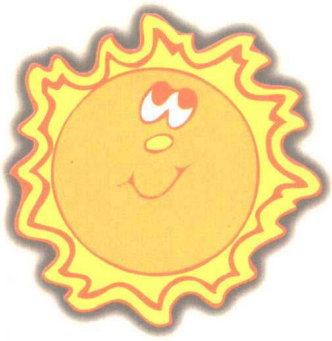
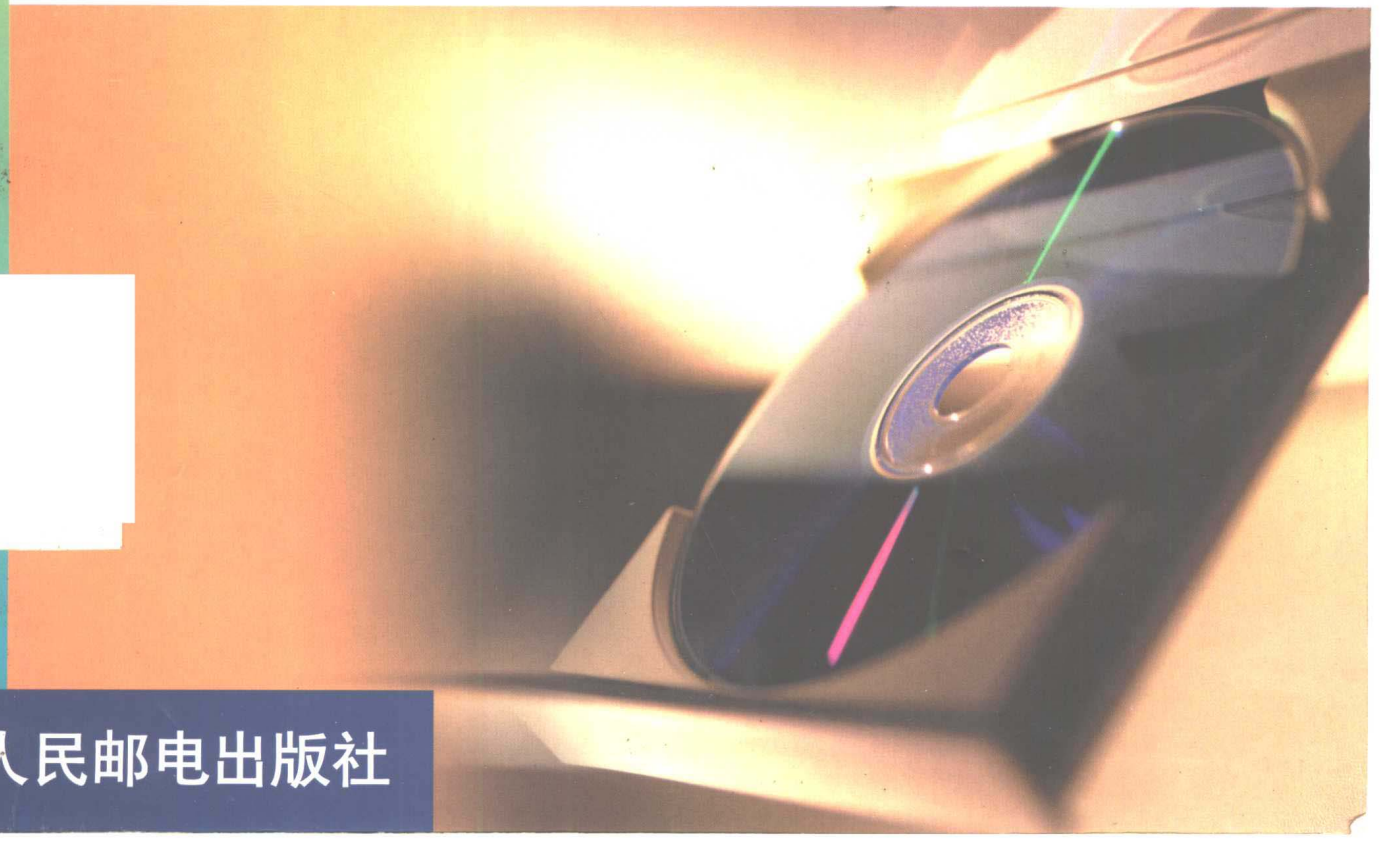


计.算.机.系.列.教.材

COMPUTER

PASCAL 语言程序设计

沈长宁 王凯 冯飞 编著



人民邮电出版社

计算机系列教材

PASCAL 语言程序设计

沈长宁 王凯 冯飞 编著

人民邮电出版社

内容提要

Pascal 语言是一种结构化的程序设计语言。本书介绍 Pascal 语言的基本概念、程序设计方法，包括 Pascal 语言的基础知识、程序结构、控制语句、子程序、数据类型、输入输出及文件操作和指针等。在每一章后面都有上机实验和习题可供读者练习使用。

通过本书的学习可以使读者了解到 Pascal 语言的基本特征，并能掌握一定的程序设计方法，为今后进一步学习和使用不同的计算机语言打下坚实的基础。可作为大、中专教材使用。

计算机系列教材

PASCAL 语言程序设计

◆ 编　著　沈长宁　王　凯　冯　飞

◆ 人民邮电出版社出版发行　北京市崇文区夕照寺街 14 号

邮编　100061　　电子函件　315@ pptph.com.cn

网址　http://www.pptph.com.cn

北京顺义向阳胶印厂印刷

新华书店总店北京发行所经销

◆ 开本：787×1092　1/16

印张：14.25

字数：349 千字　　1999 年 7 月第 1 版

印数：10 101－14 100 册　　2000 年 7 月北京第 2 次印刷

ISBN 7-115-07948-X/TP·1200

定价：19.80 元

编委会名单

序　言

为了适应“逐步实现教材多样化，增加不同品种、不同档次、不同风格、不同改革试验的教材”要求，我们组织编写了这套《计算机系列教材》，以适应大、中专计算机教学的需要。

本套教材的基本任务是系统地阐述计算机的基本概念和基本操作，这些基本概念和基本操作将是未来掌握计算机知识的基础。因此，本套教材的构架并不打算定位在不断变化的、十分活跃的研究和应用领域，而是立足于对基础知识的介绍上。本套教材共包括以下10本，计划1999年出版前5本，2000年出齐。

1. 《微型计算机应用基础》
2. 《中文Windows 98实用教程》
3. 《操作系统概论》
4. 《C语言程序设计》
5. 《PASCAL语言程序设计》
6. 《数据结构》
7. 《计算机实用软件》
8. 《计算机网络基础》
9. 《微机系统原理与维修》
10. 《汇编语言程序设计》

本套教材的书名基本上沿袭了其他版本，但无论在章节划分还是在内容选材方面几乎都是新的，因此完全可以说是一套新作。在写作方面，力求深入浅出、通俗易懂，并特别注重实例的选择和说明。为了加深对基本概念的掌握，各章的末尾均给出大量习题供学员课外练习。同时，每本教材的后面都附有实验题配合学员上机实习使用。

由于编写时间紧促，而且限于水平和经验的不足，书中肯定存在不少错误和遗漏，我们诚心希望使用本套教材的广大教师和同学们提出宝贵的批评建议。

教材编委会

1999年6月

前　言

Pascal 语言是专为讲授“程序设计”课程而研制的。这一语言本身具有逻辑严谨、适于系统地讲述相关概念的特点：它清晰且自然地反映了涉及程序设计的各个基本概念、能同时满足数值计算和非数值信息处理的需求、还兼顾了效率与可靠性，较好地满足了结构程序设计的要求。自 70 年代以来在欧美的许多国家中，许多学校都曾把 Pascal 语言当作第一语言来教初学者学习程序设计，以使学习者从一开始就能养成良好的编程习惯。

本书是供初学者使用的教材，它力图使学生学会有关程序设计的基本概念，但并没有涉及到 Pascal 语言的全貌。

我自 1981 年起曾为大学本科生、专科生和助教进修班的学员讲授“Pascal 程序设计”课近 30 次，本书是在总结教学经验的基础上写成的。它适合采用介绍语言规则与探讨算法并进的方式组织教学，力求使学生能够早上机、早会使用子程序。书中又较为全面地介绍了数据类型，既涉及了数值计算中的基本问题，也介绍了非数值信息处理的基本问题，知识点和教学要求与教育部考试管理中心制定的考试大纲相一致。

本书假定学生是在 Turbo Pascal 6.0 的集成环境中上机实习的，故在讲述上虽力求与 Pascal 语言的标准相一致，但也兼顾了 Turbo Pascal 的特点。此外书中也简介了上机环境及 Turbo Pascal 的子程序库。

我的研究生王凯、冯飞、佟君亮和董守吉分别参加了本教材的编写与校对，北京师范大学出版社的王安琳也提供了有益的资料。没有他们的参与本书将难以和大家见面。

沈长宁

1999 年 5 月

目　　录

第1章　绪论

计算机语言分为低级语言和高级语言两大类，程序设计语言 Pascal 属于高级语言。Pascal 语言从 1971 年正式推出至今，已成为世界上最广泛使用的程序设计语言之一。Pascal 语言全面地体现了结构化程序设计的概念，具有丰富完备的数据类型、简明灵活的语句和清晰明了的模块结构，书写格式自由，运行效率高，查错能力强，移植性好，程序设计风格优美。Pascal 语言适用于程序设计的教学，对于开始学习程序设计的学生，学习 Pascal 语言有助于养成结构化程序设计的良好习惯。正因为以上原因，国内外的许多学校都将 Pascal 语言作为程序设计教学的第一门语言。本章重点介绍以下内容：

- 程序设计语言的相关概念
- 结构化程序设计
- Pascal 语言的特色
- Turbo Pascal 的集成环境

1.1 程序设计语言概述

本节首先介绍了程序语言的基本概念，然后详细阐述低级语言与高级语言、编译程序与解释程序的概念，最后介绍程序设计语言的标准化以及程序设计语言的发展。

1.1.1 程序语言基础知识

在学习程序设计语言之前，先介绍几个相关的名词术语：

程序设计语言：编写计算机程序所用的语言称为程序设计语言。

源程序：用程序设计语言编写的程序称为源程序。

语言处理程序：把一种高级语言编写的程序翻译成与之等价的用另一种低级语言编写的程序系统，称为语言处理程序。常见语言处理程序有：编译程序和解释程序两种。

目标程序：源程序被编译而产生的计算机可执行的程序称为目标程序。

1.1.2 低级语言与高级语言

计算机语言可划分为低级语言和高级语言两大类。

在本世纪 50 年代初期，编写计算机程序难度很大。因为计算机只懂得以二进制数形式表示的指令与数据，而且不同机器的硬件结构、指令格式各不相同。那时，只有懂得计算机、二进制和计算方法的专家才能编程序。在某种型号的机器上运行的程序不能在其他型号的机器上运行。在编写大型程序时，总是花费巨大而又难免出错。这促使人们寻找新的方法去提高编写程序的效率和减少差错。

汇编语言和以 FORTRAN 为代表的一批程序设计语言就是在这种背景下诞生的。用汇编语言编程序时，以助记符替代了二进制的指令，用符号地址替代了以二进制数表示的内存地址。编好的汇编语言源程序，将由相应的汇编程序转换成计算机能够直接使用的二进制指令与数据。但通常助记符与计算机指令是对应的，型号不同的计算机指令系统可能很不相同，当然对应的汇编语言也就很不一样。必须具体了解该计算机中央处理机的工作原理，才能用汇编语言编好程序。

计算机的指令系统又称作机器语言。除机器语言外，汇编语言也是面向机器的语言。使用这两种语言均要求编写程序人员按照计算机的方式思考问题，好的程序员能充分利用计算机的能力，编成高质量高效率的程序，不熟悉计算机内部工作原理的人们则寸步难行。这类面向机器的语言有时称作低级语言。

为了消除低级语言对编程序带来的不便，人们从 20 世纪 50 年代起就研制了另一类以 FORTRAN 为代表的计算机语言，这类语言能较抽象地描述数据与算法，使人们不必过问计算机工作的细节。在这些语言中对问题的描述与人们的习惯相似，这不仅为不大熟悉计算机原理的人们提供了方便，也为软件工作者提高工作效率创造了条件。总之，这类语言并不是面向机器的，而是面向使用者的语言。这种语言被称作程序设计语言或算法语言。有时把面向使用者的程序设计语言称作高级语言。自 60 年代以来，大部分计算机软件都是用高级语言编写的。

用程序设计语言编写的程序是用计算机解决问题的起点和依据，称作源程序。源程序必须被转换成对应的机器码的指令序列，计算机才能实际执行。这种转换是由在计算机上运行的语言处理程序自动地完成的，也就是说，语言处理程序在计算机上建立了源程序的运行环境。

程序设计语言和相应的语言处理程序的作用是使原来只懂得机器指令的计算机以全新的面貌出现在使用者面前，成了懂得某种程序设计语言的“虚拟机器”。不同的程序设计语言就形成特性不同的虚拟机器，常可适用于不同的应用领域。在众多的程序设计语言中被广泛应用的约有 10 余种，如 FORTRAN、ALGOL、COBOL、APL、LISP、BASIC、PL / 1、Forth、 Pascal、C、ADA、PROLOG、Java……(按头一个版本出现的时间先后排列)。

从另一个观点看，任一种程序设计语言都只不过是选定了组成程序所需要的少量词汇、规定了词法符号的使用方法以及用词汇构成语句、编写程序所要遵守的语法规则。

只有完全遵守程序设计语言规则的源程序，才能被相应的语言处理程序编译成可执行的

目标程序，并能够在机器上运行。如果违背了语法规则，则使语言处理程序无所适从，不明白源程序要干什么，不能把它编译成可运行的程序。此时，计算机只能向用户报告在处理中发现了语法错误。

为了既便于使用者掌握、又有利于语言处理程序高效地去编译源程序，程序设计语言的规则都比较简单，专用词汇比较少，规定的句式也不多，且语言内各种合法成分的含义是确定的，只能以一种方式来理解，不会使语言处理程序误解。

目前在计算机上运行的绝大多数语言处理程序，只会循规蹈矩地把源程序翻译成目标程序，不会替编制程序的人改正拼写或打字错误，所以编写程序的人要先了解所用语言的规则，并认真遵守这些规则。

1.1.3 编译程序与解释程序

语言处理程序负责把源程序变换成计算机可以执行的指令序列即目标程序。语言处理程序有两种：编译程序和解释程序。

编译程序把整个源文件(源程序的文件)转换成目标文件(机器码的文件)，这个过程称作编译。编译之后目标文件的运行就与编译程序无关了。使用编译程序翻译源程序的语言是编译式的语言，一般说编译后得到的目标程序执行得较快，若无需改动程序而要再次运行时，可直接运行目标程序，不需重新编译源程序。但如果对源程序进行了改动，不论这种改动多么微小，则改动之后的源程序都得重新编译成新的目标程序才能实现这种改动。

解释程序则在工作的各个时刻都只涉及源程序中的个别部分，而不涉及整个源程序。它把源程序中待执行的那个语句转换成机器指令，并立即执行这一刚转换来的指令。在源程序执行时，解释程序必须不停地工作，边解释边执行。这使程序执行得稍慢，且不论该程序执行过多少次，每次运行时都要逐条语句地解释源程序。但这种作法使源程序的修改比较简单，只需改动单个语句，并可立即观看到改动的效果，因此便于程序的检查与调试。人们常把这类语言称作会话式语言。

从理论上讲，各种程序设计语言都能以编译方式处理，也能用解释方式处理。通常FORTRAN、ALGOL、COBOL、Pascal 和 C 总是以编译方式工作的，这些语言是编译式语言；而 LOGO、Forth、APL、BASIC 和 LISP 多是使用解释程序，被称作会话式语言。近年来 LISP、BASIC 等语言使用编译程序的版本日渐增多，有的语言还能在编写与调试时使用解释程序，而编好之后又使用编译程序将源程序转换成机器码的程序,因而可兼有编译与解释两种方式的优点。

1.1.4 程序设计语言的标准化

80 年代初，虽然许多种程序设计语言在不同型号的计算机上实现了，但还是某些程序设计语言还没有一个统一的标准，同一种语言的不同版本之间的差异相当大，使得在某种环境下编写的源程序，常常不能在其他不同类型的计算机上使用。

为了改变这种状况，人们先后为常用的程序设计语言制定了国家标准或国际标准。目前，FORTRAN、COBOL、Pascal、BASIC 等语言都有了国际标准。参照相应国际标准，我国也制定了国家标准。在语言的标准制定之后，软件生产者多会推出符合标准的语言处理程序。近年来的趋势是多数语言处理程序提供了与标准一致的基本功能，而又增加许多新的功能。完全遵照标准编写的源程序，能直接在许多不同版本的语言处理程序支持下运行。这无疑为人们在不同环境中充分利用现有软件提供了方便，也使人们能准确无误地理解用该种程序设计语言描述的算法，从而有助于算法的交流。

1.1.5 程序设计语言的发展

1.1.5.1 支持结构化程序设计

近十几年所制定的各种程序设计语言的标准中，有一点是共同的，就是力求使各种语言都支持结构化程序设计。结构化程序设计，是荷兰计算机科学家 E.W.Dijkstra 为保证程序正确性而提出的一种方法。结构化程序设计注重程序结构的合理性，以便对程序正确性加以验证。结构合理的含义有以下 3 点：

- 程序是模块化的，且各个模块间的关系简单明了。
- 程序中的各个模块都只用顺序、选择、循环这 3 种特定的结构。
- 各程序模块和程序段落都只有一个入口和一个出口。

显然，结构合理的程序层次分明、脉络清晰、易于阅读，并且常常是逐层分解成较小的模块后构造起来的。每个模块被调用时，程序运行的顺序又与源程序书写的格局自然对应着。最小的模块又都足够简单，编写者很容易确保其正确性。各模块彼此关系处理得当，又保证了整个程序的运行可靠。

1.1.5.2 程序设计语言的新进展

近年来在程序设计语言的进展中，有 3 点特别引人注目：

- 一是面向对象的技术渗透到程序设计领域，使一些语言引入了描述对象的机制，利用对象可以把数据及施加在其上的操作方法封装在一起，能使软件固有的抽象化、模块化和信息隐蔽这 3 重特性实现得较自然、较方便。这些语言支持面向对象的程序设计，并较好地解决了如何由软件元素构造应用软件的大课题。面向对象的技术在提高软件的可靠性、可理解性和可重用性等方面有显著的作用。
- 二是出现了一些比传统的程序设计语言效率更高、更加非过程化的新语言，在使用这类语言时，用户只需说明要做什么事，而无须具体指明运算的过程。这类语言提供给使用者的界面非常友好，极易使用，因而学习它所需的时间减少了许多。通常这些新语言的适用面较窄，目前一些数据库查询语言及电子表格软件包已具备了上述主要特点，且这类软件包大多既有非过程化的描述能力，又提供了过程化的编程能力，这使人们编制软件既省力又灵活。

- 三是由于人工智能新技术的发展，借助于知识库和推理机制实现的专家系统得到了广泛应用，某些新语言应运而生，利用人工智能所提供的自然语言理解能力可大大改善人与计算机的交互环境。

此外，由于用多个处理机的需要，又发展了支持并行运算的语言。

总之，新语言是层出不穷的，只有那些经得起实践的检验的语言，才能继续存在、发展并逐步完善。这些语言将使软件编制的效率更高，质量更好，使计算机能更好地为人类服务，这对于信息社会的发展是有重要意义的。

1.2 结构化程序设计与 Pascal 的特色

本节首先强调了结构化程序设计的作用，然后叙述 Pascal 的特色，最后对程序设计的风格作一些讨论。

1.2.1 结构化程序设计

在上一节给出了结构化程序设计的概念和一般方法。由于“结构化”是 Pascal 语言的最重要的特色之一，所以有必要再强调一下结构化程序设计的优越性。从表面上看，它的提出起因于对转向语句的评价。众所周知，转向语句是可改变程序执行顺序的控制语句，有人喻之为“万能语句”。但是，该语句使程序的静态结构（程序员书写程序的文字结构）与动态结构（机器执行计算时的结构）差别很大，这降低了程序的可读性、增加了调试与维护的困难。为了解决以上问题，就引进了结构化程序设计。实际上深层的原因是在 20 世纪 60 年代末，人们发现各种大型软件（不论是操作系统还是应用程序包）中总有众多的漏洞与错误，这就产生了对软件的不信任感，出现了“软件危机”。于是使软件编制的全过程科学化、工程化，摆脱工匠式的生产方式的影响，就成了摆在人们面前的大课题，结构化程序设计也就应运而生了。

结构化程序设计实际上就是为了使程序具有合理的结构，以便保证和验证程序的正确性而规定的一套进行结构程序设计的方法。用结构化程序设计的方法设计出来的程序称为结构化程序。结构化程序设计语言就是反映了结构化程序设计的要求和限制，便于用来书写结构化程序的语言。用这种语言书写的程序易于保证正确性。

1.2.2 PASCAL 语言的特色

从使用者的角度看，Pascal 语言有以下几个主要特点：

1. 它是结构化的语言

Pascal 语言是结构化的程序设计语言。Pascal 语言提供了直接实现 3 种基本结构的语句以及定义子程序（“过程”和“函数”）的功能。可以方便的书写出结构化的程序。在编写程

序时可以完全不使用转向语句。这就易于保证程序的正确性和易读性。Pascal 语言强调的是可靠性、易读性和概念的清晰性。

2. 有丰富的数据类型

Pascal 语言提供了整型、实型、字符型、布尔型、枚举型、子域型以及由以上类型数据构成的数组类型、集合类型、记录类型和文件类型。此外，还提供了指针类型。Pascal 语言所提供的丰富的数据结构和上述的结构化性质，使得它可以被方便地用来描述复杂的算法，得到质量较高的程序。

3. 能适应于数值计算和非数值信息处理领域

在 Pascal 语言出现之前，FORTRAN 语言主要处理科学计算，而 COBOL 语言则主要用于非数值信息处理。Pascal 语言则兼顾了这两个不同领域的应用。Pascal 语言可广泛应用于各种领域，还可以用于计算机辅助教育、计算机绘图等应用领域。

4. Pascal 程序的书写格式比较自由

Pascal 允许一行写多个语句，一个语句可以分写在多行上，这样就可以使 Pascal 程序写得象诗歌格式一样优美，便于阅读。

除了以上各点之外 Pascal 语言还具有简单易学的特点，许多学校把 Pascal 作为程序设计课程的第一种程序设计语言。学习 Pascal 语言有助于给学生以严格而良好的程序设计的基本训练。

1.2.3 关于程序设计的风格

对于初学者，从开始学编程就应该努力养成具有良好的程序设计风格的编程习惯，Pascal 语言是有助于培养良好的编程风格的。要养成良好的程序设计风格，应当注意以下几点：

- 尽量采用较简明的算法解题，不提倡使用带有特殊技巧和过于复杂的算法。
- 最好选用一些有明确含义的标识符。
- 用注释对程序做进一步的说明。一些正规的程序中注释占整个程序文件的三分之一到一半，甚至更多。
- 用空行隔开各个程序段，可以使程序清晰易读。按层次缩进，使同一层次各语句从同一列开始。
- 复杂的表达式最好用括号表示运算的优先次序，以免造成误解。

本书的所有程序的书写格式，都是按以上几条要求书写。初学者应注意体会这样写的好处，并在编写程序时运用之。