

信息学(计算机) 国际奥林匹克 Turbo Pascal 6.0

向期中
吴耀斌 编著
雷方桂
陈松乔 审定

中南工业大学出版社

信息学（计算机）奥林匹克

Turbo Pascal 6.0

◆ 向期中
◆ 吴耀斌 编 著
◆ 雷方桂

● 陈松乔 审 定

中南工业大学出版社

内 容 简 介

Turbo Pascal 是目前国内外最受欢迎的 Pascal 系列之一, 1993 年 Turbo Pascal 6.0 被指定为国际奥林匹克信息学(计算机)竞赛使用的三种语言之一。多年的实践证明, 使用 Turbo Pascal 6.0 的选手在奥林匹克信息学(计算机)竞赛中成绩最为理想, 并且语言又最适合于教学。

本书以 Turbo Pascal 6.0 为基础, 全面讲述了 Turbo Pascal 6.0 的各种概念与特性, 结合教学实践介绍了程序设计方法和技巧, 并重视培养学生良好的程序设计风格的养成, 同时专门介绍了上机操作和程序调试方法。全书共分 12 章, 内容充实, 编排合理, 并精心编选了大量的例题, 习题, 其中较大部分是在算法设计上有代表性和具有突出特点的例题, 习题配合例题与所学内容进行巩固练习与提高, 达到了开拓读者思路的目的。

本书既可作为 PASCAL 语言的教科书, 又可作为信息学(计算机)奥林匹克竞赛培训用书, 同时也可做为专业技术人员的参考书。书中所有例题都经 Turbo Pascal 6.0 系统上机验证。

为了方便上机, 可向读者提供 Turbo Pascal 6.0 系统盘和本书的例题与习题解答盘。

【湘】新登字 010 号

信息学(计算机)国际奥林匹克

Turbo Pascal 6.0

向期中等 编 著

责任编辑: 肖梓商

*

中南工业大学出版社出版发行

中南工业大学发强电子科技有限公司电脑排版

湖 南 大 学 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: 17 字数: 406 千字

1994 年 5 月第 1 版 1995 年 3 月第 2 次印刷

印数: 5201—11200

ISBN7-81020-650-8 / TP · 035

定价: 10.00 元

本书如有印装质量问题, 请直接与生产厂家联系解决。

序

发展计算机技术和信息技术，是当代发展科学技术和推动生产发展的关键。普及计算机教育、提高计算机教育水平对我国现代化建设具有十分重要的意义，而青少年是跨世纪人才，是下一世纪我们计算机事业的主要力量。邓小平同志指出：“计算机的普及要从娃娃做起”。明确地指出了我国计算机普及教育和计算机事业的发展方向和策略。改革开放以来，计算机技术及其应用得到了迅速发展，相应地计算机教育也正在大学、中学、甚至小学普及开来，计算机技能和外语能力一样，已成为当前我们青年就业和各单位选择人才的重要条件。广大青少年，不论是否从事计算机事业，均以极大的兴趣和进取精神，学习计算机技术，因此，帮助青少年学习和掌握计算机基本理论、基本技术，提高他们设计程序的能力，是我们计算机工作者义不容辞的任务。

国际奥林匹克信息学（计算机）竞赛是世界各国重视在青少年中普及计算机教育的产物，它是在国际范围内青少年智慧解题和操作能力的竞赛。我国参赛五届，取得了举世瞩目的优异成绩，在这些成绩中，计算机普及教育功不可抹，然而普及教育的进一步发展，需要更多的投入，需要精心组织，需要更多的优秀教材和读物，这本教材的出版，正好适应了这一迫切的需要。

Turbo Pascal 6.0 是国际奥林匹克信息学（计算机）竞赛指定使用的三种计算机程序设计语言之一，这种语言版本以其良好的结构、高超的性能和易于掌握和使用等优点，深受计算机工作者，特别是参赛青少年的欢迎。向期中等老师编著的这本书，全面系统地介绍了这一优选的语言版本，内容全面、深入浅出，文学通俗易懂，特别是精心选编了大量有代表性的例题和习题，这对于加深读者对原理的理解、开拓思路，提高解题和编程能力具有很好的作用。因此是一本很好的普及计算机教材，即可用于中学、大学计算机语言教学，又可作为奥林匹克信息学（计算机）竞赛培训用书。与国内国际的信息学奥林匹克接轨。

经验证明，青少年从学习一门计算机语言入手来学习计算机是一条可行的捷径。通过计算机语言学习，结合解题，编写程序，上机操作，分析结果，不仅能掌握计算机的基本知识和基本技能，而且能提高逻辑思维能力、抽象化形式化描述问题的能力，锻炼独立分析问题和解决问题的能力，对青少年智力和能力的培养有着十分重要的意义。

当前，改革开放给祖国带来了勃勃生机，给计算机技术和计算机应用的发展，特别是计算机普及教育的发展提供了前所未有的机遇，愿广大青少年和我国计算机工作者一道，为祖国的繁荣昌盛，为普及计算机事业，尽心尽力，奋发图强，作出贡献！

湖南省计算机学会副理事长
湖南省奥林匹克竞赛委员会计算机学分会主任
中南工业大学计算机科学系主任、教授

陈 松 乔

1994.5.1

前 言

Turbo Pascal 是目前在国内外最受欢迎的 Pascal 系列之一,它实际上已成为微机上的 Pascal 的标准。Turbo Pascal 继承了 Niklaus Wirth 教授所设计的标准 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,有利于培养良好的编程风格,易于体现结构化程序设计的思想。与此同时,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。特别是它扩充了彩色、声音、图形、窗口、类型常数、编译指示、包含文件和与汇编语言连接等许多功能,1990 年推出的 Turbo Pascal 6.0 是最新的版本,扩充并增加一些新颖的特性,较突出的是增加了调试功能,引入了面向对象的程序设计与崭新的集成开发环境和用户界面。

Turbo Pascal 6.0 是国际奥林匹克信息学(计算机)竞赛指定使用的三种程序设计语言(Quick Basic4.5、Turbo c++2.0、Turbo Pascal 6.0)之一,近几年全国竞赛的实践证明,使用 Turbo Pascal 6.0 的选手成绩最为理想,并且该语言又最适合于教学。

本书以 Turbo Pascal 6.0 为基础,全面讲述了 Turbo Pascal 6.0 的各种概念与特性,结合教学实践介绍了程序设计方法和技巧,并重视培养学生养成良好的程序设计风格。本书内容充实,编排合理,并精细编选了大量的例题,习题,其中较大部分是在算法设计上有代表性和具有突出特点的例题,习题配合例题与所学内容进行巩固练习与提高,达到了开拓读者思路的目的。它既可作为普及计算机的教材,又可以作为奥林匹克信息学(计算机)竞赛的培训用书。书中所有例题都经上机验证。

为了方便上机,可以向读者提供 Turbo Pascal 6.0 系统软盘和本书的例题与习题解答盘。

本书由湖南省计算机学会副理事长、湖南省奥林匹克选拔赛竞赛委员会计算机学分会主任委员、中南工业大学计算机科学系主任、计算机及信息技术研究所所长陈松乔教授主审。

参加本书指导和编写的还有:刘明德、陶立新、崔佩珍、曾春娥、丁为信、周喜华、符新培、吴王勺文、李树元、徐信成、梁京沙、李吉雄、郑建义。

本书得到了湖南省计算机学会、中南工业大学计算机科学系、发强电子科技有限公司、湖南省教育委员会普教处、湖南省奥林匹克选拔赛竞赛委员会计算机学分会的大力支持和帮助,在此一并表示衷心的感谢!

由于水平、时间有限,不妥之处在所难免,欢迎读者批评指正。

作 者

1994 年 1 月于长沙

目 录

1	TURBO PASCAL 基本知识	1
1.1	Turbo Pascal 6.0 简介	1
1.2	Turbo Pascal 程序的组成	2
1.3	字符集和符号	4
1.4	数据类型的概念	5
1.5	常量和变量	5
1.6	标准数据类型	7
1.7	表达式	12
1.8	上机操作	13
1.8.1	Turbo Pascal 6.0 的安装	13
1.8.2	初次在 Turbo Pascal 系统上编程	14
1.8.3	集成开发环境(IDE)使用指南	15
1.8.4	命令行编译器	22
1.8.5	Turbo Pascal IDE 编辑器	22
1.8.6	Turbo Pascal 的特殊编辑功能	24
1.9	调试程序	27
1.9.1	调试的预备知识	28
1.9.2	编译器指示	30
1.9.3	错误信息	31
	习题一	37
2	算法简介	39
2.1	算法的概念	39
2.2	简单算法举例	39
2.3	算法的特性	42
2.4	算法的表示	43
2.4.1	自然语言	43
2.4.2	程序流程图	43
2.4.3	伪代码	43
2.4.4	三种基本结构和改进的流程图	46
2.4.5	N-S 结构流程图	48
2.4.6	PAD 图	51
2.5	结构化程序设计的方法	52
2.6	程序设计风格	54
	习题二	55

3	顺序结构程序设计	56
3.1	赋值语句	56
3.2	输入和输出过程	57
3.2.1	读语句	57
3.2.2	写语句	58
3.2.3	字符类型变量输出	59
3.2.4	整数类型变量输出	60
3.2.5	实数类型变量输出	61
3.2.6	布尔类型变量输出	62
	习题三	64
4	分支结构程序设计	66
4.1	Turbo Pascal 中的逻辑运算	66
4.1.1	布尔常量和布尔变量	66
4.1.2	布尔表达式	67
4.2	复合语句	68
4.3	if 语句	69
4.3.1	条件语句概念及应用	70
4.3.2	条件语句嵌套	73
4.4	Case 语句 (分情况语句)	76
	习题四	81
5	循环结构程序设计	84
5.1	用 for 语句实现循环结构	84
5.2	用 while 语句实现循环结构	90
5.3	用 repeat—until 语句实现循环结构	92
5.4	循环的嵌套	94
5.5	goto 语句	97
	习题五	99
6	枚举与子界类型	101
6.1	Turbo Pascal 中类型概念说明	101
6.2	枚举类型	102
6.2.1	引入枚举类型的必要性	102
6.2.1	枚举类型的定义、运算规则和输入输出方法	103
6.3	子界类型	109
6.3.1	引入子界类型的必要性	109
6.3.2	子界类型的定义和运算规则	110
6.4	类型间关系	114
	习题六	117
7	数组类型	118
7.1	一维数组	118

7.1.1	引入数组的必要性	118
7.1.2	一维数组的定义	119
7.1.2	一维数组类型常数定义	121
7.1.4	一维数组的变量的赋值操作	121
7.2	多维数组	133
7.3	字符数组与字符串类型	138
7.3.1	字符数组	138
7.3.2	字符串类型	140
	习题七	148
8	过程与函数	151
8.1	过程	151
8.1.1	无参过程	152
8.1.2	带参过程	153
8.1.3	过程的引用	153
8.1.4	参数传递	154
8.2	函数	157
8.3	嵌套和标识符的作用域	160
8.3.1	过程与函数嵌套的概念	160
8.3.2	标识符的作用域	161
8.4	向前引用与递归调用	162
8.4.1	递归的概念	162
8.4.2	递归过程与函数	164
8.5	间接递归与向前引用	174
	习题八	177
9	集合与记录类型	178
9.1	集合类型	178
9.1.1	集合的基本知识	178
9.1.2	Turbo pascal 语言中的集合	179
9.2	记录类型	190
9.2.1	记录类型定义	190
9.2.2	记录的嵌套	191
9.2.3	记录类型变量及其使用	191
9.2.4	记录类型常数定义	192
9.2.5	开域语句	193
9.2.6	记录的变体简介	200
	习题九	202
10	文件类型	204
10.1	磁盘文件的逻辑组织	204
10.2	随机文件	206

10.2.1	随机文件定义	206
10.2.2	打开一个随机文件	207
10.2.3	随机文件的读写	208
10.2.4	关闭文件及文件处理函数	209
10.3	随机文件的应用举例	210
10.4	TEXT 文件	212
10.5	无类型文件	216
10.6	标准文件	217
习题十		217
11	动态数据结构	219
11.1	指针类型与动态变量	219
11.1.1	指针类型定义与变量说明	220
11.1.2	指针类型常数定义	221
11.1.3	new 标准过程与指针变量	221
11.1.4	动态变量和访问	222
11.1.5	dispose 标准过程	223
11.2	链表	224
11.2.1	链表与递归数据结构	224
11.2.2	链表的建立	226
11.2.3	链表的插入与删除	229
11.2.4	链表的检索	233
11.2.5	环形链表结构	236
11.2.6	双向链表结构	236
11.3	二叉树	239
11.3.1	二叉树的概念	239
11.3.2	二叉树的遍历	240
11.3.3	二叉树的建立与插入	241
习题十一		246
12	单元与作图介绍	247
12.1	单元介绍	247
12.2	作图	249
12.2.1	正文模式与正文窗口	249
12.2.2	图形模式与图形窗口	251
12.2.3	图形功能	253
12.2.4	龟作图	255
12.3	音响	259
习题十二		261
参考文献		262

1 TURBO PASCAL 基本知识

1.1 Turbo Pascal 6.0 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。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。1974 年, 他和 K·Jensen 联名发表了著名的修改报告和用户手册, 描述了标准 Pascal, 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。

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, 数据结构和控制结构比较丰富, 程序可读性好, 查错能力强, 有利于培养学生严谨、清晰的程序设计风格和良好习惯, 并使他们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。

在 Pascal 问世以来的 20 余年间, 先后产生了适合于不同机型的各种各样版本。其中影响最大的莫过于 Turbo Pascal 系列软件。

本书适用于 Turbo Pascal 6.0。为简化起见, 以后将用 Turbo Pascal 作为 Turbo Pascal 6.0 的代名词。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。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, 也是国际和全国青少年信息学竞赛指定的语言之

一。从历届竞赛的情况看，它是最能出成绩的语言和选手最欢迎的语言。

1.2 Turbo Pascal 程序的组成

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 12 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五部分: ①标号说明; ②常量定义部分; ③类型定义部分; ④变量说明部分; ⑤过程与函数说明部分。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 }
    usec 子句;
    说明部分;                       { 说明部分 }
begin
    语句1;
    语句2;
    ...
    语句n
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。

1.3 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 共有保留字 52 个。

(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 chr cos exp ln odd ord pred sin qrt succ trunc

标准过程：read readln write writeln

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 Year 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字英文字母, 但 GetDriverName 比 GETDRIVERNAME 与 getdrivename 更易读。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。

1.4 数据类型的概念

计算机处理的数据对象是一个广义的概念。比如 125、12.76 是数据, 'xiang qi 'zhong' 这一串字符也是数据。前者是数值数据, 后者是字符串数据, 是非数值数据。显然为了表示这些数据, 它们在内存中必须以不同方式存放, 为处理这些数据, 计算机对它们施加的运算也不同。为此 Turbo Pasca 语言建立了数据类型的概念, 对描述的数据进行分类。每一种数据类型定义了一个具有相同性质的数据集合。各种数据类型的数据具有不同的性质。程序中所用到的每一个数据, 包括常量和变量都有和它相联系的类型。由此决定了数据所具有的值, 也决定了对该数据所能进行的操作。

Turbo Pascal 语言中数据具有丰富的类型。按它们的特点可以分为简单类型、构造类型、指针类型和过程类型四大类。其中简单类型包括标准类型 (字符串类型、实数类型、整数类型、字符类型、布尔类型) 和用户自定义类型 (枚举类型、子界类型); 构造类型包括集合类型、数组类型、记录类型、文件类型、对象类型; 对于具有整数序数数据的整数类型、字符类型、布尔类型、枚举类型、子界类型合称为顺序类型。其中标准类型用语言系统预先定义的标准标识符表示; 整数类型用 integer 表示; 实数类型用 real 表示; 布尔类型用 boolean 表示; 字符类型用 char 表示。

1.5 常量和变量

(一) 常量 常量是指程序中一些具体的整数类型数、实数类型数和字符串。

(1) 整数类型数。如 3、-5、0 等;

(2) 实数类型数。如 3.1、-6.1E+2;

(3) 字符串。它是用单引号括起来的一串字符如 'book'、'96.5'、'=== '。用单引号括起来的单个字符可以看成字符数量为 1 的字符串如 'x'、'*'、'#' 等。在字符串内有单引号时要用两个连续单引号表示。如 chr(66) = "B'", 表示 chr(66) = 'B'。

以上列举的都可以作为常量在程序中使用，为了提高程序的可读性和使程序便于修改，在程序中往往用一些标识符来代表具体的常量。为此要按一定的语法形式定义它。

(二) 常量定义 在 Turbo Pascal 语言中，可以给一些常量取个名字用一个标识符代表它，这就是常量定义。经常量定义的标识符又称为常量标识符。例如， $Cost = 60$ ； $Blank = '□'$ 。常量定义要写在常量定义部分中。常量定义部分的一般形式：

Const

 <常量标识符 1> = <常量 1>;

 <常量标识符 2> = <常量 2>;

.....

 <常量标识符 n> = <常量 n>;

Const 是保留字，表示开始一个常量定义部分，其后可以有若干个常量定义，由语法图可知，这些常量定义之间要用“;”号分隔。例如：

Const

 Cost = 60;

 A = -Cost + 30;

 Pai = 3.14159;

 Blank = '□';

 Char1 = '* * *';

 Name1 = 'xiang qi zhong';

 Name2 = 'Y · H · RoseMary'

由上例可见常量可以是一个数、一个字符、一个字符串，也可以是一个常量标识符（可带有正负号）。Turbo Pascal 语言对常量定义有如下要求：

(1) 常量定义要放在程序的常量定义部分，即程序首部之后，执行部分之前。

(2) 必须遵循先定义后使用的原则，即只有已经定义的常量标识符，以后在程序中才能使用。一般称在常量定义中等号左面的常量标识符为定义性出现。以后在程序中的引用为使用性出现。例如：

Const

...

 A = 35; {这里常量标识符 A 是定义性出现}

 B = -A; {这里的常量标识符 A 是使用性出现，只是在上条中 A 已被定义了，这里才能引用它去定义 B}

如反过来 Const

...

 B = -A;

 A = 35;

则是非法的，因为常量标识符 A 使用在先，定义在后，违反了定义在前使用在后的原则。在 Turbo Pascal 语言中，只有指针类型例外，允许先使用后定义。

常量定义后，在书写程序语句时就可以引用它。即可以将程序中多次出现的常量不必直接写常量值，而以定义的常量标识符代替，这样一方面避免写错，另一方面也便于修改

程序。如要修改某一常量只修改常量定义（如 Cost = 60, 改为 Cost = 90）就将程序中所有这个常量都修改了。特别是对较大的程序这是很需要的。

Turbo Pascal 中预定义了如下几个常量标识符，它们可以直接在程序中使用：

Pi = 3.1415926536; MaxInt = 32767; False = 布尔假; True = 布尔真;

MaxLongInt = 2147483647;

（三）变量说明 在程序执行过程中其值可以改变的数据，称为变量。每个变量都要有一个名称，这就是变量名。变量名由用户自己定义，但必须符合标识符的规定。

在一个程序中，一个变量只能属于一种确定的数据类型。因此程序中出现每个变量都必须说明其数据类型。这样就规定了该变量的取值范围，也决定了对该变量所能执行的运算操作。

变量说明形式是：一个变量标识符或由逗号隔开的多个变量标识符在它的冒号“:”后面说明其数据类型。数据类型可以是标准数据类型标识符 integer、real、boolean 和 char，也可以是用户自定义的各种类型。在 Turbo Pascal 程序中，变量说明要写在变量说明部分中。变量说明部分一般形式为：

Var

〈变量说明 1〉;

〈变量说明 2〉;

.....

〈变量说明 n〉;

其中 Var 是保留字，表示一个变量说明部分开始。一个 Var 可以含有多个不同的变量说明，每个变量说明之间用分号隔开。有时称被分号隔开的变量说明为变量说明项。例如：

Var

a, b: Integer;

x, y: Real;

ch1: Char;

t, f: Boolean;

注意：不同类型的变量一般不能互相串用。这里还应指出，变量一经说明，系统就在计算机内存中为其分配一个存储空间。在程序中使用到变量时，就在相应的内存中存入数据或取出数据。一般称这种操作为变量的访问。

1.6 标准数据类型

Turbo Pascal 向程序设计者提供了丰富的数据类型，它们各用于专门的目的，但却都是由简单的、非构造型的数据类型所构成。本节介绍 Turbo Pascal 中最为基本的几种数据类型：整型、实型、布尔型和字符型。它们都是系统定义的简单数据类型，称为标准数据类型，其对应的名字称为标准标识符。

（一）整数类型 一个整型数据用来存放整数。整型的数据可以是正整数、负整数和整数零。Turbo Pascal 支持五种预定义整型，它们是短整型（ShortInt），整型（Integer），长整型（LongInt），字节型（Byte）和字型（Word），每一种类型规定了相

应的整数取值范围以及所占内存字节数（一个字节为 8 个二进制位），如表 1-1 所示。因此在具体编程定义变量类型时，要根据它们的特点选用适当的类型，以达到理想的效果。当两个不同范围类型的操作数进行运算时，得到的结果属于较小范围的类型。

表 1-1 整型数取值范围

类 型	取值范围	占字节数	格 式
ShortInt	-128..127	1	带符号 8 位
Integer	-32768..32767	2	带符号 16 位
LongInt	-2147483648..2147483647	4	带符号 32 位
Byte	0..255	1	无符号 8 位
Word	0..65535	2	无符号 16 位

Turbo Pascal 还要两个预定义整型常数标识符 MaxInt 和 MaxLongInt，它们各表示确定的数值，如表 1-2 所示，程序中可直接引用。

表 1-2 整型常数标识符

整型常数标识符	值	类 型
MaxInt	32767	Integer
MaxLongInt	2147483647	LongInt

Turbo Pascal 中的整型数必须按规定严格书写，它只能由正、负号和数字序列组成，不允许有其它符号出现。在 Turbo Pascal 语言中，数一般以十进制表示。例如 -267, 6790, 3, 31561 等都是合法的整型数。还允许整型数采用十六进制，在数字前加 \$ 表示十六进制整数。如 \$ 10，表示 $(10)_{16} = (16)_{10}$ ；\$ 1A，表示 $(1A)_{16} = (26)_{10}$ 。

Turbo Pascal 语言规定可以对整型数据进行如下操作：

(1) 允许两个整型数使用下列运算符进行运算。

① 算术运算符。+、-、*、Div、Mod。它们分别表示加、减、乘、整除和取余。这五种运算，要求参加运算的两个数都是整型，运算结果也是整型。前三种运算与一般的算术运算加、减、乘相同。Div 整除运算，是两个整型数相除取整数部分（商的整数部分），得到整型结果。Mod 取余运算，是两个整型数相除取余数。余数的符号与被除数符号相同。例如：

$$3 \text{ Div } 2 = 1; \quad 5 \text{ Div } 7 = 0; \quad (-12) \text{ Div } (-5) = 2; \quad (-14) \text{ Div } 5 = -2;$$

$$23 \text{ Mod } 3 = 2; \quad (-18) \text{ Mod } (-6) = 0; \quad 18 \text{ Mod } (-5) = 3$$

由此可见， $a \text{ Mod } b$ ，所得结果的符号与 a 相同，其值（绝对值）在 $0 \sim |b|-1$ 之间。运算符 Mod 与 Div 之间有如下关系：

$$a \text{ Mod } b = a - (a \text{ Div } b) * b \quad (b \neq 0)$$

其中 Mod 运算结果的符号与 a 的符号相同。利用以上运算可以对正整数进行分离。如：n 为三位数 531 可用下法分离出它的个、十、百位。

$$531 \text{ Mod } 10 = 1 \quad (\text{个位数})$$

$$(531 \text{ Mod } 100) \text{ Div } 10 = 3 \quad (\text{十位数})$$

$$531 \text{ Div } 100 = 5 \quad (\text{百位数})$$