

电脑程式语言RPG

译者 孙庚年

徐氏基金会出版

科學圖書大庫

電腦程式語言 RPG

譯者 孫賡年

徐氏基金會出版

原序

「報表編製程式」可能是當今資料處理中所用的最高級語言。它或許是今日可用的一種最易學習、最易使用、最易維護、編寫最快、與機器最無關、以問題為本位的電腦語言。它是所有各階層的軟體支援——從最小的IBM系統/360 20型電腦到最大型電腦——所能採用的唯一商業程式設計語言。它亦能用於IBM 1130電腦，各階層的RCA Spectra 70電腦系統，以及Univac 9200/9300系列的電腦。

所有這些系統，所用語言之規格，實際均屬相同，因此當由一個RPG系統變更至另一個時，也就毋須作重大轉換了。本書所論述之RPG，可為這些系統的任何一種使用。至於有關特殊系統使用上之重大限制，全書各節亦有提及，如20型電腦所用RPG，便是一例。此外，尚有一段參考部分，專門敘述各種RPG之獨特性質。

閱讀本書，毋需先具備太多的電腦知識。本書乃專為新進的程式設計師而著述。其主要目標乃欲使讀者一步一步地積聚RPG之知識，進而發展至使讀者能全盤了解如何去為各型電腦編寫RPG程式。

因為這個原因，本書落筆力求通俗，對於每一種特性，舉出一個或多個例子，以為示範；必要時並提出具有不同複雜性的種種樣式。在編排方面，本書分成數大部分，由介紹各型電腦，程式設計與系統分析，及RPG在資料處理中之功能開始。不需這些概論資料的讀者可以直接跳到第二篇；此篇敘述如何計劃及編寫基本的卡片RPG程式。

其次一篇包括RPG中可利用的另外一些功能與技巧，如各種表格(Tables)之處理與多元輸入案卷之處理等。當讀者讀完這篇時，他就應該能為卡片本位的系統編寫任何的RPG了。讀者之中，如僅需獲取以卡片為主的電腦系統所用的RPG的基本知識，可以由本篇轉跳到附錄中之試驗問題及資料。當他們需要另外的知識時，可以研讀另外的資料。

接下去的一篇，論及如何為磁帶卷與直接出入案卷設計程式，以及次程式與行數計算規格表之使用。最後一篇為參考篇，包括有其他樣本程式，以

及各種RPR間的異同之檢討。

採用本書為初學讀本時，讀者可在第3、7、9、14、19各章及附錄A中發現完整的樣本程式，同時亦會在第9、14、及19等章看到十一個試驗程式，以及在全書其他各章看到另外的程式。此等程式，除了在書中這些章裡是很有用的之外，讀者在學完前面各章節後，也許還要拿這些程式的一部分來做做，以測驗他對這幾章的了解程度。

舉例言之，讀者學完第4章之後，他應試著去為第9章（基本編碼問題）中的樣本程式與試驗程式設計報表。又當學完第5章時，他應試圖為某些這種問題完成輸出格式規範表。

本書在組織上係把試驗每一篇中的特殊資料之試驗問題與範例問題安排在該篇的最後一章。這些問題可以也應該被部分或全部採用，以測驗讀者對每一章之了解程度。

是以，本書對想廣泛了解RPG之新進RPG設計師們而言，算得上是理想的教材。蓋它實際包含了RPG中所有可利用的功能。再者，由於本書採用談話的方式論述，又不斷舉出說明與示例，故能使讀者了解起來極為快速。

謝詞

本書從頭至尾，承國際商業機器公司、美國無線電公司，及史貝瑞公司的Univac Division之准許，自他們擁有版權的刊物中取用了許多情報與示例資料。這些參攷指南之名稱列於本書之“參攷文獻”欄中。

還有的示例資料係經IBM公司之許可，採自該公司獲有版權，屬於下列書表編號之刊物：

R 29-0091, R 29-0092, R 29-0093, R 29-0094,
R 29-0096, R 29-0097

此外，尚有下列各項材料則是直接轉載的

	IBM書表編號
第9、14及19章中之試驗問題	R 29-8091
第9及14章中之範例程式	C 29-9001
第19章中之範例程式	C 26-3570
附錄A中之範例程式1及2	C 26-3600
附錄A中之範例程式	C 24-9001
	S.M. 伯納德

目 錄

第一篇 緒論	1
第一章 系統與程式計劃	3
第二章 程式之計劃	9
第三章 RPG 程序	15
第二篇 基本功能	25
第四章 報表之設計	27
第五章 輸出格式規格表之編寫	50
第六章 輸入規格表之編寫	69
第七章 計算規格表之編寫	91
第八章 案卷說明規格表之編寫	110
第九章 基本編碼問題	121
第三篇 其他RPG設備	129
第十章 其他功能	131
第十一章 表 卷	156
第十二章 其他邏輯	169
第十三章 RPG 程式設計循環	191
第十四章 多元案卷與表卷程式	206
第四篇 高等RPG應用	225

第十五章	磁帶案卷.....	227
第十六章	直接出入儲存器.....	243
第十七章	出口、次程式與次常規之使用.....	280
第十八章	行數計算規格表.....	289
第十九章	複雜的磁帶與磁碟問題.....	296
第五篇 附錄		315
附錄A	其他RPG程式編寫實例.....	317
附錄B	各種不同的RPG	364
參考書籍		387

第一篇

緒論

第1章 系統與程式計劃

系統是如何設計的？

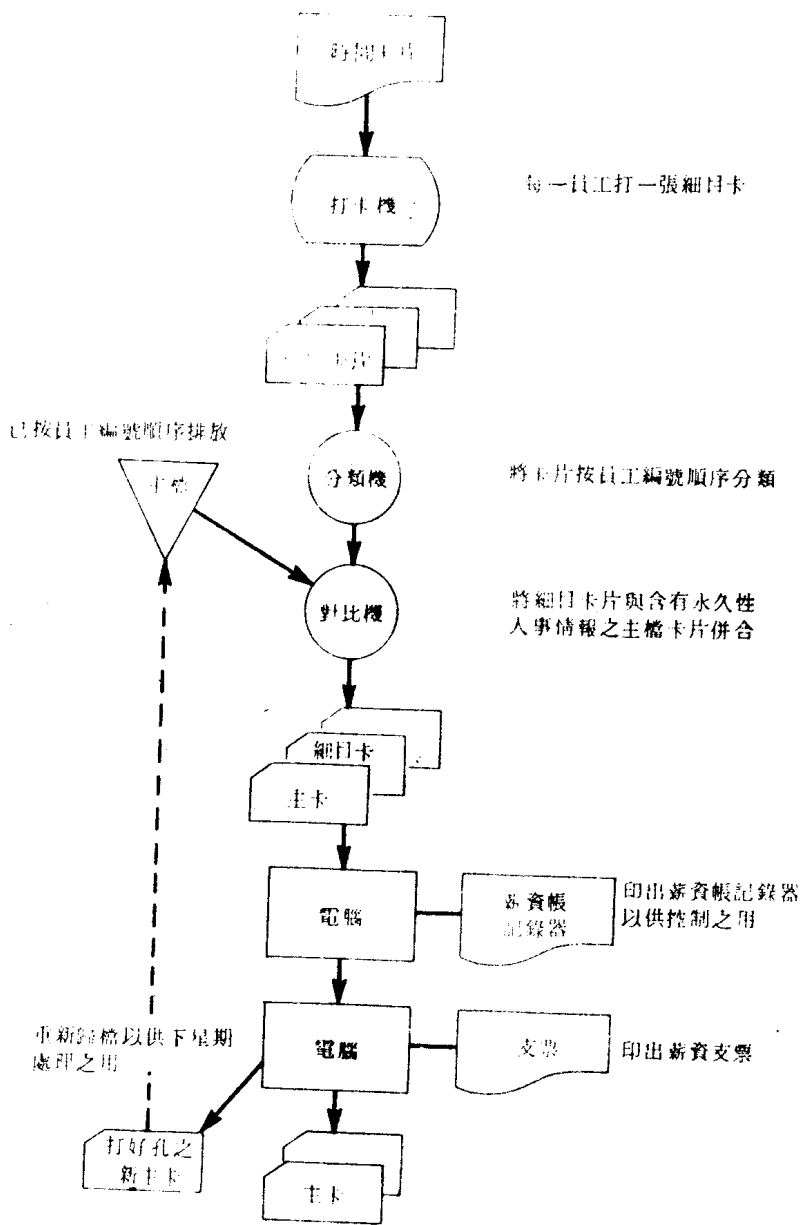
通常，系統分析師或程式設計師兼分析師為了一個公司內部特定資料處理之需要而設計出種種商業資料處理系統。舉例言之，一個分析師可能被指派去設計一種系統，使得某一個公司能為其薪資發放業務，開製薪資支票及所需的會計記錄。此分析師所擔任的工作便是要設計一個自收集員工時間卡片起，至薪資支票及相關的報表被印出為止的工作流程。

假如這個系統是要使用電腦操作，那麼，描述在使用電腦中所包括的每一步驟之輸入、輸出與處理之種種需要，也是分析師所擔任的職能之一部分。這些處理工作，每一項又需依照分析師所陳述的特殊需要，而予以規劃。

如果分析師要將薪資發放程序編寫成書面文件，他可以使用流程圖 (Flow Chart)。流程圖是把處理資料的過程中之每一步驟，以符號來代表的一種圖表。不同的機器以不同的符號代表，其情形有如圖 1-1 所示。

在這個非常簡單，而且不太完整的程序中，我們可以看到許多機器處理的步驟——

- 卡片必須依特定的格式打孔（一如分析師所設定者），以便記錄每一個員工編號及每一個員工之工作時數。在本例中，每一時間卡需打一張 IBM 卡片。
- 打好孔之卡片組，亦即代表本週之員工時間卡片（細目卡片），必須以分類機按員工編號之順序而予以分類。
- 此細目卡片組必須與包含諸如員工姓名，每小時薪資率，本年度往前各月之薪資收入等人事與歷史資料之其他卡片組併合 (Merge)。含有較永久性資料而被一再地反復使用之卡片，叫做主卡 (Master Cards)。
 - 用以將兩個依順序排列的卡片組併合（或分離）起來之機器，稱之為對比機 (Collator)。
- 由併合好之卡片組，電腦便可計算每一員工之薪資總額，各項應納稅



捐，薪資淨額以及該年度各月份之累積收入；同時電腦還可以製備書面文件，將每一員工之各項資料彙總印成一行。

接着同一打孔卡片會再被送入電腦。這一次，除了計算之外，還會為每一員工印出一張支票，同時又打一張包含所有歷史情報以及新的年度累積所得數字的主卡。以新的主卡組將在次星期的報表製作循環中使用。

注意，電腦有各種輸入與輸出機器與其相連結。即以本例中所提到的電腦而論，輸入可能為打孔卡片，輸出可能是印妥之報表或打孔卡片。

在本例中，同一電腦如何能遂行兩種全然不同之作業呢？電腦的每一項作業均由送入其中之指令在控制。依據工作類型之需要，可以將不同的指令組 (Set of instruction) 送入電腦。

每一組完整的指令叫做一個程式 (Program)；程式是程式設計師為應付某一項工作之特定需要而寫成的。

在剛剛提到的薪資帳之例子中，程式設計師須寫出兩種程式，一種供製備薪資記錄之用，另一種供開支票及打新的主檔卡片之用。此等程式之規格可由分析師加以決定，而為其設計工作之一部分。

讓我們假定，我們被指派依據分析師所指定之輸入、輸出、及處理之需要，為一個系統編寫程式。在正常的情形下，資料每通過電腦一回，我們就要寫一項程式。每一程式僅是由一組指令所構成，放入電腦後，可按一定的方式對資料加以操縱。

尚有另外一點——既然很顯然的每星期都會發生一系列相同的事件（即薪資表處理程序），我們每星期即可利用相同的程式來製備薪資表冊。如果需要改變了，那麼所用程式當然亦須根據新的需要再作修改。

RPG是什麼？

每一類型的電腦均有不同的內部結構。因之，給予指令之方式也就各不相同。如是則欲依據每一電腦之內部結構而為其編寫適用之程式，也就需要更多的技巧、訓練、與努力。為電腦編寫符合其內部結構標準之適用程式之工作，稱之為實際或機器語言程式計劃。

電腦製造商為使程式編寫工作容易進行，均有程式計劃語言之供應。這類語言使得程式設計師不必因電腦內部結構之需要而須編寫種種詳細指令，代之以一種快速的編寫方法。

例如，就 IBM 系統 /350 而言，即有下列迅捷的程式設計語言可資應

IBM INTERNATIONAL BUSINESS MACHINES CORPORATION
REPORT PROGRAM GENERATOR FILE DESCRIPTION SPECIFICATIONS
IBM System 360

Form 500-200-1
Revised 10-7-70

IBM INTERNATIONAL BUSINESS MACHINES CORPORATION
REPORT PROGRAM GENERATOR FILE EXTENSION SPECIFICATIONS
IBM System 360

Form 500-200-2
Revised 10-7-70

IBM INTERNATIONAL BUSINESS MACHINES CORPORATION
REPORT PROGRAM GENERATOR LINE COUNTER SPECIFICATIONS
IBM System 360

Form 500-200-3
Revised 10-7-70

IBM INTERNATIONAL BUSINESS MACHINES CORPORATION
REPORT PROGRAM GENERATOR INPUT SPECIFICATIONS
IBM System 360

Form 500-200-4
Revised 10-7-70

IBM INTERNATIONAL BUSINESS MACHINES CORPORATION
REPORT PROGRAM GENERATOR CALCULATION SPECIFICATIONS
IBM System 360

Form 500-200-5
Revised 10-7-70

IBM INTERNATIONAL BUSINESS MACHINES CORPORATION
REPORT PROGRAM GENERATOR OUTPUT-FORMAT SPECIFICATIONS
IBM System 360

Form 500-200-6
Revised 10-7-70

Not available on all RPG's

用：

組合語言 (Assembler Language)
 COBOL
 FORTRAN
 PL / I
 RPG

本書所論述之RPG（報表編製程式），仍是為基本的商業應用而設計的一種程式計劃語言。由於它可供產生或創出以機器語言（告訴電腦該作什麼所需之內部電碼）表示的程式，故稱為編製程式；又因其最常被用為能產生某種類別之報表的電腦程式，故也叫做報表編製程式。

RPG因係採用規格表 (Specification Sheet) 的方法來編製程式，故與其他程式設計語言相較，顯得很獨特。也就是說，我們並不按每次一個指令這樣來控制我們的程式之邏輯，而是藉RPG把我們所要電腦執行方式統統在規格表上描述出來。

圖1-2所示即是此種程序所用之表格。有些表格之名稱指出了表上所載明之功能。主表 (Major Sheets) 供作案卷說明、輸入、計算與輸出規格之用。

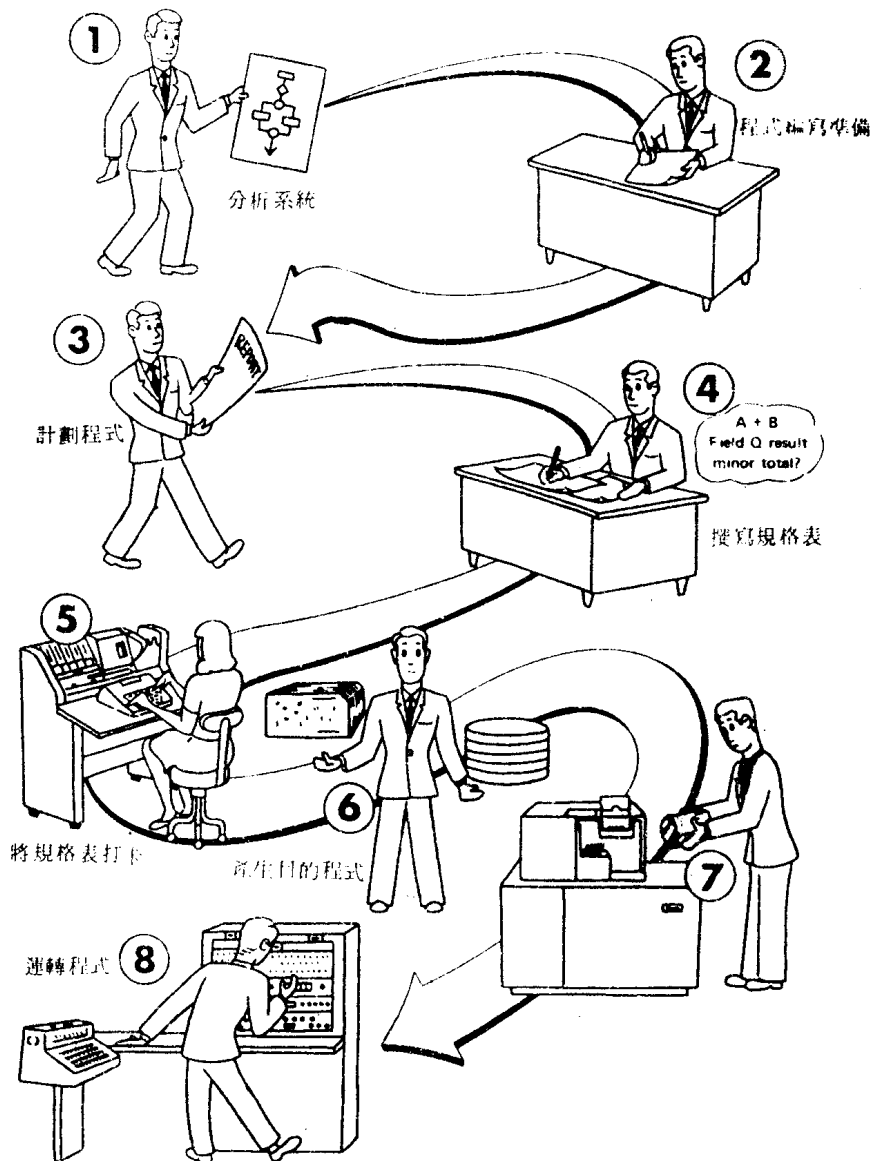
對我們所寫的每一項程式，我們每次要在這些表上記載一行。我們在規格表上所記的每一行，最後要用一張IBM卡片來打孔。由這些記載所創出之打孔卡片組就成了程式組 (Program deck)，或說得更恰當些，就是原始程式組。

讓我們來復習一下如圖1-3所示之RPG程式編寫程序：

1. 分析師使用流程圖設計出工作流程。
2. 他以輸入、輸出和處理之需要描述每一電腦運轉步驟。
3. RPG程式設計師計劃他的程式。
4. 他在RPG規格表中作適當的記載。
5. 將原始程式打卡。

另外還需要三個其他的步驟，才能完成此一程序：

6. 將原始程式組轉換成目的程式組（機器語言程式組）——此項程序將在後頭述及。
7. 將目的程式加以試驗和除錯，找出錯誤並予修正之。
8. 如果全部都做得很好，程式便可按照計劃處理資料了。



第2章 程式之計劃

程式能夠操縱輸入，並產生輸出。以符號表之，則有如圖 2-1 所示。

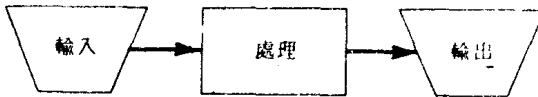


圖 2 - 1

爲了編寫程式，我們需在開始之前，對所有這三個階段的資料處理循環先予界說清楚。

一項完成的程式，其處理或執行工作之方法是獲取一單位之輸入資料，將之按某種邏輯方式加以處理，然後創造出一單位的輸出資料。這些功能的每一項，有如我們在下面所即將看到者，均有若干特別的規格。基本上，此乃係一種反復進行的程序；即一旦此一循環完成了對一單位的資料之處理後，整個循環又爲處理次一單位之資料而重新開始（見圖 2-2）。

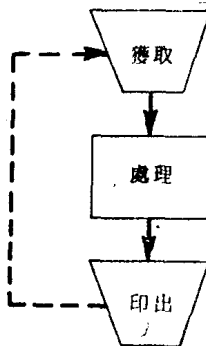


圖 2 - 2 :

拿一薪資資料處理程式（見圖 2-3 及 2-4）爲例，它可以先讀取第一位員工之時間卡片，根據他所工作之時數及薪資率計算他的薪資所得，然後

在報表上印出一行。當這一行在報表上印完後，程式又可再用相同的邏輯，為後面的員工計算。如此相同的邏輯將被一再使用，直至最後一張輸入卡片完成處理，報表的最後一行印好時為止。

在這個例子中，我們必須填寫四種規格表（見圖 2-5）：

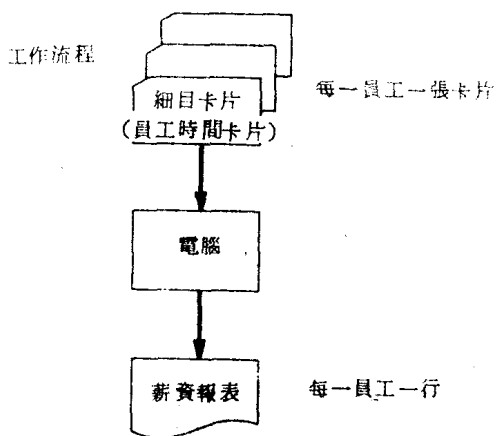


圖 2-3

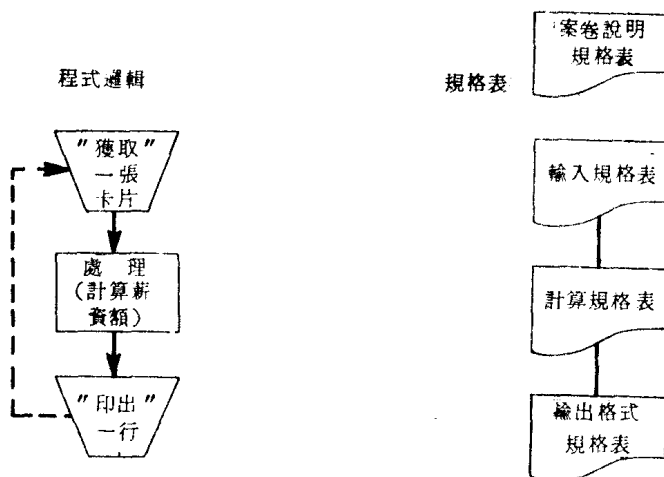


圖 2-4

圖 2-5

1. 案卷說明——敘述讀卡機與印製機
2. 輸入——說明打孔卡片之格式
3. 計算——說明各項計算工作
4. 輸出——說明印出報表之格式

於是，程式便會根據規格表的記載，反復執行它的指令。

作為一個程式設計師，我們應澈底瞭解程式之輸入處理和輸出的種種需要。因此我們在開始設計程式之前，應先有：待處理之打孔卡片之欄別分配樣卡，邏輯之彙總說明，以及所欲編製的報表之說明（見圖 2-6）。

假設這項情報有了，我們便可按如下步驟進行：

1. 檢討問題，務必確定業已完全瞭解
2. 在一張特別的規劃表上設計所欲印製的輸出資料之格式
3. 編寫輸出規格表
4. 編寫輸入規格表
5. 在計算規格表上編寫程式邏輯
6. 編寫案卷說明規格表
7. 複核所編各項規格表，確保沒有錯誤存在
8. 將原始程式打卡

至於更複雜的程式，如具有多元輸入案卷或需要用到更多的邏輯與產生更多輸出者，其工作程序亦大致相同（見圖 2-7）。

在我們原先提過的那個例子中，我們用了兩類卡片——目前薪資卡（細目卡）及歷史卡（主卡）；——以及兩種輸出——支票與新的主檔卡。還有，我們可以先在規劃表（如印製區格圖）上設計輸出格式，編寫 R P G 輸出、計算、輸入、和案卷說明表，再將我們的工作核對一遍，然後把程式交付打卡。程序進行到這裡，接着便是試驗和除錯階段，此後程式便可拿來反復用以處理資料了。

在我們閱讀一個如何正確使用 R P G 的詳細範例之前，讓我們先來復習一些我們業已提過而且對於 R P G 程式計劃之瞭解頗關重要之名詞。

系統分析師——分析及設計商業系統應用的人。

流程圖——以符號代表某一商業系統之各項處理程序之圖表。

電腦——包括中央處理機及與它相接之輸入和輸出設備。它的種種特定操作在程式之導引下獲得控制。

程式——程式設計師所寫成的整套指令，係被送入電腦的中央處理機中；乃為遂行特殊作業（一項特定工作之輸入、處理和輸出功能）而設計。