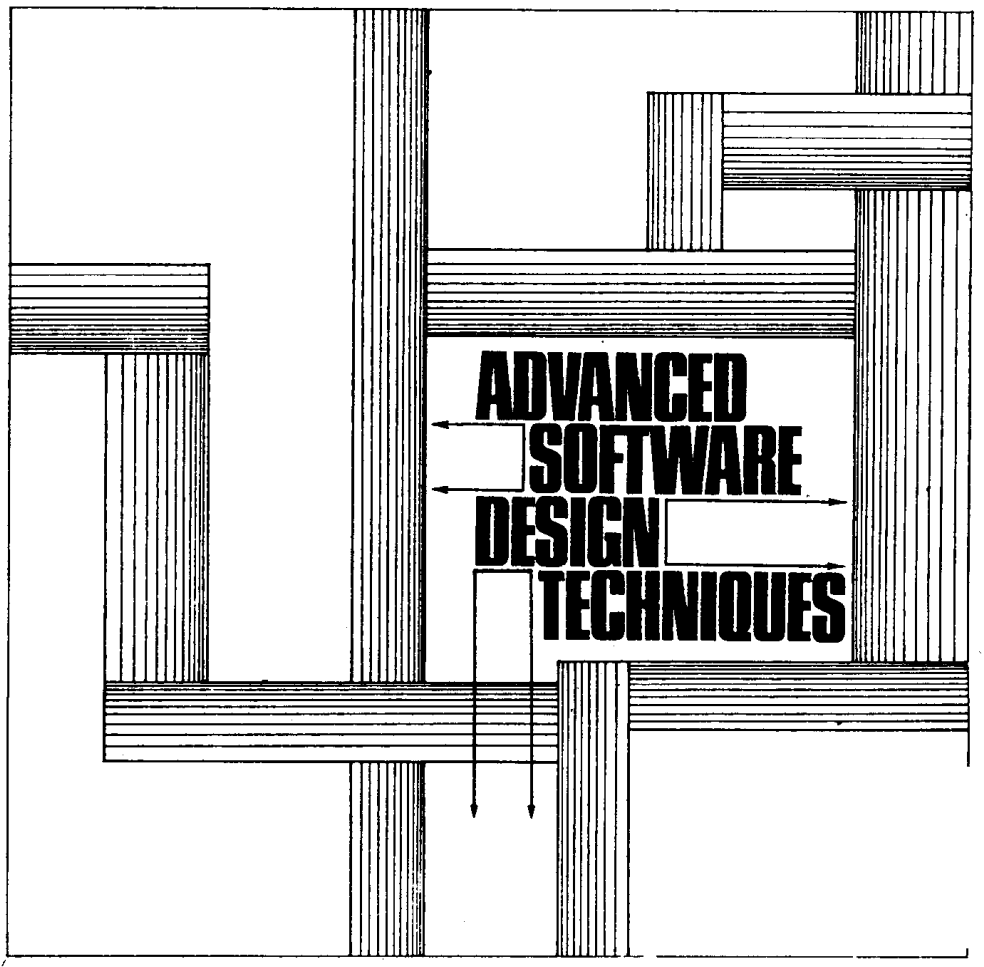
高等程式設計技巧

林睿璋·林玲慧　編譯

全華科技圖書股份有限公司 印行

高等程式設計技巧

林睿璋·林玲慧　編譯



全華科技圖書股份有限公司 印行

全華圖書　法律顧問：陳培豪律師

出版者　全華科技圖書股份有限公司
地址／台北市龍江路76巷20-2號2樓
電話／5811300（總機）
郵撥帳號／0100836－1號
發行人　陳本源
印刷者　華一彩色印刷廠

門市部　全友書局（黎明文化大樓七樓）
地址／台北市重慶南路一段49號7樓
電話／3612532●3612534

定價　新臺幣 180 元
初版／76年9月

行政院新聞局核准登記證局版台業字第〇二二三號

高等程式設計技巧

林睿璋
林玲慧　編譯

圖書編號 0111371

我們的宗旨：

推展科技新知
帶動工業升級

爲學校教科書
推陳出新

感謝您選購全華圖書
希望本書能滿足您求知的慾望

「圖書之可貴，在其量也在其質」，量指圖書內容充實，質指資料新穎夠水準，我們本著這個原則，竭心盡力地爲國家科學中文化努力，貢獻給您這一本全是精華的"全華圖書"

爲保護您的眼睛，本公司特別採用不反光的米色印書紙!!

Advanced Software Design Techniques

Robert J. Rader

RCA Government and Commercial Systems

譯者序

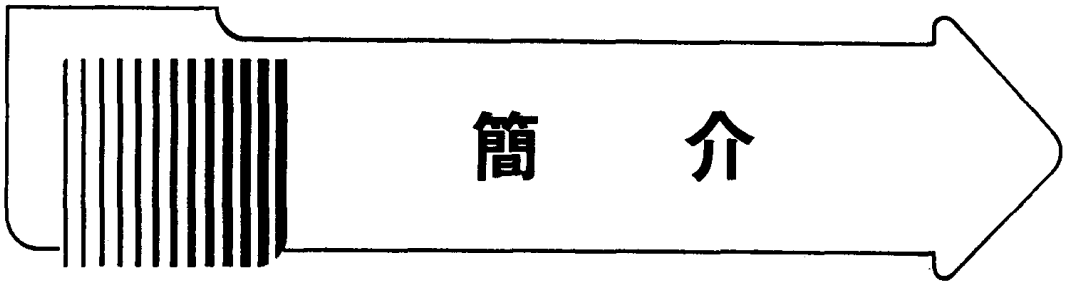
在琳瑯滿目的電腦叢書當中，您偶然的翻閱到了本書，請您珍惜這個寶貴的機緣，因爲這是一本集合了無數位資深程式設計專家，多年智慧累積的結晶。

撰寫程式與作曲、寫文章或畫畫同爲一種幾近於藝術的高度心智活動，所更爲必需的，撰寫程式還需要有清晰的邏輯與尖銳的正確或錯誤觀念，就如同眼睛裏容不下一粒砂子般，一個程式同樣也是不能允許絲毫的錯誤夾雜於其間的。

本書專注於一些常用、理論簡單、方法巧妙、威力強大，但卻大多數人都不懂得正確使用的程式設計技巧，諸如黑盒子觀念、正確的控制流程、完整的結構化設計、鏈結串列、二元樹資料結構、退回追蹤演算法、遞廻演算法、反覆式演算法等。本書的目的也是在於那些還不能靈活運用以上諸方法的程式設計師，使他們能撰寫出更"有效"的程式來。

由於本書探討的是方法的本身，雖然有實例程式的舉用，但完全並不限制於任何一種程式語言，相信這是每一位程式設計師所樂於參考引用的。

譯者　林睿璋　序
　　林玲慧



簡　介

軟體的開發是一項非常艱苦的過程。如果您不相信這句話，我可以舉出許多企業所經歷過的痛苦經驗爲明證來說服您。

目前，大部份的人對於軟體的開發已比五年前有了更多的了解，這是一種集合多數聰明的人來參與的高科技活動，所以它能在技術上有長足的進步。軟體業仍然是個年輕的行業，但一些已被肯定的基礎技術卻有其長遠的價值。本書的目的即是詳細的介紹這些技術。

我曾仔細挑選本書所應有的內容，每一種方法都非常有用而且均爲軟體設計者的一項寶藏，其中許多的技術已被電腦界作過詳細的分析和些許變動。本書適合專業的軟體設計師，但也包含有一般性的基本技術，對它們將有深入的介紹並附上範例程式。

我稱這些技術爲“高級技術”，並不是因爲它們很難學習，而是大多數的程式設計師並不知道它們，也不會發現它們，這些技術大多數是由他人處學來的。

本書的目的不僅僅是觀念的介紹，更希望能做技術的溝通。一個好的設計者一定也是一個好的實行者，這就需要具有良好的技術了。雖然本書有篇幅的限制，但仍然對每一項目做深入的說明，許多程式亦能和新發展的軟體合併使用。

軟體經常被用來解決日益增加的複雜問題，但相反的，軟體業却發現爲了解決這些複雜的問題，這些工具必須更簡化。我們的工作是處於人類心智所能控制的複雜性的

限制內。當其他的條件均相同時，更簡化的工具就使我們能處理更複雜的問題，這也是近來許多軟體發展技術革命的主要論題之一。

這個革命指的就是結構化程式設計(structured programming)。但對於結構化程式設計這個名詞卻有許多不同的看法，使得它的意義幾乎無法統一。比較有效的方法是去忘掉一些學理上的議論，而着重於學習如何來寫出更好的程式。本書所提出的演算法(algorithm)都很簡短，文中也強調結構化設計中最重要的部份爲順序(sequence)，選擇(selection)及重覆(iteration)的控制結構(control structures)的使用。所有控制結構的理論和專業的軟體設計師無關。使用少數的控制結構應能幫助您更容易的來建立程式，有效的使用它們，並與他們共存。本書中所有的設計均爲結構化程式設計的範例。

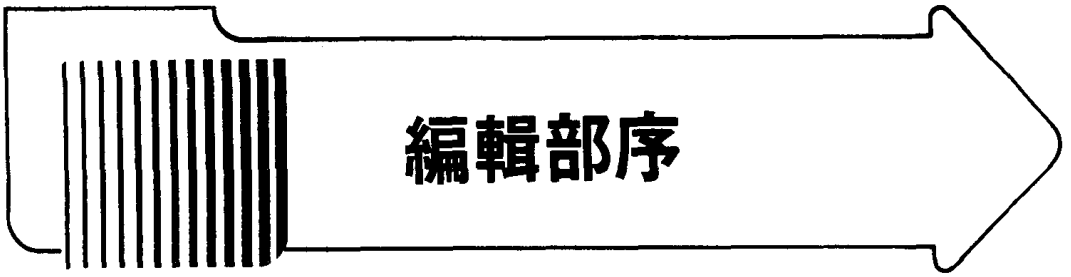
因爲本書和軟體設計有關，我們一定會想到"什麼是軟體設計？"。軟體設計是所有軟體發展活動的計劃階段。許多程式甚至沒有設計，只是被建立(built)了而已，如此所"建立"的程式，結果總難令人滿意，不是程式不能使用，就是不能解決要解決的問題，或是除了撰寫人之外，其他人都不了解，甚至不能修正。設計的目標是要產生對程式功用(目的爲何)的整體說明，以及建立這些功用的方法(如何達成)。這裏有二項回饋性的主要來源可以提示程式設計者來修正錯誤：(1)測試不能使用的設計。(2)由其他的設計者來抓出錯誤。由許多的經驗中我們得到一個結論："測試"並不是一個足夠的提示工具，它太昂貴而且太慢了。另外一方面，由有能力的同僚來作嚴密的設計覆審便能發現大多數的錯誤。當然，測試仍然是一項必需的工作——但不應該用來發現設計上的錯誤。我們要正確的設計，也就是沒有錯誤的設計。測試在本質上來說

，不能證明免於錯誤，只能表示錯誤的存在。

如果我們相信設計的覆審是正確設計的主要過程，那麼設計工作的目標就會變得非常明確；儘可能作有意義的設計而方便被覆審；儘量使設計簡單、正確；並經常想像您必須說服別人相信您的設計是正確的。

產生明確設計程式的能力是非常少見的，直到最近仍是少數人有此能力而大多數人却沒有。我們現在是在解釋一個設計工作的適當份量，使得那些不具備此能力的設計師能獲得此能力。如果某些設計師能在本書的幫助下逐漸走向精通的境界，這也就達到本書的目的了。

在本書前後所提及的演算法是以一種近似FORTRAN的語言所寫的。特別要提的是，陣列（ array ）的註標（ subscript ）是以 1，2，……N表示。註標也被用來表示一些實際不可能存在的索引值（ index value），在這種情形，0 和－ 1 是經常被選用的。在限制較鬆的語言裡，可能要作不同的選擇。



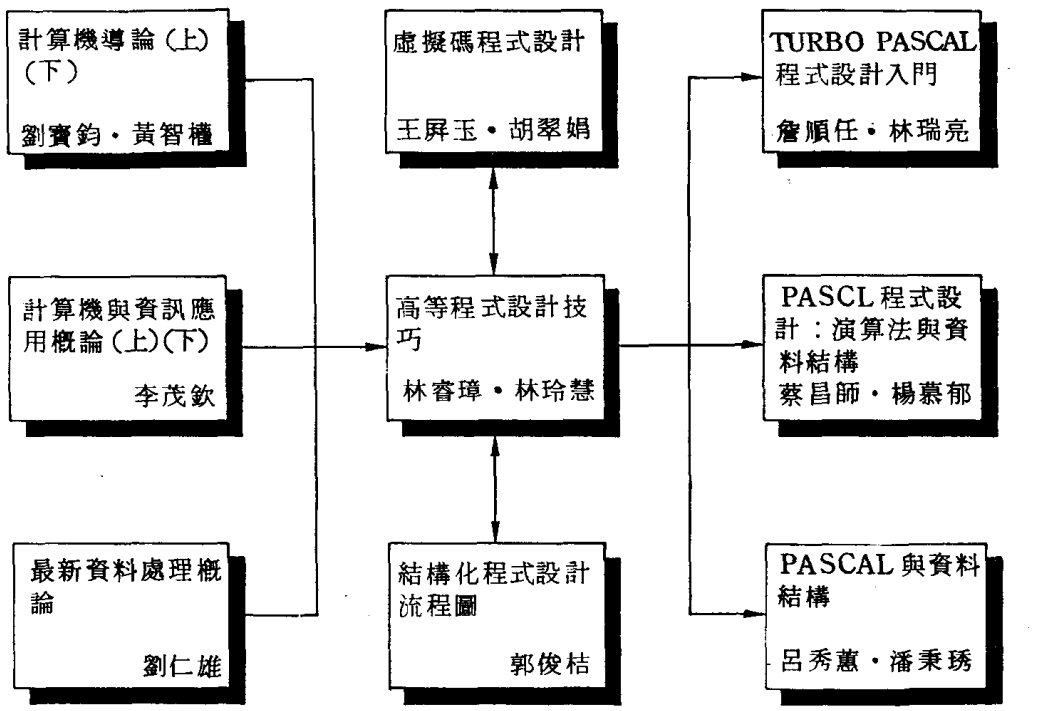
編輯部序

「系統編輯」是我們的編輯方針，我們所提供給您的，絕不只是一本書，而是關於這門學問的所有知識，它們由淺入深，循序漸進。

能夠擁有本書的讀者有福了，因爲本書集合了無數位資深程式設計專家多年來智慧累積的結晶，且書中所有的設計都是結構化程式設計的範例，除了這些優點外，書中還以深入淺出的方式來介紹一些常用、方法巧妙、威力強大，但却是大多數程式設計人員不懂得正確使用的程式設計技巧，因此本書可說是程式設計師撰寫更有效程式的最佳參考書。

同時，爲了使您能有系統且循序漸進研習程式設計方面叢書，我們以流程圖方式，列出各有關圖書的閱讀順序，以減少您研習此門學問的摸索時間，並能對這門學問有完整的知識。若您在這方面有任何問題，歡迎來函連繫，我們將竭誠爲您服務。

流程圖



全華計算機相關圖書

A057 IBM PC結構化BASIC語言教材
江麗華·曾秋蓉編譯
20K/304頁/210元

950 IBM PC/XT BASIC程式設計及應用
修廷璧編譯
20K/360頁/220元

1206 PASCAL問題與詳解
余清華編譯
20K/240頁/180元

A032 IBM PC/XT COBOL
黃展璵編譯
20K/320頁/220元

A053 IBM PC PASCAL語言
鄭林財編譯
20K/344頁/230元

A051 C語言速成
周李義編譯
20K/256頁/180元

1136 C程式語言設計
張志堅·張志偉編著譯
20K/464頁/260元

● 上列書價若有變動
請以最新目錄爲準

目　錄

1 基本的設計技術

Fundamental Design Techniques

本章中所提及的技術適用於本書所有的章節。這些技術對任何嚴謹的軟體設計者而言，應是十分適用的。

1.1 黑盒子（Black Boxes）

一個黑盒子是某些事物（通常是一段副程式）執行一項功能（盒子部份），但是却將執行該功能的方法（manner）——黑色部份隱藏起來。圖 1.1 是一最普遍的例子。在圖 1.1 中大寫字母是輸入（inputs），小寫字母是輸出（outputs），f 是被執行功能的名稱，這三部份構成了盒子的定義（definition）。爲了定義的完整，對盒子的描述一定要包括有效的輸入範圍，及在整個有效的輸入範圍內，輸出對輸入間關係的明確說明，對於超出有效範圍外的輸入的盒子行爲不需作說明。

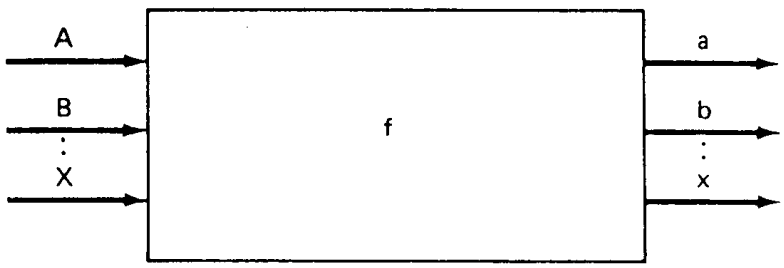


圖 1.1

現在讓我們來考慮有關此種描述的兩群典型的觀衆：第一、是盒子的使用者。一旦定義了輸入和輸出的介面而且盒子的功能也很明確，使用者便有了一切所需要的資料來衡量盒子對問題是否合適，如果合適，須如何使用它。只要盒子能產生正確的回應，使用者並不關心實際上盒子是如何建立的。另外一群便是盒子的建立者。介面和功能的描述是建立盒子的必要條件。只要有了這些介面和功能特性，任何盒子的使用都是可接受的。

我們可以得到一個結論，即只要有黑盒子的完整描述，對盒子的使用者和建立者都足夠了。對盒子內部運作的額外說明只會構成傷害。它會分散並困擾使用者的意志，也可能會全面的限制了建立者，使它遠離一個可接受的建立標準。

設計動作的主體是由黑盒子的定義所組成。使用者可覆審企劃階段所提出的盒子來更正一些誤解並使需要變得更明確。

盒子並不會永遠是黑的，一旦盒子已定義並且發現值得去建立，設計的下一步是爲盒子建立一個內部的架構，圖1.2爲一簡單的說明。現在盒子不再是黑色了，我們可以看到f是建在盒子g、h和k外。現在它們每一個都是黑盒子，而且須加以定義（如圖1.3）。由黑盒子f的觀點，來分析它的內部架構的步驟，通常被稱爲是此設計的改良(refinement)。一個更高階層的功能是由許多較低階層的功能，經過適當的連接而建立成的。更詳細的介面定義也是必須的，至少可以部份的保證盒子間是適當的聯接的。

由上至下的設計(top-down design)是由一連串改良所組成。首先，最高層的功能被描述爲一個黑盒子，然後產生一個由許多低層盒子組成

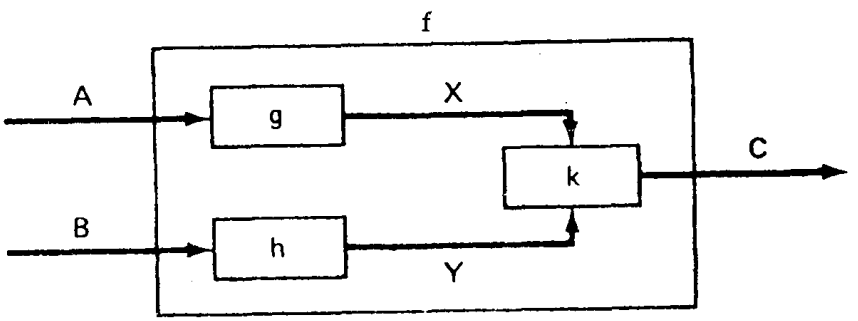


圖1.2