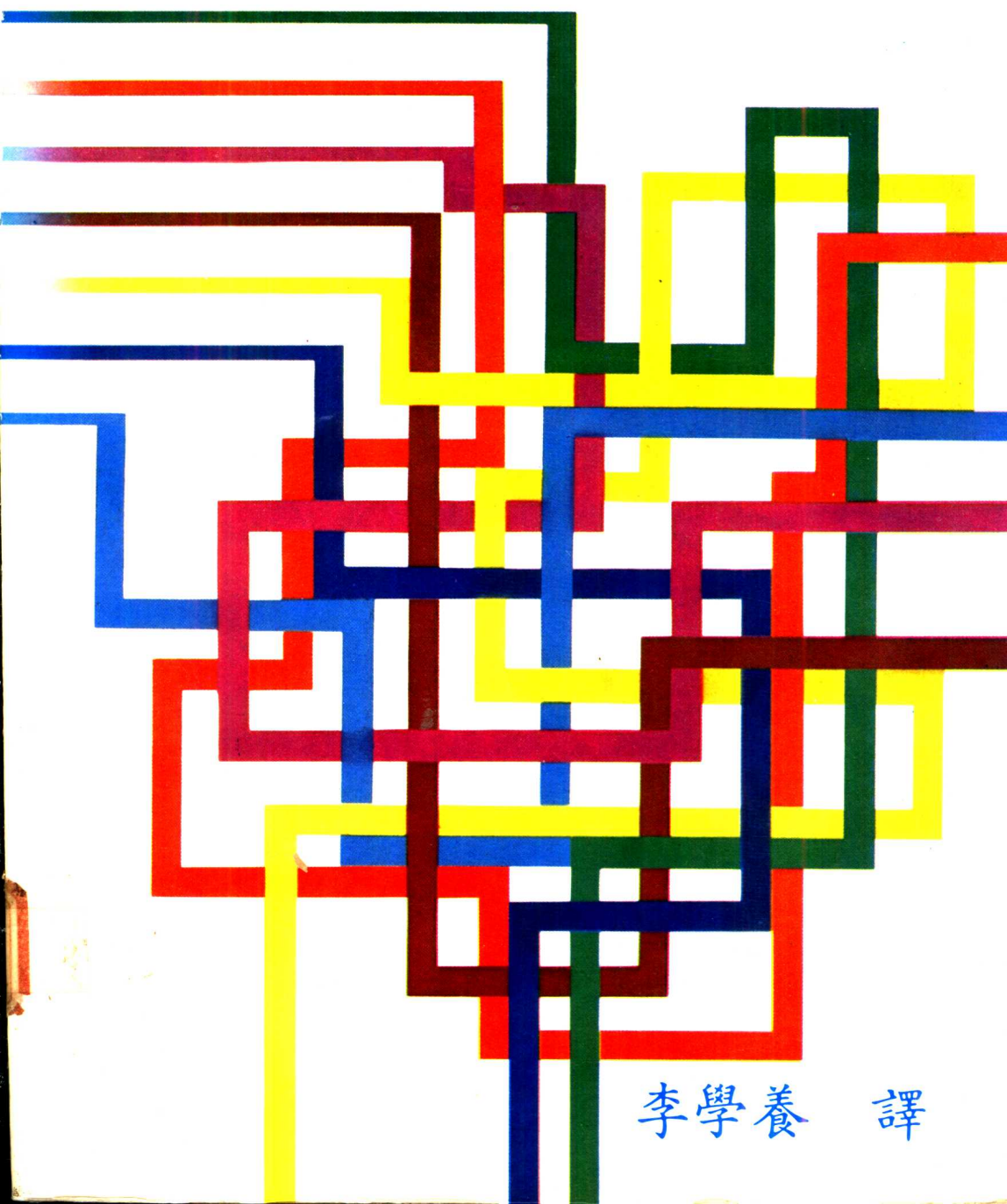


高等程式設計 PASCAL



李學養 譯

高等程式設計

PASCAL

李學養 譯

儒林圖書公司 印行

版權所有
翻印必究

高等程式設計PASCAL

編譯者：李 學 養

發行人：楊 鏡 秋

出版者：儒林圖書有限公司

地 址：台北市重慶南路一段111號

電 話：3812303 3110883 3140111

郵政劃撥：106792號

吉豐印刷廠有限公司承印

板橋市三民路二段正隆巷46弄7號

行政院新聞局局版台業字第 1492 號

中華民國七十一年八月初版

定價新台幣 250 元正

原 序

本書是為大學部計算機程式規畫的第二門課所設計的教科書。內容密切依照 CAM 1978 課程所述的 CS2 Computer Programming II 的大綱。(登載在 Communication of the ACM, 1979 年三月。)唯一的先修課是介紹計算機程式規畫的基本觀念和高級程式語言的「程式規畫」一學期的課。任何類似 ACM1978 課程 CS1 Computer Programming I 的課都適當。

我們的主要目標是使學生有準備以注意和管理程式發展過程中的所有各方面。並且在大而「現實世界」的程式規畫方案環境中覺得舒適而有生產力。出於必需，程式規畫的第一門課通常詳細討論特定語言的語法。很少對程式規劃基本部分的許多其他重要軟體的考慮給予相同的份量的討論。本書介紹給新進程式師關於軟體的規格、設計、寫碼、實現、測試、和維護等範圍廣泛的問題。假如第一門課是在教學生如何寫碼，則本書可看成是教學生如何寫程式而且好好寫程式。

本書共分五篇，每篇內容概述如下：

1. 第一篇介紹由 N. Wirth 發展而廣泛使用的方塊結構語言 Pascal 本書使用此語言教程式規畫的原理。在第一門課已經學過 Pascal 的學生可略去第一篇或把它當做語言復習。
2. 第二篇介紹程式規畫體裁、表達和優雅的原理。本書所用的每一程式、副程式片段都依照本篇的體裁指導方針。學生將看到許多我們覺得是寫得很好的碼，而且開始了解好的程式

H508/05

規劃體裁在閱讀、了解、和使用計算機程式上的重要性。

3. 第三篇介紹結構化寫碼、模組化、逐步精鍊、程式規畫隊、以及演算法和資料結構的由上而下設計等等重要的程式設計觀念。本篇的習題設計有幾個大而需要團隊合作的程式規畫方案，訓練學生使用新學到的設計技術。
4. 第四篇介紹資料結構的更高深觀念——指針和連結串列、檔、堆疊、隊列、字串、和複雜的多層資料結構。我們也提供處理這些資料結構的基本演算法（例如：添加、刪除、查尋、排序）。
5. 最後，第五篇介紹並研究在程式規劃的實現階段所產生的問題——效率、演算法分析、除錯技術、測試、形式驗證、文件說明標準、和程式維護。

第二門程式規畫課的講員常常不得不從不同而無相關的來源選取課程讀物。每種讀物又只注意上述論題之一。代表性的教材可能來自(1)語言參考手冊，(2)有關程式規畫體裁的書，(3)有關結構化程式規畫和由上而下設計的書，和(4)有關測試方法、驗證技術、和演算法分析的研討會或雜誌論文。這種方式對學生既不經濟又不方便，而且必須組織並吸收彼此無關的讀物。

我們按照 ACM 所述適宜當做第二門程式規畫課的論題收集教材並寫成一本教科書。本書就重要而選自不同來源的題材提供一致的看法，這些題材綜合起來可稱為高等程式規畫。

我們已使用並且覺得很重要的另一教學工具，即是使用複雜的實例研究以說明程式規畫的原理。有些教科書常常使用只含25到50列碼的「玩具」程式，說明一些高等設計工具、程式規畫技術、或體裁輔助器具的需要。於是學生很困惑，因為從簡化的程式全然看不出如此複雜的需要。我們試着用複雜到要求並必需使用這種發展

用工具的程式，說明本書中許多主要的地方。因此學生將能夠了解這些程式規畫技術的重要，以及不用這些技術可能導致的混亂。例如，第六章的實例研究提供含有大約 750 列碼的離散事件模擬程式之完整的規格、設計、和由上而下的文件說明。在第九章，我們設計並寫碼一文稿編譯器，說明一些有趣的資料結構之用法。那一程式含有 43 個模組和 1450 列的碼。

這些（以及其他）程式說明大的軟體計畫之複雜性，並且也提供機會解決不同於通常的「寫程式」的有趣而非傳統性的程式規畫問題。有些章末的習題要求學生閱讀和檢討這些實例研究，然後加以修改或擴充以符合新規格。這些維護計畫也可以是指派團隊工作的基礎，使學生在有效地與其他程式師工作和溝通的問題上有更多的經驗。（下略）

施奈德

G. Michael Schneider

Steven C. Bruell

目 錄

第一篇 Pascal

第一章 Pascal 語言

1.1	簡介.....	3
1.2	基本的 Pascal 觀念	6
1.2.1	無向量資料型式	6
1.2.1.1	整數	6
1.2.1.2	實數	7
1.2.1.3	布耳數	7
1.2.1.4	字元	8
1.2.2	運算子	10
1.2.3	標準函數	11
1.2.4	Pascal 程式	14
1.3	程式標題	16
1.4	宣告部	17
1.4.1	標記宣告	17
1.4.2	常數宣告	18
1.4.3	型式宣告	19
1.4.3.1	用者界定的無向量資料型式	20
1.4.3.2	子域資料型式	21

1.4.3.3	陣列資料型式	22
1.4.3.4	記錄資料型式	24
1.4.3.5	集合資料型式	26
1.4.3.6	指針和檔資料型式	28
1.4.4	變數宣告	29
1.4.5	函數和程序宣告	31
1.5	可執行的敘述	35
1.5.1	指定敘述	35
1.5.2	輸入／輸出敘述	36
1.5.3	複合敘述	40
1.5.4	條件敘述	41
1.5.4.1	if / then / else 敘述	41
1.5.4.2	case 敘述	42
1.5.5	goto 敘述	44
1.5.6	迴圈構句	45
1.5.6.1	while 敘述	45
1.5.6.2	repeat 敘述	46
1.5.6.3	for 敘述	48
1.6	範例程式	49
習題		54

第一篇的參考文獻	60
----------	----

第二篇 程式規畫的體裁和表達

第二章 程式規畫體裁

2.1	簡介	63
2.2	體裁指導方針	67
2.2.1	明晰和簡單	67
2.2.2	名稱	70
2.2.3	註解	75
2.2.4	縮格和排版	81
2.2.5	卸下用者的負擔	86
2.3	範例程式	87
習題		91

第三章 程式的跑時行爲

3.1	簡介	96
3.2	強健性	96
3.2.1	輸入確認	97
3.2.2	跑時錯誤的防止	101
3.2.3	表示法錯誤的防止	105
3.2.4	優雅的降級	108
3.2.5	例子	109
3.3	正確使用程序和函數	110
3.3.1	參數傳送機構	112
3.3.2	全盤變數與參數	116
3.3.3	當地變數	118
3.3.4	信號旗標	121
3.3.5	範例程式	124
3.4	一般性	127
3.5	可提性	132

3.6	輸入／輸出行為	137
3.7	摘要	145
	習題	147

第二篇的參考文獻	155
----------	-----

第三篇 程式設計

第四章 結構化寫碼

4.1	簡介	159
4.2	結構化寫碼	160
4.3	結構化寫碼的例子	160
4.4	結構化寫碼的要點	175
4.4.1	goto 詳論	175
4.4.2	while 和 repeat 迴圈構句詳論	183
4.4.3	for 構句詳論	185
4.5	結論	187
	習題	188

第五章 程式設計方法

5.1	簡介	193
5.2	結構化程式規畫	196
5.2.1	由上而下的程式設計	197
5.2.2	由上而下設計方法的優點	211
5.2.3	程式規畫隊	214
5.3	摘要	220

習 題	221
-----------	-----

第六章 程式模組性和實例研究

6.1 簡介	227
6.2 模組特性	228
6.2.1 邏輯一致性	228
6.2.2 獨立性	232
6.2.3 模組大小	236
6.2.4 結構化寫碼	237
6.3 實例研究	237
習 題	287

第三篇的參考文獻	293
----------------	-----

第四篇 資料結構

第七章 線性資料結構

7.1 簡介	297
7.2 綫性資料結構	298
7.2.1 陣列	298
7.2.2 記錄	306
7.2.3 堆疊	318
7.2.4 堆疊例子	323
7.2.5 隊列	329
7.2.6 隊列例子	334
7.3 實例研究：語彙掃描器	335
習 題	357

第八章 連結串列和樹

8.1 簡介	361
8.2 連結串列	365
8.2.1 Pascal 中的指針	365
8.2.2 用連結串列實現的堆疊	371
8.2.3 用連結串列實現的隊列	375
8.2.4 連結串列的簡單運算	377
8.3 哨崗節點	381
8.4 雙連結串列	384
8.5 連結串列的應用	387
8.6 樹	390
8.6.1 定義	392
8.6.2 二元樹	394
8.6.3 穿綫二元樹	403
8.6.4 樹的二元樹表示法	406
8.7 結論	408
習 題	408

第九章 檔和實例研究

9.1 簡介	415
9.2 檔	415
9.2.1 檔緩衝變數	420
9.2.2 文稿檔	422
9.3 實例研究：文稿編輯器	424
9.3.1 編輯器命令	429

9.3.1.1	改變現列的命令	429
9.3.1.2	編輯虛指令	430
9.3.1.3	系統虛指令	434
9.3.1.4	環境虛指令	435
9.3.2	編輯器用例	436
9.3.3	文稿編輯器的資料結構	439
9.3.4	編輯器列表	443
習題		482

第四篇的參考文獻	484
----------------	-----

第五篇 程式實現的工程考慮

第十章 除錯和測試

10.1	程式除錯	487
10.1.1	語法錯誤	490
10.1.2	跑時錯誤	494
10.1.3	邏輯錯誤	498
10.2	程式驗證	503
10.3	程式測試	506
10.4	結論	511
習題		511

第十一章 程式效率

11.1	簡介	513
11.2	效率的人力方面	515

11.3	演算法效率	519
11.4	寫碼效率	531
習 題		538

第十二章 文件說明和程式維護

12.1	程式的文件說明	545
12.2	用者文件說明	546
12.2.1	內容	547
12.2.2	用者文件說明的例子	549
12.3	技術文件說明	554
12.4	程式維護	558
12.5	結論	561
習 題		561

第五篇的參考文獻

附錄 A	Pascal 語言的語法	567
附錄 B	標準 Pascal 識別詞	578
附錄 C	字元集	582
有用的程式和程式片段索引		584
索 引		586

第一篇

Pascal

第一章

Pascal 語言

1.1 簡 介

計算機的程式規畫 (programming) 乃是用計算機解題所涉及的整串步驟。這當然包括寫碼 (coding)，用特定計算機語言寫敘述的過程，以及其他許多步驟。首先必須選用解題的方法，稱為演算法 (algorithm)，然後組織和略述所提的解法。將解法寫成碼之後，必須找出並移去錯誤，測試程式是否正確，並且寫完必要的文件說明 (documentation) 以便確定程式可以很容易地了解、使用、以及更改。圖 1.1 概述整個程式規畫過程的步驟。(注意這些步驟不是循序的；大多數是重疊的。)

介紹程式規畫的作業之外，將要教如何程式規畫得好和寫碼寫得好。基本的程式規畫課程往往強調正確性為單一的目標。用者開始以為只要程式產生正確的答案，他們就已滿意地完成工作。雖然正確性是最重要的，但並不是唯一判斷程式的基準。我們將介紹評