计语言。

本章的目的不在于全面系统地介绍 PC-PASCAL，而是结合到以后各章节的应用，着重介绍 IBM-PC PASCAL 的扩充，以方便读者阅读和使用本书所介绍的应用程序。

1.1 编程步骤

PC-PASCAL 是在 PC-DOS操作系统下运行的 PASCAL 编译系统。该编译系统主要包括以下几个执行文件和库文件：

PAS1.EXE	第一遍编译执行文件
PAS2.EXE	第二遍编译执行文件
PAS3.EXE	第三遍编译执行文件（不是必须的，可以省略）
PASCAL.LIB	PASCAL 库文件
LINK.EXE	连接程序执行文件
8087.LIB	若装有 8087 协处理器需要该库

当然还应该支持 PASCAL 编译器的 DOS 系统，以及编辑 PASCAL 源程序的编辑程序，如 EDLIN, WORDSTAR 等。

PASCAL 编程及运行有如下步骤：

(1) 编辑源程序

调用文本编辑程序如 EDLIN，编写或修改源程序。命令格式为

>EDLIN 文件名.PAS ↵

其中文件名为用户指定的，并建议扩展名为“PAS”。执行该命令即可进入文本编辑状态。输入相应的插入修改命令即可进行源程序的编辑工作。最后产生给定文件名的源程序文件。

(2) 第一遍编译

命令格式为

>PAS1 文件名 ↵

其中文件名即为被编译源程序的文件名。执行该命令调用第一遍编译，从文件中读出源程序检查语法正确性。该命令执行后，系统自动提示用户回答下列问题：

源文件名	Source file name [.PAS]: ****
目标文件	Object file name [****.OBJ]:

源列表文件 Sotrcce listing [Nul.LST];

目标列表文件 Object listing [NUL.COD];

用户根据规定给予必要的响应，系统就进行第一次编译。如果未发现错误，则出现如下提示：

Pass One No Errors Detected.

若有错误，则提示：

Pass One * Errors Detected

其中*位置出现的是查出的错误数目。

在上述提示的源列表文件和目标列表文件中方括号部分有 NUL.LST 和 NUL.COD，NUL 表示空文件。当用户不需要源列表文件和目标列表文件时，对这两项的响应可直接键入回车；若需要列表文件时，则应键入相应的文件名。而目标列表文件将在第三遍编译中显示。所以，若无目标列表文件，第三遍编译就不需要执行了。

在第一遍编译中还同时产生两个中间文件PASIBF.SYM和PASIBF.BIN。

(3) 第二遍编译

命令格式为：

>PAS2 ↵

第二遍编译不要求用户键入其他内容。它的作用是读入上一次编译中建立的中间文件PASIBF.SYM和PASIBF.BIN，进行优化并建立扩展名为.OBJ的目标文件；清除中间文件.SYM和.BIN；当要求目标列表文件时，还建立供第三遍编译用的PASIBF.OID文件。

(4) 第三遍编译

>PAS3

只在需要目标列表文件时才需要这一步，否则可省去。

(5) 链接

命令格式为：

>LINK 文件名

链接的目的是读出一个或多个用户的目标文件 .OBJ，与必须的 PASCAL 子程序库中的例行程序连接在一起，形成扩展名为 .EXE 的可执行文件或运行文件，并产生扩展名为 .MAP 的映象文件（可选的）。

图 1-1 绘出了各编程步骤及其所涉及到的文件。

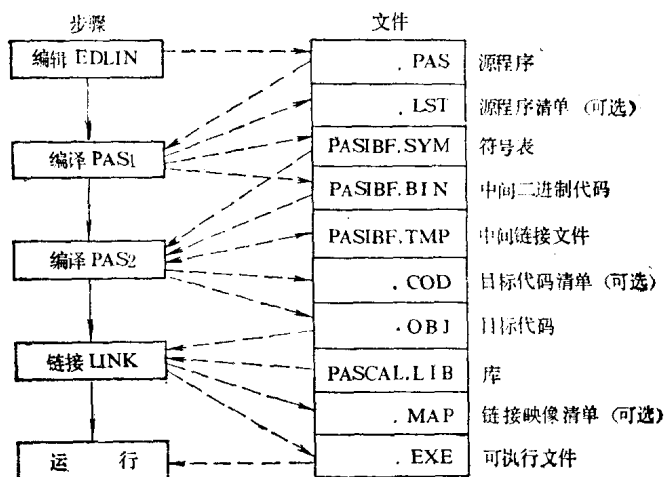


图 1-1 编程步骤和涉及的文件

(5) 运行

在 DOS 状态下直接键入 LINK 所产生的可执行文件名，命令处理程序将读入该文件并装配到内存的适当位置，然后执行之。命令格式为：

>文件名

1.2 基本词汇

从形式上看，一个 PASCAL 程序正文是由以分隔符隔开的各种记号按一定的语法规则组成的。

(1) 基本符号

字母 = A...Z | a...z

数字 = 0..9 | A..F | a..f

字符 = 256 个 ASCII 字符

(2) 标识符

标识符 = 字母 {字母 | 数字}*, 只识别前 31 个符号。

(3) 常数

常数 = 整型常数 | 实型常数 | 字符常数 | 串常数

整型常数 = [+ | -] 无符号整型常数

无符号整型常数 = 数字 {数字}*

实型常数 = [+ | -] 无符号整型常数 . 无符号整型常数
[e [+ | -] 无符号整型常数]

字符常数 = '字符'

串常数 = '{字符}*'

(4) 专用符号

专用符号 = 标准专用符号 | 扩充专用符号

标准专用符号 = + | - | * | / | = | < > | < = | > = |

> | (|) | [|] | : = | . | , | ; | : ' | ' .. | ^ | { | }

扩充专用符号 = \$ | (* | *) | (. | .) | ? | @ | # | !

这里 \$ 用作元语言的前缀; (* 和 *) 用来替换 {和}; (. 和 .) 可用来替换 [和]; ? 或 @ 可用来替换 ^ 作为指针; # 用在整型常量中表示采用的进位制; ! 用来表示行的结束和行末注释的开始; [和] 可用来替换 BEGIN 和 END。

(5) 保留字

保留字 = 标准保留字 | 扩充保留字

标准保留字 = AND | ARRAY | BEGIN | CASE

| CONST | DIV | DO | DOWNTON | 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

扩充保留字 = ADR|ADS|CYCLE|EXTERN|
EXTERNAL|FORWARD|IMPLE-
MENTATION|INTERFACE|MODULE|
OTHERWISE|PUBLIC|PURE|
READONLY|RETURN|STATIC|
SUPER|UNIT|USES|VALUE|VARS|
XOR

(6) 预说明的标识符

预说明标识符 = 标准标识符|扩充标识符

标准标识符 = ARCTAN|ABS|BOOLEAN|CHAR|
CHR|COS|DISPOSE|EOL|LN|
EXP|FALSE|FLOAT|GET|INPUT|
INTEGER|LN|MAXINT|NEW|ODD|
ORD|OUTPUT|PACK|PRED|PUT|
READ|REPEAT|REAL|RESET|
REWRITE|ROUND|SIN|SQR|SQRT|
SUCC|TEXT|TRUE|TRUNC|
UNPACK|WRITE|Writeln

扩展标识符 = ABORT|ASSIGN|BYWORD|CLOSE|
CONCAT|COPYLST|COPYSTR|
DECODE|DELETE|DISCARD|
ENCODE|EVAL|FCBFQQ|
FILLMODES|FILLC|FILLSC|
HIBYTE|INSERT|LOBYTE|LOWER
|LSTRING|MAXWORD|MOVEL|
MOVER|MOVESL|MOVESR|NULL|
POSITN|READFN|READSET|
RESULT|RETYPE|SCANEQ|
SCANNE|SEQUENTIAL|STRING|

(7) 分隔符

分隔符 = 空格 | 换行符 | 正文注释 | 行尾注释

正文注释 = {正文} | (*正文*)

行尾注释 = !串

(8) 语句标号

语句标号 = 无符号整数 | 标识符

1.3 元语言

元语言有 30 个元命令，用来设置编译程序可选项、进行条件编译以及对编译列表和程序执行进行控制。元命令只能在注释开始的地方给出，一般格式为

{元命令 1, 元命令 2, ..., 元命令 n}

实际上，每条元命令都是引用或设置一个元变量。元变量有如下几类：

- 无类型：引用时一次性作用，如：\$ 元变量
- 整数型：可置成数值，如：\$ 无变量：数值
- 开关型：可设置成开关值，如 \$ 元变量 +
- 字符串型：可置成串常量，如 \$ 元变量：'文本'

元命令可分为如下三大类：

(1) 调试和错误处理类

\$ BRAVE + / - 在屏幕上显示/不显示出错和警告信息。（省缺值 +）

\$ DEBUG + / - 打开/关闭编译程序所能执行的全部运行时检测，这些调试检测包括 \$ENTRY, \$INDEXCK, \$INITCK, \$MATHCK, \$NILCK, \$RANGECK, 和 \$STACKCK。（省缺值 +）

- \$ENTRY+/-** 调试程序生成/不生成运行时出错的过程或函数的入、出口调用。注意，**\$ENTRY-** 将使 **\$LINE+** 关闭。
(省缺值-)
- \$ERRORS:n** 设置清单每页允许的出错数。(省缺值 25)
- \$GOTO+/-** 在清单的每条 GOTO 语句旁标出警告信息 “Considened harmful”。
(省缺值+)
- \$INDEXCK+/-** 运行时检查数组的下标值范围。(省缺值+)
- \$INITCK+/-** 生成代码对所有未初始化的整数值置为 -32768、指针值置为 1。(省缺值-)
- \$LINE+/-** 运行出错产生行号调用。(省缺值-)
- \$MATHCK+/-** 运行时检测溢出及除0。(省缺值+)
- \$NILCK+/-** 检测被引用的指针是否为 NIL、未初始化、出界、指向空白块或奇地址(有效指针为偶数)。(省缺值+)
- \$RANGECK+/-** 检查子界的有效性。(省缺值+)
- \$RUNTIME+/-** 置出错位置。若为+，则为调用过程或函数的地方而不为过程或函数出现的地方。(省缺值-)
- \$STACKCK+/-** 在过程和函数的入口处或当压入多于 4 字节参数到栈时，检查栈是否溢出。(省缺值+)
- \$WARN+/-** 在清单中给出警告信息。(省缺值+)
- (2) 文件列表格式类
- \$LINE SIZE:n** 设置清单的行宽。(省缺值 79)