

跟我学用高级语言编程系列丛书

Turbo Pascal

冯博琴 黎 剑 魏恒义

电子工业出版社

- 95 个精心设计的范例
- 从入门到精通
- 具有配套的软盘
- 提供良好的在线学习环境

TP312

跟我学用高级语言编程系列丛书

Turbo Pascal

冯博琴 黎 剑 魏恒义 改编

电子工业出版社

(京)新登字 055 号

图书在版编目(CIP)数据

TURBO PASCAL/冯博琴等改编,一北京:电子工业出版社,1994.5

(跟我学用高级语言编程系列丛书)

ISBN 7-5053-2493-3

I. T... II. 冯... III. PASCAL 语言-程序设计 IV. TP312PA

中国版本图书馆 CIP 数据核字(94)第 03773 号

本书英文版由美国 Coronado Enterprises 公司所有。

本书中文版由美国 Coronado Enterprises 公司授权电子工业出版社独家改编、出版发行,未经出版者书面许可,不得以任何形式复制或抄袭本书内容。

跟我学用高级语言编程系列丛书

Turbo Pascal

冯博琴 黎剑 魏恒义 改编

特邀编辑 胡小军

责任编辑 王昌铭

*

电子工业出版社出版(北京万寿路)

电子工业出版社发行 各地新华书店经销

河北省望都县印刷厂印刷

*

开本:787×1092 毫米 1/16 印张:15 字数:384 千字

1994 年 5 月第 1 版 1994 年 5 月第 1 次印刷

印数:3000 定价:14.00 元

ISBN 7-5053-2493-4/TP·734

前 言

N. Wirth 在 1971 年正式发表了 Pascal 程序设计语言,它以崭新的设计思想在程序设计界开了一代新风。此后,“结构化程序设计原则”变成现实,程序设计风格出现了巨大变化,因而引起了各国计算机学者的普遍重视。在数以百计的语言中,它是最能体现结构化程序设计原则、且获得广泛应用的程序语言。它定义严谨,数据类型完备,语句灵活,结构清晰,运行效率高。因此在许多领域中得到广泛应用,同时也使 Pascal 向更完善、更方便的方向发展,目前广为流行的 TURBO Pascal 6.0 和面向对象的程序设计方法将使 Pascal“雅俗共赏”的特性得到更加充分的体现。

Pascal 在廿余年的发展中,在许多领域显示了它无可争辩的优势,几乎所有的计算机系都把它列为必修课,它是计算机语言教学和数据结构、算法描述的最理想工具,是指导读者进入计算机软件殿堂的带路人,它与 C 一起成为目前最流行的系统软件编写工具,因此从事计算机应用的科技工作者都把它作为必需熟悉的对象。当然,Pascal 在应用软件开发中也是十分常用的,它可以相当方便地完成科学计算、控制和管理等方面的工作。

学习计算机语言,精读一定数量的具有良好程序设计风格的程序,而且尽早坐到计算机前,边学边练是一种事半功倍的学习方法,这样的学习,使读者基本概念学得活,进入编程状态早,程序基本功扎实,能更快地编出更好的软件来。我们向读者推荐的《TURBO Pascal 指南》为实践这种方法提供了一份很好的材料。

《TURBO Pascal 指南》的体系独具一格,作者精心设计了 95 个程序例子,它们覆盖了 TURBO Pascal 6.0 的最主要内容,全书用这些例子把 Pascal 的基本概念、设计技巧深入浅出地进行了系统的介绍。该书从组织上不拘泥于一般语言书的常规章节,文字通俗,所有内容都是以例子为背景,所以学起来毫无枯燥之感,通过这 95 个例子的诱导,最后你便逐渐掌握了 Pascal 的程序设计。该书另一特色是提供了一套(三张)软盘,其中有书中 95 个例子的全部源程序及 TURBO Pascal 编译系统,读者如能边学边在机上练一遍,效果绝佳。

本书是由 CORONADO ENTERPRISES 的《TURBO Pascal TUTOR——Version 2.60》编译而成,为适应国内读者的习惯,我们对原文作了部分更动,增加了附录,对示例程序软盘建立了一个运行环境。

全书共分 16 章,第一章介绍了 Pascal 语言的概况及 TURBO Pascal 集成环境的使用。第二章至第四章介绍了 Pascal 程序的概貌以及标准数据类型和控制结构。第五章到第九章是关于 Pascal 过程和结构类型的定义及使用。第十、十一章介绍标准 I/O、正文文件及二进制文件的概念及应用。有关指针和动态存储分配及 TURBO Pascal 单元的定义及示范在第十二、十三章给出。第十四、十五章介绍目前新出现的面向对象的程序设计方法,并有详细的应用范例。最后一章采用自顶向下的程序设计方法开发一个程序。

本书由冯博琴、黎剑和魏恒义编译。本书的出版得到了电子工业出版社大力支持,在此表示衷心感谢。

由于编译者学识浅薄、时间仓促,不妥之处恳望指正。

冯博琴
1993.6

如何使用本指南提供的软盘

与本指南配合使用的有三张软磁盘,其中两张是系统盘,另一张程序盘中包含了书中所有章节使用的示例程序和练习答案及其它辅助文件。在 TURBO Pascal V6.0 环境下,这些示例程序均能正确的运行。请在您的机器上安装这三张盘,建立您的 TURBO Pascal 工作环境。

如果您是一个初学者,不了解这些例子是说明哪个方面的问题,请按照下面的步骤使用本书提供的选择指导程序。这将引导您找到有关的示例程序并将找到的程序装入 TURBO Pascal 编辑窗口供您阅读、编译、运行。

1. 需要的硬件环境及系统软件

使用指导程序需在有硬磁盘、1.2MB 5 inch 软驱、640KB 内存及 VGA 显示器的硬件环境上使用。要求有 DOS 操作系统。TURBO Pascal 6.0 可用提供的软盘系统装入,装入的方法是:

①将 TURBO Pascal 6.0 1st 盘插入 A 驱动器

②打入 A: >INSTALL

③按照屏幕提示将 TURBO Pascal 装入 C:\TP 目录

2. 装入程序盘(Program)并设置环境

①将 Program 盘插入 A 驱动器中

②打入命令 A: >Loadp

命令执行完后,要使用的程序即装入到 C:\PTUTOR 目录。若 TURBO Pascal 安装在 C:\TP 目录,请将下面几条命令加入到根目录下的 AUTOEXEC.BAT 中。

```
PATH= C:\TP;C:\PTUTOR
```

```
C:
```

```
CD\PTUTOR
```

```
FILE0A
```

```
SHOWHZ
```

```
CD\
```

在 CONFIG.SYS 文件中加入命令:

```
FILES= 20
```

然后重新启动系统就可按下面的方法使用该程序。

3. 运行指导程序

请按下面的命令进入 C:\PTUTOR 目录并启动程序。

```
C>CD C:\PTUTOR
```

 进入您的工作目录

```
C>Pascal
```

 运行选择指导程序,屏幕显示如图 0-1。

Pascal 指南

示例程序

电子工业出版社

图 0-1 系统首屏幕示意图

按任意键后,屏幕将提示本书各章所讲的内容,屏幕示意如图 0-2 所示。这时屏幕显示出本书各章的标题(屏幕上的第 i 项表示书中的第 $i+1$ 章,以此类推),光标移动到哪章,那章的内容提要将显示到屏幕的右上角,图 0-2 表示选择了第二章 Pascal 程序初步的情况。



图 0-2 选择第二章的屏幕示意图

菜单中每一项选择代表某一方面的内容,也是书中的一章。光标移动到第一项(书中第二章)时,若想看本章的程序示范,请用回车确认。这时书中第二章使用的全部程序名出现在二级菜单中(如图 0-3 所示),右上角显示了程序 trivial.pas 的功能摘要。

留心各程序文件名的拼写及屏幕中的汉字提示,就不难确定每个程序的大致内容。当用回车选定一程序后,系统将自动进入 TURBO Pascal 集成环境,并将选中的程序装入编辑窗口供您操作。有关 TURBO Pascal 集成环境下的操作,请参阅本书第一章。

若已知道要操作的 Pascal 程序文件名,可在 DOS 提示符 C:> 下打入 TURBO,直接进入

TURBO Pascal 环境,利用集成环境中的 File 子菜单装入已知的示例程序或输入您自编的新程序。



图 0-3 选中第二章中的第一个示例程序示意图

目 录

如何使用本指南提供的软盘	I ~ III
第一章 Pascal 及 TURBO Pascal	(1)
什么是计算机程序	(1)
什么是 Pascal 语言	(1)
Pascal 语言的词法单位	(1)
Pascal 程序的结构形式	(2)
Pascal 的数据类型	(2)
Pascal 的控制结构	(2)
Pascal 单元	(2)
Pascal 编译程序	(3)
本指南对 Pascal 系统的要求	(3)
TURBO Pascal 入门	(3)
如何进入 TURBO Pascal 集成环境	(3)
使用 TURBO 编辑器编辑源程序文件	(5)
使用 TURBO Pascal 的编译程序	(6)
如何使用集成环境下的运行程序	(6)
TURBO Pascal 中的出错指示	(6)
如何设置集成环境工作及控制参数	(7)
TURBO Pascal 中的动态调试	(8)
程序的装入、编译、运行及调试举例	(9)
第二章 Pascal 程序初步	(11)
第一个 Pascal 程序	(11)
什么是标识符	(11)
Pascal 程序	(12)
一个做某事的程序	(12)
有更多输出的另一个程序	(13)
程序中增加注释	(14)
执行部分的结果	(15)
好的程序格式	(15)
差的程序格式	(16)
程序设计练习	(17)
第三章 简单的数据类型	(18)
什么是数据类型	(18)
整型变量	(18)
算术表达式	(19)
多种类型的变量	(21)
布尔变量	(24)
何处使用布尔变量	(26)
布尔表达式的求值	(26)
字符型变量	(26)
扩展的整型	(28)
扩展的实型	(30)

程序设计练习	(31)
第四章 循环及控制结构	(32)
循环	(32)
Pascal 复合语句	(34)
IF 语句	(34)
IF—THEN—ELSE 块	(36)
循环和 IF 语句的结合	(36)
一个有意义的程序	(37)
REPEAT—UNTIL 循环	(40)
WHILE 循环	(41)
CASE 语句	(43)
程序设计练习	(45)
第五章 过程和函数	(46)
Pascal 程序的轮廓(outline)	(46)
过程	(46)
过程的定义	(47)
怎样定义和调用过程	(48)
Pascal 一直沿用的规则	(48)
更多的过程调用	(48)
形参和实参	(50)
值调用	(50)
引用调用	(50)
一些新的术语	(51)
“引用调用”还是“值调用”	(52)
多次定义的变量	(52)
过程调用过程	(54)
函数	(55)
递归(recursion)的奥秘	(57)
递归过程	(58)
提前调用	(58)
过程类型	(59)
程序设计练习	(63)
第六章 数组、类型、常数和标号	(64)
数组	(64)
使用数组	(65)
双重下标数组	(65)
数组的灵活性	(68)
类型定义	(68)
Pascal 数据类型的一致性	(69)
“类型”是个重要的概念	(69)
常量说明	(69)
TURBO Pascal 类型常量	(71)
标号说明	(71)
压缩数组	(73)
更多的 TURBO Pascal 扩展	(73)
程序设计练习	(73)
第七章 字符串和字符串过程	(74)

Pascal 字符串	(74)
一个字符串是一个字符型数组	(75)
TURBO Pascal 的字符串类型	(75)
字符串变量的长度	(76)
在字符串类型变量中包含什么	(76)
程序设计练习	(78)
第八章 枚举类型、子界类型和集合类型	(79)
Pascal 枚举类型	(79)
一个大的子界变量循环	(81)
子界类型	(81)
调试程序的一个技巧	(83)
三个很有用的函数	(83)
集合类型	(83)
集合的查找	(85)
程序设计练习	(87)
第九章 记 录	(88)
一个很简单的记录	(88)
一个大的记录	(90)
怎样处理记录中的数据	(92)
什么是 WITH 语句	(92)
WITH 语句能嵌套多深	(92)
高级赋值语句	(92)
程序大小的限制	(93)
变体记录	(93)
什么是特征字段	(95)
使用变体记录	(95)
包括特征字段的记录中都包含什么	(96)
程序设计练习	(96)
第十章 标准输入/输出	(97)
我们已使用过的输入/输出	(97)
多输出语句	(99)
从键盘输入	(99)
当输入数据类型不一致时	(101)
读实型数据	(101)
读字符型数据	(102)
“容错”程序设计	(103)
怎样在打印机上输出结果	(103)
程序设计练习	(105)
第十一章 文件的输入/输出	(106)
文件处理串行的数据	(106)
历史回顾	(106)
现在的状况	(106)
读和显示文件	(106)
什么是 EOF 和 EOLN 函数	(107)
可读任何文件的程序	(108)
如何复制一个文件	(109)
编译命令的使用	(111)

文件自检查	(111)
怎样从文件中读整型数据	(112)
READ 和 READLN 之间的微小差别	(114)
怎样从文件中读实型数据	(115)
二进制输入和输出	(117)
为何使用二进制文件	(118)
读一个二进制文件	(119)
文件指针,GET,PUT 语句.....	(120)
程序设计练习	(120)
第十二章 指针和动态分配	(121)
关于指针和动态分配	(121)
什么是指针,它的优点是什么	(121)
如何使用指针	(122)
指针类型	(123)
TURBO Pascal 的指针	(124)
观察动态分配	(125)
什么是堆	(126)
什么是动态分配	(126)
释放动态分配的空间	(127)
动态存储的记录	(128)
Pascal 规则的例外情况	(130)
如何给指针变量赋值	(130)
动态分配的好处	(131)
什么是链接表	(131)
程序中未定义任何变量	(133)
什么是“nil”,其用途何在	(134)
定义第二个记录	(134)
增加 10 个记录	(134)
一个完整的链接表	(135)
如何使用链接表中的数据	(135)
程序设计练习	(135)
第十三章 TURBO Pascal 的单元	(136)
单元描述	(136)
接口(interface)部分	(138)
实现(implementation)部分.....	(138)
局部的过程	(138)
单元体的用途是什么	(138)
可选择的函数和过程	(139)
另一个单元	(139)
如何使用单元	(140)
又一个使用单元的例子	(142)
同一个标识符的多次使用	(143)
为什么使用单元	(144)
信息隐蔽	(144)
程序设计练习	(144)
第十四章 封装和继承	(145)
封装	(145)

什么是方法(method)	(146)
方法(method)的实现	(147)
对象类型的实例(instance)	(147)
新的术语	(147)
我们完成了些什么	(148)
数据和代码的保护	(148)
更多的封装	(148)
专用类型	(151)
必要的几个规则	(152)
什么是构造过程(constructor)	(152)
什么是释构过程(destructor)	(153)
继承的例子	(153)
如何使用对象	(157)
为什么使用继承	(157)
单元中的对象	(157)
单元中的另一个对象	(159)
使用单元中定义的对象	(161)
数组和指针	(162)
什么是多重继承	(164)
现在该做什么	(164)
程序设计练习	(165)
第十五章 虚拟方法(Virtual Methods)	(166)
不含虚拟方法	(166)
制造一个虚拟方法	(169)
将子对象赋给父对象变量	(173)
为什么使用一个构造过程(constructor)	(173)
虚拟和指针	(173)
一个父对象	(177)
一些子对象	(178)
一个完整的雇员程序	(180)
程序设计练习	(183)
第十六章 完整的样本程序	(184)
偿还表生成程序	(184)
自顶向下的程序设计	(194)
LIST. PAS, 列出您的 Pascal 程序	(194)
TIMEDATE. PAS, 得到当天的时间和日期	(197)
SETTIME. PAS, 一个有用的应用程序	(198)
OT. PAS, 以树状列目录的程序	(199)
最重要的——您自己的程序	(212)
附录 1 示例程序索引	(213)
附录 2 TURBO Pascal 保留字	(215)
附录 3 Pascal 语言的标准函数	(216)
附录 4 TURBO Pascal 语法图	(217)

第一章 Pascal 及 TRUBO Pascal

什么是计算机程序

计算机是一个只能进行快速而精确运算的笨机器。如果没有人为其编写程序,它将不能做任何事情。如果人们编写一段将有用的数据转换成无用信息的程序,那么计算机将很听话,迅速将有用的数据转换成无用的信息。虽然在程序的编写过程中会出现错误,计算机将严格执行您认为正确的程序,得出该程序应得到的结果。程序设计的目的是为了达到期望的结果,这种结果是计算机严格按程序的安排执行得出。

计算机程序简单地象一种烹饪方法,计算机用它进行数据的输入/输出。假如这种烹饪方法是烤面包,那么,输入数据相当于配料,当然包括电炉提供的热量;程序相当于混合、搅拌、等待、加热、冷却和其它可能的烹饪操作,程序输出的结果可比喻成在柜台前出售的面包。计算机程序是由需要加工的数据和加工这些数据的数据两部分组成,二者不可分割。

什么是 Pascal 语言

Pascal 程序设计语言是瑞士苏黎世工学院教授 N. Wirth 1968 年提出来的。它以其完备的数据类型、简明灵活的通用语句和清晰明了的结构成为世界上最广泛使用的程序设计语言之一。

Pascal 语言虽比 FORTRAN, BASIC 等出现的晚,但有其独特的优点,特别是在结构化程序设计日益成熟的今天,Pascal 被更为广泛的学习及使用。Pascal 书写格式自由,运行效率高,查错能力强,它不仅适用于开发各种系统软件和应用软件,还适应于程序设计的教学,各种算法的描述,培养学生的结构化程序设计方法。在最新版本的 TURBO Pascal 语言中,还提供了面向对象的程序设计技术。

Pascal 语言的词法单位

保留字是 Pascal 中有固定意义的单独符号,如 if, then 等。TURBO Pascal 6.0 中的保留字见附录 2。

定界符是用来确定语法单位的符号,共有 21 个。如运算符 +、-、*、/, 比较运算符 =、<、> 及赋值号:= 等。

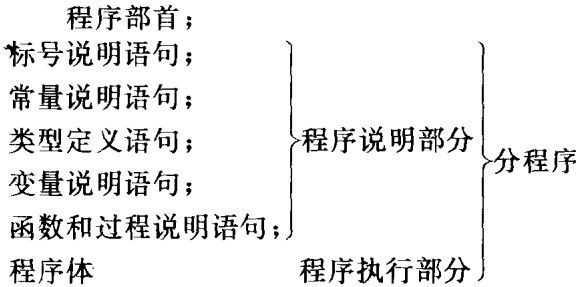
标识符是一个很重要的词法单位,用来标识 Pascal 中的各种对象的名称。可标识变量、常量及文件、过程等。标识符是由字母开头的字母数字串组成。TURBO Pascal 中有效长度为 63 个字符。有一部分系统定义的标识符,如类型定义符 integer,标准过程 Write,标准函数 sin,常量 FALSE,标准文件 input,output 等。凡系统未定义的对象均由程序设计者自定义,基本的原则是:不用保留字作标识符,不提倡用系统定义的标准标识符作为其它意义的标识符。

常数和用 { } 或 /* */ 或 (* *) 配对括起来的注释是您易于理解的词法单位。

除上述词法单位外,还有唯一的指示字 forward 和控制字符以及插入程序中的编译控制开关参数。

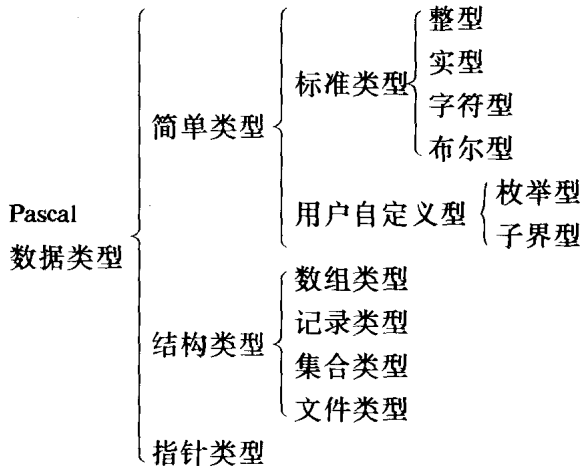
Pascal 程序的结构形式

无论什么版本的 Pascal 语言,无论程序是多么复杂,Pascal 程序的结构形式是固定的,即由程序首部和分程序组成。



Pascal 的数据类型

在 Pascal 中无论常量还是变量都必须要有一个确定的数据类型。从最初的整型、实型、字符型、布尔型等标准类型开始,程序设计者可自定义枚举及子界类型,并可用这些简单类型构造复杂的结构类型以满足解决实际问题的需要。Pascal 中的类型分类如下:



Pascal 的控制结构

Pascal 中有 for 循环结构、repeat 重复结构及 while 结构,可根据各种实际要求构造循环。分支结构和分情形结构能进行多路分支判断。过程和函数可递归调用,过程调用时参数传递有值调用和引用调用两种。可调用标准函数、自定义过程、函数及有关的 Pascal 单元。动态存储分配可充分利用机器内存,提高程序执行速度。

Pascal 单元

使用单元,可使 Pascal 系统的功能大为增强。单元是程序的子结构,作为一个单独的文件,扩展名为 TPU。单元可独立编译供其它结构调用。单元由接口部分、实现部分、初始化部分组成。程序设计者可自己设计单元,也可调用 Pascal 系统提供的标准单元。

TURBO Pascal 系统默认使用了 System 系统单元,该单元中实现了所有内部特性的低层功能,包括标准函数及标准过程等。其它的 Dos, Crt, Graph 等单元均有某一方面专用的目的,比如调用 DOS 操作系统的功能,各种基本的绘图功能等。若要使用这些单元中的程序库,须用

uses 语句在程序开头说明。具体单元中包含的程序功能请参阅相关的使用手册。

Pascal 编译程序

Pascal 使用编译程序处理 Pascal 源程序,除将源程序转变成计算机能理解的代码外,还处理各种函数调用,产生符号表,设置检查格式等。有的 Pascal 系统将编译分成若干趟来做,其最终都是形成可直接交给机器运行的代码程序。程序设计者应注意的是,要注意各种 Pascal 编译系统对标准 Pascal 的扩展,您若使用了扩展功能,请一定要在相应的编译系统下处理,否则将难以达到预期目的。

本指南对 Pascal 系统的要求

本指南是针对 Borland TURBO Pascal 而编写的。这是一高质量的编译程序。书中的大部分程序能在 TURBO Pascal v2.0 到 v4.0 的版本上编译。第 14 和 15 章是专门针对 TURBO Pascal v5.5 和 v6.0 版本来讨论面向对象程序设计和如何使用其扩展。面向对象的程序设计是一个很大的研究领域,这两章只给您一些参考及如何用它。几个完整的例子将使您学到入门的诀窍,这种新的程序设计方法有可能极大改进您的程序质量,减少系统调试时间。

如果您使用的不是 TURBO Pascal,您仍可使用大部分的程序,因为大多数例子使用标准的 Pascal 语法。请注意调整程序中有关输入输出部分,以便这些例子能在您使用的 Pascal 系统下运行。

TURBO Pascal 入门

TURBO Pascal 是美国 Borland 公司的 TURBO 系列软件之一,自 1985 年 3.0 版本问世以来,很快风靡全球。继 TURBO Pascal 5.5 之后,Borland 公司于 1990 年底又推出了最新版本 TURBO Pascal 6.0。

TURBO 系列软件最大的特点是集编辑、编译、运行及调试于一体。我们称为集成开发环境 (IDE),而且这种环境随版本的更新,对程序的调试纠错功能愈强。TURBO Pascal 经历了 v2.0, v3.0, v4.0, v5.0, v5.5 及 v6.0 版本,每一种新版本的诞生,总伴随着对原版本功能和性能的较大改进。3.0 版本基本具备了编译运行速度快,功能多,结构精巧及集编辑、编译为一体的集成环境。4.0 版本引入单元概念和智能化单元管理手段,突破了 64k 限制,使 TURBO Pascal 能编写大型程序。同时,引入下拉式菜单集成环境。5.0 版本增加调试功能,提供智能化复盖管理程序,进行 8087 协处理器仿真,速度比 3.0 提高 2 到 3 倍。5.5 版本增加面向对象的程序设计方法。6.0 版除了与早期版本兼容外,还提供了一种全新的集成开发环境 (IDE),它支持鼠标,多文件编译和多重叠窗口,并配备了增强型调试工具和功能完备的嵌入式汇编器。TURBO Pascal 6.0 还支持 286 代码生成、远程调用,并等同对待过程与函数。在面向对象的程序设计方面,增加私有域和私有方法。用户还可用 Turbo Vison 构造窗口程序,实现用户自己的 IDE。

如何进入 TURBO Pascal 集成环境

为了使您能方便的使用 TURBO Pascal 测试本指南中给出的实例,您应该了解下面几个问题。本指南关于如何使用 TURBO Pascal 不是最详细的使用手册,但可使您顺利测试书中的例子和做有关的练习。此处只介绍进入 TURBO 环境后如何工作,有关机器环境的配置及软件系

统之间的配合请参阅本指南中“如何使用本指南提供的软件”一节及有关的使用手册。

在装有 TURBO Pascal 系统软件的目录下,键入 TURBO 后就可进入如下图 1-1 所示的 TURBO Pascal 集成环境。

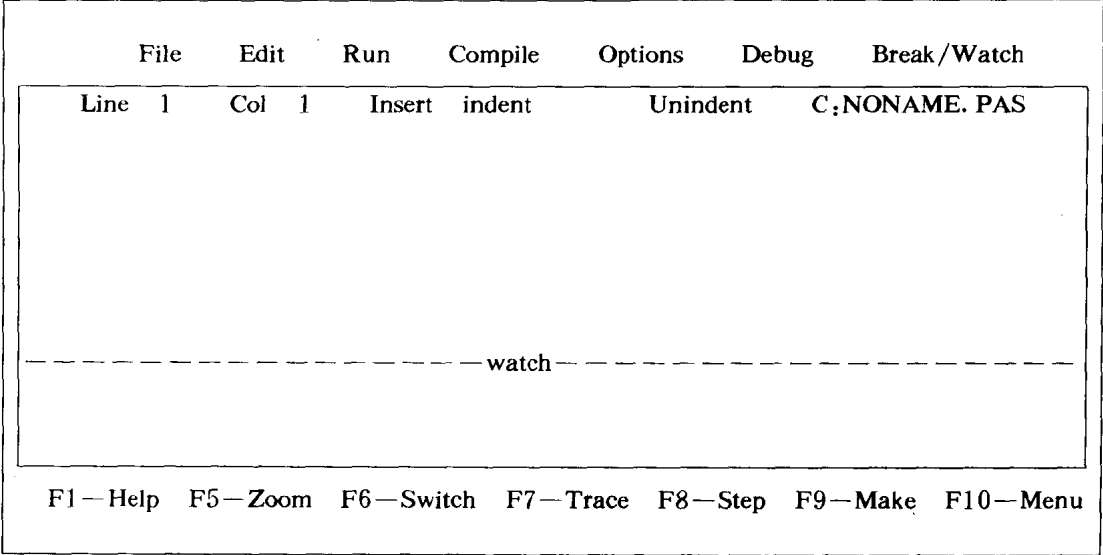


图 1-1 进入 TURBO Pascal 后屏幕示意图

进入 TURBO Pascal 集成环境后,系统自动进入编辑状态,并自动在当前路径名下建立 NONAME.PAS 文件,光标处在第 1 行的第 1 列等待输入。这时您若键入自己的程序,可用 F2 功能键存入,也可在存入时更改系统指定的文件名。方框内的第一行提示说明编辑器目前的功能设置,目的是为了方便的编排程序格式。

当您想退出编辑,进入图 1-1 中顶行提示的主菜单选择时,您可用 F10 键。在主菜单中可用→及←键任选一项进入子菜单。主菜单共有 7 项,包括了集成环境下的各种功能,它们是:

- File 文件处理子菜单。
- Edit 编辑源程序文件。
- Run 运行或单步运行程序。
- Compile 编译子菜单,产生单元文件 TPU 或 EXE 执行文件,查找错误。
- Option 选择项处理子菜单,选择工作及控制参数。
- Debug 调试处理子菜单,调试项,检查和修改变量、堆栈等。
- Break/Watch 断点/观察子菜单,设置监视窗口,处理断点。

由各子菜单回到主菜单用 ESC 键。

图 1-1 中最后一行提示称为辅助菜单。辅助菜单中的选择均用相关功能键调用,这些功能是常用的。

- F1- Help 在 TURBO Pascal 任何地方,可用 F1 得到相应帮助信息。
- F5-Zoom 在分屏模式和整屏模式之间切换。
- F6-Switch 切换当前窗口。
- F7-Trace 单步跟踪调试。
- F8-Step 单步调试,不跟踪到调用子结构。
- F9-Make Make 方式编译,只编译更新的源文件。

使用 TURBO 编辑器编辑源程序文件

在主菜单中选择 File, 进入其子菜单。您就可根据自己的意愿选择一个已有的示例程序编辑或浏览, 或自定义一个不超过 8 个字符长的新文件名输入您的程序。同时在此子菜单中还有清除当前文件内容, 存入文件内容, 更新当前子目录等一系列文件配套操作。File 的子菜单各项如下:

Load F3	装入文件
Pick Alt-F3	选择装入文件
New	创建新文件(清除当前屏幕上的编辑文件)
Save F2	存盘
Write to	写文件
Directory	查目录
Change dir	改变当前目录
Os shell	调用 Dos 环境
Quit Alt-x	退出 TURBO 集成环境

子菜单中的各项可用 ↑、↓ 键来选择, 也可使用有关的功能键及组合键。当您选择 Load 后, 屏幕上就出现了一个文件名图框, 询问要装入的文件名。您若指定的文件名不带扩展名, 则系统默认为 .PAS。当输入的文件名在机器的当前目录下存在时, 则系统自动装入供您编辑或浏览, 否则将认为是一新的程序文件名, 这时光标停在 1 行 1 列要求您输入源程序。当您输入或修改源程序时请定时用 F2 键, 或 Save 选择存入程序以免丢失输入或新修改的程序。

TURBO 编辑器提供了丰富的手段供您快速准确地修改源程序, 如块操作、查找替换、设置各种状态及开关等。但只要了解如下几点就可进行编辑工作。

编辑工作首要的一条就是很快找查要修改的字符位置。最直观的就是使用 ↑、↓、→、← 键来移动光标, 其次可用如下几个键配合。

Home 键	光标移到行头。
End 键	光标移到行尾。
PgUp 键	上翻一页。
PgDn 键	下翻一页。
Ctrl-PgUp	移到当前文件头。
Ctrl-PgDn	移到当前文件尾。

找到要更改的字符位置后, 下面的问题就是解决如何插入新内容, 覆盖老字符, 删除无用字符。

由 Ins 键控制编辑窗第一行中的 Insert 提示, 当 Insert 出现时, 您键入的字符将插在光标处, 同时光标自动后退一个字符位置。

使用 Del 键可删除光标所指的字符, Backspace 键可删除光标之前的字符。删除后右边内容左移。用 Ctrl-y 组合键可删除光标所在的一行。