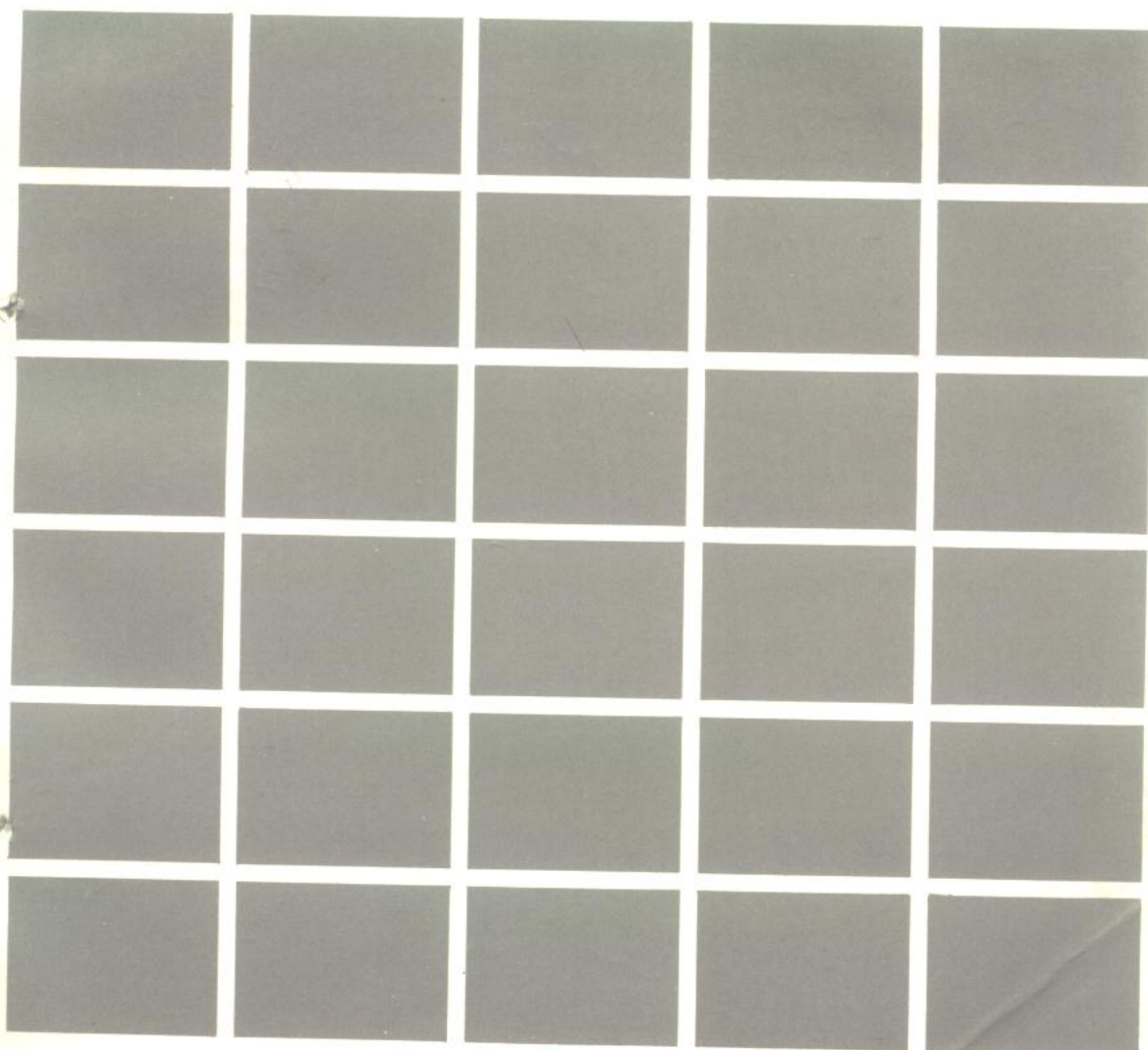


程式設計概論 與問題解析

— 以Pascal語言介紹計算機科學

彭榮生 編譯



五南圖書出版公司 印行

TP-24

程式設計概論 與問題解析

— 以Pascal語言介紹計算機科學

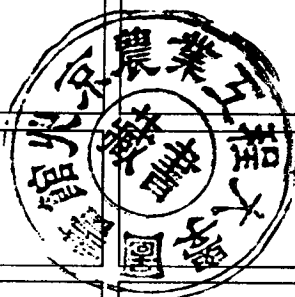
贈閱

美國奧克拉荷馬大學電腦碩士

彭榮生 編譯

北京農業工程大學圖書館

年 月 日



389548

五南圖書出版公司印行

程 式 設 計 概 論 與 問 題 解 析

中華民國 73 年 10 月 初 版

譯 者 彭 榮 生

發行人 楊 榮 川

發行所 五南圖書出版公司

局 版 臺 業 字 第 0598 號

臺 北 市 銅 山 街 1 號

電 話：3 9 1 6 5 4 2

郵政劃撥：0 1 0 6 8 9 5-3

印刷所 茂榮印刷事業有限公司

臺 北 縣 三 重 市 重 新 路 五 段 632 號

電 話：9711628・9713227

售 價 250 元

(本 書 如 有 缺 頁 或 倒 裝， 本 公 司 負 責 換 新)

新時代電腦文庫

總 主 編

范 光 陵 博士

美國斯頓蒙大學企管碩士
美國猶他州立大學哲學博士
美國哥倫比亞大學超博士研究
美國加州大學學院士

會 任

美國IBM電腦公司系統分析師
美國猶他州立大學電腦研究計劃主任
美國加州大學教授
國立中興大學企管系系主任
逢甲大學電腦系系主任
文化大學電腦研究所所長
國立成功大學商學院院長

榮 膺

美國傑出學者獎
美國國際傑出人物獎
泰國電腦成就獎

現 任

國立中興大學教授

新時代電腦文庫

總序

十年前我在美國哥倫比亞大學作超博士研究，主攻人機模控學——也就是研究人類和機器間，如何彼此模仿及有效控制的一門新科學時，接到德國國際會議中心的電話，要我擔任該會舉辦之國際研討會講座。會議是在西德的柏林市舉行。參加者有世界各國電腦專家多位。擔任講座的有美國、日本、奧國、加拿大、意大利、蘇聯、德國、英國、法國等電腦界人士。

閉幕的前一天晚上，大會執行長華特博士在高聳雲霄的自由之針上的旋轉廳，請全體講座吃德國南部名菜豬蹄，並用巨杯喝慕尼黑啤酒。酒過三巡，歌唱十遍之後，華博士說：「全世界都希望聽聽各位的高見，究竟十年後電腦會如何？廿年後會如何？」各國講座即席紛紛發言，又要我作了一個報告如下：

- 一、十年後快速成長的電腦會又小、又快、又好、又便宜，人人買得起。
- 二、十年後電腦將進入辦公室、進入社會、進入家庭，連兒童也要學電腦——電腦文庫將成必備讀物。
- 三、廿年後電腦將從無思考力變成有思考力。
- 四、廿年後電腦將使不懂電腦的人變成功能性文盲。

今天看起來，第一及第二個預測都已應驗了。而日本、英國、美國更自前年起，紛紛從事第五代電腦的設計及製造。我國有識之士，已於今年起一再研究第五代電腦之問題及發展。而即將來臨之新電腦將是一個具有智慧及思考力的機器。他可以讀書，可以與人類以語言交談；如果配在相關機器上，便近乎一位又聰明又能幹的人類。這種電腦系統預計於十年左右完成，一定更會形成新時代的科技及人文大革命。

第三項預測將在各國的大膽革新下實現，而形成對吾人生活及文化之重大衝擊。就第四項而言在那個時候認得「電」字，也認得「腦」字，而不知「電腦」二字加起來是什麼意思的人，便不再是被「新書香社會」尊敬的人士。

我國的知識水準一向不錯，一般說來大專程度以上者約有百分之五，中學程度百分之四十，初中以下百分之五十五。而臺北市之大專程度更高達百分之十四，「文盲」實在很少。在這樣漂亮的統計數字下，我們的「電腦文盲」是多是少呢？

做電腦文盲並不可怕，只要你有「三念」原則就不怕了——那就是要有「念」頭來學習電腦；學後必須要能改變舊觀「念」成為新觀「念」。

這個時代更是「電腦兒童」時代，他們生在電腦時代，所以愛電腦，不怕電腦，電腦可成為他們生活的一部份，他們與電腦在一起覺得很自然；正如許多生在農村社會的人，愛巷子、愛竹馬一樣的自然。這個時代的兒童不會成電腦文盲，也不應該成為電腦文盲；尤其是我們中國的兒童們，他們出生在電腦時代，他們將在國

際商場上為國家作一名鬥士。我們都希望子女成龍成鳳，為什麼不早讓他們學這一個最重要的工具及文化呢？我們都知道練武功要從小開始，學芭蕾舞，學鋼琴要從小開始，而且越早越好；為什麼學電腦不能從小開始？為什麼不准他們玩他們自己新時代的電腦，偏要他們玩「舊時代」的毬子和竹馬？

如果我是一位「電腦文盲」，我會自己先從事「新識字運動」；而不把「上古史」硬拿來束縛住「電腦兒童」及「電腦文盲」們的手腦。須知新時代已迅速而堅決地來了——現在是「駝鳥」飛上枝頭變「鳳凰」的最後機會。

我國電腦資訊的急速發展有目共睹；在發展及成長過程中，陣痛是免不了的，但如不能懷有「臨事而懼，好謀而成」的心理，則美國奧斯邦電腦公司、德州儀器公司及阿他雷公司、富蘭克林公司等，在電腦發展上的失敗，便是殷鑒不遠。所以我們有必要提出檢討，提出改進方法，因為自「1984」年起不過十年左右，「有思考力」的電腦便將誕生了。

要有效促進我國電腦成長，吾人必須積極從事十個新方向：

- 一、輸出要重點突破，不可兼容並包——吾人有較廉價之技術人才、聰明苦幹之知識份子，但限於國力資源及學識，還是抓住幾個重點發展為佳。
- 二、要注意「顧客為主」原則在開拓市場上之意義及價值，不可把生金蛋的鵝趕走。
- 三、造成容許發展之電腦環境及市場，不可朝令夕改；不可因噎廢食；不可過份干涉；應多獎勵學習。

- 四、電腦成長要以「行銷導向」不可以「生產導向」。
- 五、全國修訂不合時宜之法令解釋，行政管轄權及書刊，並引進新知識，以配合新時代之新需要。
- 六、由政府及民間合作成立全國性公正而客觀之電腦資訊委員會，以求統一意見，教育及導引各界，事先準備，迎接新時代。
- 七、用新人行新政——須知在電腦時代，善意的無知為害之烈勝於惡人——因惡人易為人知而加以防範。
- 八、要學習以新管理方法來管理電腦資訊之成長——要學習如何來管理電腦資訊之成長，要重視電腦成長戰略，而不可用「農業波」或「工業波」時代之舊觀念，來管理「電腦資訊波」時代之新成長。
- 九、要把握市場、原料及知識來源——不可俯仰由人，靠天吃飯，要研讀先機、未雨綢繆。
- 十、發動全民力量加入發展電腦之通盤策劃及推廣——須知以全國之力，公私合作，仍不見得能容易應付的挑戰，怎可以有限的人力挑上太重的擔子?!

所以，五南圖書出版公司發行人楊榮川先生開拓「新時代電腦文庫」的魄力與努力，是配合全民發展電腦資訊運動中，堅定而有力的一步。新時代電腦文庫將敦請最好的人才來著述及翻譯最新的學問及出版物。凡是與電腦有關，且有重要性或實用性的新知，均在網羅之列，希望「新時代電腦文庫」，將成為中國電腦發展史上，又一個新的里程碑。而個人才疏學淺，得以參與此一新時代新工作；其惶恐，其愉快，又豈這一篇序文所能表達。

發 行 者 言

范光陵博士被稱為中國電腦之父。他首先在國內揭開了電腦啓蒙運動；他舉辦了中國第一屆人造智慧會議；寫了整個中國第一本電腦書「電腦和你」——是海內外千千萬萬中國人看過的第一本電腦書，他創造了「中文電腦化」，「電腦中文化」的新觀念；舉辦了中國第一屆中文電腦會議；他和有志之士共同創辦了中國第一個全國性電腦團體，也擔任過十次國際電腦資訊會議主席；中國第一任電腦研究所所長，第一任電腦科主任及第一任電子計算機系主任，又主持過中國第一次電視電腦節目。在中國電腦史上他創造了許多第一，也使得新時代的其他新人物，更進一步創造了許多第一。

新時代電腦文庫能由范光陵博士擔任總主編，實在是一件很榮幸的事，相信在他的策劃主編之下，配合碩士級以上的電腦編譯人才，必定能夠達到「不是好書不出版，出版的都是好書」的嚴格要求，共同為中國電腦化，盡一份心力。

楊 榮 川

Programming Concepts and Problem Solving

An Introduction to Computer Science Using Pascal

Peter Linz

University of California, Davis

近年來，計算機使用快迅的成長，應用層面由科技、工業而進入個人生活領域，套裝軟體也應運而起。然而，對於想學習程式設計者而言，學習使用套裝軟體並無任何的幫助。而一般坊間的程式設計書籍大多偏重於程式語言介紹及使用細節。它只能告訴讀者如何編寫正確語法的程式，却不能引導讀者至編寫良好的程式。本書係以帕司卡語言（PASCAL）為工具，除了介紹帕司卡語言之使用外，還介紹有關的資料結構，問題演算法的產生與分析，讓讀者對程式語言能融會貫通，並且運用到實際上的需要。

鑑於計算機科學專有名詞翻譯之混亂而令學習者無所適從，本書之譯名均採用教育部公佈之「電子計算機名詞」，並提供中英對照索引，讓讀者免於混淆。本書之內容，主要分為三部份：

1. 計算機科學基本概論及帕司卡語言之介紹：提供讀者在開始編寫程式之前先瞭解計算機系統，帕司卡語言之使用及編寫程式的一些注意事項如良好習慣的養成等等。
2. 演算法的逐步細分，資料結構之介紹以及測試除錯之技巧：以漸進的方式，由淺入深來分析問題，提供解決問題步驟之演算法，結合程式語言及資料結構之使用，並指導如何測試與除去在程式編寫階段可能犯的語法或邏

輯錯誤。

3.演算法之發現，比較及分析：提供讀者一種比較深入的演算分析法，比較各種可能演算法之效率，幫助讀者選擇一種最合適的方法。

此外，並儘量以圖形及例題配合說明，反覆比較，以啓發讀者之興趣。

本書承蒙陳柏壽及沈松柏二人之試讀、校正並提供建議以及薛明鐘女士在百忙中抽空完成謄稿，謹此致謝。本書倘有疏漏或錯誤之處，尚祈諸位不吝賜教。

譯 者

一九八三年九月于臺北

程式規劃學科

最初的程式設計人員係由各種不同學科，包括工程師、物理學家與數學家等徵召而來。他們經由自行研讀或透過製造廠商代表們所講授的一些非正式課程來學習如何使用計算機。對於接受過數學以及其他科學學科的人而言，他們認為程式設計並不像一門需要仔細研讀與實際經驗的新學科，而是任何一個受過教育的人皆能利用閒暇時間來學習。

這種態度影響了許多的科學家們，並且在今日仍然有人堅持這種觀點。許多年來，在程式規劃的講授課程只集中於對某一種程式設計語言的特徵作詳細的描述。即使對於主修計算機科學的人們而言，儘管他們的需求彼此不同，亦同樣給予這種介紹性的課程。這種草率的方法產生了許多不佳的程式設計人員，而且發生了所謂的「軟體危機」(Software Crisis)。

基本課程

計算機科學的教育如同核子物理教育一般，需要長期的過程，包括正式的訓練與實際的經驗。而學習過程的開始應該是一種介紹性課程足以讓學生學習這門學科的基礎。這種

課程應包括一些計算機之一般概念，並且結合程式設計、資料結構與演算法來著重於問題的解析。這不僅提供一些基本技巧，而且最重要的是奠定以後作為更深入研究之基礎。在許多的建議課程中，對大學部的計算機系影響最甚的課程表，即ACM-78 課程表，其中兩個最基本課程（CS1及CS2），均與程式設計概論及問題解析有關。這本書即包含有講授這二門課程的所有資料。

本書使用的帕司卡語言

在一個介紹性課程裡，強調概念的重要性比強調對某一種語言使用之細節更為重要，因此我們使用虛擬語言。雖然有了虛擬語言，學生們仍然需要去學習一真正的程式設計語言來寫出適合於計算機的程式。對於學生而言，他們須學習兩種語言，若只學習其中之一種都是沒有永久價值，而其更重大的缺點就是無法討論有關的如程式設計樣式、效率、資料結構的選擇等等。幸好，帕司卡語言是一種非常好的授課工具，雖然它的語法包括了一些規則，但都是易於記憶。實際上，它並不會比合理而完全的虛擬語言難學。在教過的許多種語言如符傳、培基與PL/I，我最熱衷於帕司卡語言，並希望在未來能變成為授課的標準語言。

以帕司卡語言授課

然而，因為在這本書的許多地方都用來描述帕司卡語言的語法及語意，因此有些學生認為語言才是這個課程最重要的部份。身為講授者必需提供正確的強調與引導，建立與學

生之間的溝通，並提供本身的經驗予學生。書本只能作為這種溝通的背景。因此，我儘力要求本書的簡單化，並以最簡單的方式解釋概念。我也曾避免比較複雜的例題及一些技術性的問題，而且無意去預先討論所有可想像到的問題。如果學生們有這方面的疑問，講授者應有責任去與他們討論該問題。

本書的特徵

例題

在本書的大部份例題具有4個部份格式，分別以下列之圖形表示：

	問題敘述		演算法
	程式		討論

例題一般都很短——頂多只描述幾點——因此它們儘可能地以簡單而且有效的方式而不用太長的方式來描述概念（根據我的經驗，只有很少數的學生去讀超過一頁長度的程式）。在某些情況下，要求學生編寫的程式可以稍為長一點。

習題

習題的範圍包括有從簡單到複雜的問題，因此它們可以適合各種教育上的需求。較短的習題提供學生們對於某一特殊概念瞭解的快速測驗；中等程度的習題提供對於現成的寫碼之編輯與適應等重要任務之練習；其他較難的習題提供對複雜問題的一種有用的介紹。

附 錄

附錄給予帕司卡語法完整摘要，包括保留字、標準識別、算子居先規則之列表及在本書所使用的帕司卡語言之完整的語法圖。

感 謝

這本書所使用的資料都是從過去數年間在加州大學 DAVIS 分校上課講義所整理出。許多的學生，由於太多以致於無法記起他們的名字，曾間接地獻出他們對這本書的關懷，謹此誌謝。同時我要感謝 Joseph Autin , Grady Booch , Laurian Chirica , John Gillespie , Marc Kerstetter , Rod Southworth , Louis Steinberg , Ronald Suprenant , Elizabeth Unger , 及 Dorian Yeager 等諸位在各種不同階段中來閱讀原稿並提供建議與意見，給予我很大的影響並促成原稿之再改進。最後，我更要感謝在 Benjamin / Cummings 的工作同仁，特別是 Susan Newman 與 Susan Harrington 。他們對本書所付出的努力與意見都是很特別的。

Peter Linz

目 錄

第 1 章	計算機科學的基本概念	1
1.1	資訊處理系統	4
1.2	資料表示法	8
1.3	計算機結構	15
1.4	系統軟體	22
1.5	程式語言	23
第 2 章	程式設計與問題解析	27
2.1	演算與構成	30
2.2	敘述、變數與表示	43
2.3	指派敘述	53
2.4	輸入與輸出	56
第 3 章	簡單程式的編寫	67
3.1	簡單程式的結構	69
3.2	選定計算	76
3.3	良好的程式設計習慣	88
3.4	程式測試及除錯	89
第 4 章	重複計算與控制結構	91
4.1	迴 路	93
4.2	GOTO 敘述	103