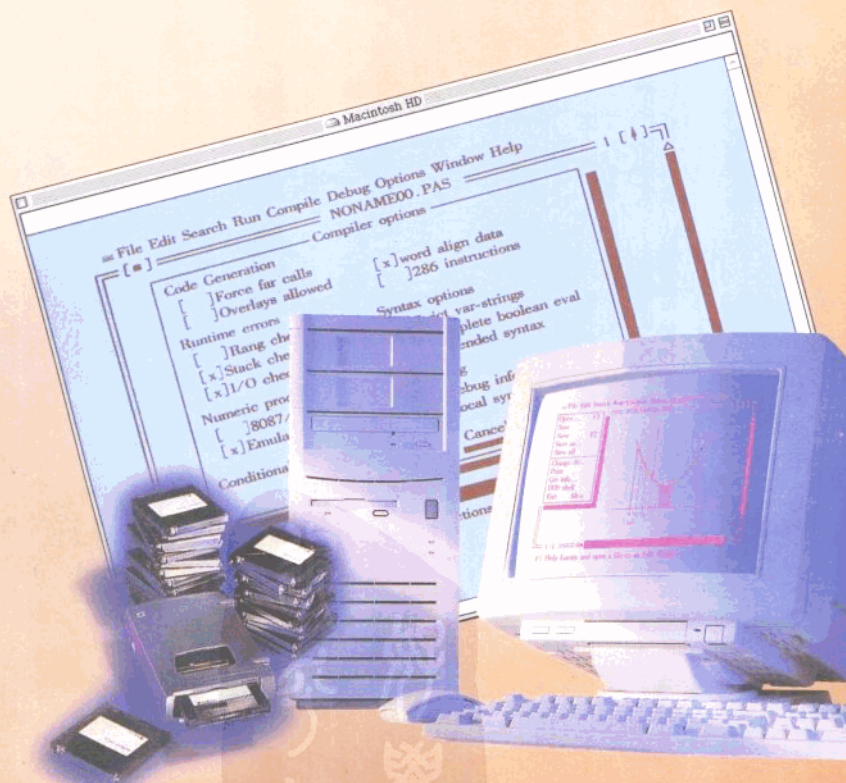


PASCAL程序设计语言(第二版)

题解分析和上机指南

陈福生 陈 锴 编著



知
PDG

同济大学出版社

前 言

对学生进行素质教育,特别是对高精尖人才的培养,程序设计过程的思路、方法和训练是非常重要的。程序设计和算法知识凝聚了很多现代的思维方式和思维观念,学过计算机和没有学过计算机的人显然不一样,仅学会操作使用计算机和能进行深入编制程序的人就更有不同,这种知识结构的差异,使得他们考虑问题的角度和全面性方面表现得大不一样。从这种角度讲,程序设计和上机调试除了有助于了解计算机是如何工作、数据在计算机中如何被处理以及人和计算机对话是怎么进行的以外,更重要的是有助于提高思维能力,开发智力;而对于培养计算机类更高层次的人才,编程就更加必要,因为他们需要对计算机的未来进行开拓,无论在硬件还是在软件方面,要有所作为,首先就要熟练地掌握编制程序。尽管现在有很多工具,但那是别人创造的,我们自己要进行创造,编程训练就是这种创造能力最好的培养方式。

PASCAL 语言长期以来一直以其结构性好、语法规则等优点著称,作为学习编程思想、技术和方法的教学语言再合适不过了,而且比较容易入门。

自从 Borland 公司推出基于 PASCAL 语言,用于开发 WINDOWS 应用程序的 DELPHI 软件后,大大拓宽了 PASCAL 的应用范围,编程人员中学习 PASCAL 语言编程的热情更高。

开发 WINDOWS 应用程序,除了 VISUAL C++ MFC 外,还有另一大类语言工具软件,就是所谓“RAD 应用程序快速开发工具”(Rapid Application Development),目前最流行的有 Microsoft 的 VISUAL BASIC 和 Borland 的 DELPHI。

BASIC 是比尔·盖茨起家的法宝,现在的 VISUAL BASIC 性能已经大大超越了当初的 BASIC。使用 VB 可视化编程确实能简化开发过程,而且还能开发出不错的程序。但是,VB 不足之处在于:解释性的执行方式导致执行速度不够快;“伪”面向对象编程无法编写可重用组件;无法支持系统级编程和调试;无法完全支持 WINDOWS API,等等。对于真正的程序员来说,VB 的功能是远远不够的。但是 DELPHI 就不同了。

DELPHI 完全采用面向对象(OBJECT PASCAL)实现可视化编程。Borland 是最早大力提倡面向对象编程的公司,DELPHI 在这方面十分出色,包括继承、封装、多态和如何建立可视和非可视组件,可重用组件(Component)是 DELPHI 的精华。

PASCAL 的编译器是 Borland 公司的“镇山宝”,TURBO PASCAL 的编译速度令所有 C++ 的程序员惊叹不已。目前 DELPHI 仍然是 WINDOWS 环境下最快的编译器。现在基本上只有 Borland 一家编译器开发商支持 PASCAL 语言,所以 Borland 可以在标准 PASCAL 上尽情发挥,无需考虑兼容性,例如 DELPHI 的类别中 Tobject 直接支持 WINDOWS 消息编程,使得整个类库设计十分简明清晰。DELPHI 的强大功能超出了一般程序员的想象。其实,正是采用了 Object PASCAL 语言,DELPHI 才具有这么强大的功能。尽管 Borland 公司 1998 年更名为 Inprise 公司,但 DELPHI 还是公司的重要产品。事实上,更名后不久就推出了 DELPHI 4.0。所有这些,也许正是广大编程人员为什么会掀起学习 PASCAL 语言编程热潮的原因。

程序设计是一门实践性很强的技术,不能想象“纸上谈兵”就会成为优秀的程序员。程序设计的技巧只有在大量的实践中得以熟练和深化。

本书共分三个部分:第一部分是解题分析,在总结了本章要点的基础上,精选了一些具有典型性、代表性的习题,作了细致的解题分析,融入了程序设计方法学的思想,并介绍了 PASCAL 程序设计风格和程序设计技巧,对各章其他所有习题都提供了参考答案。这一部分旨在对学习 PASCAL 程序设计、正在进行程序设计训练的学生和有关人员能从中得到解题思路的开扩;第二部分介绍了 TURBO PASCAL 6.0 的集成开发环境以及上机操作;第三部分是 PASCAL 程序设计的一些综合应用。

书中的 PASCAL 程序都在 TURBO PASCAL 6.0 中调试通过。编写本书的宗旨是希望对学习 PASCAL 程序设计的读者能从中得到些启迪和帮助。书中的题解并不是唯一的,也不一定是最佳的。相信广大读者能在大量的实践中一定能找到更好的答案。恳切地希望广大读者能对书中错误和疏漏之处批评指正。

在新世纪来临之际本书有幸能与广大读者见面,借此机会,衷心感谢曾无私帮助过我们的许许多多的同济人和其他朋友,并致以新世纪的良好祝愿!

编者

1999 年 7 月

目 录

第一部分 PASCAL 题解分析	(1)
第一章 程序设计基本概念	(1)
第一节 算法和程序设计	(1)
一、算法	(1)
二、程序设计	(1)
三、结构化程序设计	(2)
1. 顺序结构	(2)
2. 循环结构	(2)
3. 选择结构	(3)
四、程序设计的一般步骤	(3)
1. 确定程序的总体结构	(3)
2. 确定程序的数据结构	(3)
3. 设计主程序模块、功能模块、服务模块	(3)
第二节 PASCAL 程序结构	(3)
一、PASCAL 程序基本结构	(3)
二、数据类型	(3)
第三节 题解分析	(4)
习题 1.3	(4)
习题 1.6	(4)
第四节 其他习题参考答案	(5)
习题 1.1	(5)
习题 1.2	(5)
习题 1.4	(5)
习题 1.5	(6)
第二章 数据、表达式、赋值语句及输入输出	(7)
第一节 本章要点	(7)
一、TURBO PASCAL 专用符号表	(7)
二、TURBO PASCAL 常用保留字表	(7)
三、TURBO PASCAL 常用标准标识符	(7)
四、TURBO PASCAL 数据类型分类表	(8)

1. 整型分类表	(8)
2. 实型分类表	(8)
五、TURBO PASCAL 输出语句的标准域宽	(8)
第二节 题解分析	(8)
习题 2.3	(8)
习题 2.4	(9)
习题 2.6	(9)
习题 2.10	(10)
第三节 其他习题参考答案	(11)
习题 2.1	(11)
习题 2.2	(11)
习题 2.5	(11)
习题 2.7	(11)
习题 2.8	(13)
习题 2.9	(13)
 第三章 流程控制语句(I)	 (15)
第一节 本章要点	(15)
一、PASCAL 语句分类	(15)
二、PASCAL 流程控制语句	(15)
1. 选择语句	(15)
2. 循环语句	(15)
3. 转向语句	(15)
第二节 题解分析	(15)
习题 3.2	(15)
习题 3.3	(18)
习题 3.5	(19)
习题 3.7	(20)
习题 3.9	(22)
习题 3.10	(24)
第三节 其他习题参考答案	(25)
习题 3.1	(25)
习题 3.4	(26)
习题 3.6	(27)
习题 3.8	(28)

第四章 流程控制语句(Ⅱ)	(29)
第一节 程序设计风格	(29)
一、程序设计的风格	(29)
二、程序正文表示的风格	(29)
三、语言的应用风格	(30)
第二节 题解分析	(30)
习题 4.2	(30)
习题 4.4	(32)
习题 4.5	(33)
习题 4.8	(34)
习题 4.9	(35)
习题 4.10	(36)
第三节 其他习题参考答案	(38)
习题 4.1	(38)
习题 4.3	(39)
习题 4.6	(39)
习题 4.7	(39)
习题 4.11	(40)
第五章 函数和过程	(42)
第一节 本章要点	(42)
一、过程与函数的区别	(42)
二、变量参数与值参数的区别	(42)
三、标识符的作用域	(42)
1. 全程量和它的作用域	(42)
2. 局部量和它的作用域	(42)
第二节 题解分析	(42)
习题 5.1	(42)
习题 5.4	(44)
习题 5.5	(45)
习题 5.7	(47)
习题 5.10	(50)
第三节 其他习题参考答案	(52)
习题 5.2	(52)
习题 5.3	(53)
习题 5.6	(54)

习题 5.8	(54)
习题 5.9	(55)
第六章 枚举类型、子界类型、数组类型	(57)
第一节 本章要点	(57)
一、数组的特点	(57)
二、字符串类型	(57)
三、动态数组	(57)
第二节 题解分析	(57)
习题 6.2	(57)
习题 6.4	(59)
习题 6.7	(61)
习题 6.8	(64)
习题 6.9	(66)
第三节 其他习题参考答案	(69)
习题 6.1	(69)
习题 6.3	(71)
习题 6.5	(72)
习题 6.6	(73)
习题 6.10	(74)
第七章 集合和记录	(76)
第一节 本章要点和程序设计技巧	(76)
一、记录的特点	(76)
二、程序设计技巧	(76)
1. 数组的使用	(76)
2. 循环和递归的优化	(76)
3. 布尔变量的使用	(77)
第二节 题解分析	(77)
习题 7.2	(77)
习题 7.4	(78)
习题 7.5	(82)
习题 7.7	(84)
第三节 其他习题参考答案	(88)
习题 7.1	(88)
习题 7.3	(89)
习题 7.6	(90)

习题 7.8	(91)
习题 7.9	(92)
第八章 文件	(95)
第一节 本章要点	(95)
一、数据传送到磁盘文件的写操作步骤	(95)
二、由磁盘文件读出数据的读操作步骤	(95)
三、PASCAL 语言允许略写的语句或函数的默认含义	(95)
第二节 题解分析	(95)
习题 8.2	(95)
习题 8.4	(96)
习题 8.5	(98)
习题 8.6	(101)
习题 8.7	(102)
习题 8.11	(105)
第三节 其他习题参考答案	(107)
习题 8.1	(107)
习题 8.3	(108)
习题 8.8	(108)
习题 8.9	(109)
习题 8.10	(110)
第九章 指针—动态数据结构	(112)
第一节 本章要点	(112)
一、静态变量和动态变量	(112)
1. 静态变量	(112)
2. 动态变量	(112)
二、指针变量	(112)
第二节 题解分析	(112)
习题 9.2	(112)
习题 9.4	(114)
习题 9.5	(115)
习题 9.8	(117)
第三节 其他习题参考答案	(121)
习题 9.1	(121)
习题 9.3	(122)
习题 9.6	(122)

习题 9.7	(126)
习题 9.9	(129)
习题 9.10	(131)
 第二部分 TURBO PASCAL 上机操作使用指南	 (133)
一、TURBO PASCAL 6.0 系统的安装	(133)
二、TURBO PASCAL 6.0 集成开发环境(IDE)	(134)
三、TURBO PASCAL 6.0 集成环境的使用	(149)
四、TURBO PASCAL 程序调试	(151)
五、集成环境菜单功能和编辑命令一览表	(158)
六、TURBO PASCAL 错误信息表	(162)
 第三部分 PASCAL 综合应用	 (168)
一、打印万年历	(168)
1. 方法简介	(168)
2. 程序说明	(168)
3. 万年历完整程序	(168)
二、约瑟夫问题	(171)
1. 用数组方法求解	(171)
2. 用链表结构求解	(172)
三、课程表安排问题—集合应用	(174)
1. 方法简介	(174)
2. 程序变量说明	(175)
3. 完整程序	(175)
四、全主元高斯消去法解线性方程组	(177)
1. 方法简介	(177)
2. 子程序说明	(177)
3. 程序应用示例	(178)
五、平方根法解线性方程组	(180)
1. 方法简介	(180)
2. 子程序说明	(181)
3. 程序应用示例	(181)
六、改进平方根法	(183)
1. 方法简介	(183)
2. 子程序说明	(183)
3. 程序应用示例	(184)
七、对称矩阵的平方根法求逆	(186)

1. 方法简介	(186)
2. 子程序说明	(186)
3. 程序应用示例	(186)
八、行主元消去法求逆矩阵	(188)
1. 方法简介	(188)
2. 子程序说明	(188)
3. 程序应用示例	(189)
九、幂方法计算实矩阵绝对值最大特征值和特征向量	(191)
1. 方法简介	(192)
2. 子程序说明	(192)
3. 程序应用示例	(192)
十、雅可比(jacobi)法求实矩阵的特征值和特征向量	(194)
1. 方法简介	(194)
2. 子程序说明	(194)
3. 程序应用示例	(195)
参考文献	(199)

第一部分 PASCAL 题解分析

第一章 程序设计基本概念

第一节 算法和程序设计

一、算法

算法(algorithm),简单地讲就是解决特定问题的方法。特定的问题可分为数值和非数值两大类。解决数值问题的算法就叫做数值算法。科学和工程计算方面的算法一般都属于数值算法,如求解数值积分,求解线性方程组,求解微分方程等。解决非数值问题的算法叫做非数值算法,数据处理方面的算法都属于非数值算法。如各种排序算法、查找算法、插入算法、删除算法、遍历算法等。数值算法和非数值算法通常并没有严格的区别,一般说来,在数值算法中主要进行算术运算,而在非数值算法中则主要进行比较和逻辑运算。另一方面,特定的问题可能是递归的,也可能是非递归的,因而解决它们的算法就有递归算法和非递归算法之分。

对于解决同一问题,往往能找到多种不同的算法。为了能从解决同一问题的多种算法中选择合适的一种,或者对已有的算法进行一些改进,因此人们通常从以下几个方面来对某种算法进行评价:

- 正确性。一个正确的算法是指在合理的数据输入下,能在有限的运算时间内得出正确的结果。
- 运行时间。是指一个算法在计算机上运算所花费的时间要少。
- 占用的存储空间。一个算法在计算机存储器上需要占用的存储空间,包括存储算法本身所占用的存储空间要少。
- 简单性。算法的简单性使得证明其正确性比较容易,同时便于编写、修改、阅读和调试程序。但最简单和最直接的算法往往不是最有效的,即在运行时间和占用存储空间不是最少的算法。

二、程序设计

程序可定义为由人编写的指挥和控制计算机完成某一任务的指令序列,这个指令序列是对要解决问题的各个对象和处理规则的描述。程序与算法的区别在于:算法是解题方法的精确描述,它与“程序”还有一定差距。算法体现了一种思想方法,而不是实现方法。它可以独立于具体的程序或语言而存在,例如有了对某个问题的算法,可以用 PASCAL 语言来实现,也可以用 C, FORTRAN 等语言来实现。

程序设计(programming)是设计、编制和调试程序的过程,而把与计算机交往,用于程序设计的语言称为程序设计语言。程序设计语言的基本功能就是描述数据和对数据的处理的过程。

程序设计是一个复杂的过程,除了要掌握一些专门的程序设计技术外,还要了解求解问

题的许多细节。在程序设计过程中,程序员使用了包括推测、直觉、技巧、灵感和某些程序设计原则等。

在开始阶段,我们一般不可能就求解问题的全部细节,而是主要集中在对问题的求解方案的全局作出决策,得到第一层方案,在这一层中设计出对问题本身较为自然的,大都用自然语言来表达的抽象算法。之所以是抽象算法,是因为算法的细节还不清楚,它只是对求解问题的一般策略和解的一般结构,还不能为计算机所理解。紧接着我们就可对抽象算法中包含的抽象自然语句作进一步细化(即求精),进入更深一层次的抽象。求精的每一步都是对上一层中包含的抽象语句和数据进行进一步的分解和精确,如此一层层的求精,一直到能用计算机理解的语言表达为止。人们称这样的程序设计方法为“自顶向下”的程序设计。该方法的主导思想是:程序设计是从最能直接反映问题的体系结构的概念出发,逐步精确化、具体化,逐步展开细节,最终设计出能在计算机上执行的程序。

随着计算机硬件价格越来越便宜,软件费用急剧上升,人们对程序设计的要求发生了变化,过去主要强调程序的正确性和效率,这对小型程序来说无疑是正确的。但对大型程序,人们则倾向于首先强调程序的可维护性、可靠性和可理解性,然后才是效率。

著名的计算机专家戴克斯特拉曾指出:“程序员的任务不仅在于编制一个正确的程序,而且还应该以一种可信赖的方式证明程序的正确性”。程序调试的目的是发现程序中的错误并及时予以纠正。发现错误的方法主要有两种,即理论法和实验法。理论法是属于程序正确性证明问题,它是利用数学方法证明程序的正确性。程序正确性的证明仍处于研究之中;实验法是目前普遍使用的程序调试方法,而且卓有成效。所有程序必须经过调试才能证明其正确性,事实上只能说基本正确,要证明完全正确,还需经过一段试用。

三、结构化程序设计

编写程序应符合软件工程化的思想,软件工程的基本思想是利用工程原理,建立软件环境来提高软件开发效率。软件工程方法之一是结构化程序设计方法。这种方法指导人们用良好的思想方法去设计程序,特点是对任何程序都设计组成以下三种基本结构:顺序结构、循环结构和选择结构。



图 1-1 顺序结构

1. 顺序结构

顺序结构是一种线性、有序的结构,即依次序执行各语句或模块。

如图 1-1 所示。

2. 循环结构

循环结构是重复执行一个或几个语句或模块,直到满足某一条件为止。如图 1-2 所示。

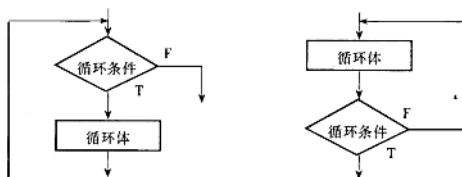


图 1-2 循环结构

3. 选择结构

选择结构是根据条件成立与否来选择程序执行的通路。如图 1-3 所示。

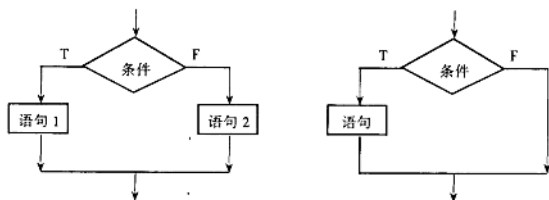


图 1-3 选择结构

四、程序设计的一般步骤

对于大型程序的设计,通常有以下步骤:

1. 确定程序的总体结构

一般程序的总体结构可以分为三层,最上一层是主程序模块,用来显示程序的功能清单,并根据用户的选择转入相应的功能模块。中间一层是功能模块,一个功能模块完成一种功能,它是程序的核心部分。最下一层是服务模块,几个功能模块相同的处理部分可以抽出来写成过程,供功能模块调用。

2. 确定程序的数据结构

在满足程序功能和设计要求的前提下,确定程序的常量定义部分、类型定义部分和变量说明部分,要尽量少读写文件,因为读写文件一般速度较慢。

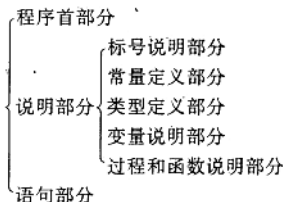
3. 设计主程序模块、功能模块、服务模块

在设计每一个程序模块时,都应将它分成若干个子问题,对于每一个子问题,如果它还不能用一个 PASCAL 语句来表达,就把该子问题再细分为若干个低一层的子问题,这样一直分下去,逐步细化到最低层的所有子问题都能用 PASCAL 语句表达为止。

第二节 PASCAL 程序结构

一、PASCAL 程序基本结构

一个完整的 PASCAL 程序可包括以下几个部分:

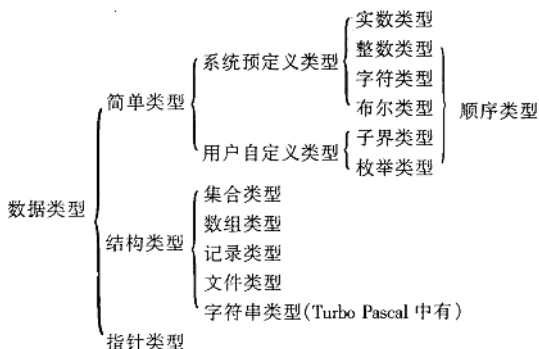


二、数据类型

数据是计算机处理的对象,程序所描述的就是数据以及对这些数据的处理步骤。数据

的性质是通过数据类型来反映,高级程序设计语言中的每一个数据都必须属于一种数据类型。数据类型从本质上定义了该类型数据的取值范围和可施加于它们的全部运算。

PASCAL 语言根据数据类型的特点,可将其分为简单类型、结构类型和指针类型三类:



第三节 题解分析

习题 1.3 PASCAL 语言的程序结构由哪几部分组成? 试述各部分的构成形式和作用。

【解】 一个 PASCAL 程序是由程序首部分、说明部分和语句部分组成。有时把说明部分和语句部分合称为程序体或程序块。整个程序以实心圆点结束。程序首部分包括三项内容,按照书写顺序,它们是:

(1) PROGRAM——PASCAL 程序标志,写在程序的开头。

(2) 程序名——由程序设计者依据“标识符”规则自行定义,起标识一个 PASCAL 程序的作用。

(3) 参数表——即本程序所用到的一些文件列表,写在程序名后面的括号中,参数之间用逗号分开。

程序体是 PASCAL 程序的主体,也称“分程序”,由说明部分和语句部分组成。

(1) 说明部分——用来描述程序中所用到的数据的属性。说明部分的顺序是:标号说明、常量定义部分、类型定义部分、变量说明部分、过程与函数说明部分。有时说明部分的内容可以全部或部分地缺省,但书写顺序不得改变。

(2) 语句部分——是程序的执行部分,用来描述程序所执行的操作。语句部分是一个复合语句,BEGIN 和 END 表示复合语句的开始和结束,其中包含一个或多个语句,语句之间用分号“;”隔开。

习题 1.6 PASCAL 语言中标识符是怎样构成的? 下列符号中哪些是正确的 PASCAL 标识符? 哪些是错误的? 为什么?

A5B, 5H4, PEL.1, A31, BEGIN, ABS, x * y

A + B, \$ 500, G1(x), CL, VAR, xzy, E - 10

【解】标识符是以字母开头的字母与数字的序列。正确的标识符有: A5B, A31, Cl, xzy。
错误的标识符有:

5H4——标识符应是字母开头的字母与数字的序列,这里是用数字开头,所以是错误的。

PEL.1——标识符中不能有小数点符号。

BEGIN——标识符不能与保留字 BEGIN 同名。

ABS——标识符应尽量避免与标准标识符(如函数名)同名,以免发生混淆。

$x * y$ ——标识符中不能有运算符“ $*$ ”。

$A + B$ ——标识符中不能有运算符“ $+$ ”。

\$ 500——标识符应是字母开头的字母与数字的序列,这里不是用字母开头,所以是错误的。

GI(x)——标识符中不能有括号,这里是函数调用。

VAR——标识符不能与保留字 VAR 同名。

E - 10——标识符中不能有运算符“-”。

第四节 其他习题参考答案

习题 1.1 计算机系统由哪几部分组成? 计算机硬件系统包括哪几部分? 各部分的作用是什么?

【参考答案】

一个计算机系统由系统硬件和系统软件组成。计算机硬件系统包括:中央处理机,主存储器 and 外围设备。

中央处理机由运算器和控制器组成,它发出控制指令并按一定的顺序执行各种操作,如算术运算和逻辑运算。

主存储器是用来存放系统的常驻信息和当前被中央处理机所处理的信息和数据的。

外围设备包括输入设备和输出设备,输入设备是用来向计算机输入数据信息的,输出设备是用来由计算机输出结果的。

习题 1.2 什么叫机器语言、高级程序设计语言? 各有什么特点?

【参考答案】

机器语言就是计算机的指令系统。高级程序设计语言是独立于机器,一般较接近数学语言和自然语言的语言。

机器语言的特点是,对于任一特定的计算机来说,它只能识别特定的机器语言,机器语言程序就是按数据处理的要求用机器语言编写的指令串,机器指令繁琐难懂,不易推广使用。高级程序设计语言的特点是,接近自然语言,语法规则简单、清晰,易于各类专业人员掌握使用、编写程序(称为源程序)。但计算机是不“懂得”这种高级语言的,必须用一个编译程序将源程序编译成机器指令组成的程序(称为目标程序)才能在特定的计算机系统中执行。

习题 1.4 PASCAL 基本字符集由哪些成份构成? PASCAL 保留字是什么? 举例说明保留字的作用。

【参考答案】

PASCAL 字符集包括英文字母、数字及一些符号,实际上是 ASCII 码字符集的一个子集。字可以分为两类:保留字和标识符。PASCAL 中的保留字对编译程序具有特殊的意义,是用来命名 PASCAL 语句、某些预定义的数据类型、某些操作符、说明段及程序首部的。例如保留字 CONST 就是用来表示开始一个常数说明段,保留字 VAR 表示开始一个变量说明段。

习题 1.5 PASCAL 语言中有哪些标准类型、标准文件、标准函数、标准过程以及标准常数。

【参考答案】

标准类型有:integer,boolean,real,char,text

标准文件有:input,output

标准函数有:abs,arctan,char,cos,eof,eoln,exp,ln,odd,ord,pred,round,sin,sqr,sqrt,succ, trunc

标准过程有:dispose,get,new,pack,page,put,read,readln,reset,rewrite,unpack,write,writeln

标准常量有:false,true,maxint

第二章 数据、表达式、赋值语句及输入输出

第一节 本章要点

一、TURBO PASCAL 专用符号表

符号	含 义	符号	含 义
+	加法运算符,并(集合)	,	参数、变量分隔符
-	减法运算符(负号),差(集合)	:	变量名与类型名分隔符
*	乘法运算符,交(集合)	'	字符常量或字符串常量限定符
/	除法运算符	..	子界说明中上界与下界分隔符
>	大于	^	指针变量指示符
<	小于	(	参数或嵌套表达式开始标志符
>=	大于等于,包含(集合)	)	参数或嵌套表达式结束标志符
<=	小于等于,包含于(集合)	[	下标或集合表达式开始标志符
=	等于	]	下标或集合表达式结束标志符
<>	不等于	或(*)	注释行开始标志符
:=	赋值号	或*)	注释行结束标志符
;	语句分隔符	.	记录域分隔符,程序结束符,小数点

二、TURBO PASCAL 常用保留字表

AND	ARRAY	BEGIN	CASE	CONST
DIV	DO	DOWNT0	ELSE	END
EXTERNAL	FILE	FORWARD	FOR	FUNCTION
GOTO	IF	IN	LABEL	MOD
NIL	NOT	OF	OR	PACKED
PROCEDURE	PROGRAM	RECORD	REPEAT	SET
STRING	THEN	TYPE	TO	UNTIL
VAR	WHILE	WITH	XOR	

三、TURBO PASCAL 常用标准标识符

标准常量	false	true	maxint						
标准类型	integer	boolean	real	char	text				
标准文件	input	output							
标准函数	abs	arctan	char	cos	eof	coln	exp	ln	odd
	ord	pred	random	round	sin	sqr	sqrt	succ	trunc
标准过程	assign	append	dispose	close	exit	newread	readln	reset	
	rewrite	write	writeln						