

全国广播电视中专通用教材  
计算机及应用专业

# 程 序 设 计

曲郁生 主编

清 华 大 学 出 版 社

(京)新登字 158 号

## 内 容 简 介

Pascal 语言是一种用于结构化程序设计的高级语言,它比较适合程序设计课程的教学。本书以 Turbo Pascal 6.0 版本为基础,系统的介绍了 Pascal 语言程序设计的基本概念和方法,同时通过一些典型的例题,重点说明了 Pascal 程序设计的一般技术。教材共分十章,各章内容安排由浅入深,循序渐进。主要内容包  
括:Pascal 语言的基础知识,结构化程序设计的主要思想;顺序结构,选择结构和循环结构程序设计的方法;模块化程序设计技术的使用,Pascal 语言中常用数据类型在程序中的应用,单元及面向对象程序设计的基本概念。每章后面均附有习题,供学习时使用。

本书是为广播电视中专计算机专业编写的教材,同时也适用于其他相关专业的学生和自学者使用。

版权所有,翻印必究。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

书 名: 程序设计

作 者: 曲郁生 编著

出版者: 清华大学出版社(北京清华大学学研楼,邮编 100084)

<http://www.tup.tsinghua.edu.cn>

印刷者: 北京牛山世兴印刷厂

发行者: 新华书店总店北京发行所

开 本: 787× 1092 1/16 印张: 13 字数: 313 千字

版 次: 2000 年 9 月第 1 版 2000 年 9 月第 1 次印刷

书 号: ISBN 7-302-00667-9/TP·228

印 数: 00001 ~ 5000

定 价: 16.00 元

# 前 言

Pascal 语言是一种结构化程序设计语言,它具有较丰富的数据类型和简洁灵活的操作语句和良好的程序设计风格,能方便地编写系统软件和应用软件,是学习程序设计课程的主选语言之一。本书采取高级语言介绍和程序设计方法学习相结合的方式,以 Turbo Pascal 6.0 版本为基础,通过简单易懂的 Pascal 程序和设计方法的介绍来说明程序设计的主要内容和基本技术。全书以结构化程序设计和模块化程序设计方法为重点,结合 Pascal 语言中的数据类型应用,逐步深入地讨论了程序设计中的基本概念和基本方法。各章节的划分主要是根据广播电视中等专业学校计算机应用专业教学大纲和学生学习的实际需要。

本书共分十章。其中第 1 章、第 2 章、第 3 章、第 4 章由四川电大陈梦老师编写,第 5 章、第 6 章由成都电大庞晓凤老师编写,第 7 章、第 8 章、第 9 章和第 10 章由四川电大曲郁生老师编写,全书由曲郁生老师主编和统稿。建立本书教学时数为 72 课时,重点放在学习程序设计的主要技术和培养学生的实际编程能力上。

本教材在编写过程中,得到了中央广播电视中等专业学校、四川广播电视大学和清华大学出版社的领导和老师们的大力支持和热情帮助,由此表示诚挚的谢意。由于编著水平有限,在书中难免存在不少缺点和错误,恳请广大师生批评指正。

作 者

2000 年 7 月

# 出版说明

广播电视中等专业教育的起点是初中毕业生,目标是培养有较强实际工作能力和综合素质的应用型中级专业技术人员。

自1985年全国广播电视大学系统相继开办广播电视中等专业教育以来,在中央广播电视大学的支持下,全国广播电视大学中专教材建设委员会(前身为全国广播电视大学电视中专教材编制委员会)从1989年12月起,经过五年的努力,已编制出版了财经、机电、农产品加工等类课程的系列文字教材及配套的练习册、实验指导书、教学指导书等,这一系列广播电视中专教材的出版,对推动全国广播电视中专教育的发展起到了积极作用。

1995年,国家教委下发了《关于加强广播电视、函授中等专业教育管理的意见》的文件。根据文件有关精神,中央广播电视中等专业学校为了进一步规范教学要求,提高教材的建设质量,与全国广播电视大学中专教材建设委员会共同组织有关学科专家和教师,对原出版的部分教材按照新编课程教学大纲进行了重新编写或修订,并针对一些适应面广、通用性强的专业,开始了新一轮的教材建设。

广播电视中专教材编制宗旨是:紧密结合培养目标,针对学生特点,充分依靠学科和媒体设计与制作专家,进行面向21世纪多种媒体的教材建设。主媒体是文字教材,同时还制有音像等现代媒体教材与之相配套。广播电视中专教材亦可供普通中专、成人中专、职业高中、职业培训等选用,还可供自学使用。

敬请广大教师和读者在使用广播电视中专系列教材过程中,对其存在的不足之处予以批评指正,以便修订再版。

中央广播电视中等专业学校  
全国广播电视大学中专教材建设委员会

1998年5月

# 目 录

前言 .....	
第 1 章 绪论.....	1
1.1 计算机程序和结构化程序设计 .....	1
1.1.1 计算机程序与程序设计.....	1
1.1.2 程序的三种基本结构.....	3
1.1.3 程序设计流程图.....	5
1.2 程序设计语言 .....	6
1.2.1 程序设计语言.....	7
1.2.2 程序设计语言 Pascal .....	8
1.2.3 Turbo Pascal 6.0 集成环境及其使用 .....	9
习题 1 .....	15
第 2 章 Pascal 程序设计基础 .....	16
2.1 Pascal 程序结构.....	16
2.2 Pascal 语言的基本符号.....	18
2.2.1 基本符号 .....	18
2.2.2 保留字 .....	18
2.2.3 标识符 .....	19
2.3 简单数据类型.....	20
2.3.1 数据类型 .....	21
2.3.2 常量与变量 .....	21
2.3.3 标准数据类型 .....	23
2.3.4 用户自定义数据类型 .....	24
2.4 运算符与表达式.....	27
2.5 标准函数及其引用.....	31
2.5.1 算术运算函数 .....	32
2.5.2 转换函数 .....	33
2.5.3 进退函数 .....	33
2.5.4 逻辑函数 .....	34
2.5.5 函数应用举例 .....	34
习题 2 .....	35

第 3 章 顺序结构程序设计 .....	38
3.1 Pascal 语言的语句 .....	38
3.2 复合语句 .....	39
3.3 赋值语句 .....	41
3.4 输入语句 .....	42
3.4.1 read 语句 .....	42
3.4.2 read ln 语句 .....	43
3.5 输出语句 .....	46
3.5.1 write 语句 .....	46
3.5.2 write ln 语句 .....	48
3.6 顺序结构的程序设计 .....	48
习题 3 .....	50
第 4 章 选择结构程序设计 .....	52
4.1 IF 语句 .....	52
4.2 CASE 语句 .....	55
4.3 多重选择结构 .....	59
4.4 标号说明与 GOTO 语句 .....	60
4.5 选择结构的程序设计 .....	62
习题 4 .....	65
第 5 章 循环结构程序设计 .....	68
5.1 WHILE 语句 .....	69
5.2 REPEAT 语句 .....	74
5.3 FOR 语句 .....	78
5.4 多重循环 .....	82
5.5 循环结构的程序设计 .....	85
习题 5 .....	89
第 6 章 构造数据类型 .....	92
6.1 集合类型 .....	92
6.1.1 集合类型概念 .....	92
6.1.2 集合的运算 .....	93
6.1.3 集合的输入与输出 .....	95
6.1.4 集合类型应用 .....	96
6.2 记录类型 .....	97
6.3 数组类型 .....	99
6.3.1 一维数值 .....	100
6.3.2 二维数值 .....	104

6.3.3	紧缩数组与字符串变量.....	107
6.4	构造数据类型的应用举例 .....	109
	习题 6 .....	112
<b>第 7 章</b>	<b>模块化程序设计.....</b>	<b>114</b>
7.1	模块化程序设计方法概述 .....	114
7.2	函数 .....	116
7.2.1	自定义函数的定义.....	116
7.2.2	自定义函数的调用.....	117
7.2.3	自定义函数使用时的几点说明.....	118
7.3	过程 .....	120
7.3.1	过程的定义.....	121
7.3.2	过程的调用.....	123
7.3.3	过程使用中的几点说明.....	123
7.4	子程序中的参数传递 .....	125
7.5	变量与标识符的作用域 .....	130
7.5.1	作用域的基本概念.....	130
7.5.2	局部变量和它的作用域.....	130
7.5.3	全程变量和它的作用域.....	131
7.5.4	标识符的作用域.....	133
7.6	向前引用和子程序的递归调用 .....	135
7.6.1	向前引用.....	135
7.6.2	递归的概念.....	137
7.6.3	递归子程序的应用.....	139
7.7	模块化程序设计的应用 .....	141
	习题 7 .....	144
<b>第 8 章</b>	<b>文件类型及应用.....</b>	<b>150</b>
8.1	文件概述 .....	150
8.1.1	文件的基本概念.....	150
8.1.2	文件的分类.....	150
8.2	文件类型的定义和文件变量 .....	151
8.2.1	文件类型的定义和文件变量.....	151
8.2.2	文件指针和文件缓冲变量.....	152
8.3	文件的操作和文件操作中常用的过程与函数 .....	153
8.3.1	文件操作的步骤.....	153
8.3.2	文件操作中常用的过程与函数.....	154
8.4	类型文件的操作 .....	158
8.4.1	类型文件的操作.....	158

8.4.2	类型文件操作应用实例.....	158
8.5	文本文件的操作 .....	159
8.5.1	文本文件中的行操作过程.....	159
8.5.2	文本文件的操作.....	159
8.5.3	标准文件和读写文本文件时数据类型的自动转换.....	161
8.6	文件应用的程序设计 .....	162
	习题 8 .....	165
<b>第 9 章</b>	<b>指针类型及应用.....</b>	<b>168</b>
9.1	动态数据类型的基本概念 .....	168
9.2	指针变量和动态变量 .....	169
9.2.1	指针变量的类型定义.....	169
9.2.2	指针变量的说明.....	170
9.2.3	指针变量的使用.....	170
9.2.4	指针变量和动态变量的运算操作.....	171
9.3	指针的应用 .....	174
	习题 9 .....	178
<b>第 10 章</b>	<b>单元及面向对象程序设计 .....</b>	<b>180</b>
10.1	Turbo Pascal 的单元 .....	180
10.1.1	单元的概念 .....	180
10.1.2	单元的结构 .....	180
10.1.3	标准单元 .....	181
10.1.4	单元的使用 .....	183
10.2	面向对象的程序设计 .....	186
10.2.1	面向对象的基本概念 .....	186
10.2.2	对象的定义 .....	187
10.2.3	方法的定义 .....	188
10.2.4	对象实例的说明 .....	188
10.2.5	继承性的使用 .....	188
10.2.6	面向对象的程序设计举例 .....	190
	习题 10 .....	191
附录	Turbo Psacal 的语法图 .....	192
	参考文献.....	198

# 第 1 章 绪 论

## 本章内容提要

- 计算机程序与程序设计, 指令, 程序, 语句与程序设计方法。
- 结构化程序, 顺序结构, 选择结构, 循环结构, 单入口, 单出口。
- 结构化程序设计。
- 流程图, 结构化流程图。
- 计算机语言, 机器语言, 汇编语言, 高级语言。源程序, 目标程序, 编辑与编译等。
- 程序设计语言 Pascal 的特点。
- Turbo Pascal 6.0 集成环境及使用方法。

计算机已经深入到了我们生活的每一个角落, 从文字处理、数据和资料管理, 到火箭、卫星的控制, 甚至电视机、电冰箱的控制等, 这都有赖于计算机系统的强大功能, 而所有这些都是由于有了丰富多彩的软件, 有了人们用各种语言设计的程序, 再由计算机按照人们的设计来执行完成的。掌握计算机操作应用的基本技能是对新世纪人才最基本的要求, 学习和掌握程序设计的基本方法, 可编写自己需要的程序, 使计算机更好地为我们服务。

## 1.1 计算机程序和结构化程序设计

在一般人看来, 计算机程序有一点神奇, 有一些神秘, 还十分复杂, 其实, 从某种意义上讲, 计算机程序就是按照某种计算机语言的语法规则, 对工作对象和工作过程进行的描述。在实际工作中, 程序设计带有较大的随意性, 也有一些十分精妙的技巧, 早期的程序设计人员也因此而常被人们誉为天才。然而, 随着计算机的普及, 计算机应用的范围也更加广泛, 单靠程序员来进行程序的开发与设计, 已远远不能满足实际应用的需要。计算机高级语言的发展也使得程序的设计与开发更为容易。同时, 基于程序设计的工程化、提高程序设计的效率以及程序设计的规范性等方面的考虑, 人们提出了结构化程序设计的原则, 从而使程序设计和软件的开发有了飞速的发展。了解计算机程序的概念、掌握结构化程序设计的原则, 是进行程序设计的基础。

### 1.1.1 计算机程序与程序设计

在日常生活中, 我们对“程序”一词已十分熟悉, 如我们常说: 办事要讲究程序, 即是说办事要遵循一定的方法和步骤。例如, 新生入学时的报到注册程序为: 首先, 凭入学通知到学籍科登记, 如果是共青团员, 还要到校团委登记组织关系, 然后, 到校医院体检, 到财务处交费, 再到总务处领取寝室卧具, 到教材处领取教材, 等等(见图 1.1)。

在这个注册程序中, 登记、交费等行为和动作是由学校的有关管理人员根据需要而设定的一个个指令, 新生则按此指令一步一步地执行, 最终完成报到注册。音乐家的乐谱可看成是另一种形式的程序: 音乐家用指定的符号谱写出不同的乐谱, 乐队按照乐谱上的音符进行

图 1.1 新生入学注册程序流程图

演奏,就可表演出优美的音乐。

用计算机进行工作时,程序设计人员首先必须将工作的步骤和方法编写成计算机可以识别的代码,即计算机指令。指令就是指挥计算机工作的命令,一条指令对应于计算机的一个操作。通常,若干指令按一定的顺序排列就组合成一个程序。

如要计算机完成在屏幕上显示:“您好!欢迎进入 Pascal 的世界!”,则可用 Turbo Pascal 编写以下程序:

```
PROGRAM Hello;  
BEGIN  
    write(您好!欢迎进入 Pascal 的世界);  
END.
```

如果这时你身边的计算机中装有 Turbo Pascal(当然还应该先运行汉字系统,不然将不能录入汉字,也看不到汉字),将此程序输入计算机,同时按下“Ctrl”和“F9”运行它,则可在屏幕上看到这一行文字。从这里可以看出:在高级语言中,一个程序是由若干条语句构成的,而一条语句又包含了若干条指令,计算机所能“看”懂的语言已与人类的自然语言十分接近了。但在计算机中,要完成这一工作并不那么简单:在计算机中执行各种语句时,必须先将这些接近自然语言的语句翻译成计算机可识别的指令,才能被执行。

在实际工作中,要让计算机完成复杂的任务,就必须由人事先把任务分解成一些指令的组合,计算机程序就是有序的计算机指令的集合,计算机一步一步地执行指令序列即完成了指定的工作。程序都是由人编写的,并由计算机来实现给出的指令,实现指令也称为“执行”或“运行程序”。

上面我们已涉及了通常的语言形式的程序、流程图形式的程序、乐谱程序及计算机程序。与现实生活中的“程序”相比可看出,计算机程序有以下几个特性:

1. 指令是顺序执行的。除非有专门的要求(如有循环或选择分支),否则将从第一条指令开始,依次执行每一个指令直到结束。

2. 程序的执行将产生一定的结果,这个结果可能是完成注册,也可能是完成一个加法

运算。计算机的运行结果常常是在屏幕上显示或通过打印机打印出来。

3. 程序总是要对某些对象进行操作, 计算机指令的操作对象称为“数据”, 如指令“交费”就要求新生要向财务人员交一定的费用。

4. 通常, 在指令前面有对操作对象的说明或定义。在计算机程序设计语言中, 编程人员在编写操作指令前必须定义对象或说明数据的属性。

5. 在某些指令中, 要由执行者进行判定。在前面的报到注册程序中, 有“如果是共青团员, 还要到校团委登记组织关系”这一判断条件, 在这种情况下, 编写指令的人并不知道在一次实际的运行过程中, 执行者会进行什么操作, 但他可设置一个执行者用于进行判定的标准。

6. 一条或一组指令可能要被执行一次以上, 当一条指令需要重复执行时, 程序员必须指明重复的次数或根据程序运行的结果来确定何时结束重复指令的标准(如当计算结果的误差小于 1% 时, 结束计算)。

7. 程序本身是静态的, 而执行指令的进程是动态的。

总之, 程序从本质上讲只是一种媒介, 它负责完成程序的编写者与程序的执行者之间都认同并识别的通讯。

由于计算机对程序有各种严格的要求, 而计算机本身不具备自动纠正错误的功能; 一个设计良好的程序应该尽量避免各种失误与差错, 程序中的小错误通常称作“Bug”。为了尽量减少错误, 人们提出了用工程的方式来进行程序设计, 并提出了许多相应的规范。因此, 将程序设计看作一个系统的过程, 它通常包括:

- (1) 对计算机所进行工作的分析与定义。
- (2) 设计出相应的方法与步骤(即程序流程图)。
- (3) 用某种语言, 按照其语法规则编写出一个个语句行, 即编写程序代码。
- (4) 通过编译、链接生成可执行文件。
- (5) 在计算机上调试、执行。
- (6) 进行结果分析并完成资料的整理等。

对程序设计的过程, 也可以用一个流程图来表示, 如图 1.2 所示。

人们常常把程序设计认为就是编写计算机程序代码, 但这仅仅是程序设计中很小的一部分工作。编写一些大型的程序或软件是不可能由一个人完成的, 通常要先由系统分析人员进行需求分析, 再进行系统设计, 完成模块的划分, 最后将软件或程序分解, 由若干程序设计人员来协作完成。

从理论上讲, 程序设计可不依赖于任何一种语言, 既可用某种高级语言进行程序设计, 也可用伪语言进行算法设计, 只是计算机不能识别执行罢了。因此, 学习程序设计的重点不在对某种具体语言的了解上, 而应放在程序设计的思路与方法上。

### 1.1.2 程序的三种基本结构

在进行程序设计时, 前面多次提到了要用工程的方法进行

图 1.2 程序设计流程图

程序设计,在实际工作中,结构化程序设计方法即是一种实用的方法。在这种方法中,规定任何程序必须按一定的结构来组织,这就是程序的三种基本结构:顺序结构、选择结构和循环结构。

### 1. 顺序结构:

如图 1.3 所示,这是一种最简单的结构。程序中的各种操作或功能是按其在程序中出现的先后顺序来执行的。如用 Turbo Pascal 来编写一个,完成从键盘上输入一个圆柱体的半径  $r$  和高度  $h$  后,计算出其体积  $v$  的程序如下:

```
PROGRAM sample1;
CONST
    pai= 3.14159;
VAR
    r, h, v : real ;
BEGIN
    write( r, h= ? );
    readln(r, h);
    v:= pai* r * r * h;
    writeln( v= ,v);
END
```

图 1.3 顺序结构

图 1.4 选择结构

图 1.5 循环结构

### 2. 选择结构:

如图 1.4 所示,它是根据指定的条件进行判断,根据判断的结果(满足条件或不能满足条件)在两个分支路径中选择其中的一个来执行。下面是一个 Turbo Pascal 的程序:从键盘上输入一个数,判断其正、负,如果是非负数,则输出其平方根值,反之则在屏幕上显示“输入数据不能为负”。

```
PROGRAM sample2;
VAR
    x: real;
BEGIN
    read(x);
    IF x >= 0
        THEN write(平方根为: ,sqrt(x))
        ELSE write(输入数据不能为负);
END.
```

### 3. 循环结构:

如图 1.5 所示,它是在给定的条件下重复执行某种操作。循环结构中重复执行的部分称为“循环体”。在 Turbo Pascal 中,实现循环结构十分方便。在上一个程序中,只能执行一次计算程序就结束了,可将上一个程序改为输入某个数(如 0)后程序结束,通过这种方式形成一个循环,这样可计算多个数的平方根。

```
PROGRAM sample3;
VAR
  x: real;
BEGIN
  REPEAT
    read(x);
    IF x >= 0
      THEN write(平方根为: ,sqrt(x))
      ELSE write(输入数据不能为负)
    UNTIL x = 0
END.
```

在结构化程序设计中,任何程序都可按这三种基本结构进行组织,在程序中不存在无规律的转移或分支,只在基本结构内才有向前或向后的跳转或分支。由此设计出的程序有以下一些共同点:

- (1) 单入口,单出口。在三种基本结构(即图 1.3、图 1.4、图 1.5 中的各虚线框中的部分中,只有一个入口和一个出口)。
- (2) 无“死语句”。在结构化程序设计中,程序中的每一个语句都有可能被执行到(只是有些语句被执行到的几率较小)。
- (3) 无“死循环”。在基本结构中,必存在满足终止循环体执行的条件,没有永远也执行不完的语句。

在结构化程序设计中,任何一个正常的可运行的程序都可归结到三种基本结构中,或是三种基本结构的有机组合。

### 1.1.3 程序设计流程图

在进行程序设计时,人们常把一些问题进行分解,形成一些基本的顺序与操作的步骤。为了用简明的方法表示要进行的操作与步骤,人们提出了流程图的方法。流程图是用一些有固定形状的图框来表示不同类型的操作,使程序的设计更明确,表达更直观。流程图的用处在前面的例子中已看得很清楚了。在程序设计中,有多种形式的流程图,最常见的即是本节所采用的形式。流程图是形象地描述程序设计思想的好方法。它比较直观。通常用一个框表示某一段程序(一般包含一个或几个语句),用一个箭头表示执行的路径及方向。在这种流程图中,各种图形框的功能和意义见图 1.6。

用流程图表示的程序形象直观、灵活、自由。在结构化程序设计中,不允许流程毫无限制地转移。

在结构化程序设计中常采用 N-S 图(由美国学者 Nassit B. Shneiderman 提出的一种

图 1.6 程序设计流程图例

无流线的流程图)或 PAD 图(Problem Analysis Diagram, 问题分析图)的方式进行程序的描述, 本书不做进一步说明。

下面看一个用流程图进行设计的实例。

[例 1] 计算自然数 1 到 N 的和, 并将结果显示出来。

```
PROGRAM sample4;
  VAR
    sum, i, n: integer;
  BEGIN
    read(N);
    sum = 0;
    i = 1;
    WHILE i <= n DO
      BEGIN
        sum = sum + i;
        i = i + 1;
      END;
    writeln( sum = , sum)
  END.
```

图 1.7 程序流程图

## 1.2 程序设计语言

在计算机发展过程中, 出现了各种各样的计算机程序设计语言, 所有语言的共同目的就是使计算机能正确理解并执行人们发出的指令。正如人与人交流信息时需要通过语言一样, 人与计算机之间的通讯也必须通过某种语言, 这就是计算机程序设计语言。在现实生活中, 已有许多的自然语言, 如汉语、英语等, 但对计算机而言, 这些语言不能完全满足程序设计任务的需要, 从而出现了众多的程序设计语言。这与现实生活中提到的程序有相似之处: 各种语言编写的程序通过各自的语言与控制的对象进行通讯, 如音乐家通过乐谱与乐队进行通

讯、舞蹈家通过形体语言向人们传达他的信息, 程序人员则通过某种程序设计语言来控制计算机进行工作。

### 1.2.1 程序设计语言

计算机语言的种类非常的多, 总的来说, 可以分成机器语言、汇编语言、高级语言三大类, 那我们学习计算机语言有什么用呢? 因为计算机每做的一次动作, 一个步骤, 都是按照已经用计算机语言编好的程序来执行的。程序是计算机要执行的指令的集合, 而程序全部都是用我们所掌握的语言来编写的, 所以, 人们要控制计算机, 一定要通过计算机语言向计算机发出命令。

#### 1. 机器语言

从电子计算机的基本工作原理上看, 计算机能识别和接受的只能是数字 0 和 1 这一类二进制指令。指令是在设计计算机的时候编排的, 某种型号的计算机指令的全体叫该机型的指令系统, 计算机只能理解和执行它自己的指令, 这种指令也称为该 CPU 的机器语言, 它完全由数字 0 和 1 的序列构成。但通常人们编程时, 不采用机器语言, 因为它非常难于记忆和识别, 并且用机器语言编写的程序通用性很差。

#### 2. 汇编语言

为解决机器语言难于使用的问题, 人们提出了用助记符的方式来表示机器指令, 这就是汇编语言。在汇编语言中, 加法指令用“ADD”来表示, 循环控制指令为“LOOP”等, 这样, 用一些有意义的英文单词或缩写符号来代替对人而言难记难懂的数字 0 和 1 所组成的机器指令。在汇编语言中, 每一条汇编指令对应一条机器指令。汇编语言比机器语言更便于记忆和理解, 用户在程序中可直接用二进制数或助记符指令进行计算。汇编语言的实质和机器语言是相同的, 都是直接对硬件操作, 只不过指令采用了英文缩写的标识符, 更容易识别和记忆。它同样需要编程者将每一步具体的操作用命令的形式写出来。汇编程序通常由三部分组成: 指令、伪指令和宏指令。汇编程序的每一句指令只能对应实际操作过程中的一个很细微的动作, 例如数据传送、数据移动, 因此汇编源程序一般比较冗长、复杂、容易出错, 而且使用汇编语言编程需要有更多的计算机专业知识。但汇编语言的优点也是显而易见的, 用汇编语言所能完成的操作不是一般高级语言所能实现的, 而且源程序经汇编生成的可执行文件不仅比较小, 而且执行速度很快。

从机器语言到汇编语言尽管只是小小的一步, 但计算机不能直接识别并执行最简单的助记符, 要先将助记符指令转换成二进制的机器指令, 计算机才能执行, 这种转换是由专门的“汇编程序”软件来完成的, 转换的过程就叫“汇编”。

由于汇编语言的指令语句与机器指令是一一对应的, 所以它是依赖于某种型号的计算机, 也还是一种“低级”的语言。

#### 3. 高级语言

这是从低级的机器语言抽象而来的计算机语言, 也常称为算法语言。高级语言通常使用与英语单词类似的关键词, 这些关键词可以被翻译成多条机器指令。实际上, 比汇编语言高级的计算机语言都是高级语言。和汇编语言相比, 它不但将许多相关的机器指令合成为单条指令, 并且去掉了与具体操作有关但与完成工作无关的细节, 例如使用堆栈、寄存器等, 这样就大大简化了程序中的指令。同时, 由于省略了很多细节, 编程者也就不需要有太多的专业

知识。高级语言主要是相对于汇编语言而言,它并不是特指某一种具体的语言,而是包括了很多编程语言,如目前流行的 Visual Basic、C/C++、Pascal、FoxPro、Delphi 等,这些语言的语法、命令格式都各不相同。

高级语言是一种与机器无关的语言,用高级语言所编制的程序不能直接被计算机识别,必须经过转换才能被执行,按转换方式可将它们分为两类:

(1) 解释类: 执行方式类似于我们日常生活中的“同声翻译”,应用程序源代码一边由相应语言的解释器“翻译”成目标代码(机器语言),一边执行,因此效率比较低,而且不能生成可独立执行的可执行文件,应用程序不能脱离其解释器,但这种方式比较灵活,可以动态地调整、修改应用程序,典型的解释型的高级语言有 QBasic。

(2) 编译类: 编译是指在应用源程序执行之前,就将程序源代码“翻译”成目标代码(机器语言),因此,目标程序可以脱离其语言环境独立执行,使用比较方便、效率较高。但应用程序一旦需要修改,必须先修改源代码,再重新编译生成新的目标文件(\*.OBJ)才能执行,只有目标文件而没有源代码,修改很不方便。现在大多数的编程语言都是编译型的,例如 Basic、C、Visual FoxPro、Delphi 等。

用汇编语言或高级语言编写的程序叫源程序,通常是 ASCII 方式的文本文件,用户可用通用的文本编辑软件对其进行编辑、修改。源程序经过汇编或编译程序“翻译”成可被计算机接受的目标程序,再链接到对应的操作系统即可执行,并得出结果。

### 1.2.2 程序设计语言 Pascal

1968 年,瑞士苏黎世联邦工业大学 N. 沃思(Niklaus Wirth)教授设计了 Pascal 语言,于 1970 年推出了第一个编译程序。Pascal 语言是一种较完备的满足结构化程序设计的语言,这是因为 Pascal 语言有如下一些特征:

1. 变量说明是强制性的。
2. 某些基本词语(如: BEGIN, END, REPEAT 等)是“保留”的,不能用作标识符。
3. 分号“;”用作语句的分隔符,而不是语句的终止符。
4. 标准数据类型是整数、实数、字符和布尔类型。基本的数据结构有数组、记录、集合和文件。这些结构可组合和嵌套以形成集合的数组、记录的文件等。数据还可以动态地分配并由指针访问。
5. 数组可具有任意维数与任意的界,数组的界是常量,即没有动态数组。
6. 常量、变量等所有对象必须在它们被引用之前加以说明。
7. 与其它语言一样,存在转移语句,也有循环语句。
8. 过程和函数可递归调用。
9. 具有强大的图形功能,支持多种显示器和驱动器。而且计算功能、逻辑判断功能也比较强大,可以实现决策目的。
10. 适用范围大,既适用于多种操作系统如 DOS、UNIX 等,也适用于多种机型。

总之,Pascal 语言是按照结构化程序设计原则而设计的语言,有丰富的数据类型,简明的通用语句,清晰的程序结构,书写格式自由,编译紧凑,Pascal 语言的这些特点,为程序设计的基本概念和结构提供了简明的、系统的、精确的和合理的描述方法,使计算机科学中的数据结构、编译技术、离散数学等课程中涉及的算法都可通过它来进行描述。

目前,国际上有多种版本的 Pascal 语言,国际标准化组织 ISO 对其进行了规范。Turbo Pascal 是在国际标准的基础之上对其进行了扩展而推出的一个 Pascal 编程的集成化开发环境(IDE)。

Turbo Pascal 是美国 Borland 公司的产品, Borland 公司是一家专门从事软件开发、研制的大公司。该公司相继推出了一套 Turbo 系列软件,如 Turbo Basic、Turbo Pascal、Turbo C 等编程与开发工具,这些软件很受用户欢迎。该公司在 1990 年推出的 Turbo Pascal 6.0,其中使用了全新的集成开发环境,运用了一系列下拉式菜单将文本编辑、程序编译、连接以及程序运行一体化,大大方便了程序的开发。而其提供的面向对象(Object Oriented)程序设计的支持,使得读者可较方便地将学习的知识迁移到新的设计风格上。

### 1.2.3 Turbo Pascal 6.0 集成环境及其使用

Turbo Pascal 6.0 提供的集成环境将编程中涉及的文本文件的编辑、修改,源程序的编译、运行、调试等功能全部集中在集成开发环境(IDE)中。在安装了 Turbo Pascal 6.0 的计算机中,进入 Turbo Pascal 6.0 的目录(通常为 TP 子目录),执行“turbo”命令(注意:如果在程序中要使用汉字,则应先进入汉字系统,如进入希望汉字系统,要先执行“ucdos”),启动 Turbo Pascal 6.0 的集成开发环境。进入集成开发环境后,屏幕上显示如图 1.8 所示。

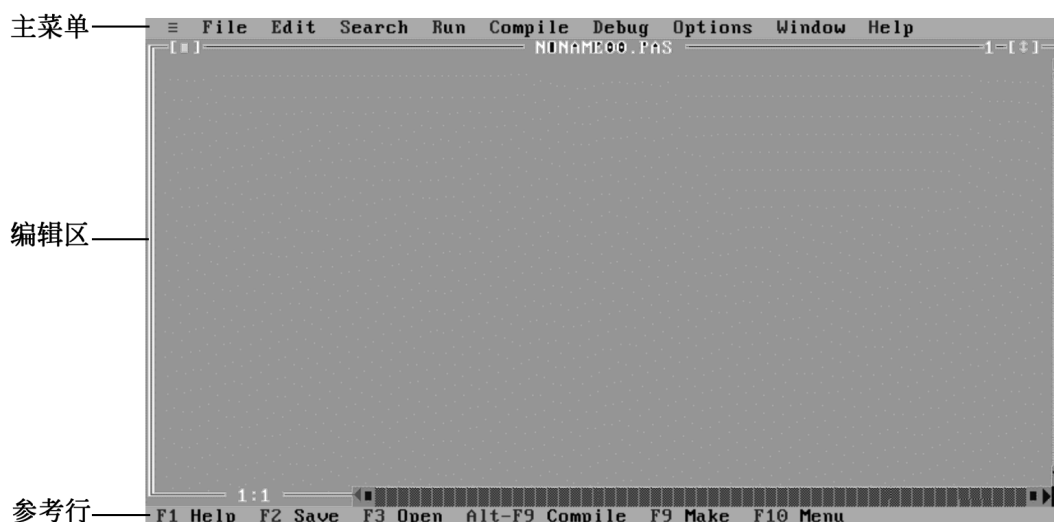


图 1.8 Turbo Pascal 6.0 的集成开发环境

其中顶上一行为 Turbo Pascal 6.0 的主菜单,中间窗口为编辑区,最底下一行为参考行(如果运行了汉字系统,则再底下一行为汉字系统的状态行),这三个部分构成了 Turbo Pascal 6.0 的主屏幕,以后的编程、编译、调试以及运行都将在这个主屏幕中进行。还可使用鼠标来进行操作,本书在后面的介绍中只以键盘操作为主。下面详细介绍 Turbo Pascal 6.0 的集成开发环境的内容。

Turbo Pascal 6.0 主菜单显示下列内容:

File Edit Search Run Compile Debug Options Window Help

每个菜单项下均有子菜单,只要用 Alt 加上某项中第一个字母(即大写字母),就可进入该项的子菜单中。

#### 1. File(文件)菜单

按 Alt+ F 键可进入“File”菜单,该菜单内容如图 1.9 所示:

(1) Open(打开): 打开一个文件,可用类似 DOS 的通配符(如\* .PAS)来进行列表选