

计算机语言实用程序 与编程技巧

Turbo Pascal

语言

任朝阳
吴耀斌 编 著
雷方桂

王云宜 审 定

中南工业大学出版社



计算机语言实用程序与编程技巧

Turbo Pascal 语言

任朝阳
吴耀斌 编著
雷方桂
王云宜 审定

中南工业大学出版社

湘人新登字010号

计算机语言实用程序与编程技巧

Turbo Pascal 语言

任朝阳 吴耀斌 雷方桂 编著

王云宜 审定

责任编辑：肖梓高

*

中南工业大学出版社出版发行

中南工业大学出版社印刷厂印装

新华书店总店北京发行所经销

*

开本：787×1092 1/16 印张：12 字数：298千字

1994年11月第1版 1994年11月第1次印刷

印数：0001—7000

*

ISBN 7-81020-690-7/TP·041

定价：9.00元

本书如有印装质量问题，请直接与生产厂家联系解决。

《计算机语言实用程序与编程技巧》丛书

编委会成员

编委会主 任: 王云宜

编委会副主任: 田荣璋

编委会成 员 (以姓氏笔画为序):

王云宜 毛先成 田荣璋 向南平

任朝阳 陈福盈 杨路明 肖梓高

吴耀斌 张后苏 崔秀梅 彭先定

雷方桂

前 言 一

《计算机语言实用程序与编程技巧》丛书暂包括五个分册。它们是：《C语言》；《Turbo Pascal语言》；《80386汇编语言》；《BASIC语言》；《FOXBASE+数据库语言》。

编写这套丛书的目的是为了帮助已经掌握了以上语言基本编程方法的读者进一步提高程序设计技术，学习高级程序设计的方法和技巧。在应用程序设计工作中，借鉴于已有的、经过优化的现有实用程序与实用程序段，使用到应用程序中去，节省编程周期，不断提高应用程序的质量和效果。它是大专生及相关人员最好的教学参考书和工具书。

近年来，各种程序设计语言，从BASIC到C都在不断的完善和丰富自己，版本不断更新，各种语言在保持自己特色的同时，功能上均在相互接近，互相渗透，取长补短，不断改进。掌握好任意一门程序设计语言都能满足绝大部分应用程序设计的需要。本丛书选材正是从各种语言的最新版本功能出发，叙述各种语言的高级程序设计技术和实用程序，使读者缩短掌握语言新功能的途径。

《C语言》不仅叙述了基本的静态、动态数据结构，还系统地阐述了内存管理、视频系统及应用、图形、资源辩认、常驻内存程序、数值计算等方面高级程序设计方法，有很多都是作者最新科研成果的总结。

《Turbo Pascal语言》由于其结构化、模块化特点，不仅是大专学生的基础课程，也是当今中学生奥林匹克竞赛的规定语言之一。本书不仅适合大专学生提高的需要，也适合竞赛需要，适合中学生提高PASCAL语言程序能力的参考书。

《80386汇编语言》则以8088/8086为基础，重点阐述80386、80486和80387、80487的程序设计技术，供程序员参考使用。

《BASIC语言》是传播最早、最广泛的计算机语言。它对Quick Basic和Ture Basic功能作了详细叙述，其矩阵语句、图形功能、动画能力、计算能力、I/O功能及人机图形、窗口界面丝毫不弱于其它任何语言，编程者可以设计出具有各种特色的应用程序。

《FOXBASE+数据库语言》总结了目前最常用和实用的程序设计技术，在菜单设计、窗口设计、查询搜索、工具应用等方面都进行了系统的阐述，还有不同类型管理系统的实用程序示例，供读者直接引用或模仿。它是数据库管理系统开发人员和应用程序设计人员最好的一本参考书。

参加本丛书编写的人员都是长期从事教学和科研工作的教师，他们具有扎实的基础理论和实践能力。全书所有程序均经过上机验证，不少是科研成果的总结汇编而成，写法上注意了先进性、实用性，尽量符合读者学习的思维过程，因而可读性好，极具实用价值。

由于计算机软件的发展一日千里，丛书不可能全部包含当前应用中最新内容，不当之处，尚望批评指正，并愿为广大读者提供咨询服务。

编委会

1994年10月

前 言 二

程序设计是计算机及相关专业的基础课，学生对这门课程掌握的程度和能力的培养对今后其它课程的学习以至于从事计算机事业的能力都至关重要，影响深远。计算机科学中，用 Pascal 语言描述算法简明清晰，取得的成功已被公认，它对操作者养成良好的程序设计习惯，提高对现实数据对象及算法的描述能力上所起的作用越来越被人们重视。Turbo Pascal 是目前国内外最受欢迎的 Pascal 系列之一，它继承了标准 Pascal 的许多优点，整个语言紧凑整齐，概念简洁，数据结构和控制结构丰富，程序结构性和可读性好，查错能力强，易于体现结构化程序设计的思想。同时，Turbo Pascal 本身还具有设计精巧，编辑方便，编译与目标代码运行效率高，菜单驱动，用户界面好，操作简便和使用灵活等许多鲜明的特点，特别是扩充了彩色、声音、图形、窗口、类型常数、编译指示、包含文件和汇编语言连接等许多功能。

本书以 Turbo Pascal 为背景，按标准 Pascal 教程的次序，在简单介绍 Turbo Pascal 语言的基本知识和语法规则的基础上，着重描述了算法设计思想，程序设计技巧以及具体程序编码。其中第一章介绍了 Turbo Pascal 基础知识；第二章说明了 Pascal 语言的算法设计思想及有关方法；第三章介绍了结构、分支、循环等 3 种基本程序设计方法；第四、五、七章分别从数据结构的角度来说明各类实际问题的数据描述及对应的程序设计方法；第六章介绍了 Turbo Pascal 函数、过程以及递归的程序设计；第八章介绍了 Turbo Pascal 新增功能：单元、作用和音响等的程序设计方法；第九章介绍了 Turbo Pascal 语言与其它语言和 DOS 的接口程序设计。第十章介绍了 Turbo Pascal 语言程序设计技术。本书各章的实例，具有一定的代表性和实用性，这些经典的程序实例可供读者直接应用到各种实际的程序设计和编码中。

本书具有一定的通用性、实用性和可读性，它可以作为大中专学生学习 Turbo Pascal 语言的一本较好的参考书，也可以作为程序员编写程序的一本实用工具书。

本书由任朝阳（第 2、9、10 章）、吴耀斌和雷方桂（第 1、3、4、5、6、7、8 章）编写，由王云宜等审定，参加编写的还有吴王匀文、王建新、向期中等同志。书中有错误或不足之处，请广大读者和同行批评指正。

作 者
1994 年 9 月

目 录

0-1	概述	(1)
1.1	Pascal 语言简介	(1)
1.2	Turbo Pascal 语言概述	(2)
1.3	Turbo Pascal 和标准 Pascal 的比较	(3)
1.4	Turbo Pascal 程序结构的组成	(6)
1.5	字符集和符号	(8)
1.6	常量和变量	(9)
1.7	标准数据类型	(11)
1.8	表达式	(17)
2	算法与程序设计方法	(22)
2.1	程序设计步骤	(22)
2.2	算法	(23)
2.3	算法设计的基本方法	(24)
3	程序设计基础	(38)
3.1	顺序结构程序设计	(38)
3.2	分支结构程序设计	(42)
3.3	循环结构程序设计	(49)
4	基本数据结构	(58)
4.1	数据类型	(58)
4.2	枚举	(58)
4.3	子界	(62)
4.4	一维数组	(64)
4.5	多维数组	(71)
4.6	字符数组	(73)
4.7	字符串类型	(74)
5	复杂数据结构	(79)
5.1	集合	(79)
5.2	记录类型	(83)
5.3	文件类型	(89)
6	过程、函数与递归程序设计	(99)
6.1	过程	(99)
6.2	函数	(103)
6.3	递归	(105)
7	动态数据结构	(116)
7.1	指针	(116)

7.2	链表	(119)
7.3	二叉树	(131)
8	单元与作图	(137)
8.1	单元	(137)
8.2	作图	(146)
8.3	音响	(156)
9	接口程序设计	(158)
9.1	Turbo Pascal 与汇编语言的接口	(158)
9.2	Turbo Pascal 与 DOS 和 BIOS 的接口	(164)
9.3	Turbo Pascal 与 Turbo C 的接口	(174)
10	程序设计方法与技术	(176)
10.1	结构化程序设计	(176)
10.2	结构化程序设计典型实例——算术表达式处理与计算	(176)
10.3	小结	(181)
	参考文献	(182)

1 PASCAL 语言的基本知识

1.1 Pascal 语言简介

Pascal 是一种计算机通用的高级程序设计语言。它是由瑞士 Niklaus Wirth 教授于 60 年代末设计并创立的。它是在 ALGOL60 的基础上发展的一种结构程序设计语言。取名 Pascal 是为了纪念 17 世纪法国著名哲学家和数学家 Blaise Pascal (1623~1662 年)。第一个 Pascal 编译程序于 1970 年开始运行。1971 年, Wirth 教授在瑞士的《ETH》杂志上正式发表了 Pascal 程序设计语言的初始报告。1975 年, 他和 J. Jensen 联名发表了著名的修改报告和用户手册, 描述了标准 Pascal 语言, 并提供了一些用于说明 Pascal 特性的例子, 被 Pascal 的系统实现者和用户们视作基本的指南和权威性著作。国际标准化组织 ISO (International Standard Organization) 在进一步的一些修改后, 已将它定为 ISO 标准 Pascal 语言, 我国制定的 Pascal 标准与 ISO 标准一致, 不同的计算机系统在实现此标准时可能作了一些增删 (例如扩充了某些功能, 拓宽了某此限制)。

Pascal 语言问世以来, 在 20 余年间, 先后产生了适合于不同机型的各种各样的版本。目前, 国际上 Pascal 语言有许多种版本, 例如有:

(1) 标准 Pascal 语言。1971 年, 沃斯教授提出并定义了 PASCAL 语言。1975 年, 他和金埃森 (J. Jensen) 对 Pascal 语言进行了修改, 修改报告作为“标准 Pascal 语言”。

(2) Pascal 8000 语言。它是标准 Pascal 的扩展版本。1976 年, 由日本东京大学的 HIKITA 和 ISHIHATA 用 PASCAL 语言编译程序, 它能在 OS7 操作系统操作下应用于 HITAC8800/8700 计算机系统 (与 IBM/370 指令系统相同)。后来, 这个版本经过澳大利亚原子能委员会 (AAEC) 的 COX 和 TOBIAS 的修改和扩展, 能在 OS 操作系统控制下应用于 IBM360/370 计算机。

(3) OMSI Pascal 语言。它是由美国俄勒冈州小型计算机软件公司 (OMSI) 确定, 在 RT-11 操作系统操作下应用于 PDP-11/03 等计算机系统上。

(4) PDP-11 Pascal 语言。它是由美国 DEC 公司 1983 年推出的在该公司生产的 PDP-11 系列机上实现的 PDP-11 Pascal V0.1 版本。该版本除了包含标准的 Pascal 所有语言之外, 还增加了一些扩充语言成份。PDP-11 Pascal 与 DEC 公司在 VAX-11 系列机实现的 VAX-11 Pascal 是向上兼容的, 即 PDP-11 Pascal 是 VAX-11 Pascal 的一个子集。一个用 PDP-11 Pascal 写的程序可以在 VAX-11 系统上正确的编译和运行。它对标准的 Pascal 语言语法部分的扩充包括: % include 指令; 常数定义的扩充; 标识符定义的扩充及变量的初始化和扩充运算符、标准函数、标准过程及一些语法规则。它对标准的 Pascal 语言关于文件的处理的扩充包括: 文件的分量、存取方法和文件的模式; 用于文件处理的标准过程。

(5) UCSD Pascal 语言。UCSD Pascal 是美国加州大学圣地亚哥分校开发研制的。它在很多方面对 Pascal 进行了扩充。在 APPLE-II 微型机上运行的 APPLE UCSD Pascal 系

统中，它对标准的 Pascal 语言语法部分的扩充包括：定义了一些新的变量类型和有关字符串的处理的内部函数。关于文件处理的扩充包括：过程 rewrite 和 reset 的扩充和 close 过程、seek 过程的定义和扩充，以及 blockread 和 blockwrite 函数、ioresult 函数的定义和扩充。在 APPLE UCSD Pascal 系统中还有其它不少的预定义过程和函数，并增加了一些图形，都很具有特色，颇受欢迎的。

(6) Turbo Pascal 语言。它是美国 Borland 公司在 80 年代末期推出的最新产品。它的功能更强，使用更方便，提供了 Pascal 语言的崭新环境，实际上已成为微机上 Pascal 语言的主流，是目前在国内外最受欢迎的 Pascal 系列之一。

Wirth 教授所设计的 Pascal，是一种优良的程序设计教学语言，整个语言紧凑整齐，概念简洁，可以方便地书写出结构化的程序，它强调的是可靠性、易于验证性、概念的清晰性和实现的简化。

Pascal 具有丰富的数据结构和控制结构，可以用来描述复杂的算法，得到质量较高的程序，程序可读性好，查错能力强，有利于培养严谨、清晰的程序设计风格和良好习惯，并使从中领会和理解精细的程序设计技巧。尤其引人注目的是引入了结构化程序设计的思想。

Pascal 能适用于数值运算和非数值运算，能广泛应用于各种领域，还可以用于辅助设计和作图。

1.2 Turbo Pascal 语言概述

自从 Turbo Pascal 语言问世以来，它对 N.wirth 教授原报告的标准 Pascal 做了较大的改进和扩充。Turbo Pascal 语言的说明部份可以任意次序和次数进行，变量可以用有类型常量初始化，CASE 语句增加了 else 分支，可发直接与汇编语言连接并增设了与系统底层接口的方法。特别的 4.0 版的单元实现了分离编译的模块化结构。5.0 版的 OOP 程序设计及 6.0 版的 Turbo Vision 使其步入了最高新程序设计方法行列，Turbo Pascal 提供的大量标准过程和函数使编程者得心应手，这些“半成品”稍加拼凑便可达到设计目标，这一特点使 Turbo Pascal 具有较高的实用价值。

另外十分友好的使户界面，它将 Pascal 源程序编辑、编译、链接及调试集于一个环境之下 (IDE)，使编程者在窗口菜单方式及 Help 信息的帮助下从设计到调试一气呵成等等，弥补了标准 Pascal 的不足。是当今影响最大的一种算法语言。

在短短的几年中，美国 Borland 公司推出了 Turbo Pascal V3.0、V4.0、V5.0 和 V6.0，Pascal 6.0 版与过去的版本高度兼容。它具有以下特点：

①提供了集成调试器，用户可以在集成开发环境下使用它调试程序。它可完成对程序的追踪、单步执行至光标、重新执行、设置断点、中断执行、检查和修改变量及内存地址等功能。并提供了在集成开发环境之外利用 Borland 公司的独立调试器 Turbo Debugger 调试程序的选择。

②支持以单元为基础的覆盖模块，提供智能化的覆盖管理程序，使覆盖更容易和更快地执行。支持 Lotus / Inter / Microsoft 扩展内存规范 (Expanded Memory Specification, 简记为 EMS)。一旦加载到 EMS 内存，就关闭覆盖文件，减少后续覆盖的加载，从而加快内存转换。

③提供了 8087 浮点仿真模式，即使没有配置 8087 数学协处理器，亦可使用 IEEE 浮点

数据类型, 进行优化的浮点运算。

④支持面向对象的程序设计 (Object-Oriented Programing, 简记为 OOP) 这一新颖的程序设计手段, 跟上时代潮流。

⑤新一代用户界面, 能支持鼠标器操作, 多层覆盖窗口, 多文件编辑, 扩展联机求助系统, 条件断点等; 以及编辑器能使用扩展内存以建造大型应用软件。

Turbo Pascal 系列软件作为开发系统软件与应用软件及实施科学计算和教学的有力工具, 正在发挥其越来越大的作用。

1.3 Turbo Pascal 和标准 Pascal 的比较

现把 Turbo Pascal (V4.0) 和由《美国国家标准 Pascal 计算机程序语言》书中介绍的 ANSI/IEEE 7703X3.97-1983 的 ANSI Pascal 作一比较。

(一) Turbo Pascal 与 ANSI Pascal 的差别

Turbo Pascal 和 ANSI Pascal 不同点如下:

(1) 在 Turbo Pascal 中, 标识符可以任意长度, 并且所有字符都有效。在 Turbo Pascal 中标识符可以任意长度, 但只有 63 个字符有效。在 ANSI Pascal 中, @ 符号是 ^ 的代替符号。在 Turbo Pascal 中, @ 符号是标识符, 从不作 ^ 符号识别。

(2) 在 ANSI Pascal 中, 注释语句可以用 { 开始 * } 结束, 或用 (* 开始) 结束, 在 Turbo Pascal 中必须用相同的符号进行注释。{ 和 } (* 和 *) 配对。

(3) 在 ANSI Pascal 中, 在变体部分标志类型的值只能出现一次。在 Turbo Pascal 中没有这种类型。

(4) ANSI Pascal 中, 文件类型的分量类型不能有文件类型的结构类型。Turbo Pascal 无此限制。

(5) ANSI Pascal 函数语句体至少含有给函数标识符赋值的语句。Turbo Pascal 没有这种要求。

(6) 在 ANSI Pascal 函数语句体中, 变体选择域不能是实际变量参数。Turbo Pascal 没有这种限制。

(7) ANSI Pascal 允许把过程和函数作为参数, 在 Turbo Pascal 中不允许。对 5.0 这条不再适用。

(8) ANSI Pascal 标准过程 RESET, REWRITE 不要求初始化文件变量。在 Turbo Pascal 中文件变量传递给 RESET, REWRITE 前文件变量必须是 ASSIGN 过程把文件变量和外部文件连接起来。

(9) ANSI Pascal 定义了用来文件输入、输出的 gets, put 标准过程。这些过程在 Turbo Pascal 中没有定义。

(10) ANSI Pascal 语法 NEW (P, C1, ... Cn) 创建指定变体的动态变量。Turbo Pascal 不允许。

(11) ANSI Pascal 定义了用来紧缩和拆开紧缩变量的 PACK, UNPACK 标准过程。Turbo Pascal 没有定义。

(12) ANSI Pascal 中由 $i \bmod j$ 总是计算正值, 如果是 0 或负是错误的。Turbo Pascal

中 $i \bmod j$ 等价于 $i - (i \div j) * j$, j 为负不是错误。

(13) ANSI Pascal 块中的 goto 语句可引用包含块中的标号。Turbo Pascal 不允许。

(14) ANSI Pascal 中 CASE 语句选择表达式的值不等于任何分支常量是错误的。Turbo Pascal 中, 这不是错误, 除非有 else 子句, 否则不执行条件分支。

(15) ANSI Pascal 不允许修改 for 语句控制变量。Turbo Pascal 没有这种限制。

(16) ANSI Pascal 中 Read 前如果 Eoln 为真, 从文本文件读取字符付给一空格。Turbo Pascal 赋回车码 (\$ B)。

(17) ANSI Pascal 从文本文件读取整数或实数, 遇到下一个字符不是整数或符号数的组成部分为止; Turbo Pascal 遇到空格字符时的读数。

(18) ANSI Pascal 如果指定宽度小于字符串, 输出到文件中的紧缩字符串被截断。Turbo Pascal 输出完整的串。

(二) Turbo Pascal 对 ANSI Pascal 的扩展

Turbo Pascal 下述功能是对 ANSI Pascal 的扩展:

(1) Turbo Pascal 增加了保留字

absolute, interfacw, string, external, interrupt, unit,
implementation, shl, uses, inline, shr, xor

(2) 标识符可含有一出现在第一位置的下划线, 标识符可作标号。(3) 整数常量可用十六进制表示, 常量前面冠以 \$。

(4) Turbo Pascal 串常量和串类型兼容, 可含有控制字符和其它非打印字符。

(5) 块中标号, 常量, 类型, 变量, 过程和函数允许以任意顺序出现多次。

(6) Turbo Pascal 实现了短整型、长整型、字节型和双字节型整数; 单精度型、双精度型、扩展型、十进制组装型实数。

(7) Turbo Pascal 实现了不同于 ANSI Pascal 紧缩数组的字符串节型, 字符串包含可动态变化的长度属性。扩展了类型兼容规则, 使得字符类型, 紧缩字符串类型和字符串类型兼容。

(8) 变量可由 absolute 子句说明在绝对内存地址上。

(9) 字符串变量可由下标存取单个字符。变量可通过类型强制转换, 变成另一类型。

(10) Turbo Pascal 实现了除文件类型外的任何类型变量的类型常量。

(11) Turbo Pascal 实现了三个新的逻辑操作符 Xor, Shl 和 Shr。

(12) '+' 操作符可用来连接字符串, 关系运算可用来比较字符串。

(13) Turbo Pascal 实现了用来获得变量, 过程或函数地址的操作符。

(14) 通过值类型强制转换, 表达式类型可转变成另一种类型。

(15) Case 语句允许在分枝号表中使用常量范围, 并提供可选择的 else 子句。

(16) 可用 external, inline 和 interrupt 指令说明过程和函数, 支持汇编语言程序, 内部机器码和中断处理过程。

(17) 变量参数可以是无类型参数, 这样可用任何变量作为参数。

(18) Turbo Pascal 实现了 ANSI Pascal 所没有的文件处理过程和函数:

Append Close Flush RmDir BlockRead Erase
GetDir seek lockWrite FilePos Mkdir Seekof

ChDir fileSize Rename SeekEoln

(19) 字符串值可用 Read, Readln, Write, Writeln 标准函数进行输入、输出。

(20) Turbo Pascal 实现了下述 ANSI Pascal 没有的标准过程和函数:

Addr FreeMem MaxAvail Randomize Cseg GetMem

MemAvail Release Concat Halt Move Sptr

Copy Hi Ofs Sseg Dseg Inc ParamCount Seg

Dec Insert ParamStr SizeOf Delete Int Pi Str

Exit Length Pos Swap

FillChar Lo Ptr Upcase Frac Mark Random Val

(21) Turbo Pascal 单元实现了更多标准常量、类型、变量、过程和函数。

(三) 依赖现实特性

Turbo Pascal 没有指定 ANSI Pascal 中定义的依赖实现特性会产生什么结果。除非使用了某些依赖实现特性, 通常程序不应依赖于任何使用有依赖特性的方法。依赖实现特性如下:

- 变量引用下标表达式的计算顺序;
- 集合构造符表达式运算的顺序;
- 二进制操作数的运算顺序;
- 函数调用中实际参数的运算顺序;
- 过程语句中实际参数的运算顺序;
- 用 Page 读文本文件的影响;
- 外部参数的结合。

(四) 错误处理

下面列出 Turbo Pascal 不自动检测的错误。错误码是 ANSI 使用的编号。

6, 19-22, 25-31 号错误, Turbo Pascal 不会遇到。

2 号、如果 t 是变体部分的标志域, f 是活动体的一个域, 当引用 f 存在改变 t 是一个错误。Turbo Pascal 不检测这种错误。

3 号、P 是指针变量, 如果 p 为 nil, 用户引用是错误的。Turbo Pascal 不检测这种错误。

4 号、P 是指针变量, 当引用 P 存在时改变 P 的值是一种错误。Turbo Pascal 不检测这种错误。

42 号、如果 Eof(f)为真, 调用 Eoln(f)是错误的。这种情况 Turbo Pascal 不认为错误的。

43 号、如果变量值没有定义, 在表达式中引用变量是错误的。Turbo pascal 不检测这种错误。

46 号、I mod j 如果 j 是 0 或负是错的。Turbo Pascal 中 j 允许负值。

48 号、如果函数标识符没有赋值是一个错误。Turbo Pascal 不检测这种错误。

51 号、Case 语句选择表达的值不是任何分枝常量是错误的。在 Turbo Pascal 中不是错误。

1.4 Turbo Pascal 程序结构的组成

对于标准 Pascal 教程，而上机实践时又要学习一种具体版本，给读者学习增加了环节和难度；而对于一种 Pascal 的使用手册，初学者读起来内容杂乱，不容易掌握要领，甚至有时读不懂。因此在编写本书时，我们将 Turbo Pascal 语言符合标准的部份提取出来，按常规顺序向读者介绍，并可以边学习边上机实践。对于与标准紧密相关的内容穿插在相应的章节中，作到使读者潜移默化的接受，而对于其特殊功能将分类整理专门说明。这样，旨在使读者既了解和学习标准 Pascal 语言，同时又掌握 Turbo Pascal 这一优选语言版本。

首先举几个最简单的 Turbo Pascal 程序的例子。

【例 1】program Hello;

begin

Write('How do you do?')

end.

这个程序的作用是打印出“How do you do?”这个英文句子。

【例 2】已知圆的半径，求圆的直径、周长和面积。

program CircleDemo (input, output);

{calculate diameter, circumference and area of a circle}

const

Pai=3.14159;

var

Radius: integer;

Diameter, Area: real;

begin

Write('Enter radius of a circle: '); {whole number}

ReadLn (Radius);

Diameter: =2 * Radius;

Area: =Pai * Radius * Radius;

Writeln('Diameter of the circle is : ', Diameter: 12: 5);

Writeln('Circumference is : ', 2 * Pai * Radius: 12: 5);

Writeln('Area of the circle is : ', Area: 12: 5);

ReadLn

end.

这个程序要求从键盘键入一个正整数作为圆半径，计算它的直径、周长及面积，并将结果显示出来。

从以上两个简单例子可以看出：Turbo Pascal 程序是由程序首部和程序体组成，程序体又分为说明部分和执行部分，它还允许在程序首部和程序体之间加 uses 子句，说明被程序直接和间接使用的单元。关于单元的概念见第 8 章。

(一) 程序首部

程序的第一行称为程序首部，它包括一个 Turbo Pascal 专用词汇 program，接着是用户给该程序取的名字，最后以分号表示程序首部的结束，下面将是程序主体的开始。程序首部在一个 Turbo Pascal 程序中并非必须出现，即它是可选的。写上它仅起文档作用，不过建议最好把它写上，养成良好习惯。

对程序的首部可作如下理解：首先是保留字 program，接着是标识符；再下去有两个通路，一个是直通过去，该语法部分结束；另一个是到圆括号“(”，接着是标识符；再下去，路径又有两个通路，一个是直接到圆括号“)”结束该部分，另一个是到逗号“，”，然后又到标识符，这可以经过多次循环，每循环一次，就增加一个标识项，再到圆括号“)””，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，其中有一个标识项（参数），也可以有用逗号分隔的多个标识项；也可以根本没有用圆括号括起来的参数表。

(二) 程序说明部分

说明部分用于定义和说明程序中要用到的数据，如常数、变量等，是对数据和过程的描述。说明部分的内容可以非常丰富，它除了可包含常数的定义、变量说明外，还可能有类型定义、标号说明、子程序（过程与函数）说明；若程序要用到单元，则还应书写相应的 uses 子句等。但是，一个简单的 Turbo Pascal 程序亦可不包含说明部分，即说明部分也是可选的。如上面的例 1 的程序即是如此。程序的说明部分分为：标号说明、常量定义、类型定义和变量说明、以及过程与函数说明五部分。Turbo Pascal 允许这些部分在程序中出现任意次，也允许它们以任何顺序出现。

(三) 执行部分

执行部分描述了程序要执行的操作。它必须以一个 Turbo Pascal 专用词汇 begin 开始，以另一个专用词汇 end 跟句点结束，其间是一些执行具体操作的语句，以分号作为语句间的分隔符。begin 与 end 应配对出现这是每一个 Turbo Pascal 程序必须有的。紧跟 end 后的句点既表示执行部分的结束，亦表示整个程序的结束。此后若还有任何语句编码，Turbo Pascal 编译器将不予理会。另外 Turbo Pascal 规定：紧随 end 之前出现的分号允许省略。说明部分和执行部分合在一起称为程序块（亦称为程序体）。因此，Turbo Pascal 程序是一个块结构的程序。

一个完整的 Turbo Pascal 程序形如图 1-1 所示。

```
Program 程序名(参数表);           { 程序首部}
  use 子句;
  说明部分;                         { 说明部分}
begin
  语句1;
  语句2;
  ...
  语句n }                           { 执行部分}
end.
```

图 1-1 程序组成图

Turbo Pascal 使用计算机能够识别和检验（通过 Turbo Pascal 编译器）的、类似于日常英语的专用词汇、术语及规则。它们构成了 Turbo Pascal 语言的语法。用户使用它们编写 Turbo Pascal 程序时，可不必去管它们究竟是如何在计算机内转换并实现的，而只需致力于正确理解和使用这些词汇、术语，并严格遵循既定的规则。本书并不企图一开始就去完整地、从头至尾地介绍各种定义、词汇、规则、条文、限制等，因为这往往会使初学者不得要领从而对这种语言望而生畏。我们将采用符合人们理解力的习惯做法，即从最基本、最易于被接受的开始，然后由简到繁逐步深入，陆续引出一些新的特性、规则，以解决新的问题。由于 Turbo Pascal 中的结构清晰，语句形式直观，因此易于阅读与理解。

1.5 字符集和符号

每种程序设计语言都有一套自己的基本符号。符号又是由若干基本字符组成，用这些符号按着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被视为是非法的，计算机将不能识别。Turbo Pascal 基本符号由三部分组成。

(一) 基本字符

- (1) 字母: A 到 Z, a 到 z 和 (下划线)。
- (2) 数字: 0、1、2、3、4、5、6、7、8、9。
- (3) 特殊字符: +、-、*、/、=、^、<、>、(、)、[、]、{、}、...、:、;、'、
#、¥、@;

赋值运算符: =;

子界分隔符: ..;

关系运算符: >、<、<>、=、>=、<=;

括号: (.和.) 可以用来代替[和];

注释号: (* 和 *) 可以用来代替{和}。

(二) 符号

(1) 保留字。保留字是由字母拼成的字，Turbo Pascal 语言预先规定了它们的意义，不允许被再定义；因此不能用作下节介绍的用户自定义的标识符，否则 Turbo Pascal 编译器将指示出错信息。Turbo Pascal 保留字如下：

```
absolute*   and    array   begin   case    const   div     do
downto     else   end     external*  file   for     forward
function   goto   if      implementation  in     inline*
interface  interrupt  label  mod     nil    not     overlay
of         or     packed  procedure  program  record  repeat  set
shl*      shr*   string*  then    to     type    unit    until
uses      var    while   with    xor*
```

其中不带 * 号的是标准 Pascal 的保留字。

(2) 标识符。标识符是用来表示程序、常量、变量、过程、函数、文件和类型等名称的符号。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。在 Turbo Pascal 中，

标识符形成的规则是：它必须由一个英文字母或下划线开头，后面可跟英文字母、数字和下划线的任意组合。

标识符中的英文字母大、小写不予区分。一个标识符的长度由行的长度 127 个字符所限制，但只有前 63 个字符有效，其中不允许有空格。

标识符可以分标准标识符和用户自定义的标识符。标准标识符是预先给标准函数、标准过程、标准常量、标准类型、标准文件定义的标识符，即 Turbo Pascal 语言已确定了标准标识符的名称和性质，如标准输入文件、标准输出文件等，在书写程序中可直接使用，不必事先定义。下面列出标准标识符的一部分：

标准常量: false true maxint

标准类型: integer real boolean char text

标准文件: input output

标准函数: abs arctan chr cos eof eoln exp ln odd ord pred round
sin sgr sqrt succ qrt succ trunc

标准过程: read readln write writeln page pa unpack reset rewrite
new dispose get gotoxy put val

用户自定义的标识符是用户按标识符定义的规则自己定义的。常选用能说明某种意义的英文单词。用户标识符必须事先定义或说明才能在程序中使用它。用户自定义的标识符不能用保留字，最好也不要选用标准标识符。下列用户定义的标识符是合法的：

Name BookOne Grade ArrayTwo Yeal a b U123 char point

下列用户定义的标识符是非法的：

ab□x: 在标识符中不能有空格符 (□用来代表空格符)；

end: 保留字不能作标识符；

6xy: 标识符不能以数字开头；

ABC%: 标识符不能有非字母非数字型字符。

由于 Turbo Pascal 不区分英文大、小写字母，所以平常编程时可以用我们习惯的小写英文字母，但 GetDriverName 比 GETDRIVERNAME 与 getdrivername 更容易读。因此本书为了增加可读性，所用的标识符绝大部分采用大小写字母区分字段的混合形式。

(三) 分隔符

在 Turbo Pascal 语言中，把空格符行结束符作为分隔符。规定程序中任何两个相邻的保留字、标识符、数或符号之间都必须插入一个或几个分隔符。但在一个保留字、标识符或数的内部不允许出现分隔符。另外逗号“，”、分号“；”、括号“()”、子界分隔号“..”和注释等也可起分隔符作用。

Turbo Pascal 语句之间也要分隔开，一般用分号“；”分隔，一条语句可以分几行来写，一行也可以写几条语句；但一个程序行最多可以有 127 个字符。

1.6 常量和变量

(一) 常量