

张 铭  
黄维通 编著

# Pascal

到

# Delphi 4.0

计算引论与结构化程序设计

上



北京大学计算机科学技术系教材

# 从标准 Pascal 到 Delphi 4.0(上册)

——计算引论与结构化程序设计

张 铭 黄维通 编著

北 京 大 学 出 版 社  
北 京

TP312

## 内 容 简 介

本套书由上、下两册组成。上册主要介绍计算机科学的基本知识和标准 Pascal 的结构化程序设计方法；下册主要介绍面向对象的程序设计方法。作为上册的本书内容全面，算法清晰，叙述由浅入深，系统性强，并配有丰富而深刻的例题和习题，实为计算机和软件专业学生“计算引论”课的理想教材。

## 图书在版编目(CIP)数据

从标准 Pascal 到 Delphi 4.0(上册)/张铭,黄维通编著. —北京:北京大学出版社,1999.01  
ISBN 7-301-04005-9

I. 从… II. ①张… ②黄… III. 结构化语言-程序设计 IV. TP312

书 名: 从标准 Pascal 到 Delphi 4.0(上册)

著作责任者: 张 铭 黄维通

责 任 编 辑: 沈承凤

标 准 书 号: ISBN 7-301-04005-9/P·0436

出 版 者: 北京大学出版社

地 址: 北京市海淀区中关村北京大学校内, 100871

网 址: <http://cbs.pku.edu.cn/cbs.htm>

电 话: 出版部 62752015 发行部 62754140 编辑部 62752037

电 子 信 息: [zpup@pup.pku.edu.cn](mailto:zpup@pup.pku.edu.cn)

排 版 者: 兴盛达激光照排中心

印 刷 者: 北京飞达印刷厂印刷

发 行 者: 北京大学出版社

经 销 者: 新华书店

787毫米×1092毫米 16开本 23.25印张 579千字

1999年1月第一版 1999年1月第一次印刷

定 价: 28.00元

## 前 言

在 21 世纪即将来临之际,人类已经跨入了知识经济的信息时代。这个时代的标志就是计算机与信息处理技术的广泛应用。正因为如此,掌握和运用计算机成为人们的共识。计算机基本知识和计算机编程的基本技巧成为许多人的必修课。计算机专业的学生更是需要合适的导引课程“计算引论”。课程内容应该包括计算机科学的简单基础知识,使用计算机系统的基本方法,程序设计的基本方法和技巧。这些基础知识对于其他后继课程的学习是十分必要的。

根据“计算引论”课程教学需要,我们编写了一套上、下两册的教材——《从标准 Pascal 到 Delphi》。上册主要介绍计算机科学的基本知识和标准 Pascal 结构化程序设计方法。下册则介绍 Windows 操作系统下的 Delphi 4.0 可视化编程环境,以及 Object Pascal 的面向对象程序设计方法。这套教材作为计算机软件专业一年级学生两个学期的导引课程非常合适。

每一个计算机工作者都必须有程序设计能力,这是一个基本功。计算机学科的特色大都寓于程序设计符号之中,程序设计语言是涉足计算机科学领域的基本工具。当前程序员应用较多的是面向对象程序设计语言和面向过程的算法语言。面向对象程序设计中,底层的对象操作方法,仍然是由算法语言中的顺序结构、条件选择结构、循环结构来实现的。面向对象方法并不排斥结构化程序设计方法。因此,我们认为计算机专业学生仍然有必要首先进行严格的结构化程序设计训练,然后再进行面向对象程序设计的训练。

Pascal 语言作为结构化程序设计的首选语言,以其清晰的结构,丰富的类型,编译、运行的高效性,以及方便的可移植性,被广泛地用于各类教学环境和软件开发中。Pascal 语言清晰、自然地表达了程序设计的基本概念,适于讲授程序设计课程。由于 Pascal 语言具有良好的结构化特点,算法简明、清晰,便于接受,程序的书写格式自由,程序格式优美、便于阅读,因此 Pascal 语言很适宜于初学者学习,能系统地给学生以严格而良好的程序设计概念的基本训练。为此,作为这套书的上册,本书将以标准 Pascal 语言为媒介,讲授结构化程序设计的基本原理与方法。

本书介绍了 Pascal 语言的数据说明、数据运算特点、控制流程语句,引入了新运算的函数和过程机制,并通过大量的编程实例讲授了顺序、条件、循环三种基本结构的程序设计方法,涉及到了蒙特卡洛法(模拟法)、穷举法、分治法、递归法、回溯法等基本算法和一些典型的排序算法,并引入了栈、队列、二维数组、图、链表等基本数据结构。

由于 Turbo Pascal 不支持标准 Pascal 的可变数组(即保形数组),Turbo Pascal 和标准 Pascal 对于函数和过程作为参数的规定差别较大,而这两部分的内容对于初学者来说难度较大,本书不予介绍,感兴趣的读者可以阅读相关书籍。为了便于普通读者尽快适应具体的操作环境,又能帮助需要应付各种考试的读者对于标准的把握,我们在程序设计的各个章节中详细地比较了 Turbo Pascal 与标准 Pascal 的差别。由于 Turbo Pascal 与标准 Pascal 在文件方面的差别较大,我们特意开辟了一节重点讲解 Turbo Pascal 的文件特点。

对于大部分例题,我们都先从分析入手,首先找到解决问题的算法思想,然后设计该算法所涉及的数据结构,并用伪算法语言粗略地描述出主要算法框架,最后给出具体的 Pascal 程

序。我们强调自顶向下、逐步求精、模块化的结构化程序方法。在许多例题中都有关于算法的讨论,以开阔学生的思路,使得学生更注意从算法的角度来考虑程序设计问题的不同实现方法。每一章的习题能够帮助学生掌握本课程的重点内容并灵活应用,而其中尤以上机环节更为重要。学习计算机语言的钥匙就是上机操作。

Turbo Pascal 语言是 Borland(现已改为 Inoprise)公司 1983 年推出的,以其独特的优点在短短的几年内赢得了广大用户的青睐。它可以在 IBM PC 机及其兼容机上快速运行,是较为实用的 Pascal 编译系统。它在标准 Pascal 语言功能的基础上,又扩充了一些过程和函数,实现了一些屏幕、窗口、图形、声音的控制及绝对地址存取,并且能通过直接调用 DOS 软件中断以实现控制。

在 Turbo Pascal 的基础上,Borland 公司又推出了 Delphi 可视化编程环境,以其快捷方便、友好的界面和最优化的编译执行代码而成为热门的编程语言和运行环境。Delphi 的出现使 Pascal 获得了第二次生命,也使软件开发人员看到了摆脱繁重工作的希望。

本套书的下册重点介绍 Delphi 4.0 可视化编程环境和面向对象程序设计方法,主要引导读者理解面向对象程序开发的概念、Object Pascal 程序设计、各种对象的使用、Delphi 4.0 在网络上的应用以及在数据库系统上的应用等。同时我们还将把在学习和应用 Delphi 过程中编写的一些样本程序融入书中。

应该感谢北京大学出版社的沈承凤老师,本套书可以说是在她的热心呵护下问世的。北京大学计算机科学技术系汪琼博士在使用本书清样给北大计算机科学技术系软件专业本科一年级学生授课过程中,为本书提供了不少建议性意见。作者在写书的过程中与北大计算机科学技术系的龙晓苑老师、丁文魁副教授、孙家骥副教授、刘从义副教授和张宁副教授进行了广泛而深入的讨论。作者对他们深表感谢。另外,感谢计算机科学技术 96、97 和 98 级的学生,他们对书中的例题和习题提出了许多宝贵意见。

当然,书中难免有许多不足之处,欢迎读者提出宝贵意见和建议。作者的 E-mail 地址是:

zhm@ibmrs1.pku.edu.cn

hwt@mail.cic.tsinghua.edu.cn

张 铭 黄维通

1998 年 9 月于北京大学、清华大学

# 目 录

|                      |      |
|----------------------|------|
| <b>第一章 信息时代的计算学科</b> | (1)  |
| 1.1 信息时代             | (1)  |
| 1.2 信息与信息科学          | (2)  |
| 1.3 计算学科             | (2)  |
| 1.4 本课程的目的           | (5)  |
| 1.5 学习方法             | (5)  |
| 1.6 小结               | (7)  |
| 1.7 习题               | (7)  |
| <b>第二章 计算机系统发展史</b>  | (8)  |
| 2.1 人类计算工具与方法的发展初始阶段 | (8)  |
| 2.2 完整的近代计算机器        | (9)  |
| 2.3 现代计算机的萌芽         | (10) |
| 2.4 现代计算机的发展         | (12) |
| 2.5 计算机系统的组成         | (15) |
| 2.6 计算机的主要应用领域       | (16) |
| 2.7 计算学科的重要研究方向      | (17) |
| 2.8 小结               | (18) |
| 2.9 习题               | (18) |
| <b>第三章 计算机硬件系统</b>   | (19) |
| 3.1 输入/出设备           | (19) |
| 3.2 存储器              | (20) |
| 3.3 中央处理器 CPU        | (24) |
| 3.4 计算机系统的总线结构       | (26) |
| 3.5 计算机的分类           | (27) |
| 3.6 计算机的主要特性及指标      | (29) |
| 3.7 微机的一般硬件配置        | (31) |
| 3.8 小结               | (33) |
| 3.9 习题               | (33) |
| <b>第四章 计算机软件系统</b>   | (34) |
| 4.1 软件的分类和层次关系       | (34) |
| 4.2 操作系统             | (35) |
| 4.3 计算机软件工程简介        | (41) |
| 4.4 小结               | (44) |
| 4.5 习题               | (44) |

|                                    |              |
|------------------------------------|--------------|
| <b>第五章 计算机语言和程序设计方法 .....</b>      | <b>(45)</b>  |
| 5.1 语言的基本成分.....                   | (45)         |
| 5.2 语言的分类.....                     | (45)         |
| 5.3 其他的特色语言.....                   | (47)         |
| 5.4 几种常见的语言.....                   | (47)         |
| 5.5 Pascal 语言产生的背景及其特点.....        | (50)         |
| 5.6 算法和程序设计步骤.....                 | (51)         |
| 5.7 语法描述.....                      | (56)         |
| 5.8 小结.....                        | (58)         |
| 5.9 习题.....                        | (58)         |
| <b>第六章 Pascal 语言的词汇和语法构架 .....</b> | <b>(60)</b>  |
| 6.1 Pascal 的词汇集.....               | (60)         |
| 6.2 Pascal 的程序构架.....              | (62)         |
| 6.4 小结.....                        | (65)         |
| 6.5 习题 .....                       | (66)         |
| <b>第七章 顺序结构程序设计 .....</b>          | <b>(67)</b>  |
| 7.1 数据类型.....                      | (67)         |
| 7.2 常量和变量.....                     | (73)         |
| 7.3 表达式.....                       | (75)         |
| 7.4 赋值语句.....                      | (80)         |
| 7.5 类型的一致性.....                    | (80)         |
| 7.6 输入输出语句.....                    | (81)         |
| 7.7 复合语句和空语句.....                  | (84)         |
| 7.8 顺序程序设计举例.....                  | (85)         |
| 7.9 小结.....                        | (86)         |
| 7.10 习题 .....                      | (86)         |
| <b>第八章 条件语句与选择结构程序设计 .....</b>     | <b>(89)</b>  |
| 8.1 IF 语句 .....                    | (89)         |
| 8.2 CASE 语句 .....                  | (96)         |
| 8.3 小结 .....                       | (100)        |
| 8.4 习题 .....                       | (100)        |
| <b>第九章 循环结构程序设计.....</b>           | <b>(103)</b> |
| 9.1 WHILE 语句 .....                 | (103)        |
| 9.2 REPEAT 语句.....                 | (107)        |
| 9.3 FOR 语句 .....                   | (110)        |
| 9.4 多重循环 .....                     | (115)        |
| 9.5 循环语句的比较和注意事项 .....             | (119)        |
| 9.6 用 GOTO 语句实现循环.....             | (122)        |
| 9.7 小结 .....                       | (125)        |

|             |                         |              |
|-------------|-------------------------|--------------|
| 9.8         | 习题 .....                | (125)        |
| <b>第十章</b>  | <b>函数与过程程序设计 .....</b>  | <b>(132)</b> |
| 10.1        | 自顶向下的分析方法 .....         | (132)        |
| 10.2        | 子程序 .....               | (133)        |
| 10.3        | 函数 .....                | (133)        |
| 10.4        | 过程 .....                | (141)        |
| 10.5        | 函数与过程的比较 .....          | (146)        |
| 10.6        | 值参数与变量参数 .....          | (147)        |
| 10.7        | 函数与过程的递归 .....          | (151)        |
| 10.8        | 层次结构和标识符的作用域 .....      | (161)        |
| 10.9        | 函数与过程的副作用 .....         | (168)        |
| 10.10       | 小结 .....                | (169)        |
| 10.11       | 习题 .....                | (169)        |
| <b>第十一章</b> | <b>枚举与子界 .....</b>      | <b>(175)</b> |
| 11.1        | 类型定义的一般形式 .....         | (175)        |
| 11.2        | 枚举类型 .....              | (176)        |
| 11.3        | 子界类型 .....              | (181)        |
| 11.4        | 小结 .....                | (184)        |
| 11.5        | 习题 .....                | (184)        |
| <b>第十二章</b> | <b>数组 .....</b>         | <b>(188)</b> |
| 12.1        | 数据的抽象 .....             | (188)        |
| 12.2        | 一维数组的定义及意义 .....        | (188)        |
| 12.3        | 一维数组的应用 .....           | (193)        |
| 12.4        | 字符数组和字符串 .....          | (205)        |
| 12.5        | 多维数组 .....              | (210)        |
| 12.6        | 数组的重要数据结构 —— 栈和队列 ..... | (221)        |
| 12.7        | 数组的注意事项 .....           | (229)        |
| 12.8        | 小结 .....                | (230)        |
| 12.9        | 习题 .....                | (231)        |
| <b>第十三章</b> | <b>集合类型 .....</b>       | <b>(240)</b> |
| 13.1        | 集合的概念 .....             | (240)        |
| 13.2        | 集合类型的定义和运算 .....        | (241)        |
| 13.3        | 集合类型的应用 .....           | (242)        |
| 13.4        | 注意事项 .....              | (250)        |
| 13.5        | 类型间的关系 .....            | (250)        |
| 13.6        | 小结 .....                | (254)        |
| 13.7        | 习题 .....                | (254)        |
| <b>第十四章</b> | <b>记录类型 .....</b>       | <b>(257)</b> |
| 14.1        | 记录类型的定义和引用 .....        | (257)        |

|             |                                  |              |
|-------------|----------------------------------|--------------|
| 14.2        | 开域语句.....                        | (262)        |
| 14.3        | 变体记录.....                        | (266)        |
| 14.4        | 记录的应用.....                       | (269)        |
| 14.5        | 记录的注意事项.....                     | (271)        |
| 14.6        | 小结.....                          | (273)        |
| 14.7        | 习题.....                          | (274)        |
| <b>第十五章</b> | <b>文件类型.....</b>                 | <b>(280)</b> |
| 15.1        | 标准 Pascal 的 FILE 类型文件.....       | (281)        |
| 15.2        | 标准 Pascal 的正文文件(Text 类型文件) ..... | (291)        |
| 15.3        | 标准 Pascal 文件的内部文件和外部文件 .....     | (297)        |
| 15.4        | FILE 类型文件和正文文件的比较 .....          | (298)        |
| 15.5        | 标准 Pascal 文件的综合应用 .....          | (298)        |
| 15.6        | 标准 Pascal 文件小结 .....             | (304)        |
| 15.7        | Turbo Pascal 文件的特点.....          | (306)        |
| 15.8        | 习题.....                          | (315)        |
| <b>第十六章</b> | <b>指针与动态数据结构.....</b>            | <b>(317)</b> |
| 16.1        | 动态变量与动态数据结构 .....                | (317)        |
| 16.2        | 指针类型.....                        | (318)        |
| 16.3        | 动态变量及链表.....                     | (323)        |
| 16.4        | 栈与队列.....                        | (333)        |
| 16.5        | 二叉树.....                         | (335)        |
| 16.6        | 小结.....                          | (345)        |
| 16.7        | 习题.....                          | (347)        |
| <b>附录 A</b> | <b>Pascal 的字汇表.....</b>          | <b>(351)</b> |
| A.1         | 保留字 .....                        | (351)        |
| A.2         | 标准标识符 .....                      | (351)        |
| A.3         | 标点符号 .....                       | (352)        |
| <b>附录 B</b> | <b>Pascal 语法.....</b>            | <b>(353)</b> |
| B.1         | 语法图 .....                        | (353)        |
| B.2         | 巴科斯·瑞尔范式(BNF) .....              | (357)        |
| <b>附录 C</b> | <b>ASCII 码.....</b>              | <b>(362)</b> |

# 第一章 信息时代的计算学科

## 1.1 信息时代

信息技术革命正以迅猛之势改变着我们的社会。人类社会以其持续增长的加速度,由既往奔向未来。百万年蒙昧,数万年游牧,几千年农耕,几百年工商,如今正经历着一场前所未有的巨变,由工业时代迈向信息时代。微电子、通信、计算机是信息时代的三大技术支柱。这个时代的标志就是计算机与信息处理技术的广泛应用。以信息产业为龙头的知识经济(或称“信息经济”,“高技术经济”)则是信息时代的主流。

资源、工具、生产力、时代的连锁反应,是时代变迁的奥秘。

物资、能源、信息是人类可以利用的三项战略资源。人类利用物资以加工成材料;利用能量以转换成动力;利用信息以提炼成知识和智慧。古人类只能利用物资资源,使用锄头等人力工具,停留在农业社会;近代人类能够利用能源,发明了蒸汽机等动力工具,进入工业社会;现代人类充分利用信息资源,运用计算机等智能工具,从而跨入了现代信息社会。

从1946年2月5日第一台电子计算机ENIAC的诞生之日起,现代计算机就以其快速、通用、高精确性的算术运算,极强的逻辑处理能力和大存储量而迅速成为人类信息处理的主要工具,广泛地应用于科学计算和数据处理等领域,如核物理聚变模拟计算、天文计算、气象数据处理等。

80年代微型计算机的出现和发展,使计算机这个昔日“王家之燕”飞入了寻常百姓家,应用于各个领域,渗透到人们的日常生活当中。从办公室自动化应用到几乎所有的小家电,到处都有计算机的影子,甚至国内外一些条件较好的幼儿园都开始利用微机进行辅助教学。可以说,微机的出现是计算机真正走向成熟的标志。

80年代末、90年代初因特网(Internet)的迅速发展,使得计算机从一个个功能强大的孤岛连接成为一个全球性的巨大的信息共享的资源库。信息成了社会的粘合剂,生存在地球各个角落的人们借助因特网而密切地联系在一起,跨越了区域的地理障碍,如同生活在一个村庄一样,有人称之为“地球村”。地球村的人们畅游在由电子信息而形成的一个虚拟世界中,人们称之为赛百空间(Cyberspace)。

展望未来,预计计算机将向微型化、巨型化、智能化、集成化、网络化发展,多媒体技术、多学科综合,将使得计算机向更宽广、更深入的前景迈进。以“信息高速公路计划”为蓝图的信息基础结构将逐渐形成。大规模的先进发达的信息网络逐步建立,与此同时,保障信息网络有效运转的社会支持环境也走向成熟。人们将更方便、高速地利用信息。信息化程度的高低是衡量一个国家国力的新尺码。21世纪即将来临,各国政府都制定了相应的信息发展计划。

全球信息化的过程,给我们每个人、每个国家带来了巨大的挑战和机遇。回顾中国的历史,当人类文明从农业社会跨入工业社会之时,我们正处于晚清鸦片战争的落后挨打局面;当人类从工业时代向信息时代转折的初期,我们正陷入“文革”的泥潭。改革开放之后,中国成了世界上最有希望、最具活力的国家。我们年轻的一代,应该把握时代的脉搏,丰富自己的知识与能

力,为中国巨龙在信息时代腾飞作出应有的贡献。

## 1.2 信息与信息科学

任何一门科学都有其研究的对象和方法。数学作为最基本的基础科学,主要研究数与形的表示和处理。计算机科学研究信息(数据、知识)及其处理。

信息是可以传递的知识,它来源于数据。数据是指文字、数字、图像等各种符号对客观事实的表达,数据是信息的载体。信息的产生、存储、显示、通信等处理过程都要依靠数据来进行,是从数据中提取、解释的。信息与任务有关,要根据上下文并利用知识才能推导理解。

信息的处理可简可繁,处理过程最集中地体现在运算、逻辑判断与存储上,而这正是计算机的特长。随着计算机科学以及相关学科的发展与运用,信息科学得以飞速地发展,并对人类社会产生了深远的影响。

信息科学是一门研究各种系统中信息过程(如信息的产生、采集、存储、交换、传递、处理和使用等)的一般规律,并能动地加以利用的科学。其研究对象是信息;其研究内容是信息及其运动规律;其研究方法是信息科学方法论;其研究的目的是扩展人的信息功能,尤其是智力功能。

与信息科学相关的分支领域有:① 信息论:研究信息基本理论;② 材料科学:研究信息过程基本物质手段;③ 微电子学:研究基本技术手段;④ 计算机科学:研究信息处理工具计算机;⑤ 系统与控制科学:研究系统信息运动规律;⑥ 通信理论与技术:研究信息传递;⑦ 智能科学:研究和模仿人脑中信息过程。

以微电子、计算机、软件、通信技术为主导的信息技术,已经渗透到人类的一切生活和工作中,对整个社会的发展起着巨大的推动作用。

## 1.3 计 算 学 科

### 1.3.1 计算学科的定义

计算机到底是一门科学,还是一门工程科学?其实,称作“计算机科学”,还是称作“计算机工程”,这二者之间并没有什么本质的区别,只是侧重点不同而已。有人建议使用“计算学科”这一名词,因为计算机处于数学(理论)、科学(抽象)和工程(设计)的交汇点。

计算学科是对描述和交换信息的算法过程——其理论、分析、设计、效率、实现和应用——的系统的研究。全部计算学科的基本问题是:什么能(有效地)自动进行?

就理论来说,计算机可以做定理证明,检查技术条件的相容性,检查反例,演示测试等,这是数学的特点。就抽象与模型化来说,超级计算机能评估数学模型,并作出对现实世界的预言,从而具备科学的特点。从设计方面来说,复杂的 VLSI(超大规模集成电路)设计和模拟系统使我们能进行微型电路的有效和正确的设计,程序设计环境使我们能有效地进行软件设计,计算机的工程特点是一目了然的。

### 1.3.2 计算学科的分支领域

计算学科可以划分为以下几个分支领域:

### 1. 算法与数据结构

研究对问题的特殊分类及有效算法,包括算法的性能特征以及与不同需求有关的数据结构。

### 2. 程序设计语言

定义虚拟机及执行算法的表示形式,从高级语言到机器代码的有效翻译,以及对编程语言提供各种扩充机制。

### 3. 体系结构

研究怎样将硬件(和相应软件)有效而可靠地组织起来。例如,处理机、存储器、通信、软件接口的实现,以及大型可靠的计算机系统的设计与实现。

### 4. 数值和符号计算

该领域的核心是从数学模型出发,利用计算机来高效而精确地解各种方程。它包括各种方程解法的可行性和效率,开发高质量的数学软件包等。

### 5. 操作系统

处理各种控制机构,以便在执行程序时,能有效地协调多种资源的服务。它包括对用户请求的相应服务,对资源控制的有效策略,以及对分布式计算的支持。

### 6. 软件方法学和软件工程

基本问题是大型软件系统的规格说明、设计和实现。研究软件开发与编程的原理,软件的验证与确认,特别是对安全、保密、可靠、可信赖的软件系统的规格说明与实现。

### 7. 数据库和信息检索系统

研究信息的组织、有效存取的算法以及已存储信息的修改。基本问题包括数据关系的建模,共享环境中信息的安全与保护,以及外存的特性。

### 8. 人工智能和机器人学

研究人和动物的基本行为模型,并建造实际的或虚拟的机器去模拟动物与人类的行为。包括演绎、模式识别、知识表达、知识工程等内容。

### 9. 人机通信

问题的焦点是人与计算机之间信息的有效传递。它研究影响有效交互作用的人类因素,信息的组织与显示,特别是图形学的技术。

各个分支领域都有其理论、抽象和设计的内容。下面用一个表格来定义计算学科的研究范围模式,事实上这也是我们课程设置的根据。

| 计算学科分支领域 | 理论                                               | 抽象                           | 设计                                      |
|----------|--------------------------------------------------|------------------------------|-----------------------------------------|
| 算法与数据结构  | 可计算性理论<br>计算复杂性理论<br>并行计算<br>密码学、离散数学<br>微积分、概率论 | 数据结构<br>算法技术<br>并行与分布式算法     | 算法设计<br>分布式算法设计<br>网络协议<br>存储管理<br>密码协议 |
| 程序设计语言   | 形式语言与自动机<br>图灵机、形式语义<br>离散数学、时态逻辑<br>近世代数、数学归纳   | 语言的分类<br>语言抽象实现模型<br>编译理论与方法 | 各类程序设计语言<br>程序设计环境<br>编译器<br>编译器的产生器    |

续表

| 计算学科分支领域   | 理论                                                  | 抽象                                            | 设计                                                     |
|------------|-----------------------------------------------------|-----------------------------------------------|--------------------------------------------------------|
| 体系结构       | 数字逻辑<br>有限状态机理论<br>可靠性理论、排队论<br>离散数学、数论             | 计算机组成原理<br>计算机体系结构<br>微机原理<br>VLSI(大规模集成电路)设计 | 硬件设计与实现<br>VLSI 的 CAD 系统                               |
| 数值和符号计算    | 数论、线性代数<br>数值分析、微积分<br>非线性力学<br>实变、复变函数             | 连续问题的离散逼近<br>符号积分与微分                          | 形式化系统<br>数学软件工具包<br>符号运算                               |
| 操作系统       | 并发性理论<br>调度理论、存储管理<br>概率论、排队论<br>通信和信息论<br>时态逻辑、密码学 | 操作系统<br>分布式计算<br>安全控制<br>网络                   | 操作系统设计<br>网络结构<br>网络协议                                 |
| 软件方法学和软件工程 | 程序验证和证明<br>时态逻辑、离散数学<br>可靠性理论<br>认知心理学              | 软件方法学<br>程序开发自动化方法<br>工具与程序设计环境<br>程序与系统的测评   | 软件工具<br>质量管理、软件测试<br>用户接口<br>大型软件工程系统                  |
| 数据库和信息检索系统 | 关系代数、并发理论<br>统计学、密码学                                | DBMS 数据库管理系统<br>SQL 查询语言<br>超文本模型             | 数据库设计技术<br>信息检索系统<br>超文本、数据仓库                          |
| 人工智能和机器人学  | 结构力学、离散数学<br>形式语言、语言学<br>哲学、心理学                     | 知识表示与处理<br>自然语言识别与表示<br>推理和学习模型<br>仿生物的机器结构   | 专家系统、定理证明<br>自然语言理解系统<br>神经网络、机器人<br>策略性游戏程序           |
| 人机通信       | 高维几何、着色理论<br>认知心理学<br>线性代数、离散数学<br>自动机、物理学          | 画面显示算法<br>CAD 模型、图像处理<br>实体对象的表示<br>人机通信      | 图形标准、用户接口<br>多媒体、超媒体<br>Virtual Reality<br>CAD 系统、图像系统 |

计算机科学家关心的是表格的理论和抽象两栏,而计算机工程师关心的是后两栏(抽象和设计)。

另外,还有一些跨越若干领域的研究方向。例如,并行性渗透到所有分支领域(包括并行算法、并行语言、并行结构等),也涉及到理论、抽象和设计。类似的情况还有安全性、可靠性和性能评估。

与计算机有关的还有一些社会的、道德的以及职业的论题。例如计算机与社会历史联系,专业人员职责,知识产权等。

### 1.3.3 计算学科的定义

如果从理论、抽象和设计的分类角度来看,计算机系软件专业开设的专业课程一般有:

#### 1. 理论类

数学分析,高等代数,近世代数,概率论,离散数学(集合论、图论、数理逻辑),数字逻辑,可计算性理论,计算复杂性理论,形式语言与自动机,并行计算等。

## 2. 抽象类

数据结构与算法分析,编译原理,计算机语言学概论,计算机体系结构,操作系统,分布式计算与计算机网络,软件工程,面向对象的分析与设计,数据库概论,人工智能,计算机图形学等。

## 3. 设计类

计算引论(包括 Pascal 语言),C++ 语言,面向对象的 Smalltalk 语言,Internet 与 Java 语言,程序设计实习,微机原理与实习,Unix 程序设计环境,Windows 编程,TCP/IP 协议等等。

计算学科是一门新兴的交叉学科,既具有很强的理论基础,又需要扎实的实验训练。设计类的课程自然会包含大量的上机实验,抽象类的课程一般也具有相当多的实验设计。只有有效地、深入地进行理论与实验的学习才可能真正掌握学科的精髓,培养出本学科的工作能力。

# 1.4 本课程的目的

作为一门导引性的课程,我们将简单介绍计算技术的发展历史,计算学科的一般性概念和方法,计算机系统的软件、硬件环境,以求引出本领域的学习目的和显示本领域的研究内容。

程序设计是本学科标准的实践活动的一部分,每一个计算工作者都必须有程序设计的能力,这是一个基本功。计算学科的特色大都寓于程序设计符号之中,程序设计语言是涉足本领域特色的基本工具。Pascal 语言作为结构化程序设计的首选语言,以其清晰的结构,丰富的类型,编译、运行的高效性,以及方便的可移植性,被广泛地用于各类教学环境和软件开发中。因此,我们的入门课程将以标准 Pascal 语言为媒介,讲授结构化程序设计的基本原理与方法,并且配合以适量的上机实习。

为此,我们的教学目的是:

- (1) 了解计算学科的历史、概貌,掌握计算机系统的基本概念;
- (2) 掌握高级语言程序设计的原理与方法,培养结构化程序设计的风格和习惯;
- (3) 能够编写一些简单的非数值计算、非信息处理等有关的程序,并在机器上运行调通。

# 1.5 学习方法

四年的本科培养的主要目的是面向学科的思维能力与工作能力。计算机的技术发展日新月异,我们真地需要“活到老,学到老”。学习和研究能力的培养,不仅仅在于知识的积累,更在于知识的灵活应用,以及开拓与创新能力。

能力的培养有三个境界。其一为“吸收”阶段,完成知识的汇集、整理、剪裁,这是依葫芦画瓢的模仿,主要是学知识、做练习。其二为“融会”阶段,从知识中找出方法,从方法中体会思想,并且利用现成的方法解决一些问题。能够做到这一点,就已经具备了很好的工程技术人员的素质。其三为“贯通”阶段,触类旁通,应用适当的知识和方法来发现问题、分析问题,从而解决问题。这三个阶段在我们的学习过程中是有机地结合在一起的。而只有能够达到第三阶段的人,才是真正有创意的开拓型人才。这种人最善于思考,在以后的工作中,他们甚至能够创造新的方法,或者开辟新的方向,是具有研究能力的人员。

另外,计算学科是一个富有集体劳动特性的行业,合作与交流是极其重要的。

计算学科的学习培养过程是非常富有挑战性和创造性的,这种扎实的基本功训练对于将来仍然从事计算领域的研究和工作的入自然是一笔财富。那些转而从事其他领域工作的人,也把学习与创新的能力带到了新的岗位。

对计算学科的一般原理和方法以及计算机硬件、软件知识的理解,可能会随着其他课程的介绍而有更深入的感性认识。至于程序设计的原理与方法,尤其是结构化程序设计方法,以及 Pascal 程序设计,则是我们必须掌握的内容。

程序是在数据的某些特定的表示方法和结构的基础上对抽象算法的具体表述。合理有效的数据组织是程序设计的基础,而算法设计是程序设计的核心。Pascal 的设计者 Wirth 提出的著名公式“程序=算法+数据结构”充分说明了两者的的重要性。作为计算机软件专业的初学者,要注意保持一定的高度,不要一下子就钻入语言的细节,应该首先从算法的角度去思考问题。学习到一定的深度后,更要从算法和数据结构两方面去把握问题。当然准确掌握标准 Pascal 语言的语法、语义,有助于我们书写和阅读程序,体现数据结构,实现和改进算法。

我们应该注意,养成“自顶向下、逐步求精、模块化”的结构化程序设计风格,注意遵循结构化程序设计“抽象、分解(子目标)、局部化和信息隐蔽、一致性、完整性、可验证性”的原则。Pascal 语言有助于我们养成良好的编程习惯,设计出逻辑清晰简明,运行效率高,可读性好,易于维护的程序。

另外,高瞻远瞩,注意遵循软件工程的一般原则,注意程序的文档与注释,将为我们未来设计大型软件打下有益的基础。

具体到同学们的学习,通用的学习方法是:主动学习,提前简单预习,独立完成练习(包括书面练习题和上机实习题)。课上课外多与老师和同学交流。

提高学习兴趣这是最主要的。学习时要勤于思考,最好是能够自己超前地预习,在预习时多问为什么,带着这些问题再听老师讲课,收获自然就大了,兴趣也就上来了。而且爱提问题的人,将来才能发现问题,从而解决问题,成为有创造力的人才。上课时集中精力,跟上老师的进度,有问题及时提出来与大家一起讨论。对于初学者来说,怎样建立程序设计的思想是有一定难度的,掌握枯燥的语法规定也有一定的困难,要解决这两个问题,必须要有大量的阅读程序、做习题、编写程序、上机调试程序的工作。

做完习题,最好能借几个学习比较好的同学的习题本参考一下,可以发现别人的好方法,开拓自己的思路,也提高阅读程序的能力。切记一定要自己先做,千万不要一个问题想不出来就直接去抄别人的答案。有问题想不出来,可以与老师、同学讨论,以启发自己的思路。

程序设计的途径是:学习语法、研究例题(带着简单实例模拟计算机运行的过程)、总结方法、上机调试。其中最不可少的是亲自上机调试自己编写在纸面上的程序,因为要学会游泳,一定要亲自下水实践,仅仅在陆地上学习游泳的理论知识是学不会的。编制程序是一项非常富有创造性的劳动,当收获劳动的果实时,人们的心情是十分喜悦的。第一次看到神秘的计算机按照自己编写的程序正确地按预定的运算次序运行,初学者一定非常有成就感。

老师可能只会布置几道大程序作为上机检查题,但是如果有条件,一定要多上机练习一些小习题,尤其对于自己没有把握的程序。每次上机前要静静地思考一下你准备干什么,最好在分析完算法后,把程序完整地写在纸上再去上机操作。否则,面对计算机,脑子里一片空白,时间白白浪费了,还徒生了对这个神秘而陌生机器的恐惧。上机前要先思考,作好充分准备,下机后更要及时总结,设计好下次上机的工作流程。

调试程序时应该先做基本功能,再加界面。界面最好和功能分开,界面部分可以作为子程序甚至编写成为 Turbo Pascal 的单元。加界面时,对于可运行的版本,应该保留一个拷贝(用 Save As 或 Copy 命令),在一个修改版本上反复修改,直到运行结果较理想才把它作为永久版本保留。对于算法不好或不正确的程序,不要“敝帚自珍”,要敢于抛弃,重新精心设计算法,推倒重来。

计算机是你最忠实、最可靠的老师和朋友,你编的程序正确,它会回报你以预期的正确答案;你编的程序有错误,它一点也不客气地给你指出来,而且它还提供调试工具(Debug)来帮助你查找错误。通过自己上机而找出来的错误,你的印象最深刻,有关的语法和算法就牢牢地掌握了。

## 1.6 小 结

作为本书的引子,我们介绍了信息时代的主要特点,强调了计算学科在信息科学中的作用和地位,并介绍了课程的教学目的和学习方法。

人类正处于信息时代,这个时代的标志就是计算机与信息处理技术的广泛应用。

信息科学是一门研究各种系统中信息过程(如信息的产生、采集、存储、交换、传递、处理和使用等)的一般规律,并能动地加以利用的科学。其研究对象是信息;其研究内容是信息及其运动规律;其研究方法是信息科学方法论;其研究的目的是扩展人的信息功能,尤其是智力功能。

计算学科是对描述和交换信息的算法过程——其理论、分析、设计、效率、实现和应用——的系统研究。全部计算科学的基本问题是:什么能(有效地)自动进行?

计算学科的主要分支领域有 9 个:算法与数据结构,程序设计语言,体系结构,数值和符号计算,操作系统,软件方法学和工程,数据库和信息检索系统,人工智能和机器人学,人机通信。

“计算引论”这门课程的教学目的是:

- (1) 了解计算学科的历史、概貌,掌握计算机系统的基本概念;
- (2) 掌握高级语言程序设计的原理与方法,培养结构化程序设计的风格和习惯;
- (3) 能够编写一些简单的非数值计算、非信息处理等有关的程序,并在机器上运行调通。

学习的基本途径是钻研,勤于思考,举一反三,多与别人交流。通过阅读和编写并调试程序,而掌握程序设计的基本方法,并准确掌握标准 Pascal 语言的语法、语义。程序设计的环节最要注意的是问题分析、算法设计和合理有效的数据组织。

初学者必须养成“自顶向下、逐步求精、模块化”的结构化程序设计风格,把“可读性”放在第一位,注意程序的清晰、易懂。注意学习 Pascal 语言嵌套格式的书写方法,添加适当程序的空格与注释,书写相应的程序文档。

## 1.7 习 题

- 1-1 什么是数据?什么是信息?二者有何异同与联系?
- 1-2 请给出信息科学的定义。
- 1-3 请给出计算学科的定义。计算学科的主要分支领域有哪些?

## 第二章 计算机系统发展史

社会需要是一切发明之母。计算机的发展是人类社会需要的产物。

人类计算工具的历史演变过程经历了五个阶段:机械式计算器,机械式逻辑器,机械式输入输出装置,完整的近代计算机器,现代的计算机系统。

### 2.1 人类计算工具与方法的发展初始阶段

#### 2.1.1 机械式计算器

原始人类的生产力有限,剩余物质不多,最自然的计算工具便是手指,但是其缺点是不能存储。结绳记事,甲骨文等记载方法,是最初的数据存储方式。经过漫长的发展,才出现了计数方法和算术。

随着畜牧业的发展,出现了“算筹”(细圆棍)。中国数学家祖冲之用算筹计算出圆周率 $\pi$ 的数值在 3.1415926 到 3.1415927 之间,比西方早了 1000 年。但是算筹太大,携带与运算都不方便。传说秦始皇爱随身携带算筹。有一次,他的算筹掉入东海,就变成了后来的乌贼鱼。

公元 13 世纪(东汉),人们把算筹固定下来,就成了算盘,这是最早的计算工具。这种重要的运算工具至今仍广为使用。

17 世纪,数学和计算工具的重心转移到欧洲,随着天文学与数学的需要,各式各样的计算器发展起来。1621 年美国数学家奥垂德(William Oughtered)根据对数原理发明的圆形计算尺,是最早的模拟计算工具。

1642 年,法国物理学家帕斯卡(Blaise Pascal)发明了齿轮式加法器。为了纪念他,瑞士的沃思(Niklaus Wirth)教授把自己所设计的一个结构化的编程语言命名为 Pascal 语言。

#### 2.1.2 机械式逻辑器

1854 年,英国数学家布尔(George Boole)出版了《布尔代数》(《Boolean Algebra》)。他的逻辑理论建立在两种逻辑值 0、1,三种运算 and(与)、or(或)、not(非)之上。这种“二值逻辑”为数字计算机的二进制数、开关逻辑元件和逻辑电路设计奠定了基础。

1777 年,英国的马洪(Chales Mahon)发明了逻辑演示器。

1869 年,杰文斯(William Jevons)发明了逻辑机。

#### 2.1.3 机械式输入输出装置

1804 年,法国的雅各(Joseph Marie Jacquard)发明了穿孔卡织布机。这为穿孔卡输入输出装置的发明打下了基础。

1886 年,美国的霍勒瑞斯(Herman Hollerith)受穿孔织布机启发,发明了第一台机电式穿孔卡系统——造表机,用于美国的人口普查工作。后来早期的计算机输入输出设备就是穿孔机。他所创立的造表机公司是赫赫有名的 IBM 公司(International Business Machine Corpora-