

● 国防科技大学出版社

Turbo Pascal
Turbo Pascal Turbo Pascal Turbo Pascal
Turbo Pascal Turbo Pascal Turbo Pascal
Turbo Pascal Turbo Pascal Turbo Pascal

TP

TP

● 姚庭宝 编著

Turbo Pascal

语言及编程技巧



73.763
409

Turbo Pascal 语言及编程技巧

姚庭宝 编著

国防科技大学出版社

· 长沙 ·

2001025

图书在版编目(CIP)数据

Turbo Pascal 语言及编程技巧/姚庭宝. —长沙:国防科技大学出版社,1998.12
ISBN 7-81024-510-4

- I. Turbo Pascal 语言及编程技巧
- II. 姚庭宝
- III. 计算机—程序语言—教材
- IV. TP312

国防科技大学出版社出版发行

电话:(0731)4555681 邮政编码:410073

E-mail:gfkdcbs@public.cs.hn.cn

责任编辑:何晋 责任校对:石少平

新华书店总店北京发行所经销

湖南大学印刷厂印装



*

787×1092 1/16 印张:34 字数:786 千

1998年12月第1版第1次印刷 印数:1-3000册

*

定价:38.00 元

内 容 简 介

本书通过介绍和剖析大量程序示例,系统地论述 Turbo Pascal 各种类型的数据结构与语句特性,以及用 Turbo Pascal 进行程序设计的基本方法、技巧及应用,努力使读者既能熟悉一种优良的语言工具,又能掌握如何发挥这一工具效能的方法与技巧,并着力于良好编程风格与习惯的养成,以便为读者今后进一步程序设计实践打下较为扎实的基础。全书主要内容分为上、下两编。上编是语言编,共十章。下编是技巧编,共五章。上编着重于阐述 Turbo Pascal 的诸多概念与特性,下编致力于发展编程技巧,增强灵活运用语言的能力。全书内容翔实,信息量大,结构严谨,编排合理,文字流畅,示例众多,实用性强。文字叙述力求体现启发性、针对性、直观性、技巧性。每章末一般都有小结,并给出一个程序设计问题以综合运用本章概念。除第一章外,每章都配置了练习题。本书程序示例适用于 Turbo Pascal 5.5 以上版本。

作者在选材和编写中尽可能顾及各个层次人士的广泛需求,不仅为大学生们提供了一本学习 Turbo Pascal 语言及编程技巧的教材,而且也给全国广大微机 Turbo Pascal 爱好者提供一本信息量大、非常实用、从入门到精通的读本。

JS237/18

3501005

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上 Pascal 语言的主流,是在教学、科研、开发系统软件和应用软件中的一个颇为理想的工具。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。Borland 公司于 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 1992 年相继推出的 Turbo Pascal 5.0,5.5,6.0 以及 7.0,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;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;而 Turbo Pascal 7.0 则是一个面向对象的程序设计系统,它提供了可分别在实模式和保护模式下运行的两套集成开发环境,并大大地加强了面向对象应用程序框架 Turbo Vision 的能力,推出了 Turbo Vision 2.0 版。近几年来,随着 Microsoft 公司的 Windows 操作系统风靡全球,以及可视化开发语言如 Visual Basic、Visual C++ 等的崛起,Borland 公司紧追潮流,将其系列软件以适用于 DOS 转向适用于 Windows。Borland 公司相继于 1995,1996 及 1997 年推出了其可视化的开发平台 Delphi 1.0、2.0 及 3.0。Delphi 将可视化设计界面、可视化部件和强大的数据库访问能力集成于面向对象的 Pascal 语言开发平台之中。Delphi 集中了可视化编程、面向对象程序设计、程序自动生成、可视化部件类库、客户机/服务器前端应用开发平台等方面最新的风格和优点,是可视化的全功能开发平台。因此,可以将 Delphi 比喻作 Visual Pascal。Delphi 的核心语言是 Object Pascal。目前,Delphi 已在国内迅速推广,正为越来越多的微机用户和软件设计、开发人员所选用。Delphi 的最新版本是 4.0 版。

本书专为大学生入学后以 Turbo Pascal 作为学习的计算机第一教学语言而编写。作者认为,大学低年级是很重要的打基础阶段,其所选计算机语言应能充分体现适宜于大学低年级加强基础、培养与训练计算机语言编程及应用能力、拓宽知识面的特色。Turbo Pascal 能很好地满足这些要求。这是由 Turbo Pascal 自身众多的良好特性以及所提供的卓越的集成开发环境所决定的。本书通过介绍和剖析大量程序示例,系统地论述 Turbo Pascal 各种类型的数据结构与语句特性,以及用 Turbo Pascal 进行程序设计的基本方法、

技巧及应用,努力使读者既能熟悉一种优良的语言工具,又能掌握如何发挥这一工具效能的方法与技巧,并着力于良好编程风格与习惯的养成,以便为读者今后进一步程序设计实践(如应用于《数据结构》等课程)打下较为扎实的基础。

全书由上编、下编和附录组成。上编是语言编,共十章。下编是技巧编,共五章。另设十个附录。上编着重于阐述 Turbo Pascal 的诸多概念与特性,下编致力于发展编程技巧,增强灵活运用语言的能力。故上编在于打下较扎实的语言基础,而下编则重视语言使用技巧及应用。附录中除了介绍 Turbo Pascal 5.5,6.0 与 7.0 的集成开发环境并给出 Turbo Pascal 标准子程序库之外,还对 Borland 公司在自 Turbo Pascal 7.0 版之后推出的 Borland Pascal for Windows 和 Delphi 作简要介绍。另外还集中选编了一批颇富于技巧性及启发性的上机练习题与作业题。为了更能体现大学生教材的特点,每章末一般都有小结,并给出一个程序设计问题以综合运用本章概念。除第一章外,每章都配置了练习题。对于初学者来说,从学习和使用 Turbo Pascal 5.5 开始是较为理想的选择。本书即是以 Turbo Pascal 5.5 为起点。今后书中所提 Turbo Pascal 均指 5.5 以上版本。

作为讲授 Turbo Pascal 的大学生教材,本书计划教学时数为 60 学时。还应安排充足的上机实践机时。需要说明的是,本书无论从内容深度上还是从篇幅上来看,均较所定教学大纲有所充实与加强,这一方面是为了力图给学有余力的同学们提供更为丰富的、便于自学的阅读及上机实践的材料,另一方面也是为了给全国广大微机 Turbo Pascal 爱好者提供一本信息量大、非常实用、从入门到精通的读本。在具体实施教学的过程中,可酌情取舍教学内容,量力安排上机实践计划。

张帆、卢中辉、陈建民、周军民等为本书撰写了部分内容。刘青宝、陈洪辉曾参与本课程的教学,并提出了不少有益建议。国防科技大学出版社何晋、陆荣斌同志及电脑室的工作人员为本书的出版付出了辛勤劳动,社领导一直给予关心与支持,在此一并表示感谢。

对于本书在内容选材和文字叙述上的不当之处,热诚欢迎广大读者特别是使用本书作为教材或教学参考书的大学生们提出批评、建议。

国防科技大学 姚庭宝

1998 年 11 月

目 录

◁ 上编 语言编 ▷

第一章 导引

- 1.1 形形色色的编程语言 (1)
- 1.2 关于 Turbo Pascal (4)
- 1.3 怎样学好用好本书 (8)

第二章 初阶

- 2.1 程序基本结构 (10)
- 2.2 基本字符集与标识符 (13)
 - 2.2.1 基本字符集 (13)
 - 2.2.2 常字 (13)
 - 2.2.3 标识符 (14)
 - 2.2.4 保留字与标准标识符 (14)
- 2.3 简单数据类型 (15)
 - 2.3.1 整数类型 (16)
 - 2.3.2 实数类型 (16)
 - 2.3.3 布尔类型 (17)
 - 2.3.4 字符类型 (17)
 - 2.3.5 枚举类型 (18)
 - 2.3.6 子界类型 (19)
- 2.4 注释、常数定义和变量说明 (20)
 - 2.4.1 注释 (20)
 - 2.4.2 常数定义 (21)
 - 2.4.3 变量说明 (22)
- 2.5 简单类型常数 (22)
- 2.6 表达式 (24)
 - 2.6.1 运算符和优先级 (24)
 - 2.6.2 算术运算符 (24)
 - 2.6.3 逻辑运算符 (25)
 - 2.6.4 关系运算符 (26)
 - 2.6.5 常用预定义算术函数 (27)
 - 2.6.6 常用预定义标量函数 (28)
 - 2.6.7 常用预定义转换函数 (29)
 - 2.6.8 常用预定义杂类函数 (29)
 - 2.6.9 类型转换 (30)
 - 2.6.10 算术表达式、关系表达式和布尔表达式 (30)
 - 2.6.11 常数表达式 (31)

2.7 简单输入、输出及赋值语句	(32)
2.7.1 输入过程 Read 和 Readln	(32)
2.7.2 输出过程 Write 和 Writeln	(32)
2.7.3 整型数据的输出格式	(33)
2.7.4 实型数据的输出格式	(34)
2.7.5 布尔型数据的输出格式	(34)
2.7.6 字符型数据的输出格式	(35)
2.7.7 赋值语句	(35)
2.8 程序设计风格	(36)
2.9 小结与程序设计问题	(38)
2.9.1 小结	(38)
2.9.2 程序设计问题	(38)
练习题	(39)
第三章 程序控制结构	
3.1 简单 if 语句	(43)
3.2 复合语句	(44)
3.3 扩展 if 语句与嵌套 if 语句	(46)
3.4 case 语句	(49)
3.5 while 语句	(50)
3.6 repeat 语句	(52)
3.7 for 语句	(54)
3.8 多重循环结构	(56)
3.9 goto 语句与标号	(62)
3.10 小结与程序设计问题	(64)
3.10.1 小结	(64)
3.10.2 程序设计问题	(65)
练习题	(68)
第四章 字符串和数组	
4.1 字符串类型定义和变量说明	(73)
4.1.1 字符串类型定义	(73)
4.1.2 字符串变量说明	(74)
4.1.3 字符串动态长度	(74)
4.1.4 字符串与字符	(74)
4.2 字符串常数定义和类型常数定义	(75)
4.2.1 字符串常数定义	(75)
4.2.2 字符串类型常数定义	(75)
4.3 字符串表达式和赋值语句	(76)
4.3.1 字符串表达式	(76)
4.3.2 字符串赋值语句	(76)
4.4 字符串变量的输入输出	(77)
4.5 预定义字符串标准函数和过程	(79)
4.5.1 预定义字符串标准函数	(79)

4.5.2 预定义字符串标准过程	(81)
4.6 一维数组类型定义和变量说明	(83)
4.6.1 一维数组类型定义	(83)
4.6.2 一维数组变量说明	(84)
4.7 一维数组类型常数定义	(84)
4.8 一维数组变量的赋值操作	(85)
4.9 使用一维数组	(85)
4.10 多维数组	(91)
4.11 小结与程序设计问题	(95)
4.11.1 小结	(95)
4.11.2 程序设计问题	(96)
练习题	(98)
第五章 记录与集合	
5.1 记录类型定义和变量说明	(101)
5.1.1 记录类型定义	(101)
5.1.2 记录变量说明	(102)
5.2 记录类型常数定义	(103)
5.3 记录变量的赋值操作	(104)
5.4 with 语句和嵌套记录	(106)
5.4.1 with 语句	(106)
5.4.2 嵌套记录	(108)
5.5 变体记录	(110)
5.5.1 变体记录类型定义和变量说明	(110)
5.5.2 判别式联合和自由联合变体记录类型	(113)
5.6 集合类型定义和变量说明	(114)
5.6.1 集合类型定义	(114)
5.6.2 集合变量说明	(115)
5.6.3 集合的值	(116)
5.7 集合类型常数定义	(116)
5.8 集合表达式和集合变量的赋值操作	(117)
5.8.1 集合构造符	(117)
5.8.2 集合运算符	(117)
5.8.3 集合表达式	(118)
5.8.4 集合变量的赋值操作	(119)
5.9 集合的输出	(120)
5.10 小结与程序设计问题	(121)
5.10.1 小结	(121)
5.10.2 程序设计问题	(122)
练习题	(126)
第六章 过程和函数	
6.1 过程及其调用	(131)
6.1.1 过程说明的基本形式	(131)

6.1.2	过程调用——过程语句	(133)
6.1.3	参数传递(实在参数和形式参数,值参数和变量参数)	(133)
6.1.4	全程变量和局部变量	(138)
6.2	函数及其调用	(140)
6.2.1	函数说明的基本形式	(141)
6.2.2	函数的调用	(145)
6.2.3	函数调用的副作用	(147)
6.3	过程或函数的嵌套调用和向前引用	(148)
6.3.1	嵌套调用	(148)
6.3.2	向前引用	(149)
6.4	过程或函数的递归调用	(152)
6.4.1	递归的概念	(153)
6.4.2	递归函数说明与递归过程说明示例	(154)
6.4.3	递归算法与非递归算法	(155)
6.5	无类型参数	(156)
6.5.1	绝对变量与 absolute 子句	(156)
6.5.2	无类型参数——应用绝对变量	(157)
6.5.3	无类型参数——应用强制数据类型转换	(159)
6.6	过程类型、过程变量与过程类型参数	(160)
6.6.1	过程类型定义	(160)
6.6.2	过程变量说明	(161)
6.6.3	过程类型常数定义	(162)
6.6.4	过程类型参数	(162)
6.7	小结与程序设计问题	(164)
6.7.1	小结	(164)
6.7.2	程序设计问题	(164)
	练习题	(167)

第七章 文件

7.1	文件类型定义和变量说明	(171)
7.1.1	文件类型定义	(171)
7.1.2	文件变量说明	(173)
7.2	文件标准过程和函数	(174)
7.2.1	适用于所有文件的标准过程和函数	(174)
7.2.2	类型文件的标准过程和函数	(175)
7.2.3	正文文件的标准过程和函数	(175)
7.2.4	无类型文件的标准过程和函数	(176)
7.3	类型文件操作	(176)
7.4	正文文件操作	(183)
7.5	无类型文件操作	(188)
7.6	I/O 出错检查与 I/O 重定向	(190)
7.6.1	I/O 出错检查	(190)
7.6.2	I/O 重定向	(192)

7.7 Turbo Pascal 的外设	(195)
7.8 小结与程序设计问题	(196)
7.8.1 小结	(196)
7.8.2 程序设计问题	(197)
练习题	(200)
第八章 指针	
8.1 动态数据结构	(201)
8.2 指针类型定义和变量说明	(202)
8.2.1 指针类型定义	(202)
8.2.2 指针变量说明	(203)
8.3 指针类型常数定义	(204)
8.4 用于堆管理的指针标准过程和函数	(205)
8.4.1 New 和 Dispose	(205)
8.4.2 Make 和 Release	(207)
8.4.3 GetMem 和 FreeMem	(208)
8.4.4 MemAvail 和 MaxAvail	(209)
8.5 指针变量的赋值操作	(210)
8.6 线性链表概念	(213)
8.7 线性链表的生成与遍历	(214)
8.7.1 线性链表生成——插表头	(214)
8.7.2 线性链表生成——插表尾	(216)
8.8 在已知线性链表中的插入与删除操作	(219)
8.8.1 插入一个新表元素至已知线性链表中某表元素之后	(220)
8.8.2 插入一个新表元素至已知线性链表中某表元素之前	(220)
8.8.3 在已知线性链表中删除一个表元素	(222)
8.8.4 考虑通用性——对更新操作语句编码的改善	(222)
8.9 双向链表结构与双向链环结构	(223)
8.10 小结与程序设计问题	(225)
8.10.1 小结	(225)
8.10.2 程序设计问题	(226)
练习题	(229)
第九章 单元	
9.1 单元的基本结构	(231)
9.1.1 什么是单元	(231)
9.1.2 单元的基本结构	(232)
9.2 标准单元概述	(234)
9.3 System 单元	(234)
9.3.1 System 单元类型常数和变量	(235)
9.3.2 System 单元标准过程和函数	(237)
9.3.3 System 单元 I/O 过程和函数	(237)
9.4 Dos 单元	(237)
9.4.1 Dos 单元常数、类型和变量	(238)

9.4.2 Dos 单元标准过程和函数	(238)
9.4.3 Dos 单元使用示例	(240)
9.5 Crt 单元	(241)
9.5.1 Crt 单元的若干特征	(241)
9.5.2 Crt 单元常数和变量	(243)
9.5.3 Crt 单元标准过程和函数	(244)
9.5.4 Crt 单元使用示例	(244)
9.6 Printer 单元	(248)
9.7 Graph 单元	(249)
9.7.1 基本概念与术语	(249)
9.7.2 Graph 单元常数、类型和变量	(251)
9.7.3 Graph 单元标准过程和函数	(253)
9.7.4 Graph 单元使用示例	(255)
9.8 Overlay 单元	(257)
9.8.1 覆盖管理	(257)
9.8.2 Overlay 单元变量	(258)
9.8.3 Overlay 单元标准过程和函数	(259)
9.8.4 Overlay 单元使用示例	(259)
9.9 Turbo3 和 Graph3 单元	(260)
9.10 编写用户定义单元	(260)
9.11 使用单元	(262)
9.11.1 编译单元	(262)
9.11.2 使用单元示例	(262)
9.11.3 使用单元时的引用规则	(264)
9.12 单元和大程序	(265)
9.13 小结和程序设计问题	(266)
9.13.1 小结	(266)
9.13.2 程序设计问题	(267)
练习题	(270)
第十章 结构化程序设计思想	
10.1 什么是结构化程序设计	(271)
10.2 结构化程序设计示例	(272)
10.3 再谈程序设计风格	(282)
练习题	(284)

◁ 下编 技巧编 ▷

第十一章 编程技巧示例

11.1 显示图案	(286)
11.2 逻辑判断	(292)
11.3 数据模拟	(297)
11.4 整数问题	(304)

11.5 数据处理	(309)
练习题	(315)
第十二章 通用程序设计	
12.1 一般数值计算	(317)
12.2 关于求素数	(329)
12.3 集合结构应用	(336)
12.4 分类算法概述	(341)
12.4.1 直接分类算法	(341)
12.4.2 改进的分类算法	(342)
12.4.3 分类算法程序示例	(342)
12.4.4 内部分类算法的比较和选择	(346)
练习题	(347)
第十三章 递归技术	
13.1 递归与分治	(348)
13.2 递归算法典型示例	(357)
13.3 应用递归技术描绘平面曲线图案	(360)
13.4 探索策略和回溯算法	(368)
练习题	(378)
第十四章 动态数据结构的基本应用	
14.1 一个简单的指针应用问题	(380)
14.2 直接表和有序表的检索与插入	(385)
14.3 树结构与二叉树结构	(392)
14.4 二叉树的生成与遍历	(394)
14.5 二叉搜索树的检索与插入	(399)
14.6 指针在二叉树结构中的应用示例——构造给定正文的相互参照索引程序	(402)
练习题	(409)
第十五章 面向对象的程序设计入门	
15.1 概述	(410)
15.2 面向对象的基本概念与特征	(411)
15.3 对象	(412)
15.3.1 什么是对象	(412)
15.3.2 对象类型和对象类型实例	(414)
15.3.3 继承	(416)
15.4 方法	(417)
15.4.1 什么是方法	(417)
15.4.2 代码/数据封装	(418)
15.4.3 定义方法	(419)
15.4.4 方法的作用域与 Self 参数	(420)
15.4.5 在单元中定义对象	(422)
15.5 静态方法和虚拟方法	(425)
15.5.1 继承静态方法	(425)
15.5.2 虚拟方法和多态性	(428)

15.5.3 兼容性	(432)
15.6 动态对象与动态分配	(433)
15.6.1 动态对象	(433)
15.6.2 New 的扩展形式	(434)
15.6.3 释放动态对象	(435)
15.6.4 析构方法	(435)
15.6.5 动态对象分配示例	(437)
练习题	(440)
附录一 Turbo Pascal 5.5 集成开发环境	(442)
附录二 Turbo Pascal 6.0 集成开发环境	(447)
附录三 编译指示一览表	(467)
附录四 Turbo Pascal 标准子程序库	(470)
附录五 编译与运行出错信息	(496)
附录六 Turbo Pascal 7.0 集成开发环境	(502)
附录七 Borland Pascal 7.0 for Windows	(513)
附录八 Delphi 简介	(515)
附录九 ASCII 码表及键盘返回代码	(518)
附录十 上机练习题与作业题选编	(521)

主要参考书目

◁ 上编 语言编 ▷

第一章 导 引

1.1 形形色色的编程语言

软件开发当今是一个热门行业,国内外有数以十万计的软件开发工作者正在从事这项复杂的智力活动,并对国计民生产生着深远的影响。他们使用着形形色色的编程语言。对于较小的软件开发项目,其最终产品可以是使用一种编程语言写成的一个软件系统,加上必要的文档。但事实上,许多较大规模的软件系统是用多种编程语言实现的。在实现软件开发的过程中,编程语言的选择是很关键的。编程语言的优良特性加上良好的编程风格,将强烈地影响着软件开发的进程,对确保软件的可靠性、可读性、可测试性、可维护性以及可重用性等等将起着重大的作用。

为能保证程序编码的质量,软件开发工作者必须深刻地理解、熟练地掌握并正确地运用编程语言的诸多特性。然而,软件开发项目对程序编码的要求不仅仅是其源程序语法上的正确性,也不只是源程序中没有各类错误,还要求源程序具有优良的结构性和良好的编程风格。程序的正确性并非是对程序质量的唯一要求。当今对软件开发的质量要求中,至少应当包括容易阅读以及方便使用。如果源程序编码包括相应文档既便于阅读,又便于测试和排除所发现的程序故障,就能够有效地在软件开发期间消除绝大多数程序中的隐患,大大减少软件运行期间失效的可能性,从而提高软件的可靠性。假如所编写的源程序在运行过程中发现了问题或者错误时很容易修改,而且当软件在使用过程中,能根据计算机技术发展的需要或用户的需求,很容易地扩充其功能,改善其性能,则这样的软件系统就具有较好的可维护性,维护人员可以很方便地对它进行修改、扩充和移植。而软件可重用性更是当今计算机业界所关心的热点之一。它对于提高软件生产率与质量、降低软件成本有着特殊重要的意义。

编程语言大致可以分为汇编语言和高级语言两大类。从应用特点来看,高级语言可以分为基础语言、结构化语言和专用语言三类。从语言的内在特点来看,高级语言可以分为系统实现语言、静态高级语言、块结构与有限的动态高级语言以及动态高级语言等四类。

汇编语言是依赖于具体计算机的语言。用它编写出来的程序,执行效率高,但在其他方面优点甚少。因此,只是在高级语言无法满足设计要求时,或者不具备支持某种特定功能(如特殊的输入/输出)的技术性能时,才被使用。以往在系统软件这个层次上使用汇编代码颇为广泛,即编译程序、操作系统以及解释程序等常用汇编代码编写。但近些年来,许多系统级的程序,包括编译程序和操作系统等,都已经使用高级语言来编写了。现代优化的高级语言编译程序所产生的代码,其运行速度已能与汇编语言代码的运行高速相当。汇编语言的主要缺点是可读性差、难以维护,编程错误在汇编语言程序中频繁出现。

基础语言是通用语言,其特点是历史悠久、应用广泛、有大量软件库,已为多数人所熟悉和接受。基础语言的适用性强,应用面广。它们是传统的计算机高级语言。具代表性的有 BASIC、FORTRAN、COBOL、ALGOL 等等。

BASIC(Beginner's All-purpose Symbolic Instruction Code,初学者通用指令代码)是一种解释执行的交互式会话语言,其特点是简单易学。目前较流行的有 True BASIC、Turbo BASIC、Quick BASIC、VB(Visual BASIC)等。

FORTRAN(FORmula TRANslation,公式翻译)是科学和工程计算中使用的主要语言,有大量成熟、实用的子程序。较新版本是 FORTRAN90。

COBOL(COMmon Business Oriented Language,面向商业的通用语言)是商业数据处理中广泛使用的语言。它与 FORTRAN 均是美国国防部标准语言。由于本身结构上的特点,使得它能够有效地支持与商业处理有关的、范围广泛的过程技术。它的缺点是不简洁。

ALGOL(ALGOrithmic Language,算法语言)是所有结构化语言的前驱,它具有丰富的过程和数据结构,语法严谨。ALGOL 在欧洲使用广泛,但并没有被国际上广泛采用,这是由它本身和历史的原因所造成的。

结构化语言是在 20 世纪 60 年代末国际上针对如何解决“软件危机”而提出软件工程的概念后于 70 年代开始迅速发展起来的一批计算机高级语言,它们具有很强的过程功能和数据结构功能,并提供结构化的逻辑构造。其中最具代表性的有 ALGOL(前已叙述)、PL/1、Pascal、Modula、C、Ada 等等。

PL/1(Programming Language One,程序设计语言 1)是一个用途广泛的语言。它能支持通常的科学工程和商业应用,能描述复杂的数据结构、多重任务处理、复杂的输入输出和表格处理等。但它仅应用于较大型的计算机系统中。

Pascal 是 70 年代初发展起来的一种结构化程序设计语言,它具有特别丰富的数据结构类型和优良的结构化特性。它原先是被作为一种教学语言而推出,但自其问世后,得到了交口赞赏,也得到了软件开发者的广泛支持。Pascal 除作为优良的教学语言外,还广泛应用于系统程序设计以及科学工程、信息处理技术之中。我国教育部计算机专业教育委员会会议曾把 Pascal 定为计算机专业程序设计语言。目前 Turbo Pascal 实际上已成为微机上的 Pascal 的现实标准。关于 Turbo Pascal,下节将有详述。

Modula 是 Pascal 设计者 N. Wirth 教授继 Pascal 后设计的另一种计算机高级语言。它

具有很强的模块化特色。它不仅支持信息隐蔽、抽象和强数据类型等设计特性的直接实现,而且还提供了支持递归和并发控制结构。较广泛使用的是 Modula-2。

C 是作为 UNIX 操作系统的主要使用语言。由于 UNIX 操作系统的成功,现今 C 语言也得到了广泛的应用。C 是为有经验的软件工程师所设计的,它具有很强的功能以及高度的灵活性。它和其他结构化语言一样,能够提供丰富的数据类型、广泛使用的软件以及一组很丰富的、供计算和数据处理的运算符。目前广泛流行的 C++ 是当今极受欢迎的面向对象的程序设计语言之一。今有 AT&T C++、Turbo C++、Borland C++、Microsoft C++、Visual C++(简称 VC)等。

Ada 是由美国国防部大力扶植而发展起来的新语言。它是遵循软件工程的目标和原理而设计出来的,体现了软件工程的诸多特色。Ada 在语言结构上是 Pascal 型的,但它有完善的实时处理特性,包括中断处理、多重任务和进程通信等。它适用于嵌入式实时计算机设计。美国军方强制所有与国防部有关的软件系统均必须以 Ada 作为主体语言,并已载入法律条文。目前 Ada 的最新国际标准是 Ada95,它具有面向对象程序设计的功能,并支持系统编程、实时系统、分布系统、信息系统、数字化、安全与加密以及语言接口等各种类型需求。

专用语言是一类为特殊的应用而设计的计算机语言,通常具有自己特殊的语法形式,面对特定的问题,其输入结构及词汇表与该问题的相应范围密切相关。专用语言种类繁多,主要有 APL、SNOBOL、BLISS、FORTH、LISP、PROLOG、Smalltalk 等。

APL(A Programming Language,一种程序设计语言)是专为数组和向量运算而设计的一种简洁而功能很强的语言。

SNOBOL 是一种符号处理语言,它适用于符号串的处理操作。

BLISS 是一种为开发编译程序和操作系统而设计的计算机语言。

FORTH 是为开发微机处理机软件而设计的计算机语言,这是一种中级语言(所谓低级、中级、高级系指该语言与机器级相接近的程度),其特点是以面向堆栈的方式执行用户定义的函数,因此能提高执行速度和节省存储。

LISP(LISt Processing,表格处理)是一种人工智能领域专用的语言,它广泛应用于人工智能领域的各类问题求解。

PROLOG(PROgramming in LOGic,逻辑程序设计)是一种逻辑推理语言,它特别适合于求解逻辑推理性质的问题,广泛应用于专家系统构造。

Smalltalk 是首先实现真正面向对象的程序设计语言之一,它适合于面向对象的程序设计。

系统实现语言允许程序员直接访问机器操作,因而许可程序员实现机器级的设计。典型的系统实现语言也提供许多高级语言中常用的特征,诸如条件控制结构(如 IF THEN ELSE)和变量类型检查等。系统实现语言的主要应用领域是编译程序和操作系统。主要的系统实现语言是 C。

静态高级语言并不为程序员提供直接控制机器上的操作。它通常提供控制结构和数据类型。静态高级语言的关键是静态存储分配。两个最流行并广泛使用的静态高级语言是 COBOL 和 FORTRAN。