# 電腦程式設計
# BASIC

李惠明 編著

萬人出版社 出版

# 電腦程式設計
# BASIC

李 惠 明 編著

萬人出版社 出版

# 電腦程式設計 BASIC



萬縮卡電字第 78-1001 號

編著者：李 惠 明
出版者：萬人出版社有限公司
發行人：呂 芳 烈
電 話：3144895 3939280
地 址：台北市重慶南路一段 20 號二樓
郵 撥：119410 萬人出版社
出版證：局版台業第 1882 號
印刷者：啓盟印刷廠股份有限公司
台北市長泰街 121 巷 10 弄 2 號

中華民國六十九年三月二版

特價 NT$ 150.00

# 序

BASIC 程式語言的特點是易於學習和使用，而且具有一般對話的特性，爲當今中小型電腦程式語言中，使用廣泛者之一，是學習其他電腦程式語言的基礎。

坊間有關BASIC 程式語言的中文書籍雖有一、二本，但是大都僅介紹BASIC程式語言中各種敍令的格式，對於敍令的活用是很少加以詳細說明。著者有鑑於此，乃將歷年教學心得寫成此書，以供教學及自修之用。本書的編寫要旨如下：

1. 目的在使讀者儘早能夠設計完整的BASIC 程式。
2. 對於各種敍令的格式及用法，均作詳細的介紹，同時附有例題，以期使讀者能夠設計正確而有效的程式。

在編寫本書時，承蒙中國市政專科學校創辦人上官業佑先生、校長李瑞衡先生的鼓勵和支持，謹此致最高的謝意。蒙中華電腦中心系統分析師陳憲彰先生、財稅資料處理中心程式設計師黃瑞屏先生提供甚多寶貴的意見，特此致謝。

本書之編著係利用公餘之暇，漏誤之處在所難免，尙祈各先進不吝指示。

李　惠　明　識

# 目　錄

# 第一章　程式設計概述

## 1—1　前　言

電腦對於我們日常生活的影響越來越大；舉凡水電、瓦斯、帳單、稅單、薪資表等，現在都可用它處理，又如工程結構分析、工程估價分析，以及預測氣象等都可藉著電腦計算處理。但是電腦只是套構造複雜的機器，當我們希望電腦爲我們處理許多不同的工作時，還需要運用「工具」來指揮它；如果缺少這種「工具」，我們就難指使電腦去處理所面對的問題，它亦將無法爲我們效勞。

上述所謂「工具」，那是指「程式語言」，如欲靈活運用電腦，必先要會程式語言來設計，因爲每部電腦需要程式來指示必要做的每一不同的步驟。換句話說沒有程式的電腦就像一個「死的機器」而已，不能作任何一件工作。

程式就是一串有邏輯順序而能被電腦接受的敍令所構成。

## 1—2　程式語言的種類

電腦發明之初，程式設計人員必須用機器碼來寫程式，這種機器碼結構簡單，使用起來却相當麻煩，非但不容易記憶而且學習困難。因而乃逐步發展出目前接近人類語言（英語）的程式語言。所以我們可依其接近適用英文之程度，大致分爲二大類：低階語言(Low Level Language)及高階語言（High Level Language）。

### 1-2-1 低階語言

**低階語言**包括了**機器語言**及**組合語言**。

**I. 機器語言**

是由二進位數字來代表電腦所要作的工作。由於這種語言不易了解，而且書寫不易，因而對使用者極其不便，目前已不採用。

**II. 組合語言**

由於機器語言完全是由二進位數字所代表，對程式設計人員當然是件極煩且易出錯的工作。爲了解決使用機器語言書寫程式的困難，進而有所謂組合語言的產生。組合語言的每一個敍令，對應一個機器語言，存放敍令及資料的儲存位置也可以用符號來表示，這樣對於記憶及了解敍令的意義，有極大的幫助。例如：相加的敍令，可以用A（ Add 的縮寫）來表示。同時亦使程式的書寫修改更爲方便。

組合語言不是電腦能夠直接接受的語言，必須經過編譯程式編譯使它成爲機器所能接受的機器語言。

組合語言較機器語言更爲方便，且能控制中央處理機的每一細節動作，在控制機器作業的系統程式及一些講求使用電腦效率的程式，常使用組合語言書寫。但由於機器不同，組合語言也不同，因此不易普遍使用。即是組合語言是隨著電腦之廠牌不同而異，像ＣＤＣ　CYBER 機型的組合語言叫 COMPASS；IBM360 機型及 370 機型就叫 ASSEMBLER，因此不易普遍使用。

## 1-2-2 高階語言

組合語言雖具有優點，但對一般使用者而言，並非容易學習、便於使用之工具。爲克服其缺點、高階語言遂應運而生。此種語言之特點爲接近於使用者習用之表達方式，且不受機種的限制。

高階語言是無法被機器直接接受的，必須經過其編譯程式編譯，

成爲機器語言才能執行。它與組合語言不相同，組合語言與機器語言之關係差不多是一對一的，也就是每一個組合語言的敍令對應產生一個機器語言的敍令。但是高階語言就不同了，一個簡單的敍述可能要由數十個以上的機器語言敍令來完成。因此，以高階語言寫程式時，設計人員更能專注於處理的邏輯，減少對機器內部許多細節的考慮。

目前高階語言已有一百多種，除了具有特殊用途的語言外，現被廣泛使用的語言有：商業資料處理的語言一 COBOL，科學計算用的 FORTRAN，具有商業用及計算用優點的 PL/1，易於學習使用的 BASIC 等。

## 1—3　電腦作業概述

電腦作業人員的職能包含了下列三項：

1. **系統分析**（System Analysis）
2. **程式設計**（Programming）
3. **電腦操作**（Operation）

通常從事電腦工作人員，由於工作環境的不同，工作範圍往往混淆不清，也許包括上述前二項或全部，或者只作專一的工作。使用小型電腦的機構，工作的界限比較不容易劃分清楚；使用大型電腦的機構，就分得十分清楚，而且分工合作的安排很微妙。

**系統分析**（System Analysis)的工作可分爲兩個不同的方向。一種是應用分析（Application Analysis），它不太需要對電腦結構有很深的認識，但對應用方面的知識却需十分透徹了解，諸如工作方法、方程式、參考文件流程等是。另一種是電腦運用分析，需要對電腦能力、容量、速度及如何有效運用等專門知識，需有透徹的了解，並給予應用分析者有關的建議與幫助。

**程式設計**（Programming）的工作，就是用一種特定的格式、符

號或字句的組成書寫成使電腦能確實地去執行應用工作的每一詳細的步驟。

**電腦操作**（Operation）的工作就是將程式輸入進電腦，並安排需要的輸出／入的裝置，使電腦運轉，使程式能順利地完成工作。

## 1—4 程式設計的步驟

現在我們再詳細分析一下程式設計的步驟。

1. **了解程式之規範**
2. **計劃解決程式的邏輯分析**
3. **依邏輯繪製程式流程圖**
4. **利用所規定的程式語言書寫**
5. **測試與除錯**
6. **編寫程式說明**

### 了解程式規範

所謂程式規範（Specification），就是包括了輸出／入的檔案佈置（Layout）或單據、報表規格圖及處理過程的要點，或計算公式，亦就是它的功能。

在程式設計之前，須對規範有透徹的了解，並明白其相互間的關係。

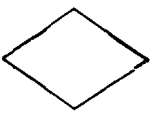
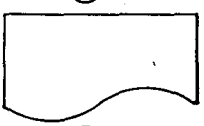
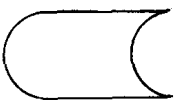
例如：庫存作業輸入單據有進貨單、發貨單…等；檔案有庫存主檔（Master File），異動檔（Transaction File）等；有庫存狀況表等。處理要點：進貨時庫存主檔增加，發貨時庫存主檔減少，如果低於安全存量，即列印採購單等。

### 計劃解決程式的邏輯分析

雖然對於「程式規範」有所了解，但須安排分析選擇其解決問題的步驟的最佳方案，如果遇到非常複雜的邏輯，則需借助決策表（Decision Table）來解決。

**繪製流程圖**

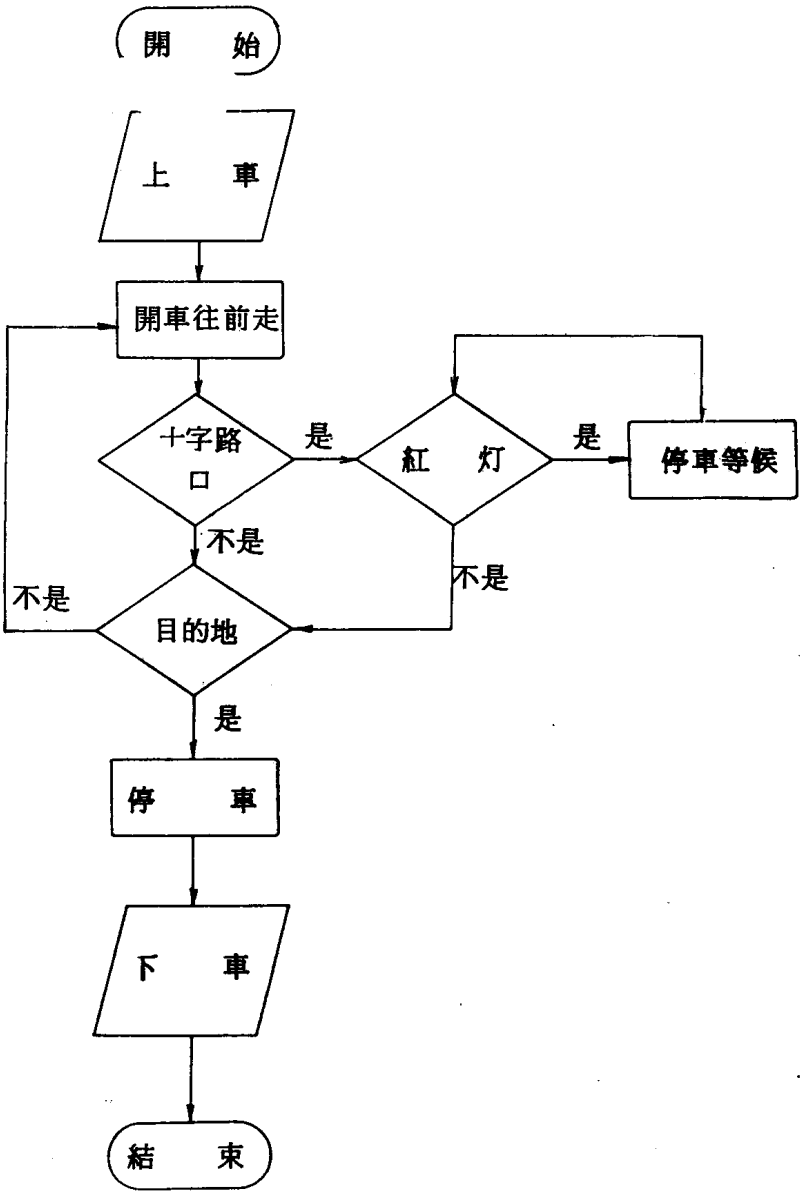
有了最佳的方案後，則需借助流程圖來表示其邏輯步驟。在此將介紹幾個一般常用的**基本流程圖符號**：

| 符　號 | 符號名稱 |
| --- | --- |
|  | 端點 |
|  | 輸入／輸出 |
|  | 處　理 |
|  | 決　擇 |
|  | 連　接 |
|  | 文件；報表 |
|  | 磁　帶 |
|  | 磁　碟 |
|  | 單據；紙帶 |
|  | 流　向 |

**說明：**

1. **端點符號**：端點亦即是起訖點，用來表示流程圖邏輯上的「開始」(Start)，「結束」(End)，「中斷」(Interrupt)等。
2. **輸入／輸出符號**：輸出／入符號是表示資料之輸入或輸出。也就是對電腦提供資料以備處理，或者將已處理之資料予以記錄或列印出。
3. **處理符號**：是表示一個或一組運算的執行，以將資料之數值、形式或位置加以改變。
4. **抉擇符號**：是表示決定性的處理。也即是根據處理的結果，決定下一步應走的路綫。
5. **連接符號**：是用以連接流程圖的二部份。繪圖時因受紙張大小的限制，或其他需要將流向轉移到較遠的地方時，可以利用連接符號表示流綫的連續。而且可以與其他很多條流線會合。
6. **文件、報表符號**：用來表示輸出／輸入資料的媒體是一般的紙張或表册之類的文件。
7. **磁帶符號**：表示輸出／輸入資料的媒體係用磁帶。
8. **磁碟符號**：表示輸出／輸入資料的媒體係用磁碟。
9. **單據、紙帶符號**：表示輸出／輸入資料的媒體爲紙帶或代表紙帶機。
10. **流向符號**：用來表示處理步驟的順序，流綫介於兩個其他符號之間，用流綫上的箭頭表示方向。

**例題**：今以開車上班程序爲例、繪一流程圖。(見次頁)


開　始
上　車
開車往前走
十字路口
是
紅　灯
是
停車等候
不是
不是
不是
目的地
是
停　車
下　車
結　束

**說明：**

(1) 任何的程式邏輯都有開始與結束。

(2) 此例中以上車作爲開始。

(3) 開車行駛碰到紅綠灯的機會最大，故爲行駛中的重點之一，如遇綠灯時車子可暢行無阻，如遇紅灯時則需暫停等候綠灯再行。

(4) 如未到目的地需繼續行駛，否則到達時即表示停車結束。

**書寫程式**

書寫程式是利用所規定或熟悉的程式語言遵循其規定的方法，依流程圖的邏輯步驟和處理方法，逐一書寫在程式用表上。書寫完成後，可打在卡片或錄上磁帶或在端末機上直接輸入磁碟內，然後由電腦執行處理。程式卡片或磁帶需經電腦轉錄輸入磁碟，因爲程式儲存在磁碟內在作業上較爲方便、經濟。

**測試與除錯**

程式產生後，一般情形不能立刻使用，因爲書寫程式時難免發生錯誤，當電腦在編譯(Compile or Interprete)程式時會自動指出程式語法上的錯誤，所以須先除錯(Debug)，予改正後，再加以測試(Test)，一直到其產生的結果與預期的結果完全符合爲止。

程式發生錯誤的原因可歸納爲以下幾點：

1. 工作上的疏忽，例如書寫不合的語法(Syntax)，登打時的錯誤。
2. 邏輯步驟的錯誤，可能分析不完全或處理的方法不當。
3. 未能十分了解程式規範而有所誤解。

測試時可用假設情況的資料測試，然後再以眞實的資料測試，以符合其正確性。

**編寫程式說明書**（Documentation）

程式設計人員應於測試完畢後，編寫程式說明書，其作用如下：

1. 證明程式已能正式作業，即符合程式規範的要求。
2. 指示電腦操作人員如何使用和控制程式的運轉。
3. 使用者易了解程式的性質。
4. 便於將來的維護及異動交接。

程式說明書的標準如下：

1. 程式規範（如前述）包括輸出輸入檔案的佈置等。
2. 整個程式的流程圖。
3. 原始程式的列表（LIST）。
4. 列出測試資料與結果（報表等）。
5. 其他補充說明，如規範之修改單等。