

高等学校试用教材

COBOL 程序设计

管理信息系统课程教材之二

哈尔滨工业大学 黄梯云 主编



GAO DENG XUE
XIAO JIAO CAI

机械工业出版社



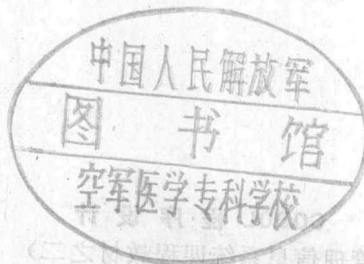
空军医专610 2 0054305 2

高等学校试用教材

COBOL 程序设计

管理信息系统课程教材之二

哈尔滨工业大学 黄梯云 主编



58563



机械工业出版社

本书内容包括:COBOL程序概述、COBOL数据部(Ⅰ)、COBOL过程部(Ⅰ)、COBOL结构描述及标识部和设备部、COBOL数据部(Ⅱ)、COBOL过程部(Ⅱ)、表处理、子程序和排序、COBOL的文件处理及结构COBOL程序设计。

COBOL 程 序 设 计

(管理信息系统课程教材之二)

哈尔滨工业大学 黄梯云 主编

*

机械工业出版社出版(北京阜成门外百万庄南里一号)

(北京市书刊出版业营业许可证出字第117号)

机械工业出版社印刷厂印刷

新华书店北京发行所发行·新华书店经售

*

开本 $787 \times 1092 \frac{1}{16}$ · 印张 $9\frac{3}{4}$ · 字数 232 千字

1986年5月北京第一版·1987年6月北京第二次印刷

印数 13,001—18,500 · 定价 1.70元

*

统一书号: 15033·6151

前 言

本书是根据 1983 年 4 月在杭州召开的机械工业管理工程类专业教材编审委员会第二次全体会议制定的教学计划和《管理信息系统》课程教学大纲编写的。管理信息系统课程教材分《管理信息系统导论(管理信息系统课程教材之一)》和《COBOL 程序设计(管理信息系统课程教材之二)》两册编写。

COBOL 语言是一种具有强大数据处理能力的计算机程序设计语言。目前,在国际上,已广泛地应用于商业及经营管理领域。本书的目的是使学员初步掌握 COBOL 程序设计方法,为把电子计算机这种先进工具应用于管理领域准备好一种基本的技能。

《COBOL 程序设计》共分十章,包括 COBOL 程序概述、COBOL 数据部(I)、COBOL 过程部(I)、COBOL 结构描述及标识部和设备部、COBOL 数据部(II)、COBOL 过程部(II)、表处理、子程序和排序、COBOL 的文件处理和结构 COBOL 程序设计。

本书可作为高等院校管理工程专业的试用教材,也可供企、事业管理干部、电子计算机应用软件人员等作为参考书。还可作为在职干部培训教材。

本书第一、二、三章由浙江大学王燮臣编写;第四、五、六、七、八、九、十章由西安交通大学周金琪编写。哈尔滨工业大学黄梯云任主编。清华大学薛华成任主审。

参加本书审稿会的有:华中工学院邓汉源、北京工业学院刘振兴、大连工学院龙连文、陕西机械学院李枫、吉林工业大学杨杰、北京化纤学院王美华等同志,并对本书提出了许多宝贵意见。机械工业部教材编辑室对编写本书给予大力帮助,谨在此一并表示感谢。

由于编者水平所限,编写时间仓促,书中难免有不当之处,敬请读者指正。

编者

1984年6月

目 录

第一章 COBOL 程序概述	1	§ 5-5 带有 66 和 88 层号的数据描述	56
§ 1-1 COBOL 程序的特点	1	第六章 COBOL 过程部 (II)	60
§ 1-2 COBOL 程序的组成	1	§ 6-1 ACCEPT 语句和 DISPLAY	
§ 1-3 成绩程序的标识部和设备部	4	语句	61
§ 1-4 成绩程序的数据部	6	§ 6-2 GOTO 语句	63
§ 1-5 成绩程序的过程部	8	§ 6-3 ALTER 语句	65
第二章 COBOL 数据部 (I)	13	§ 6-4 PERFORM 语句和 EXIT 语句	66
§ 2-1 编写程序的初步工作	13	§ 6-5 条件和 IF 语句	71
§ 2-2 COBOL 的数据部	15	第七章 表处理	76
§ 2-3 COBOL 中数据项的初始化和		§ 7-1 表处理功能	76
VALUE 子句	18	§ 7-2 表和定义一个表	77
§ 2-4 COBOL 的数值编辑	19	§ 7-3 表元素的查找	82
第三章 COBOL 过程部 (I)	21	§ 7-4 表处理实例	86
§ 3-1 过程部的输入/输出动词	21	第八章 子程序和排序	90
§ 3-2 过程部的算术运算与数据传送		§ 8-1 COBOL 子程序	91
动词	23	§ 8-2 COBOL 排序	96
§ 3-3 过程部 IF 语句	27	第九章 COBOL 的文件处理	104
§ 3-4 过程部的次序控制语句	28	§ 9-1 文件和文件种类	104
§ 3-5 用伪指令的方法编制学生		§ 9-2 顺序文件的处理	105
成绩程序	30	§ 9-3 索引顺序文件的处理	110
§ 3-6 COBOL 程序的书写格式、		第十章 结构 COBOL 程序设计	117
编译和运行	35	§ 10-1 结构化程序设计原理	117
第四章 COBOL 结构描述及标识部和		§ 10-2 结构化程序设计方法	119
设备部	38	§ 10-3 结构化 COBOL 的模块描述	122
§ 4-1 COBOL 程序的结构	38	§ 10-4 结构化 COBOL 程序实例	124
§ 4-2 标识部和设备部	38	附录一 ANSI COBOL X3.23-1968	129
第五章 COBOL 数据部 (II)	43	附录二 ANSI COBOL 保留字表	130
§ 5-1 数据的表达方式	43	附录三 ANSI COBOL X3.23-1974	
§ 5-2 COBOL 的数据组织	45	的语言格式表	134
§ 5-3 PICTURE (字形) 子句	48	参考文献	149
§ 5-4 USAGE (用法) 子句	53		

第一章 COBOL 程序概述

§ 1-1 COBOL 程序的特点

COBOL, 全称 Common Business Oriented Language (面向管理的通用语言), 是一种最广泛地应用于企业经营管理领域的计算机程序设计语言。目前, 世界上计算机语言的发展日新月异, 各国科学研究人员研制了很多功能强大、使用方便的新的计算机语言, 如 PASCAL, C, PL/1 等语言。但迄今为止, COBOL 语言仍不失为应用最广泛的语言之一。

COBOL 最早的文本是 COBOL-60。1968 年美国标准化协会 (ANSI) 通过了 ANSI COBOL-68 文本。此后, 国际标准化组织 (ISO) 在 1972 年通过了 ISO COBOL-72 文本, 1978 年通过了 ISO COBOL-78 文本。近年来又出版了新的版本, 不断扩大其功能。

COBOL 最主要的特点是具有强大的数据处理能力。正因为如此, 它特别适用于商业、银行、服务行业、运输、工矿企业、政府机关等管理领域。

COBOL 是一种类似英语的程序设计语言。写一个程序如同写一篇用简单英文句子拼成的文章。因而这种语言容易看懂, 不必多作解释, 只要学过一点英语的人就能很快掌握它。

COBOL 也有缺点, 它的语句比较冗长、繁琐, 即使很简单的程序, 写起来也要面面俱到, 并且必须按严格规定的格式书写程序。COBOL 的另一缺点是科学计算能力较差。

COBOL 内容繁多, 要求教学过程由浅入深, 从最基本的语句着手, 逐步掌握较深入的内容, 因此, 本书把教学内容分为两大部分。一是基础部分, 它通一个具体的程序实例, 讲述 COBOL 最基本的构成和语句, 以便初步掌握 COBOL 程序设计的方法。第二部分则充实更多的语句, 使读者能够系统地掌握 COBOL 语言编写程序的技术。

学习 COBOL 语言要注意理论联系实际。可以说, 看许多条有关语句的具体规定, 还不如弄懂一个程序实例更能掌握要领。只有亲自编写程序, 并通过计算机这个最严格的老师考试上机通过程序, 才能真正学好 COBOL 程序设计。

§ 1-2 COBOL 程序的组成

每个 COBOL 程序由以下四个部 (DIVISION) 组成:

标识部 IDENTIFICATION DIVISION.

设备部 ENVIRONMENT DIVISION.

数据部 DATA DIVISION.

过程部 PROCEDURE DIVISION.

这四个部必须按上面的次序相继出现在程序中。各个部都有其特有的功能。

标识部 用于对程序命名。此外还可写一些备注性的内容, 如作者的姓名、编写与编译程序的日期等。

设备部 是程序中描述计算机硬件的部分。它指出使用的计算机主机和外围设备。

数据部 这是一个用于描述程序的数据结构。这些数据包括输入、输出及中间数据。

过程部 是最重要的一个部，它的功能是对数据进行处理及计算。

部以下可分成几个节 (Section)，但也可以不分节。部和节由段 (Paragraph) 组成。段由句子 (Sentence) 组成。句子由一个或几个语句 (Statement) 组成，一个句子以句 号结束。句子和语句由字 (Word) 组成。最后，字由字符 (Character) 组成。

COBOL 程序的组成 结构可概略地用图 1-1 表示。

表 1-1 是 COBOL 各组成部分的示例。

字是组成 COBOL 的基本单元。COBOL 的字由字母、数字与连字符组成。COBOL 字需符合的规定是：字符的个数最多不超过 30 个；至少有一个字母；连字符不能出现在字符组合的头与尾；字与字之间要用空格隔开。

COBOL 有两类字，保留字与用户字。保留字是具有专门语法意义的字，它要按 COBOL 语法规则来使用，不能作为用户字使用。例如保留字 ADD 只能出现在加法语句的开头位置。

COBOL 保留字又分为两类：一类叫基字；一类叫任选字。基字是在语句格式中必须书写的字，而任选字是在语句格式中可以省略的字。使用任选字的目的，仅是为了增加程序的可读性。

例如：读语句格式中：

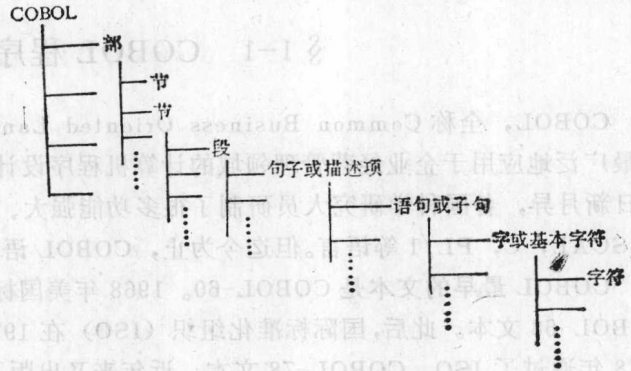


图1-1 COBOL的组成结构

表1-1 COBOL各组成部分示例

字	MOVE
语 句	MOVE 'Y' TO END-OF-FILE-FLAG
句 子	READ GRADE-CARD-FILE AT END MOVE 'Y' TO END-OF-FILE-FLAG.
段	READ-GRADE-RECORD. READ GRADE-CARDS-FILE AT END MOVE 'Y' TO END-OF-FILE-FLAG.
节	CONFIGURATION SECTION. SOURCE-COMPUTER. ... OBJECT-COMPUTER. ... SPECIAL-NAMES. ...
部	DATA DIVISION, FILE SECTION. ... WORKING-STORAGE SECTION. ...

<u>READ</u>	文件 名	<u>RECORD</u>	[<u>INTO</u>	数据名]
	{ <u>AT</u>	<u>END</u>	强制语句	
	<u>INVALID</u>	KEY	强制语句	

其中 RECORD, AT, KEY 是任选字，书写时可以省略。其他的是基字。

用户字是程序员自己定义的字，只要符合 COBOL 字组成的规定即可。用户字主要用来命名程序中的数据名 (即变量名)、文件名、记录名以及过程部中的节名和段名，还可命名条件名 (后面有专门介绍)。下面用表 1-2 举例说明 COBOL 的字：

COBOL 保留字举例：

ADD、READ、SECTION、MOVE 及 NOT 等。

用户字举例：

表1-2 COBOL 字举例

正 确 的 COBOL 字	非 COBOL 字
WANG 1	WANG □ LIN(是两个COBOL字)
XY	X·Y
MIS	MIS.
X-12	1-12
12X-1	12X·1

WANG-PROGRAM01、READ-CARD、PAY、HOURS 及 WORKS 等

字符是组成语句的符号，也是构成程序的最小元素。COBOL 源程序与 COBOL 字都由字符组成。

COBOL 字符集由下面 51 个字符组成，它包括五类字符，见表 1-3。

表1-3 COBOL 字符集

数字与英文字母	算术运算符	标 点 符 号	关 系 运 算 符	货 币 号
0, 1, 2...9	+(加)	, (逗号)	>(大于)	\$
A, B, C...Z	-(减)	; (分号)	<(小于)	
	*(乘)	.(句号)	= (等于)	
	/(除)	.(引号)		
		((左括号)		
		) (右括号)		
		□(空字符)		

为了正确地描述 COBOL 语句格式，通常作如下习惯上的规定：

1. 凡保留字用大写英文字母，基字下面划一横线，无横线者为任选字。
2. 凡用汉字或用英文小字指明的语法单位是由程序员根据相应语法规则填写的语法单位。
3. 凡用方括号[]括起来的内容是任选的，其选用与否可由程序员决定。
4. 凡用花括号{}括起来的内容都是必须的。当有几个并列的语法成分出现时，应必选其中之一。
5. 省略号“……”表示其前的内容可以重复，若省略号跟在[]或{}后面，则表示该方括号或花括号中的整个说明项可以重复。
6. 格式中的逗号“，”和分号“；”都可以省略；如果写上它们，其后至少跟一个空格。为了强调该处要用空格，可以用一个空格符“□”表示。
7. 语法格式中的足码，只表示某同一语法单位的多次出现次数。

例：

<u>MOVE</u>	{ 标识符-1 文 字 }	TO 标识符-2 [标识符-3]……
-------------	------------------	--------------------

习 题

1. 下面名词中哪些是保留字？

PAYROLL LEFT WAIT WRITE RIGHT

TWENTY BRANCH MIDDLE ZERO TEST
 TRY CALL ANSWER OVERFLOW STOP
 GO CODE LOOK COMPUTE CALCULATE

2. 下列项目中有多少字?

PROCEDURE DIVISION.

INITIALIZATION.

OPEN INPUT TRANSACTION.

IF CONTROL-TEST-ON

THEN PERFORM MONTH-END.

3. 下列项目中有几个句子?

PERFORM READ-CARDS-ROUTINE UNTIL

EOF-FLAG IS EQUAL TO 'STOP'.

IF CUST-NUMBER IS EQUAL TO CUST-HOLD-AREA

THEN NEXT SENTENCE

ELSE MOVE DETAIL-LINE TO OUTPUT-AREA

PERFORM WRITE-ROUTINE.

PERFORM ACCUMULATE-ROUTINE.

4. 将下列COBOL术语从最小单元到最大单元按次序排列。

SECTION PARAGRAPH SENTENCE WORD

DIVISION STATEMENT

§ 1-3 成绩程序的标识部和设备部

学习 COBOL 的最好方法是具体编写程序,因此,下面以学生成绩程序为例介绍 COBOL 程序的最基本内容。

学生成绩程序要求打印出一张学生的成绩报表(参见表 1-4),其内容包括每个学生的学生号和成绩两项信息。报表的末行打印全班中 A 等成绩的总人数。图 1-2 是它的系统流程图。由图可知,该程序的输入是每个学生的一张输入卡片,卡片上穿的数据是学生号与该学生的成绩。

下面是成绩程序的第一个部,叫做标识部:

IDENTIFICATION DIVISION.

PROGRAM-ID. GRADEDS.

AUTHOR. K. DOLAN.

标识部中程序员首先给程序命名,其次可写上某些备注,如著者姓名、程序编写日期等。上述语句中 IDENTIFICATION DIVISION、PROGRAM-ID 与 AUTHOR 都是保留字,而 GRADES 与 K. DOLAN. 是由程序员确定的用户字。程序名不长于 8 个字符。(CROMEMCO 系统程序名不长于 6 个字符),但著者姓名的长度不限。

第二个部是设备部。

ENVIRONMENT DIVISION.

CONFIGURATION SECTION.

SOURCE-COMPUTER. IBM 370.

表1-4 成绩程序打印输出表

	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2
1																																
2											G	R	A	D	E	S																
3																																
4					X	X	X	X	X	X	X	X	X	X	X				X													
5																																
6					X	X	X	X	X	X	X	X	X	X	X				X													
7																																
8					X	X	X	X	X	X	X	X	X	X	X				X													
9																																
10					X	X	X	X	X	X	X	X	X	X	X				X													
11																																
12																																
13	N	U	M	B	E	R			O	F			A			G	R	A	D	E	S		I	S	Z	Z	9					
14																																

OBJECT-COMPUTER. IBM 370.

SPECIAL NAMES. C01 IS TOP-OF-PAGE.

设备部包括两节，配置节 COFIGURATION SECTION 与输入-输出节 INPUT-OUTPUT SECTION.

在配置节中描述了计算机的主机系统，如源计算机（编译源程序的计算机）是 IBM370，目标计算机（执行程序的计算机）也是 IBM370。配置节中有一个专用名段 SPECIAL NAMES，它起注释作用，如 C01 是该机器系统的一个功能名，它的功能是指示打印机在新的一页上打印，由于 C01 功能名不易记忆，因而在专用名段上把 C01 注释为 TOP-OF-PAGE 便于记忆。以后写程序时就可用 TOP-OF-PAGE 代替 C01。

在输入-输出节描述了程序的输入-输出文件。

INPUT-OUTPUT SECTION.

FILE-CONTROL.

SELECT GRADE-CARDS-FILE ASSIGN TO SYSIN.

SELECT GRADE-REPORT-FILE ASSIGN TO PRINTER.

本例的输入-输出节中有一个文件控制段 FILE-CONTROL，程序员用它定义程序中要使用的每个文件的名称，并指出与文件有关的输入-输出设备。

上述语句中文件名 GRADE-CARDS-FILE 与 GRADE-REPORT-FILE 是两个用户

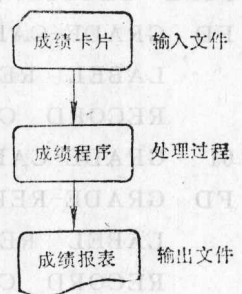


图1-2 成绩程序系统流程图

字。其他全是保留字。还需说明两个设备的名称 SYSIN 与 PRINTER。SYSIN 即系统的输入设备, PRINTER 即打印机。这两个字不是 COBOL 语法所规定的字,它是根据特定的计算机系统规定的该系统的外部设备名,如西门子 7738 系统的读卡机命名为 SYSIN;宽行打印机命名为 PRINTER。不同的机器系统有它特定的设备名称。编写程序时只要查阅机器的有关手册,就可确定输入-输出设备的名称。

习 题

1. 设备部的哪一节识别计算机系统?
2. 设备部的哪一节命名输入与输出文件?
3. 下列 COBOL 项目中哪些是文件名?

```
SELECT ACCTS-RECEIVEABLE-MAST ASSIGN TO DA-2314-S-MAST.
SELECT PRINT-REPORT ASSIGN TO UR-1403-S.
```

4. 写出一个 COBOL 段。用来命名一个名为 UPDATE 的程序。

§ 1-4 成绩程序的数据部

成绩程序中的数据,都必须在数据部定义和描述。COBOL 是用文件的形式来描述数据的。上述 GRADE-CARDS-FILE 是输入文件, GRADE-REPORT-FILE 是输出文件。

输入文件 GRADE-CARDS-FILE 由一叠成绩卡片上的数据组成。

由程序产生的打印输出也以文件的形式描述,称为打印输出文件,文件名即 GRADE-REPORT-FILE,每次打印一行,就是输出文件的一个记录。

成绩程序的数据部描述如下:

DATA DIVISION.

FILE SECTION.

FD GRADE-CARDS-FILE

LABEL RECORDS ARE OMITTED

RECORD CONTAINS 80 CHARACTERS.

01 GRADE-CARD-RECORD...

FD GRADE-REPORT-FILE

LABEL RECORDS ARE OMITTED

RECORD CONTAINS 133 CHARACTERS.

01 GRADE-REPORT-RECORD.

05 FILLER...

05 PRINT-LINE...

数据部的第一节是文件节 FILE SECTION。在设备部的 SELECT 语句中所命名的每一个文件必须在文件节中更完整地描述。字母 FD 是 FILE DESCRIPTION 的缩写,它标明了这是有关该文件的描述项。文件名紧接在 FD 的后面。LABEL RECORD 子句(子句是语句中的一个短句 Clause)指出文件的特定标记。如卡片文件的标记可以省略,写作

“LABEL RECORD IS OMITTED”，因为它不需要特定的标记，而对磁盘文件则应写上“LABEL RECORD IS STANDARD”。它说明了文件的标记是标准的。计算机会自动地确定文件的标记。

下一个子句是 RECORD CONTAINS 80 CHARACTERS。它指出记录的长度由80个字符组成。卡片的规格是80列，因而记录以80个字符为限。整个FD句子的结束处有一个句号。

FD描述体要求COBOL的编译程序在计算机的储存中留出一个储存空间，称为缓冲区(Buffer)。缓冲区用于计算机的输入与输出，也就是数据可以写入缓冲区，也可以从缓冲区读出。程序员往往频繁地将输入缓冲区的数据，复制到一个中间的工作区，然后进行有关的数据处理过程。经过处理的数据，再由工作区复制到输出缓冲区，然后通过输出设备将数据输出(参见图1-3)。

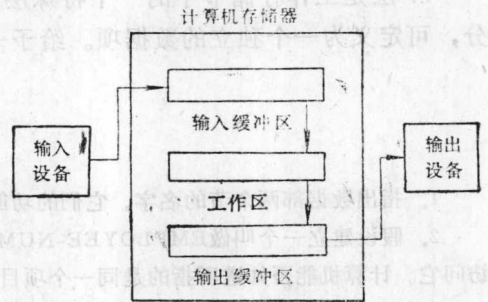


图1-3 计算机中数据的传送

RECORD CONTAINS 133 CHARACTERS子句指出每个打印记录有133个字符。其中第一个字符用于控制打印机走纸的信息。而其他132个字符是打印的字符。它们都应在输出文件中说明。

第二节是工作存储节 WORKING-STORAGE SECTION。

学生成绩程序的工作存储节举例如下：

WORKING-STORAGE SECTION.

77 NUMBER-OF-A-GRADES...

01 HEADER-LINE.

05 FILLER...

05 FILIER...

05 EILIER...

01 GRADE-LINE.

05 FILLER...

05 STUDENT-ID-OUT...

05 FILLER...

05 GRADE-OUT...

05 FILLER...

NUMBER-OF-A-GRADES是累加A成绩的数据项。在成绩处理中，凡是A等成绩都在该数据上累加。它是一个独立的数据项，不再细分。

HEADER-LINE设置了报表头中需打印的字符。

GRADE-LINE组成了一个输出报表的一行，包括学生号与学生成绩。

GRADE-CARD-WORK-AREA是一个工作区，当每个输入记录读入计算机进行处理之前，应把该记录复制到一个工作区，它共有三项，STUDENT-IN，GRADE-IN，与FILLER。FILLER是一个保留字，它在数据描述中作为一个不需专门命名的数据项，我们称之为充填项，它在程序中经常使用，特别是在工作储存节中经常使用。

01, 05 是数据项的层号。01 是记录的最高层, 它定义了一个数据集合, 即一个记录, 02 到 49 层号用于记录中所属的各数据项。

77 层是工作存储节中的一个特殊层号。在工作存储中某些无层次的项, 它们不需要细分, 可定义为一个独立的数据项。给予一个专门的层号, 77 层。

习 题

1. 指出数据部两个节的名字。它们的功能是什么?
2. 假设建立一个叫做EMPLOYEE-NUMBER的数据项, 并在过程部中以 EMPLOYEE-NO 的名来访问它。计算机能否知道所指的是同一个项目? 为什么是或为什么不是?
3. 给一个名为CHAGE-ACCT-TRANSACTIONS的卡片文件编写文件描述项(FD)。
4. 在下面COBOL项中, 哪些是文件名? 哪些是记录名? 哪些是数据名?

FD PAYROLL

LABEL RECORD ARE OMITTED

RECORD CONTAINS 80 CHARACTERS.

01 PAYROLL-INFO PIC X(80).

FD PRINTOUT

LABEL RECORD ARE OMITTED

RECORD CONTAINS 133 CHARACTERS

01 PRINT-LINE.

03 VERTICAL-CONTROL PIC 9.

03 OUTPUT-AREA PIC X(132).

§ 1-5 成绩程序的过程部

编制过程部的方法很多, 主要的方法有两种。一是绘制流程图的方法, 另一种是编写伪指令(Pseudocode)的方法。下面先介绍绘制程序流程图编写成绩程序过程部的方法。关于编写伪指令的方法将在后面第 § 3-5 中介绍。学生成绩程序的程序流程图如图 1-4 所示。

学生成绩程序的过程部分为五个段:

1. 起始段;
2. 打报表头段;
3. 处理段;
4. 总计段;
5. 工作结束段。

这五个程序段的内容和上述程序流程图是一致的。第一段是启动程序打开文件。第二段是打印报表的表头。第三段是处理一张输入卡片, 判别该学生的成绩是否 A 等, 若是 A 等, 则将其计数, 然后打印该学生的成绩。每处理一张输入卡片, 重复地执行该处理段的处理过程, 一直处理完所有的输入卡片为止。第四段是打印 A 成绩总数的打印行。第五段是任务结束段, 内容包括关闭文件和程序停止运行。

整个学生成绩程序的源程序如下,

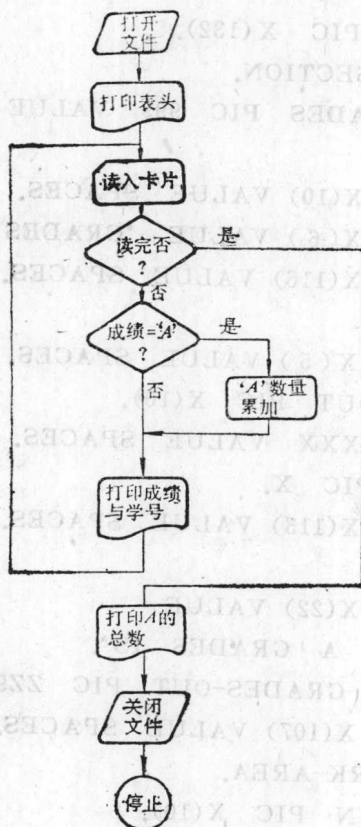


图1-4 学生成绩程序的程序流程图

IDENTIFICATION DIVISION.

PROGRAM-ID. GRADES.

ENVIRONMENT DIVISION.

INPUT-OUTPUT SECTION.

FILE-CONTROL.

SELECT GRADE-CARDS-FILE ASSIGN TO SYSIN.

SELECT GRADE-REPORT-FILE ASSIGN TO PRINTER.

DATA DIVISION.

FILE SECTION.

FD GRADE-CARDS-FILE LABEL RECORDS ARE OMITTED
RECORD CONTAINS 80 CHARACTERS.

01 GRADE-CARD-RECORD PIC X(80).

FD GRADE-REPORT-FILE LABEL RECORDS ARE OMITTED
RECORD CONTAINS 133 CHARACTERS.

01 GRADE-REPORT-RECORD.

05 FILLER PIC X.

05 PRINT-LINE PIC X(132).

WORKING-STORAGE SECTION.

77 NUMBER-OF-A-GRADES PIC 999 VALUE 0.

01 HEADER-LINE.

05 FILLER PIC X(10) VALUE SPACES.

05 FILLER PIC X(6) VALUE 'GRADES'.

05 FILLER PIC X(116) VALUE SPACES.

01 GRADE-LINE.

05 FILLER PIC X(5) VALUE SPACES.

05 STUDENT-ID-OUT PIC X(10).

05 FILLER PIC XXX VALUE SPACES.

05 GRADE-OUT PIC X.

05 FILLER PIC X(113) VALUE SPACES.

01 TOTAL-LINE.

05 FILLER PIC X(22) VALUE
'NUMBER OF A GRADES IS'.

05 NUMBER-OF-A-GRADES-OUT PIC ZZ9.

05 FILLER PIC X(107) VALUE SPACES.

01 GRADE-CARD-WORK-AREA.

05 STUDENT-ID-IN PIC X(10).

05 GRADE-IN PIC X.

05 FILLER PIC X(69).

PROCEDURE DIVISION.

A1. OPEN INPUT GRADE-CARDS-FILE.

OPEN OUTPUT GRADE-REPORT-FILE.

A2. MOVE HEADER-LINE TO PRINT-LINE.

WRITE GRADE-REPORT-RECORD AFTER
ADVANCING TOP-OF-PAGE.

PROCESS.

READ GRADE-RECORDS-FILE INTO GRADE-CARD-WORK-AREA
AT END GO TO TOTAL.

IF GRADE-IN EQUAL TO 'A'

ADD 1 TO NUMBER-OF-A-GRADES.

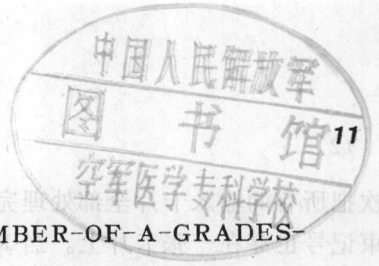
MOVE STUDENT-ID-IN TO STUDENT-ID-OUT.

MOVE GRADE-IN TO GRADE-OUT.

MOVE GRADE-LINE TO PRINT-LINE.

WRITE GRADE-REPORT-RECORD AFTER

ADVANCING 2 LINES GO TO PROCESS.



TOTAL.

MOVE NUMBER-OF-A-GRADES TO NUMBER-OF-A-GRADES-
OUT.

MOVE TOTAL-LINE TO PRINT-LINE.

WRITE GRADE-REPORT-RECORD AFTER
ADVANCING 3 LINES.

JOB-END.

CLOSE GRADE-CARDS-FILE GRADE-REPORT-FILE.

STOP RUN.

现在对照这个源程序，对过程部作概要的说明：

第一段，起动段是二个 OPEN 语句，它分别打开了 GRADE-CARDS-FILE 与 GRADE-REPORT-FILE 两个文件。

第二段，打印报表头段。第一语句是 MOVE 语句。它将已按排好的数据项 HEADER-LINE 传送到另一个数据项 PRINT-LINE 中，这样就形成了打印报表头的行。下一句是 WRITE 语句，意为在打印机上打印一个打印的记录，该记录名是 GRADE-REPORT-RECORD。AFTER ADVANCING TOP-OF-PAGE. 即要求输出报表从一张打印纸的页头开始打印，也就是另起一页打印。

第三段处理输入数据的程序段。首先是 READ 语句。它要求计算机从输入文件中取得一个记录。这是读入一张卡片记录，把它写在计算机内存的输入区。INTO GRADE-CARD-WORK-AREA 是由计算机将输入记录的数据复制到数据项 GRADE-CARD-WORK-AREA 上。AT END 是由计算机去判断被读的记录，是数据记录，还是文件的结束记号，如果是文件的结束记号，则执行 AT END 后面的语句 GO TO TOTAL，即要求程序的执行次序转移到名为 TOTAL 的程序段去。程序从该段的第一条语句开始往下运行。若 READ 语句读到的记录是一个数据记录，则执行 READ 语句的下一条语句，即 IF 语句。

IF GRADE-IN IS EQUAT TO 'A' THAN
ADD 1 TO NUMBER-OF-A-GRADES.

这是一个条件语句，它将数据项 GRADE-IN 的值与 A 作比较，GRADE-IN 的值是否等于 A，这就是条件语句的一个条件。若该条件为真则执行 THAN 以后的强制语句，若条件为假，则执行 IF 语句后的下一条语句。

上述语句中应注意有两项数据 1 与 'A'，1 与 'A' 都称为文字 (Literal)，1 是一个数值文字，而引号内的 'A' 是一个非数值文字。文字都具有值。数值文字是一个数值，而非数值文字则是一串在引号中的字符串。

后面是三条 MOVE 语句与一条 WRITE 语句。三条 MOVE 语句是将有关输入的数据传送到成绩报表记录中的有关数据项中去，并形成了一个学生成绩的打印行的数据。WRITE 语句是打印一个成绩报表的记录。AFTER ADVANCING 2 LINES, 意为打印时空一行打印一行。该段的最后是一个 GO TO PROCESS 语句。意为程序运行次序转到 PROCESS 段。即从 PROCESS 段的第一条语句开始运行。

通过 GO TO PROCESS 语句，该程序重复执行 PROCESS 程序段，也就是开始读第二张卡片，并重复执行全部 PROCESS 程序段的语句。这样重复地进行这个处理循环，依

次把所有的输入卡片全部处理完毕。最后程序将读到该输入卡片中的文件结束记号。文件结束记号也穿在一张卡片上。当 READ 语句读到文件结束记号时则执行 GO TO TOTAL 语句，程序的运行次序即转移到 TOTAL 程序段。

第四段 TOTAL 程序段是建立一个统计 A 成绩总人数的打印行。它也由两个 MOVE 语句建立打印行的数据，由 WRITE 语句在报表之末打印一行。至此整个程序要进行的工作已经做完，即打印出一张符合设计要求的学生成绩报表。

第五段 JOB-END 是一个程序结束段。程序最后要求把所有的文件关闭，并停止运行。这里使用了 CLOSE 与 STOP RUN 语句。