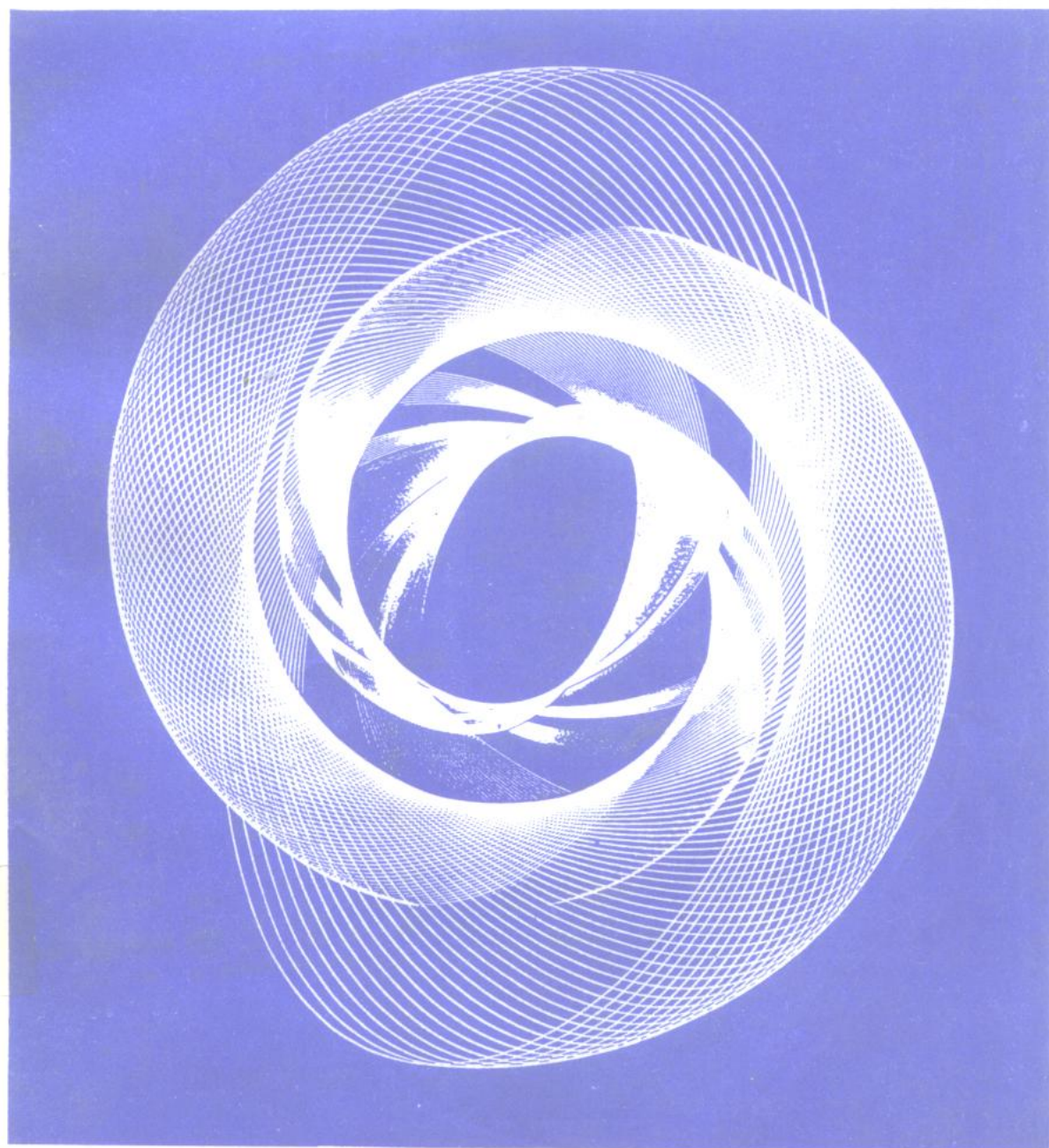


# Turbo Pascal

# 大全

姚庭宝 张 帆 编著



電子工業出版社

# Turbo Pascal 大全

姚庭宝 张 帆 编著

电子工业出版社

(京)新登字 055 号

## 内 容 简 介

本书全面而详尽地介绍了 Turbo Pascal 语言的程序设计技术与具体应用。书中包含众多的实际示例,实用性极强。全书共十六章,它们是: Turbo Pascal 初阶;基本元素与特性;条件控制与循环控制结构;字符串和数组;记录与集合;文件;指针;过程和函数;结构化程序设计思想;通用程序设计示例;单元;图形及其应用; Turbo Pascal 高级特性;与汇编语言的连接;覆盖技术;面向对象的程序设计。另设七个附录: Turbo Pascal 集成开发环境;编译指示一览表; Turbo Pascal 实用程序; Turbo Pascal 标准子程序库;编译与运行出错信息;一些流行的 Turbo Pascal 工具箱软件简介; ASCII 码表及键盘返回代码。大多数章末都有小结,并给出一个程序设计问题,各章均配置了练习。

本书在选材和编写中顾及了各个层次的需要,不仅可供广大软件开发人员作为参考书,也可供大专院校学生及其他学习计算机的人员作为教材或教学参考书。

### Turbo Pascal 大全

姚庭宝 张 帆 编著

特约编辑 贾良红

责任编辑 潘 海

\*

电子工业出版社出版(北京市万寿路)

电子工业出版社发行 各新华书店经售

北京市顺新印刷厂印刷

\*

开本: 787×1092 毫米 1/16 印张: 28 字数: 716.8 千字

1993 年 2 月第 1 版 1993 年 2 月第 1 次印刷

印数: 8000 册 定价: 16.00 元

ISBN7-5053-1849-7/TP·438

## 前 言

Turbo Pascal 是目前在国内外最受欢迎的 Pascal 系统之一。它实际上已成为微机的 Pascal 标准。Turbo Pascal 继承了 Niklaus Wirth 教授所设计的标准 Pascal 的许多优点,如整个语言紧凑整齐,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,有利于培养良好的编程风格,易于体现结构化程序设计思想,适合于用作教学语言,等等。与此同时,Turbo Pascal 本身还具有许多鲜明的优点。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,如彩色,声音,图形,窗口,类型常数,编译指示,包含文件,与汇编语言的连接等等。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开发环境,并引入单元的概念以及各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。1988年至1990年相继推出的 Turbo Pascal 5.0,5.5 以及 6.0,每一种新的版本,都扩充并增加了一些新颖的特性。较突出的是,Turbo Pascal 5.0 增加了调试功能,Turbo Pascal 5.5 引入了面向对象的程序设计,而 Turbo Pascal 6.0 给出了崭新的集成开发环境用户界面,并提供一种面向对象的应用框架和库——Turbo Vision,以高效地开发实用的、具有连贯一致用户界面的应用程序。Turbo Pascal 已经成为在教学、科研、开发系统软件和应用软件中的一个颇为理想的工具。

本书主要以 Turbo Pascal 5.5 为基础,部分内容引入了 Turbo Pascal 6.0 的新增特性。全书既全面讲述 Turbo Pascal 最新版本的各种概念与特性,又致力于介绍程序设计的基本方法与技巧,并重视学生良好程序设计风格的养成。本书内容充实,编排合理,示例众多,实用性强。全书共十六章,另设七个附录。为了更能体现大学本科生教材的特点,大多数章末都有小结,并给出一个程序设计问题以综合运用本章概念。各章均配置了练习。

为方便用户,促进交流,我们已将书中所有程序示例的源程序装入软盘。需要者可直接与我们联系。

在本书的编写过程中,杜秀荣和董定君两位女士给予了我们全力的支持,在此表示衷心感谢。

贾良红对全书进行了文字审校和编辑加工。限于我们的水平,本书定有不当之处,竭诚欢迎广大专家、读者对本书提出批评、建议。

姚庭宝 张 帆

一九九二年五月

长沙国防科技大学七系(邮编 410073)

# 目 录

<b>第一章 Turbo Pascal 初阶</b> .....	(1)
1.1 关于 Turbo Pascal .....	(1)
1.2 Turbo Pascal 系统安装 .....	(2)
1.3 使用集成开发环境 .....	(3)
1.4 编辑、编译和运行一个 Turbo Pascal 程序 .....	(6)
1.5 程序调试 .....	(6)
1.6 编译指示简介 .....	(7)
练习 .....	(8)
<b>第二章 基本元素与特性</b> .....	(9)
2.1 程序基本结构 .....	(9)
2.2 基本字符集与标识符 .....	(11)
2.2.1 基本字符集 .....	(11)
2.2.2 标识符 .....	(11)
2.3 标准标量类型 .....	(12)
2.3.1 整型 .....	(12)
2.3.2 实型 .....	(13)
2.3.3 布尔型 .....	(13)
2.3.4 字符型 .....	(14)
2.4 注释、常数定义和变量说明 .....	(14)
2.4.1 注释 .....	(14)
2.4.2 常数定义 .....	(15)
2.4.3 变量说明 .....	(16)
2.5 枚举类型与子界类型 .....	(16)
2.5.1 枚举类型 .....	(16)
2.5.2 子界类型 .....	(18)
2.6 简单类型常数 .....	(19)
2.7 表达式 .....	(20)
2.7.1 运算符和优先级 .....	(20)
2.7.2 一些常用的预定义标准函数 .....	(22)
2.7.3 类型转换 .....	(24)
2.7.4 算术表达式、关系表达式和布尔表达式 .....	(25)
2.7.5 常数表达式 .....	(25)
2.8 简单输入、输出及赋值语句 .....	(25)

2.8.1	输入过程 Read 和 ReadLn .....	(26)
2.8.2	输出过程 Write 和 WriteLn .....	(26)
2.8.3	标准标量类型数据的输出格式 .....	(26)
2.8.4	赋值语句 .....	(27)
2.9	程序设计风格 .....	(28)
	练习 .....	(29)
<b>第三章</b>	<b>条件控制与循环控制结构 .....</b>	<b>(31)</b>
3.1	简单 if 语句 .....	(31)
3.2	复合语句 .....	(32)
3.3	if 语句的扩展与嵌套 .....	(33)
3.4	case 语句 .....	(35)
3.5	while 语句 .....	(36)
3.6	repeat 语句 .....	(38)
3.7	for 语句 .....	(38)
3.8	多重循环 .....	(40)
3.9	goto 语句及标号 .....	(42)
3.10	小结与程序设计问题 .....	(43)
3.10.1	小结 .....	(43)
3.10.2	程序设计问题 .....	(44)
	练习 .....	(45)
<b>第四章</b>	<b>字符串和数组 .....</b>	<b>(48)</b>
4.1	字符串类型定义和变量说明 .....	(48)
4.1.1	字符串类型定义 .....	(48)
4.1.2	字符串变量说明 .....	(48)
4.1.3	字符串长度 .....	(49)
4.1.4	字符串与字符 .....	(49)
4.1.5	字符串的常数定义和类型常数定义 .....	(49)
4.2	字符串表达式和赋值语句 .....	(50)
4.2.1	字符串表达式 .....	(50)
4.2.2	字符串赋值语句 .....	(50)
4.2.3	字符串变量的输入输出 .....	(51)
4.3	字符串标准函数和过程 .....	(51)
4.3.1	字符串标准函数 .....	(51)
4.3.2	字符串标准过程 .....	(52)
4.4	一维数组类型定义和变量说明 .....	(54)
4.4.1	一维数组类型定义 .....	(54)
4.4.2	一维数组变量说明 .....	(55)
4.4.3	一维数组类型常数定义 .....	(55)

4.4.4	一维数组变量的赋值操作 .....	(56)
4.5	一维数组应用 .....	(56)
4.6	多维数组 .....	(58)
4.7	小结与程序设计问题 .....	(61)
4.7.1	小结 .....	(61)
4.7.2	程序设计问题 .....	(61)
练习	.....	(64)
<b>第五章</b>	<b>记录与集合 .....</b>	<b>(66)</b>
5.1	记录类型定义和变量说明 .....	(66)
5.1.1	记录类型定义 .....	(66)
5.1.2	记录变量说明 .....	(67)
5.1.3	记录类型常数定义 .....	(67)
5.2	记录的赋值 .....	(68)
5.3	with 语句和记录的嵌套 .....	(69)
5.3.1	with 语句 .....	(69)
5.3.2	记录的嵌套 .....	(71)
5.4	变体记录 .....	(72)
5.4.1	变体记录类型定义和变量说明 .....	(72)
5.4.2	判别式联合与自由联合变体记录类型 .....	(74)
5.5	集合类型定义和变量说明 .....	(75)
5.5.1	集合类型定义 .....	(75)
5.5.2	集合变量说明 .....	(76)
5.5.3	集合的值 .....	(76)
5.5.4	集合类型常数定义 .....	(77)
5.6	集合表达式 .....	(77)
5.6.1	集合构造符 .....	(77)
5.6.2	集合运算符 .....	(78)
5.6.3	集合表达式 .....	(78)
5.7	集合的赋值与输出 .....	(79)
5.7.1	集合的赋值 .....	(79)
5.7.2	集合的输出 .....	(80)
5.8	小结与程序设计问题 .....	(81)
5.8.1	小结 .....	(81)
5.8.2	程序设计问题 .....	(81)
练习	.....	(84)
<b>第六章</b>	<b>文件 .....</b>	<b>(87)</b>
6.1	文件类型定义和变量说明 .....	(87)
6.1.1	文件类型定义 .....	(87)

6.1.2 文件变量说明 .....	(88)
6.2 文件标准过程和函数 .....	(89)
6.2.1 适用于所有文件的标准过程和函数 .....	(89)
6.2.2 类型文件的标准过程和函数 .....	(89)
6.2.3 无类型文件的标准过程和函数 .....	(89)
6.2.4 正文文件的标准过程和函数 .....	(90)
6.3 类型文件操作 .....	(90)
6.4 正文文件操作 .....	(94)
6.5 无类型文件操作 .....	(99)
6.6 I/O 检查与 I/O 重定义 .....	(100)
6.6.1 I/O 检查 .....	(100)
6.6.2 I/O 重定向 .....	(102)
6.7 小结与程序设计问题 .....	(103)
6.7.1 小结 .....	(103)
6.7.2 程序设计问题 .....	(104)
练习 .....	(106)
<b>第七章 指针</b> .....	(107)
7.1 动态数据结构 .....	(107)
7.2 指针类型定义和变量说明 .....	(107)
7.2.1 指针类型定义 .....	(107)
7.2.2 指针变量说明 .....	(108)
7.2.3 指针类型常数定义 .....	(109)
7.3 指针标准过程和函数 .....	(109)
7.4 指针变量的赋值操作 .....	(112)
7.5 线性链表结构 .....	(114)
7.5.1 线性链表概念 .....	(114)
7.5.2 线性链表的生成与遍历 .....	(115)
7.5.3 在已知线性链表中插入一个新元素 .....	(118)
7.5.4 在已知线性链表中删除一个表元素 .....	(119)
7.6 双向链环结构 .....	(120)
7.7 树结构 .....	(121)
7.8 小结与程序设计问题 .....	(125)
7.8.1 小结 .....	(125)
7.8.2 程序设计问题 .....	(125)
练习 .....	(127)
<b>第八章 过程和函数</b> .....	(129)
8.1 过程及其调用 .....	(129)
8.1.1 过程说明的形式 .....	(129)



8.1.2	过程语句	(131)
8.1.3	参数传递	(131)
8.1.4	全程变量和局部变量	(134)
8.2	函数及其调用	(135)
8.2.1	函数说明的形式	(135)
8.2.2	函数的调用	(137)
8.3	嵌套调用和向前引用	(139)
8.3.1	过程或函数的嵌套调用	(139)
8.3.2	过程或函数的向前引用	(141)
8.4	过程和函数的递归调用	(143)
8.5	无类型参数,过程类型与过程变量,以及过程类型参数	(145)
8.5.1	无类型参数	(145)
8.5.2	过程类型与过程变量	(146)
8.5.3	过程类型参数	(148)
8.6	小结与程序设计问题	(149)
8.6.1	小结	(149)
8.6.2	程序设计问题	(149)
	练习	(152)
<b>第九章</b>	<b>结构化程序设计思想</b>	(155)
9.1	什么是结构化程序设计	(155)
9.2	结构化程序设计示例	(156)
9.3	再谈程序设计风格	(164)
	练习	(165)
<b>第十章</b>	<b>通用程序设计示例</b>	(166)
10.1	一般数值计算	(166)
10.2	关于求素数	(172)
10.3	集合结构应用	(178)
10.4	分类算法概述	(181)
10.4.1	直接分类算法	(182)
10.4.2	改进的分类算法	(182)
10.4.3	分类算法程序示例	(183)
10.5	递归算法典型示例	(186)
10.5.1	梵塔游戏	(186)
10.5.2	探索策略和回溯算法	(189)
10.6	动态数据结构的应用举例	(193)
	练习	(200)
<b>第十一章</b>	<b>单元</b>	(202)

11.1	单元的基本结构	(202)
11.2	使用单元	(203)
11.3	标准单元及其功能	(207)
11.4	SYSTEM 单元	(208)
11.5	DOS 单元	(210)
11.6	CRT 单元	(217)
11.7	PRINTER 单元	(224)
11.8	TURBO3 单元	(225)
11.9	GRAPH3 单元	(226)
11.10	小结与程序设计问题	(227)
11.10.1	小结	(227)
11.10.2	程序设计问题	(227)
	练习	(233)
<b>第十二章 图形及其应用</b>		(235)
12.1	图形子程序分类和数据结构	(235)
12.1.1	图形子程序分类	(235)
12.1.2	图形子程序涉及的常数	(237)
12.1.3	图形子程序使用的数据类型	(240)
12.1.4	图形单元说明的变量	(241)
12.2	图形系统初始化	(241)
12.3	视口与坐标	(245)
12.4	使用图形功能	(249)
12.4.1	画点和读点	(249)
12.4.2	画直线和改变当前点 CP	(250)
12.4.3	颜色和调色板	(251)
12.4.4	线型、线宽和写模式	(255)
12.4.5	矩形、直方图和多边形	(257)
12.4.6	填充	(260)
12.4.7	圆、椭圆和圆弧	(263)
12.4.8	位映象	(267)
12.4.9	显示正文	(269)
12.5	动画设计原理	(273)
12.6	小结与程序设计问题	(275)
12.6.1	小结	(275)
12.6.2	程序设计问题	(276)
	练习	(278)
<b>第十三章 Turbo Pascal 高级特性</b>		(280)
13.1	绝对变量与无类型参数	(280)

13.1.1	绝对变量 .....	(280)
13.1.2	无类型参数 .....	(280)
13.1.3	应用绝对变量 .....	(281)
13.2	直接访问内存(Mem, MemW 和 MemL) .....	(281)
13.3	访问数据端口(Port 和 PortW) .....	(281)
13.4	获取命令行参数(ParamCount 和 ParamStr) .....	(282)
13.5	获取系统环境参数(GetEnv, EnvCount 和 EnvStr) .....	(283)
13.6	调用操作系统命令(Exec) .....	(284)
13.7	调用 BIOS 和 Dos 功能(Intr 和 MsDos) .....	(285)
13.8	中断子程序与驻留(Interrupt, Keep, GetIntVec 和 SetIntVec) .....	(286)
13.9	声音(Sound, NoSound 和 Delay) .....	(288)
13.10	小结与程序设计问题 .....	(291)
13.10.1	小结 .....	(291)
13.10.2	程序设计问题 .....	(291)
	练习 .....	(294)
<b>第十四章</b>	<b>与汇编语言的连接 .....</b>	<b>(296)</b>
14.1	四种连接方法 .....	(296)
14.2	调用约定 .....	(296)
14.2.1	参数传递方法 .....	(296)
14.2.2	函数结果返回方法 .....	(297)
14.2.3	远调用和近调用 .....	(297)
14.2.4	入口和出口代码 .....	(298)
14.2.5	寄存器保存的约定 .....	(298)
14.3	inline 语句 .....	(298)
14.4	inline 命令 .....	(299)
14.5	inline 汇编 .....	(300)
14.5.1	asm 语句 .....	(300)
14.5.2	汇编语句的语法 .....	(301)
14.5.3	标号 .....	(302)
14.5.4	指令前缀 .....	(302)
14.5.5	指令操作码 .....	(303)
14.5.6	操作数 .....	(304)
14.5.7	assembler 过程和函数 .....	(308)
14.6	外部过程 .....	(310)
14.7	小结 .....	(315)
	练习 .....	(315)
<b>第十五章</b>	<b>覆盖技术 .....</b>	<b>(316)</b>
15.1	覆盖 .....	(316)

15.2	覆盖管理	(316)
15.3	了解 OVERLAY 单元	(318)
15.4	编写覆盖程序	(321)
15.4.1	产生覆盖代码	(321)
15.4.2	远调用要求	(321)
15.4.3	初始化覆盖管理模块	(321)
15.4.4	覆盖单元的初始化部分	(323)
15.4.5	不能用作覆盖的单元	(324)
15.4.6	调试覆盖模块	(324)
15.4.7	覆盖模块的外部子程序	(325)
15.4.8	.EXE 文件中的覆盖	(325)
15.5	小结与程序设计问题	(326)
15.5.1	小结	(326)
15.5.2	程序设计问题	(326)
	练习	(327)
<b>第十六章</b>	<b>面向对象的程序设计</b>	<b>(328)</b>
16.1	对象	(328)
16.2	继承	(331)
16.3	封装	(332)
16.4	静态方法和虚方法	(332)
16.5	兼容性	(338)
16.6	动态分配	(338)
16.7	小结	(341)
	练习	(341)
<b>附录</b>		
附录一	Turbo Pascal 集成开发环境	(342)
附录二	编译指示一览表	(360)
附录三	Turbo Pascal 实用程序	(363)
附录四	Turbo Pascal 标准子程序库	(377)
附录五	编译与运行出错信息	(422)
附录六	一些流行的 Turbo Pascal 工具箱软件简介	(428)
附录七	ASCII 码表及键盘返回代码	(431)
<b>主要参考书目</b>		<b>(434)</b>

# 第一章 Turbo Pascal 初阶

## 1.1 关于 Turbo Pascal

Pascal 是一种计算机通用的高级程序设计语言。它由瑞士 Niklaus Wirth 教授于六十年代末设计并创立。取名 Pascal 是为了纪念十七世纪法国著名哲学家和数学家 Blaise Pascal(1623—1662)。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。这本书描述了标准 Pascal, 并提供一些用于说明 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的习惯, 并使他们从中领会和理解精细的程序设计技巧。尤其引人注目的是其中引入了结构化程序设计的思想。

在 Pascal 问世以来的二十余年间, 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。

1983 年 11 月, 美国 Borland 公司推出一种适用于微机的崭新 Pascal 编译系统, 称为 Turbo Pascal(1.0 版)。Turbo 与 Turbine 同义, 意即透平(机)或涡轮(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。1987 年, Borland 公司开始销售 Turbo Pascal 4.0, 并立即得到用户的广泛欢迎。Turbo Pascal 4.0 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 Pascal 系统的分钟级降到了秒级, 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、易学好用的集成开发环境(Integrated Development Environment, 简记 IDE), 它集编辑、编译、连接、运行、求助、查错于一身, 采用多窗口技术、多级菜单式驱动, 上下文敏感式求助, 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 4.0 在前几个版本的基础上又进一步扩充功能, 提供了更为丰富的内部预定义标准子程序(约 200 个), 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 引入了单元(UNIT)的概念, 每个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。这就给应用程序的开发带来方便, 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驱动器。

Borland 公司继 1987 年推出 Turbo Pascal 4.0 后, 又先后于 1988 年推出 Turbo Pascal-5.0, 1989 年推出 Turbo Pascal 5.5。两种新版本展现了若干新颖特色, 使它们在微机程序设计中成为更加卓

有成效的开发环境和工具。Turbo Pascal 5.0 和 5.5 主要新增了以下诸功能：①提供了集成调试器，用户可以在集成开发环境下用它调试程序。它可完成对程序的追踪、步进、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；②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；③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，使覆盖可以更容易和更快地执行；④支持 Lotus/Intel/MicroSoft 扩展内存规范 (Expanded Memory Specification, 简记 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；⑤提供了 8087 浮点仿真模式，即使未配置 8087 数字协处理器，亦可使用 IEEE 浮点数据类型，进行优化的浮点运算；⑥Turbo Pascal 5.5 支持面向对象的程序设计 (Object-Oriented Programming, 简记 OOP) 这一新颖的程序设计手段，跟上时代的潮流。

此外，自 Turbo Pascal 5.0 起，增加了一个标准单元 OVERLAY，用以实现强有力的覆盖管理功能；还提供了支持新版本各项功能的若干实用程序和单元；新增了一批 (约 30 个) 内部预定义标准子程序，使标准子程序的数目达约 230 个之多；对某些编译指示、编译与运行出错编码及信息作了增、删或修改。

从 Turbo Pascal 4.0 开始，Borland 公司就率先使用了集成开发环境 IDE，令用户感到耳目一新，使用起来非常便利。IDE 为用户提供了一个十分友好的界面。Turbo Pascal-5.0 和 5.5 在这方面又有了显著的提高，它反映在用户界面的变化上。主菜单增加了两个可选菜单项——Debug 和 Break/watch，并以 Watch 窗口取代了 4.0 版中的 Output 窗口。各级下拉菜单中亦相应增、改了若干选项，使功能更加丰富、完善。

1990 年，Borland 公司又进一步推出了最新版本 Turbo Pascal 6.0，该系统提供了许多新的特性，主要有：①新一代的 IDE 用户界面，使能支持诸如鼠标器操作，多层覆盖窗口，多文件编辑，扩展的联机求助系统，条件断点等等功能；②Turbo Vision，它是个面向对象的应用框架和库，用它可高效地开发实用的、具有连贯一致用户界面的应用程序；③允许将汇编语言直接写入 Turbo Pascal 源程序中；④其编译器能使用扩充内存以建造大型应用程序。

我们相信，Turbo Pascal 系列软件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，正在发挥其越来越大的作用。它们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 系统之一。

本书内容适用于 Turbo Pascal 5.5，部分内容引用了 Turbo Pascal 6.0 新增特性。为简化计，以下将 Turbo Pascal 作为 Turbo Pascal 5.5 的代名词。对于其它版本，则注明版本号以示区别，如 Turbo Pascal 4.0, 5.0 或 6.0 等。

## 1.2 Turbo Pascal 系统安装

Turbo Pascal 系统软件存储在 4 张容量各为 360K 字节的  $5\frac{1}{4}$  英寸双面双密度软盘上。它包括了运行集成开发环境和命令行编译版本所需要的一切文件和程序。为了便于将 Turbo Pascal 系统软件安装到 IBM PC 系列微机或其兼容机上，在该系统软件中装有一个专门的安装文件，其文件名为 INSTALL.EXE。INSTALL.EXE 将引导你逐步完成整个安装过程，你只要仔细阅读显示屏上对每一步骤所给出的指示信息并遵循执行之就行了。

运行 INSTALL.EXE 文件可实现三个功能：一是可将 Turbo Pascal 系统软件安装到硬盘上，此时，硬盘必须留有足够存储空间；二是可将 Turbo Pascal 系统软件安装到软盘上，此时，必须先准备

好 4 张容量各为 360K 字节的空软盘；三是可将硬盘上的 Turbo Pascal-4.0 或 5.0 版升级成 5.5 版，此时，需使用 INSTALL 中的 Upgrade(升级)选项。

现在假定有一台带有硬盘的 IBM PC 机。我们希望将 Turbo Pascal 系统软件安装到硬盘的当前目录上。可按如下步骤执行：①把 Turbo Pascal 系统软件中标记为 Installation Disk 的软盘插入 A 驱动器中；②键入 A: \，这里 \ 表示按回车键；③键入 INSTALL \，并遵循显示屏上给出的指示信息作出响应，直至完成整个安装过程为止。这台 IBM PC 机上就配有 Turbo Pascal 系统软件了。

Turbo Pascal 6.0 系统软件存储在 3 张容量各为 1.2 兆字节的  $5\frac{1}{4}$  英寸高密软盘上。其安装方法与 Turbo Pascal 5.5 相似。

### 1.3 使用集成开发环境

本节描述 Turbo Pascal 用户界面并简单介绍如何使用 IDE，关于 IDE 的详情可参见附录一“Turbo Pascal 集成开发环境”。那里给出的是最新的 Turbo Pascal 6.0 集成开发环境。

Turbo Pascal 之所以深受广大用户欢迎，其主要原因之一就是它率先提出 IDE 概念并付诸应用，经过不断改进，为用户提供了一个十分友好的界面。

为启动 Turbo Pascal，在 Dos 操作系统状态提示符下键入 TURBO \，Turbo Pascal 就进入了 IDE，显示屏第一屏包括主菜单屏幕和产品版本信息，可按任一健使版本信息消失。此时，显示屏呈现如图 1-1 所示的主菜单屏幕：

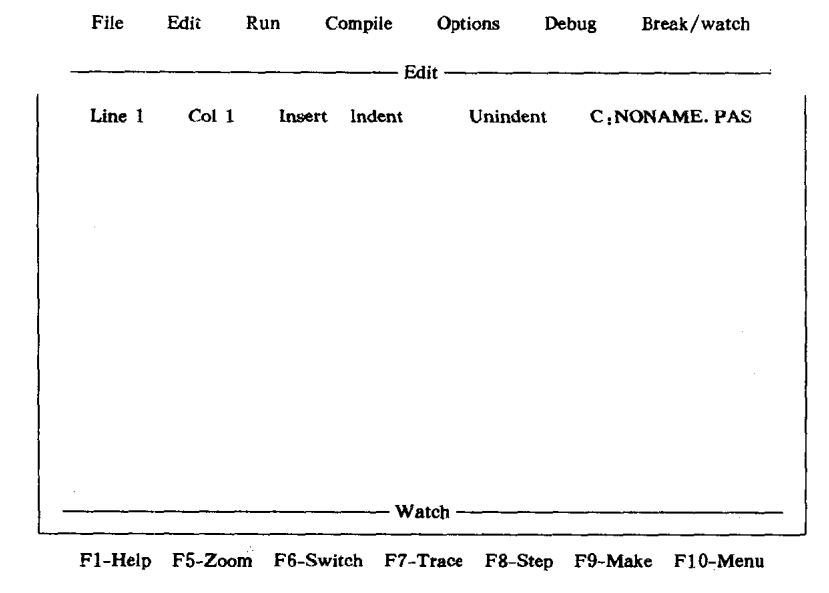


图 1-1 Turbo Pascal 主菜单屏幕

由图 1-1 可见，IDE 呈现的主菜单屏幕把显示屏分成四个部分：主菜单，Edit 窗口，Watch 窗口，以及底部状态行。

显示屏顶端是主菜单，它包含 7 个可选菜单项：File，Edit，Run，Compile，Options，Debug 以及 Break/watch。除了 Edit 选项外，其余 6 个均含有其相应的下拉式可选菜单项，而某些下拉式菜单项本身又含有再下一级的可选菜单项，从而构成了 IDE 独特的多级菜单结构形式。

主菜单中 7 个可选菜单项的功能简述如下：

File	处理与文件有关的操作,如文件的装入、选择、建立、存储、写至新文件等,还有目录列表、修改,切换至 Dos 操作系统状态,以及从 Turbo Pascal 出口退出 IDE。
Edit	进入 Edit 窗口,并激活 Turbo Pascal 编辑器。此时可编辑源文件。
Run	在 IDE 下用集成调试器运行或调试一个程序。
Compile	对当前源文件进行编译,提供对重编译的选择,设置目标代码的去向,发现运行错误,选定主文件,以及获得当前源文件的有关信息等。
Options	确定在 IDE 下工作的配置。这些配置将对编译器选择项,单元文件、目标文件和包含文件的目录,以及程序的运行参数等产生影响。
Debug	可检查和修改变量、表达式,显示当前调用堆栈,在源程序中快速定位过程和函数,打开或关闭集成调试程序,管理显示屏的切换等。它还提供了允许在 IDE 外使用 Borland 公司的独立调试器 Turbo Debugger 调试程序的状态开关。
Break/watch	允许添加、删去或编辑位于 Watch 窗口中的变量,以及在源程序中设置、清除、移动或观察断点。

在主菜单行下面是 Edit 窗口,即编辑窗口。程序源代码在 Edit 窗口中进行输入及编辑加工,它使用了 Turbo Pascal 的编辑器中各项功能。编辑器类似于一个简单的字处理器,可用它来方便地编写和修改程序。

Edit 窗口顶端有一个报告有关编辑器信息的状态行。状态行各项元素的含义如下:

Line n	显示当前源文件中光标所在行的行号。
Col n	显示当前源文件中光标所在列的列号。
Insert	当出现该指示符,表示为插入方式,此时所键入的任何字符均被插入当前源文件中光标所在位置。若指示符消失,则表示为覆盖方式,此时所键入的任何字符均替代了当前源文件中光标所在位置的内容。
Indent	当出现该指示符,表示为自动缩进,此时按 \ 键可使光标返至下一行中与上行开始处对齐的位置。若指示符消失,表示取消自动缩进,此时按 \ 键光标返至下一行第一列处。
Tab	表明已激活了 Tab 方式,即制表键(Tab 键)有效。在 Tab 方式下,按 Tab 键产生一个制表符(^ I)并将光标右移一定数目的空格至某位置。图 1-1 所示,表示未激活 Tab 方式。
Fill	仅当 Tab 方式下工作。Fill 特性使得 Turbo Pascal 每行均以制表符与空格的组合开始,从而减少了所使用的总字符数,优化了编码。图 1-1 所示,表示未启动 Fill 特性。
Unindent	当出现该指示符,表示为自动退格,此时按回退键可使光标回退至与当前行之上的第一个非缩进行对齐的位置。若指示符消失,表示取消自动退格,此时按回退键光标返至上一字符位置。

最后一项元素 C:NONAME.PAS 表示驱动器 C 中正在编辑的文件名 NONAME 和扩展名 PAS。

在 Edit 窗口之下有 Watch 窗口,即监视窗口。一旦将源程序输入 Edit 窗口,就可以在程序执行时使用 Watch 窗口来检验程序。在 Watch 窗口中可以显示、跟踪或改变程序中变量或表达式的值。



Turbo Pascal 主菜单屏幕中的最底下部分是底部状态行。它显示了 7 个功能键(称作热键)及相应功能,即:

- |           |   |
|-----------|---|
| F1-Help   | F1 键调用求助功能,它打开 Help 窗口,并显示有关求助信息。                             |
| F5-Zoom   | F5 键可将当前活动窗口扩大至整屏,或者将整屏缩小至原先大小。这是个跟斗式状态开关。                    |
| F6-Switch | F6 键可切换当前活动窗口,即可以从 Edit 窗口和 Watch 窗口之间进行跟斗式切换。因此,这也是个跟斗式状态开关。 |
| F7-Trace  | F7 键可用来单步跟踪所执行的程序或子程序。每按一次 F7 键,执行程序或子程序中的一步。                 |
| F8-Step   | F8 键的功能与 F7 键相似,但它对子程序的调用视作一步执行。                              |
| F9-Make   | F9 键用于重新编译当前源文件,但它仅编译为更新整个源文件所必须的最少数量模块。                      |
| F10-Menu  | F10 键用于激活主菜单。   |

由于 IDE 中有许多频繁使用的菜单命令,IDE 为用户提供了 30 多个热键,这些热键用于直接执行某种功能,相当大一部分热键还与某些菜单命令等价。上述主菜单屏幕的底部状态行就显示了其中 7 个热键及相应功能。但底部状态行的内容并非固定不变。比如,按 ALT 键并保持数秒,则底部状态行显示由 ALT 键与其它几个功能键组合而成的一些热键及相应功能:

Alt; F1-Last help F3-Pick F6-Swap F9-Compile X-Exit

下面以回答问题的形式介绍 IDE 的一些基本操作。要真正熟悉 IDE 并能灵活使用,尚需大量上机实践及悉心体验。

①如何选择主菜单中某一选项?——若在如图 1-1 所示的主屏幕状态,则可键入所需选项中高亮度显示的首字母(大、小写均可),如 F 表示选 File 选项,E 表示选 Edit 选项等。亦可使用光键←或→在各选项间移动光条,当移至所需选项时,按↵键即可。另外,不论当前活动窗口在何处,均可使用同时键入 Alt 键与某选项首字母,达到上述同一目的。例如,用 Alt-R 组合键(同时按 Alt 键与 R 字母键)即可选择主菜单中 Run 选项。

②如何装入、保存 Turbo Pascal 源文件?——简单的办法是按 F3 键装入源文件,此时应回答 IDE 询问的文件名;按 F2 键保存当前在 Edit 窗口的源文件。也可使用相应的菜单命令,如组合键 Alt-F-L 与 F3 键等价,Alt-F-S 与 F2 键等价。

③怎样获得联机求助信息?——IDE 提供了上下文敏感式联机求助功能,并包含大量有关信息。可在除了运行程序以外的任何时刻按 F1 键,以得到光标当前位置条目的求助信息,它显示在一个窗口中,本身还包含一个或多个可以得到更详细信息的关键字,它们以高亮度显示,可移动箭头键至某一关键字并按↵键。在求助系统中再按 F1 键,可得到求助索引。顺着求助索引进行选择,亦可获得各方面的信息。此外,当使用编辑器时还可以按组合键 Ctrl-F1,以得到光标所在位置的标准单元、子程序、变量、常量、类型等有关信息。

④怎样扩大或缩小当前活动窗口?——可以使用 F5 键,它是个跟斗式状态开关。

⑤怎样切换当前活动窗口?——可以使用 F6 键,它亦是个跟斗式状态开关,使当前活动窗口为 Edit 窗口或者 Watch 窗口。

⑥如何返至主菜单?——不管在什么窗口,按 F10 键即可返回主菜单。但 F10 键亦是个跟斗式状态开关,它在主菜单与当前活动窗口之间切换。

⑦怎样离开某一菜单?——IDE 采用多窗口技术、多级菜单式驱动。为离开某一菜单,可按 Esc