

● 姚庭宝 编著
● 国防科技大学出版社

4.0

4.0

Turbo Pascal

4.0

语言、技巧及应用

77.37429
C 148
73.87429
C148

● 姚庭宝 编著

国防科技大学出版社



Turbo Pascal

4.0

语言、技巧及应用

内 容 简 介

本书通过语言介绍与程序示例剖析相结合的方式,系统地论述了 Turbo Pascal 4.0 各种类型的数据结构与语言特性、程序设计的基本技巧与方法以及它在各个领域中的应用。全书由上、下两篇及附录组成。上篇着重阐述 Turbo Pascal 4.0 的概念与特性;下篇则致力于发展程序设计技巧及应用;附录描述了 Turbo Pascal 4.0 的特色、集成开发环境以及上机操作的方法与步骤,同时还介绍了近200个预定义标准子程序以及编译与运行时的出错信息。

本书是作者为高等院校本科生编写的计算机语言课程教材,但在内容选取与叙述方式上考虑了各方面的要求,因而,本书亦是广大微机用户的一本极好的参考书。

Turbo Pascal 4.0 语言、技巧及应用

姚庭宝 编著

责任编辑 钟 平

*

国防科技大学出版社出版发行

(湖南省长沙市 邮政编码: 410073)

湖南省新华书店经销

国防科技大学印刷厂印装

*

开本: 787×1092 1/16 印张: 22 字数: 510千字

1989年10月第一版 1989年10月第一次印刷 印数: 0001—7000

ISBN 7-81024-080-3

TP·15 定价: 7.80元

前 言

Turbo Pascal 是当前在国内外最受欢迎的Pascal系统之一,已售出拷贝五十万份以上。国际上一些软件评论家认为它实际上已成为微机的Pascal标准。Turbo Pascal具有许多鲜明的特点。它恪守并实现了N. Wirth的标准Pascal,它设计精巧,编辑方便,编译与目标代码运行效率高,菜单式驱动,用户界面良好,操作简便,使用灵活。特别是它扩充了许多功能,如色、声、图、窗,类型常量,编译指示,包含文件,与其它高级语言的灵活交接,实现机器级能力等等,从而成为在教学、科研、开发系统软件和应用软件中的一个颇为理想的工具。1987年推出的 Turbo Pascal 4.0 在编译速度、目标代码质量以及易于使用等方面更向前推进了一步。它提供了一个自给自足的集成软件开发环境,集编辑、编译、连接、运行、求助、查错于一身,并采用多窗口技术,实现多级菜单式驱动、上下文敏感式求助系统以及与操作系统切换等功能。它还引入库单元(unit)概念以及各库单元可分别编译的机制,为设计和开发应用程序带来了极大方便。

本书通过介绍和剖析大量程序示例,较系统地论述了 Turbo Pascal 4.0 各种类型的数据结构和语言特性、程序设计的基本技巧与方法以及它在各个领域中的应用。全书由上、下两篇及附录组成。上篇着重于阐述 Turbo Pascal 4.0 概念与特性,它共分十章,即:基础;控制结构;字符串和数组;过程和函数;记录和集合;文件;指针;库单元(unit);一些扩充特性;结构化程序设计思想。下篇致力于发展程序设计技巧及应用,它共分五章,即:通用程序设计实例剖析;程序设计的有力手段——递归技术;动态数据结构在程序设计中的基本应用;一个综合性图形功能演示程序;应用程序的设计和开发。大多数章末配有一定数量的练习题。八个附录对 Turbo Pascal 4.0 的特色、集成开发环境等作了较详尽的描述,并通过示例介绍在 IBM PC 机上机操作步骤。另外,还汇集了 Turbo Pascal 4.0 近200个预定义标准子程序的首部形式及基本功能,介绍了建立用户自定义库文件的方法,以及给出编译与运行时的出错编码及信息等。

在编写本书时,力求体现下列特色:一是把计算机语言和程序设计技巧综合在一起讲述,并通过各类程序示例,以适应广大读者对多方面应用的需求;二是文字叙述深入浅出,简明扼要,明确直观,示例众多,并重视良好程序设计风格的养成;三是内容安排易于入门,坡度适宜,难点分散,目标较高。

本书可作为理工科院校本科生计算机语言课程教材,教学时数包括上机实践在内拟为60—80学时。亦可供广大教师、科技工作者、研究生、行政干部、工程技术人员以及业余爱好者作为了解、熟悉 Turbo Pascal 4.0 及其应用的一本良好入门书,或视作便捷的参考使用手册。打*号的章、节,初学时可先略去。

书中所有程序示例均在 IBM PC 机上用 Turbo Pascal 4.0 编译系统经编译、运行

通过，其源程序均已装入软盘。为方便用户，促进交流，作者愿提供软盘拷贝。需要者请联系。

在编写本书过程中，得到张丁、钟平两同志的不少指点和帮助，获益匪浅；朱建民、李俊等同学亦做了一些工作。特向他们表示深切谢意。对各级领导以及国防科技大学出版社、印刷厂的大力支持，一并致谢。

限于作者的水平，难免在内容选材和叙述上有不当之处。竭诚欢迎广大读者对本书提出批评、建议。

姚庭宝

一九八九年六月

联系地址：长沙国防科技大学系统工程与数学系

目 录

上 篇 Turbo Pascal 4.0 —— 概念与特性

第一章 基 础

1.1	Pascal发展简史	1
1.2	程序基本结构	3
1.3	基本字符集与标识符	4
1.4	标准标量类型	6
1.5	注释、常量定义和变量说明	7
1.6	表达式	9
1.7	简单输入、输出及赋值语句	14
1.8	程序设计风格	16
	练习题	17

第二章 控制结构

2.1	if 语句	19
2.2	case 语句	22
2.3	while 语句	24
2.4	repeat 语句	25
2.5	for 语句	26
2.6	嵌套循环	27
2.7	goto 语句及标号	29
	练习题	30

第三章 字符串和数组

3.1	程序员自定义类型：枚举类型和子界类型	32
3.2	字符串	36
3.3	一维数组	40
3.4	多维数组	44
	练习题	46

第四章 过程和函数

4.1	过程及其调用	47
-----	--------	----

4.2 函数及其调用	51
4.3 嵌套调用和向前引用	54
4.4 递归调用	57
练习题	60

第五章 记录和集合

5.1 记录类型定义	61
5.2 记录的赋值	62
5.3 with 语句和记录的嵌套	63
5.4 变体记录	68
5.5 集合类型定义	71
5.6 集合表达式	72
5.7 集合的赋值与输出	74
练习题	76

第六章 文 件

6.1 文件类型定义	78
6.2 文件标准过程和函数	79
6.3 类型文件操作	81
6.4 正文文件操作	84
6.5 无类型文件操作	88
6.6 Turbo Pascal 的外设	89
练习题	90

第七章 指 针

7.1 动态数据结构	91
7.2 指针类型定义和指针变量说明	91
7.3 指针标准过程和函数	92
7.4 指针变量的赋值操作	95
7.5 线性链表结构	96
7.6 双向链环结构	104
练习题	106

第八章 库单元(unit)

8.1 库单元的基本结构	108
8.2 标准库单元及其功能	110
8.3 用户定义库单元	124
8.4 怎样使用库单元	125

第九章 一些扩充特性

9.1 类型常量	129
9.2 编译指示	133
9.3 包含文件	142
*9.4 绝对地址变量	143
*9.5 与机器代码或汇编语言代码的连接	144

第十章 结构化程序设计思想

10.1 什么是结构化程序设计	148
10.2 结构化程序设计示例	149
10.3 再谈程序设计风格	158
练习题	158

下 篇 程序设计技巧及应用

第十一章 通用程序设计实例剖析

11.1 一般数值计算	159
11.2 关于求素数	169
11.3 编译初阶	176
*11.4 集合结构应用	183
11.5 分类算法概述	187
练习题	195

第十二章 程序设计的有力手段——递归技术

12.1 递归算法典型示例	197
12.2 应用递归技术描绘平面曲线图案	201
12.3 探索策略和回溯算法	210
练习题	218

第十三章 动态数据结构在程序设计中的基本应用

13.1 直接表和有序表的检索与插入	220
13.2 二叉树的遍历、检索与插入	226
*13.3 动态数据结构的应用举例	234
练习题	243

*第十四章 一个综合性图形功能演示程序

14.1	演示程序框架	244
14.2	辅助子程序说明	246
14.3	初始化和显示主菜单过程说明	251
14.4	图形功能演示过程说明	254

***第十五章 应用程序的设计和开发**

15.1	建立公用库单元	271
15.2	定义应用	280
15.3	组织数据	287
15.4	实现程序	292

附 录

附录一	ASCII 代码	301
附录二	了解 Turbo Pascal 4.0	302
附录三	Turbo Pascal 4.0与标准Pascal的比较	305
附录四	Turbo Pascal 4.0集成开发环境	308
附录五	上机操作步骤示例	317
附录六	Turbo Pascal 4.0 预定义标准子程序	319
附录七	怎样使用 TPUMOVER	335
附录八	出错编码及信息	337

参考文献	344
------	-----

上 篇

Turbo Pascal 4.0 —— 概念与特性

第一章 基 础

1.1 Pascal 发展简史

Pascal 是七十年代初期由瑞士苏黎世联邦技术学院的 Niklaus Wirth 教授设计并发表的一种计算机高级程序设计语言。取名Pascal是为了纪念十七世纪法国著名数学家 Blaise Pascal(1623—1662)。Pascal 由ALGOL60 发展而来,属于ALGOL 家族的一个成员。第一个Pascal 编译程序于1970年开始运行。一年以后,即1971年,Wirth 教授正式发表了Pascal 程序设计语言的初始报告。1974年,他和K.Jensen 联名发表了著名的修改报告 和 用户手册*。这本书描述了标准 Pascal,并提供一些用于说明 Pascal 特性的例子,直至今日仍被Pascal 的系统实现者和用户们视作基本的指南和权威性著作。

一种新的计算机程序设计语言的设计与研制,通常是以某些特殊要求为动力的。Wirth 教授正是由于对当时那些主要用于程序设计教学的计算机语言感到不满意而推出 Pascal 的。他感到那些语言无论是其特性,还是其构成成分,总是不能从逻辑上给人以令人信服的解释,亦难以给出系统的推理。他坚信,“教给学生们用以表达他们思维的语言,会深深地影响他们思维和创造发明的习惯;而正是这些语言本身的混乱直接地影响着学生们程序设计的风格。”^[5]因此,他所设计的 Pascal,是一种优良的程序设计教学语言,可用来培养学生严谨、清晰的程序设计风格和良好的习惯,并使他们从中领会和理解精细的程序设计技巧。目前,Pascal 已逐渐被许多大学选为第一计算机教学语言。

Wirth 教授设计 Pascal 时,有如下三个主要目标:支持现代程序设计研制的方法学;作为教学语言对学生们来说简单易学;易于在各类计算机(包括小型机甚至微型机)上既可靠而又有效地实现。这三个目标,Pascal 都实现了。加拿大的 Peter Grogono 在其所著《PASCAL 程序设计》一书关于Pascal 的评价中写道:“Pascal 的一个重要特性是它的

* K.Jensen and N.Wirth, "Pascal User Manual and Report," Springer-Verlag, Heidelberg, 1974.

稀疏性,它只有相当少量的基本结构,但由于能用许多方法加以组合,从而使这个语言的功能很强。特别是,算法和数据结构二者都能按分层结构设计。Pascal 具有节省的特点,这是它最重要的性质,它至少在四个不同方面使该语言从中得到益处。第一, Pascal 的编译相当简单, Pascal 程序能在大多数计算机上有效地运行。第二,正确编写的 Pascal 程序是很容易阅读和理解的。第三, Pascal 结构的递归特性允许以简单而自然的方法应用自顶向下研制技术。第四,除少数情况外, Pascal 的控制结构是容易检验的,因为它们的语义简单。” [8]

与其它流行的程序设计语言一样, Pascal 也有各种各样的版本。它的早期版本,如 Pascal 6000,是在 CDC 公司的 Cyber 和 6000 系列计算机上实现的,它被开发用于大型计算机,其修订版本沿用至今。在小型机上,如 DEC 公司的 VAX 11/780,也有 Pascal 编译系统在运行。微型计算机在各个领域特别是教育及商业方面的普及应用,促进了研制适用于微机的 Pascal 编译系统。它们利用各类微机的硬件特殊性能,对标准 Pascal 进行扩充或限制。仅在 IBM PC 机上,就可使用多种 Pascal 的“方言”,其中广为流行的一种是 DOS Pascal,运行于 IBM PC 机的 DOS 操作系统之下,另一种是 UCSD P- 系统 Pascal,运行于 IBM PC 机的 P- 系统之下。随着微机应用领域的不断扩大,还陆续推出各种更新版本。

然而,近几年来,这些版本受到了强烈的冲击。1983 年 11 月,美国一家当时很小且鲜为人知的公司——Borland 国际公司,推出了一种适用于微机的崭新 Pascal 编译系统,称为 Turbo Pascal (1.0 版)。Turbo 意即“透平(机)”,似乎表明这一系统将是强力、高效的。不久,1.0 版被更新为 2.0 版。与此同时,又引入了在 Intel 8087 数字协处理器支持下适用的 Turbo-87 Pascal 编译系统,它提高了实数的运算速度并扩大了值域。1985 年,该公司又推出 Turbo Pascal 3.0。同时,能提高实数精度、提供格式化输出形式的 Turbo BCD Pascal 编译系统问世,它特别适合于商业应用。于是,在不到两年内,Borland 国际公司相继推出三种不同类型的 Turbo Pascal 编译系统,其来势之迅猛实为罕见。1987 年,Borland 国际公司开始销售 Turbo Pascal 4.0,并立即得到用户的广泛欢迎。Turbo Pascal 系统与其它 Pascal 系统相比,有如下几个突出的优点:①编译速度快。Turbo Pascal 1.0 和 2.0 的编译速度由通常的分钟级降到了秒级,而 Turbo Pascal 3.0 又比 2.0 提高了一倍。Turbo Pascal 4.0 的编译速度更快,达到每分钟可编译 27000 行源代码(在 8MHz 的 IBM AT 机上);②操作环境好。Turbo Pascal 4.0 提供了一个设计精巧、易学好用的集成软件开发环境,它集编辑、编译、连接、运行、求助、查错等于一身,提供多级窗口、菜单式驱动,可自由进入操作系统状态并返回,有良好的输入输出陷阱控制和完善的交互式编辑功能,使编辑和修改源程序十分方便;③系统功能强。Turbo Pascal 具有许多扩充功能,提供了丰富的标准过程和函数,它们功能齐全而且使用灵活,从而大大增强了其通用性。特别是 Turbo Pascal 4.0 引入了库单元(unit)的概念,每个库单元就是一段目标代码或称一个模块,它们可单独编译,供多处使用,需要时将它们连接起来。这就为应用程序的开发提供了方便,并能有效地缩短开发周期。此外, Turbo Pascal 4.0 还提供了与汇编语言的接口以及直接把汇编指令写入程序的方式,支持多种新的数据类型及各类图形设备驱动器。

Turbo Pascal 的出现是 Pascal 发展史上的一个里程碑。它作为开发系统软件与应用软件以及实施科学计算和教学的有力工具，正在发挥着越来越大的作用。

为简化计，以下将 Turbo Pascal 作为 Turbo Pascal 4.0 的代名词。对于其它版本，则注明版本号以示区别，如 Turbo Pascal 3.0。

1.2 程序基本结构

计算机语言，例如 Turbo Pascal，用于人机通讯，实现一定的算法。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。一个 Turbo Pascal 程序，就是数据描述和一组必须由计算机一步一步地执行的“语句”或规则的汇集。程序员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。下面就是一个简单而完整的 Turbo Pascal 程序，它要求从键盘键入一个正整数作为圆半径，计算圆的周长及面积，并将结果显示输出。

```
program sample01;
{calculate circumference and area of a circle}
const
  pi=3.14;
var
  radius:integer;
  circumference,area:real;
begin
  write('Enter radius of a circle: '); {whole number}
  readln(radius);
  circumference:=2*pi*radius;
  area:=pi*radius*radius;
  writeln('Radius of the circle is: ',radius);
  writeln('Circumference of the circle is: ',
    circumference:12:5);
  writeln('Area of the circle is: ',area:12:5);
end.
```

上述示例描述了一个 Turbo Pascal 程序的基本结构。它形如

```
program 程序名;
  说明部分
begin
  语句1;
  语句2;
  :
end.
```

第一行称为程序首部，它包含了一个 Turbo Pascal 专用词汇——program，接着是程序员给该程序所取的名字，如 sample01，最后是分号“;”，它表示程序首部结束，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。

用花括号“{”与“}”括起的内容是注释。

说明部分用于定义或说明程序中要用到的数据，如常量、变量等。本程序第三

行是常量定义，第五至七行是变量说明。今后将会看到，在一个程序的说明部分，除了常量定义、变量说明外，还可有类型定义、标号说明、子程序（过程或函数）定义等。若程序要用到已知的库单元，则还应书写相应的 `uses` 子句。但是，一个简单的程序可以不包含说明部分，即程序的说明部分亦是可选的。

紧接着程序说明部分的是程序的语句部分。语句部分定义了该程序所需执行的动作。它必须以一个 Turbo Pascal 专用词汇 `begin` 开始，以另一专用词汇 `end` 结束，其间是一些执行具体动作的语句，以分号作为语句分隔符。`begin` 与 `end` 应配对出现，这是每一个 Turbo Pascal 程序都必须有的。`end` 之后紧跟的一个句点“.”既表示语句部分的结束，亦表示该程序的结束。`end` 前的分号允许省略。

在计算机上通过键盘键入上述程序，经编译后便可运行。下面是运行示例。为直观起见，在程序运行中由程序员所键入的信息下面加一横线，而其余部分则都是计算机显示输出的内容。

```
Enter radius of a circle: 10↵ (↵表示回车键)
Radius of the circle is: 10
Circumference of the circle is: 62.80000
Area of the circle is: 314.00000
```

1.3 基本字符集与标识符

1.3.1 基本字符集

Turbo Pascal 的基本字符集由字母、数字和专用字符构成，它们均属于 ASCII 字符集（参见附录一）中的字符。

字母：英文大写字母 A 至 Z，小写字母 a 至 z

十进制数字：阿拉伯数字 0 至 9

十六进制数字：阿拉伯数字 0 至 9，英文字母 A 至 F，或 a 至 f

专用字符：`+` `-` `*` `/` `=` `<` `>` `[` `]` `.` `,` `(` `)` `:` `;` `^` `@` `{` `}` `$` `#`

某些运算符和分隔符系由两个专用字符组成，如

赋值运算符：`:` `=`

关系运算符：`<>` `<=` `>=`

子界分隔符：`..`

还有一些替代符号，亦由两个专用字符组成，如：

`(` `*` `)` `(` `.` `)`

它们分别等价于专用字符

`{` `}` `[` `]`

此外，常用到的字符还有空格符，单引号字符，下划线字符等。

1.3.2 标识符

标识符用以标记名字。程序中要用到许多名字，表示诸如常量、变量、标号、类型、过程、函数、记录字段、库单元等，这些名字都是用标识符来标记的。在 Turbo Pascal 中，标识符的形成规则是：它必须以英文字母或下划线打头，后面可跟英文字母、阿拉伯数字或下划线的任意组合。下面是一些符合规则的标识符示例：

X	U235	sample01	Department.of.Defense
BirthDate	S1000-000-000-000		
big----	number		不限制下划线连用
distance-from-	Changsha-		不限制以下划线结尾
1st			不限制以下划线打头

一个标识符的长度是任意的，但仅前63个字符有效。下述标识符
this-identifier-is-very-long-but-is-still-quite-legal
由53个字符组成，它们都是有效字符。

非法的标识符示例：

1st	不允许以数字打头
What-is-this?	不允许问号符
Department of Defense	不允许空格符

Turbo Pascal 对标识符中的英文大、小写字母不予区分，故写 sample01, SAMPLE01 或者 Sample01 都视作同一名字。本书大部分场合均使用小写字母，在一些文字信息或注释中出现大写字母，某些语句编码中则使用大小写混合形式。

有一些标识符在 Turbo Pascal 中具有特定含义，称它们为保留字。程序员不能将它们另作别用。程序 sample01 中的 program, const, var, begin 和 end 均为保留字。在标准 Pascal 中，这样的保留字共有 36 个，但 Turbo Pascal 的保留字有所扩充。以下就是 Turbo Pascal 的保留字表，其中打*号的是扩充的保留字。

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

编译器若发现程序中有使用上述保留字作为用户定义的标识符时，则将发布出错信息。

Turbo Pascal 还定义了一些标准标识符，用来标识系统预定义的类型、常量、变量、过程、函数等。它们与保留字略有不同，程序员可对它们重新定义。但最好不要这样做，以免混淆。上述程序示例中的 integer, real, write, readln, writeln 均为标准标识符。

1.4 标准标量类型

Turbo Pascal 有丰富的数据类型。程序中所用到的变量,需在说明部分加以说明。除了以某标识符作为它的名字之外,还必须指明该变量所属的数据类型。数据类型约束了该变量所能取的值的集合以及对它所能执行的操作。本节所涉及的仅是Turbo Pascal 中最基本的几种数据类型:整型(integer)、实型(real)、布尔型(boolean)和字符型(char)。它们都是系统预定义的简单数据类型,称为标准标量类型。integer, real, boolean和char均是标准标识符。

1.4.1 整型(integer)

一个整型数据用来存放整数。Turbo Pascal支持五种预定义整型数据类型: integer, byte, shortint, word, longint。每一种类型规定了相应整数的取值范围以及所占内存字节数(一个字节为8个二进位):

类 型	取 值 范 围	占字节数
integer	-32768..32767	2
byte	0..255	1
shortint	-128..127	1
word	0..65535	2
longint	-2147483648..2147483647	4

Turbo Pascal还预定义了两个与整型数据取值有关的常量标识符maxint和maxlongint,其值分别为32767和2147483647。程序中可直接引用它们。此外, Turbo Pascal 允许使用十六进制的整型数据。为规定一个整型值是十六进制,需在它之前加一个字符"\$"。例如\$27即表示一个十六进制的整数,其值等于十进制的39。又\$3F的值等于十进制的63。

1.4.2 实型 (real)

一个实型数据用来存放实数。Turbo Pascal支持五种预定义实型数据类型: real, single, double, extended, comp。其中 single, double, extended和comp仅适用于配置有8087数字协处理器的计算机。每一种类型规定了相应实数的取值范围;其所占内存字节数,以及它们所能达到的精度即有效数字位数:

类 型	取 值 范 围	占字节数	精 度
real	2.9×10^{-39} 至 1.7×10^{38}	6	11 至 12
single	1.5×10^{-45} 至 3.4×10^{38}	4	7 至 8
double	5.0×10^{-324} 至 1.7×10^{308}	8	15 至 16
extended	1.9×10^{-4931} 至 1.1×10^{4932}	10	19 至 20
comp	$-2^{63}+1$ 至 $2^{63}-1$	8	19 至 20

上表中前四种类型给出的是其绝对值的取值范围。对于此类实型数据，若其绝对值大于上界，则产生上溢；绝对值小于下界，则产生下溢。下溢导致结果为0。

comp类型的取值范围是 $-2^{63}+1$ 至 $2^{63}-1$ 之间整数，相当于十进制的 -9.2×10^{18} 至 9.2×10^{18} 。由于comp类型的数据表示成二进制形式的数，这种类型的变量有时处理起来比较方便。

1.4.3 布尔型 (boolean)

一个布尔型数据用来存放逻辑值，或称布尔值。总共只有两个可能的布尔值：false和true，它们分别表示逻辑假和逻辑真。false, true都是标准标识符。每个布尔型数据在内存中占1个字节。规定false < true。

1.4.4 字符型 (char)

一个字符型数据用来存放ASCII字符集内的某个字符。该字符集共有256个字符，但其中有一些字符不能在标准显示设备如视屏显示器或宽行打印机上显示或印刷输出，即非印刷字符。在计算机内部，字符集的元素是以该元素在字符集内的顺序位置来标记的，位置取值范围为0至255。称这些整数为字符在字符集内的序数值或序号。每个字符型数据在内存中占1个字节。字符用单引号括起，表示字符常数，如'a'，'3'，'?'。字符常数可按字符的序数值确定大小关系，如'Y' < 'Z'。

Turbo Pascal 提供了另外两种方式来表示一个字符常数：使用"^"符，或者"#"符。首先，ASCII字符集中序号为0至31的那些字符称为控制字符，它们具有特定的意义。Turbo Pascal允许以"^"符后跟一个相应字母（或字符）来表示某一控制字符，比如^G表示序号为7的控制字符，^M表示序号为13的控制字符。这种方式仅适用于表示控制字符。其次，可以用"#"符后跟某字符在ASCII字符集中的序号来表示该字符，例如#7与^G一样，#65表示字符'A'，#218表示某一特殊图形字符'r'。

1.5 注释、常量定义和变量说明

1.5.1 注释

为了增加程序的易读性，可以插入若干注释。注释用花括号{"和"}括起，可加在程序中任何能使用分隔符的地方。编译器对注释的内容予以跳略。注释可跨越若干行。记号{"和"}可分别用"(*和*)"代替。例如程序sample01第二行注释亦可写成

```
(* calculate circumference and
   area of a circle *)
```

写注释应注意花括号配对。如果键入时误将"}"打成")"或"]"，则编译器会认为下面所有内容均属于注释，因为它未遇到"}"。

由于表明注释的记号有两套，这就提供了注释嵌套的有限形式，因为若注释以"(*"开始，它将忽略所有的花括号，反之亦然。下面是注释嵌套的示例

```
(* This is a comment. {This is also a comment.
  It terminates with a matching brace like this.}
  Here is the end of this comment. *)
```

在一个注释中,若符号"{"或"("后紧跟一个符号"\$",则表示这是一个编译指示。关于编译指示,参见9.2节。

1.5.2 常量定义

常量定义引入标识符作为常数值同义词。它由保留字 `const` 开始,后随一个或一系列常量赋值,其间用分号隔开。每个常量赋值由一个标识符后跟等号及常数。程序 `sample01` 中仅定义一个常量赋值,即引入标识符 `pai` 作为常数值 3.14 的同义词。称 `pai` 为常量标识符。

多个常量赋值的常量定义示例:

```
const
  minimum = 1,
  maximum = 256,
  fourth = 0.25,
  first_char = 'A',
  last_char = 'Z',
  comm_port = $3F8;
```

这些常量标识符被分别赋以整常数、实常数、字符常数等。常量标识符 `comm_port` 被赋以十六进制常数 `$3F8`, 等于十进制的 1016。

常量标识符在程序中可当变量标识符一样被引用,但不允许改变其值。

1.5.3 变量说明

Turbo Pascal 规定,程序中所用到的每个变量,在使用之前必须命名并指明其所属数据类型,即需进行变量说明。变量说明以保留字 `var` 开始,后跟一个或若干个标识符,它们之间用逗号隔开,然后是冒号和类型。变量一经说明,编译器就会为它们按所属类型分配一定的内存单元。程序 `sample01` 中,标识符 `radius` 被说明为 `integer` 类型变量,而 `circumference`, `area` 均被说明为 `real` 类型变量。

变量说明的书写格式是比较灵活的。比如,下述变量说明

```
var
  radius: integer;
  circumference, area: real;
```

亦可写成

```
var
  radius: integer;
  circumference: real;
  area: real;
```

或者

```
var
  radius: integer; circumference, area: real;
```

等等。

变量标识符一经说明便可被引用,并可不断改变其值,但以最后一次赋值为该变量