计算机基础教育丛书

COBOL 语 言（上册）

（修订版）

谭浩强 编著

清 华 大 学 出 版 社

(京)新登字 158 号

内 容 提 要

本书是作者在其编著的《COBOL 语言》一书的基础上修订补充而成。本书根据 ANSI COBOL 1974(即 ISO COBOL-78)标准的规定,介绍了 COBOL 语言及其程序设计。作者针对初学者在学习 COBOL 语言时所遇到的问题,对各部分内容作了合理的安排。本书的叙述通俗易懂,循序渐进,例题丰富,启发性强,便于初学者理解。

本书可作为高等学校和计算机学习班的教材,也可供计算机程序设计人员和企事业单位管理人员及其他初学者自学参考。

计算机基础教育丛书

COBOL 语 言(上册)

(修订版)

谭浩强 编著

☆

清华大学出版社出版

北京 清华园

通县宏飞印刷厂印刷

新华书店总店北京科技发行所发行

☆

开本:787× 1092 1/16 印张:13 字数:320 千字

1994 年 5 月第 2 版 1994 年 5 月第 1 次印刷

印数:0001—6000

ISBN 7-302-01434-5/TP·559

定价:9.80 元

计算机基础教育丛书

出　版　说　明

近年来，我国的计算机应用事业迅速发展，大批科技人员、大中学生、管理人员、以及各行各业的在职人员都迫切要求学习计算机知识，他们已经认识到，计算机知识是当代知识分子的知识结构中不可缺少的重要部分。

计算机应用人才的队伍由两部分人组成：一部分是从计算机专业毕业的计算机专门人才，他们是计算机应用人才队伍中的骨干力量；另一部分是各行各业中从事计算机应用的人才，他们既熟悉本专业的业务，又掌握计算机应用的技术，人数众多，是计算机应用人才队伍的基本力量。他们掌握计算机知识情况和应用计算机的能力在相当大程度上决定了我国计算机应用的水平。因此，在搞好计算机专业教育的同时，在广大非计算机专业中开展计算机基础教育是十分必要的。

非计算机专业中的计算机教学，无论就目的、内容、教学体系、教材、教学方法等各方面都与计算机专业有很大的不同，它以应用为目的，以应用为出发点。如果不注意这个特点，将会事倍功半。广大非计算机专业的师生、在职干部迫切希望有一套适合他们的教材，以便循序渐进地迈入计算机应用领域，并且不断地提高自己的水平。我们在前几年陆续编写了一些适合初学者使用的教材，受到广大群众的欢迎。许多读者勉励我们在此基础上进一步摸索和总结规律，为我国的广大非计算机专业人员编写一整套合适的教材。

近年来，全国许多专家、学者在这个领域作了有益的探索，写出了一批受到群众欢迎的计算机基础教育的教材。特别是全国高等学校计算机基础教育研究会作了大量的工作，在集思广益的基础上，提出了在高等学校的非计算机专业中进行计算机教育的四个层次的设想，受到广泛的注意和支持。我们认为：计算机的应用是分层次的，同样，计算机人才的培养也是分层次的；非计算机专业中各个领域的情况不同，也不能一律要求，在进行计算机教育时也应当有不同的层次。对于每一个学习计算机知识的人，还有一个由浅入深，逐步提高的过程。

我们认为，编辑出版一套全面而有层次的计算机基础教育的教材，目前不仅是十分必要的，而且是完全有条件的。在全国高等学校计算机基础教育研究会和许多同志的积极推动和清华大学出版社的大力支持下，我们决定编辑《计算机基础教育丛书》。它的对象是：高等学校非计算机专业的学生、计算机继续教育或培训班的学员、广大在职自学人员。

本丛书包括计算机科学技术的一些最基本的内容，例如计算机各种常用的高级语言、微机系统应用基础、计算机软件技术基础、计算机硬件技术基础、微型计算机的原理与应用、算法与数据结构、数据库基础、计算机辅助设计基础、微机网络与应用、系统分析与设计等，形成多层次的结构，读者可以根据需要与可能选学。

本丛书的宗旨是针对广大非计算机专业的需要和特点来组织教材，敢于破除框框，从实

际出发，用读者容易理解的体系和叙述方法，深入浅出、循序渐进地帮助读者更好地掌握课程的基本内容。希望我们的丛书能在这方面闯出自己的风格，在实践中接受检验。

本丛书的作者大多数是高等学校中有较丰富教学经验的教师。但是，由于计算机科学技术的飞速发展以及我们的水平有限，丛书肯定会存在许多不足，丛书的书目和内容也应当不断发展和更新。我们热情地希望得到社会各界和广大读者的批评指正。

主编　谭浩强　林定基　刘瑞挺

前　　言

用计算机进行事务管理,是计算机最重要的用途之一。COBOL 语言是国际上最广泛流行的用于数据处理的一种计算机高级语言,在商业、银行、财会、企业管理、行政事务等领域中得到广泛的应用。但由于 COBOL 语言的规定比较繁琐,很多初学者学习起来感到困难和枯燥,不少读者迫切希望有一本介绍 COBOL 语言的通俗而实用的入门书。为了推动计算机在管理方面的应用,作者于 1984 年编写了《COBOL 语言》一书,由清华大学出版社出版。在编写该书的过程中,作者深入研究了 COBOL 语言初学者的基础和认识规律,根据作者学习和讲授 COBOL 语言的体会,改革了传统的 COBOL 教材的写法,重新组织了易于初学者接受和理解的教材体系。该书出版后受到社会各界和广大读者的欢迎和好评,认为是介绍 COBOL 语言的较好的书,为初学者学习 COBOL 语言减少了许多障碍。该书出版后已重印多次,累计印数达 60 多万册。中央电视台还以该书为教材,组织了"COBOL 语言"电视讲座,由作者主讲。该书的出版对推动 COBOL 语言在我国的应用起了积极作用。

根据计算机技术的发展,根据读者的要求,作者对《COBOL 语言》一书进行了较大的修订补充,并改名为《COBOL 程序设计》,仍由清华大学出版社出版,以期抛砖引玉,促进计算机语言在我国的进一步推广普及。

本书的对象是 COBOL 语言的初学者。写法上力求通俗易懂,深入浅出,不从规则定义出发罗列一系列的规定而使人望而生畏,而是通过大量具体实例来说明问题,然后给出基本的语法规则。

COBOL 程序包括四大部分(标识部、环境部、数据部、过程部),每一部分又有许多规定,有的初学者往往分不清其主次和它们之间的关系,感到头绪较乱,无从入手。作者认为,这四个部分中的核心部分是过程部。本书的写法是以过程部为主要线索,说明其它三个部分如何与过程部配合组成一个完整的 COBOL 程序。

COBOL 的语法规则很多,特别是"可选项"很多,如果都一一介绍,将会使本书十分繁杂而重点不突出。考虑到本书是一本入门教科书而不是 COBOL 使用手册,作者认为,略去那些次要的、繁琐的"可选项"对初学者可能是适宜的。教科书应该用读者最容易理解的方式使读者建立起本课程的基本概念和掌握程序设计的一般技巧,为今后进一步学习、使用和提高打下基础。而手册的任务是给出一个包罗万象的规定和说明,以备查阅,它的对象是对本门课程已有一定知识的读者。

为帮助一部分从未接触过计算机的读者在学习计算机语言时不致发生困难,在"绪论"一章中介绍了关于计算机的一般知识。同时,考虑到不同程序读者的需要,以及考虑到初学者今后的提高,本书也包括了一些较深入而实用的内容,例如:表处理的位标(足标)引用方法和 SEARCH 语句、COPY 语句以及 USE 语句等,对磁带文件和磁盘文件亦作了较详细的介绍。在目录中打"*"符号的是初学者第一次学习时可以不必选学的部分,需要时再选学。

在每章中都有程度适当的例题,帮助读者更好地理解和运用所学的知识,使读者不致感

到枯燥。即使从未接触过计算机的读者,也能通过学习本书掌握 COBOL 语言的基本内容,并编写出一般的 COBOL 程序。

为了使读者对 COBOL 语言的全部内容有一个粗略的了解,在本书的附录中印出了 ANSI COBOL 1974(即 ISO COBOL-78)语法规则表。但读者开始时不需要对它的每一部分内容都搞得很清楚。随着编制程序技巧的逐步提高,不断地扩充自己的知识范围。为使初学者编制简单的程序时查阅语法格式的方便,本书附录中有一个"简单的 COBOL 程序语法格式索引"。

关于 COBOL 语言中术语的译名,本书尽量采用使读者易于理解而含义确切的译法。例如,Working-storage section 译为"工作单元节",表处理中的 Index 译为"位标",Mnemonic-name 为"助忆名"等等。同时也照顾到已广泛流行的一些译法,并在文中注明英文原文和国内不同的译法,以便读者在阅读其它书籍时不致发生困难。

由于实用的 COBOL 源程序一般都比较长,因此本书只能介绍一些最基本的 COBOL 例题。我们对这些例题都作了比较详细的分析。有了这个基础,读者不难看懂或编写较复杂的程序。

清华大学出版社出版的《COBOL 语言习题和例题集》(谭浩强主编)一书提供了本书习题的参考解答,并给出三十多个程序例题(包括 COBOL"报表生成"功能的使用),可作为学习本书时的参考。

最后,作者还想说明:COBOL 语言是一种很有实用价值的计算机高级语言,它特别适合于数据处理,它的报表生成和输出的功能丰富,是其它高级语言难以比拟的。用惯了 COBOL 语言的同志都感到 COBOL 语言功能强,十分好用。有人以为有了 dBASE 等小型数据库系统以后,COBOL 语言就没有用了。这是一种误解。用数据库组织数据,用 COBOL 程序去调用数据和按人们要求的格式来输出数据,是十分有效的。事实上,COBOL 语言仍然是有生命力的。

不久前,国外已推出 COBOL 语言的最新版本 COBOL 85。但目前国内还未普遍使用 COBOL 85,在一些计算机系统上还未提供 COBOL 85 的编译系统。因此本书仍根据 ISO COBOL-78 标准介绍 COBOL 程序设计。有了这个基础,再学习和掌握 COBOL 85 是不困难的。

在本书的修订工作中,高志强同志帮助我对本书进行全面的整理和修改,并增写了部分内容。

由于作者的水平和经验有限,本书无疑存在不少缺点甚至错误,恳切地希望从事 COBOL 语言教学和实际工作的同志以及广大读者给予批评指正。

谭浩强

1993.5

第一章　COBOL 语言概述

§1.1　COBOL 语言的发展概况

COBOL 是 Common Business Oriented Language(通用商业语言)的缩写。实际上,COBOL 不仅是商业数据处理的理想语言,而且广泛应用于数据管理领域,例如财会工作、统计报表、计划编制、情报检索、人事管理等。因此 COBOL 语言也被称为"用于管理的语言"。

在计算机的应用领域中,数据处理(data processing)是应用最广泛的一个领域。数据处理的日益广泛应用要求人们设计出能满足实际数据处理工作中各种要求的一种计算机语言。COBOL 语言就是在这种形势下应运而生的。

1959 年 5 月,美国国防部召开了一个有政府机关、企业、计算机厂家代表参加的会议,各方面都认为有必要设计出一种数据处理专用的计算机语言。会上确定了常设机构,以研究这种语言。这个会议称为 CODASYL(Conference on Data Systems Languages),意为数据系统语言会议。1959 年 12 月提出了世界上第一个 COBOL 语言文本,次年 4 月由美国政府印刷局正式发表,因此称 COBOL-60。后来进一步扩充和完善,出现了 COBOL-61,扩展 COBOL-61。它们为后来的版本提供了基础。

1965 年美国出现了更完善的版本,即 COBOL-65,但直到 1968 年 8 月才由美国国家标准化协会 ANSI(American National Standard Institute)通过批准了这个语言的标准版本,作为各厂家的依据。这就是 ANSI COBOL X3 23-1968。1972 年国际标准化组织 ISO (International Standard Organization)决定把它作为 ISO COBOL-72 国际标准 COBOL 文本,该文本已为美、英、法、日、苏等 21 个会员国承认。

1974 年,美国 ANSI 对 COBOL-68 作了修改扩充,发表了 ANSI COBOL X3 .23-1974 文本。1978 年 ISO 宣布 ANSI COBOL X3 23-1974 作为国际标准文本,即 ISO COBOL-78。

标准版本的出现,为 COBOL 语言的推广应用创造了一个有利的条件。多年来各国计算机厂商都以 ISO COBOL 72 (即 ANSI COBOL 1968)或 ISO COBOL-78(即 ANSI COBOL 1974)作为设计软件的依据。几年前,美国又提出新的 COBOL 版本,即 COBOL 85。它在 ANSI COBOL 1974 的基础上,扩充了功能,适合于结构化程序设计。但目前国内大多数计算机系统还未配置 COBOL 85 编译系统,而仍普遍使用 ANSI COBOL 1974。因此本书的叙述仍然以 COBOL 1974 为基础,并在附录中给出 COBOL 85 的语法一览。掌握了 COBOL 1974 之后,再学习 COBOL 85 是不会有什么困难的。

应该说明,尽管 COBOL 标准化程度比较高,但各个计算机厂家在实现它时还是有一些差别的,在具体用计算机时应查阅该计算机系统的 COBOL 说明书。

§1.2　COBOL 语言的特点

COBOL 语言的主要特点有:

(一) 最适于数据处理领域。所谓数据处理是指对大量数据的收集、统计、分类和加工。例如企业管理、库存管理、报表统计、帐目计算、信息情报检索等方面的应用都属于数据处理。

数据处理的特点是：算术计算量少而逻辑处理多；输入输出量大；数据间存在着一定的逻辑关系(数据项间有清晰的层次关系，例如职工工资中包括应发工资、扣除部分、实发工资几部分，应发工资又包括基本工资、附加工资等)；大量的分类排序(如按年龄大小排名单，按受教育程度分类……)；对打印报表要求较高、多样化等等。

在企业(如银行、商业、工厂)和其它部门(如领导机关、业务管理部门)的管理工作中，一般并无很复杂的计算公式，不要求太高深的数学基础，但是处理数据的量很大。

COBOL 正是针对数据处理要求而设计的。COBOL 所处理的问题具有数据繁多而运算简单的特点。COBOL 中也有加、减、乘、除、乘方等运算以及表达式的概念，但这些不是 COBOL 的重点。它的主要功能是描述数据结构和分析处理大批量的数据。

COBOL 对数据的处理过程，与人工处理的过程是相似的，即与人们的思维过程比较接近，因此，一般的管理人员是比较容易理解和掌握 COBOL 语言的。

(二) COBOL 比较接近于自然语言(指的是英语)。COBOL 程序看起来很像一篇用英语写的文章。例如，用 ADD A TO B 来表示 A + B → B(A 加 B，结果放在 B 中)，用 MOVE C TO D 表示将变量 C 的值传送到变量 D 中。COBOL 大量采用普通英语词汇和句型，学过英语的人看 COBOL 程序感到通俗易懂。也就是说它的特点是：成文自明。看它的英文意思就可以大致懂得程序的含义。

(三) 通用性强，由于 COBOL 语言的标准化程度较高。不同厂家生产的计算机系统所提供的 COBOL，是 COBOL 标准的全集或一个子集。一个计算机上的 COBOL 程序向另一计算机系统上移植，是比较容易实现的。

(四) COBOL 的结构严谨，层次分明。每个程序分四大部分(称为部，division)，每个部下面又分为若干节(section)，节下面又分为若干段(paragraph)。每一部分都有固定的程式。这个特点使初学者比较容易通过摹仿别人程序中的有关部分，从而较快地写出自己的程序。

(五) COBOL 的缺点是比较繁琐。如同中国古代的八股文一样，程序无论大小简繁，一律都要写齐四大部分，对每个部进行必要的定义和说明。因此源程序显得比较冗长。

据国外统计，在大、中型计算机系统上运行 COBOL 程序所占用的计算机时间为全部机时一半以上，超过了任何一种其它语言，是目前世界上使用得最多的一种计算机语言。

§1.3 最简单的 COBOL 程序介绍

为了使初学者从一开始就了解 COBOL 源程序的格式以及它的组成，建立起一个整体的概念，我们在这一节中先介绍两个最简单的 COBOL 源程序。

【例 1.1】

1　　6	7	8　　12
		IDENTIFICATION DIVISION.(标识部)
		PROGRAM-ID. EXAM1.　　(程序标识段)

```
      ENVI    RONMENT                    DIVISION .(环境部)
      DATA                               DIVISION .(数据部)
      PROC    EDURE                      DIVISION .(过程部)
      S .     DISPLAY          'THIS IS A COBOL PROGRAM .'
              STOP     RUN .
```

这个程序的目的是使计算机在指定的外部设备(终端显示器或打印机)上显示(或打印)出“ THIS IS A COBOL PROGRAM。”这样一串字符,然后停止运行。这些操作是在“ 过程部 ”中指定的。程序倒数第二行的“ S ”是段名。在本例中过程部只包括一个段,即 S 段。在 S 段中有两个句子,每个句子以句点“ .”和空格结束。

【例 1 .2】 将 A 和 B 的值相加,其结果放在 B 中。

```
1     6  7  8                  12
      IDEN    TIFICATION       DIVISION .(标识部)
      PROG    RAM- ID .        EXAM2 .  (程序标识段)
      ENVI    RONMENT          DIVISION .(环境部)
      DATA                     DIVISION .(数据部)
      WORK    ING- STORAGE     SECTION . (工作单元节)
      77      PICTURE     IS   9(3) .   (对 A 进行描述)
      77      PICTURE     IS   9(3) .   (对 B 进行描述)
      PROC    EDURE            DIVISION .(过程部)
      S .     ACCEPT      A                (输入 A 的值)
              ACCEPT      B                (输入 B 的值)
              ADD         A    TO   B      (A + B→B)
              DISPLAY     A,   B .         (显示 A 和 B 的值)
              STOP     RUN .               (停止运行)
```

这个程序的名字叫“ EXAM2 ”。与例 1 .1 不同,在本例中数据部下面有一个 WORKING-STORAGE SECTION(工作单元节,或称工作存储节),用它来描述程序中用到的中间工作单元。今有两个数据项 A 和 B,用“ PICTURE IS 9(3) ”来说明(描述)A 和 B 的类型是数值型的,“ 9 ”代表数值型,“ (3) ”代表数据长度为三位,即 A 和 B 的值是三位整数(有关数据描述将在以后详述,在此只要求大体知道它们的作用即可)。在过程部中,只有一个 S 段。在 S 段中有两个句子。每个句子以句点和空格为结束标志。第一个句子中包含四个语句,每个语句完成一个特定的操作。ACCEPT A 和 ACCEPT B 是从指定的外部设备上先后接收两个数值给 A 和 B(指定的外部设备可以是控制台或终端的键盘,也可以是读卡机或软磁盘机,由具体的计算机系统规定)。例如,如果从键盘上打入“ 012 ”和“ 024 ”二个数值给计算机,则 A 的值为 12,B 的值为 24。第三个语句是加法语句,ADD A TO B 表示“ 将 A 的值加到 B 上面去 ”,即 A + B B,也就是 12 + 24 = 36,将 36 存放在 B 中,因此,B 的值为 36。第四

个语句为显示语句,显示出 A 和 B 的值,在指定的外部设备上输出 A 和 B 的值(12 和 36)。至此,第一个句子完了。第二个句子是 STOP RUN,停止程序运行。

对这两个例子,可以暂时“不求甚解”,即只要求大体上知道它们的意思即可,有关各部分将在下面各章中详细说明。

§1.4 COBOL 程序的结构

1.4.1 部

从上面两个例子中可以看到,每一个程序都应包括以下四大部分,每个部(Division)有自己的部名,每一个部的作用如下:

IDENTIFICATION DIVISION	(标识部)主要用来指定源程序名字,也可以写入其它用作备忘的某些信息(如日期、作者等)。
ENVIRONMENT DIVISION	(环境部)指出程序中用到的数据文件名与计算机系统的设备的对应关系,即把某一文件名与一外部设备联系起来。此外还指定目标程序中使用的专门控制方法及程序所用内存区的大小等。
DATA DIVISION	(数据部)程序中所用到的全部数据(包括输入输出的数据和中间数据)都应在数据部中说明它们的类型和所占内存的情况。
PROCEDURE DIVISION	(过程部)用来给出程序要执行的指令,使计算机产生相应的操作。例如进行运算或其它处理。

在以上四个部分中,只有过程部是执行部分。计算机的任何一个操作都是由过程部中的指令给出的。因此,过程部是整个程序的核心部分,由它决定程序的每一步操作。也就是说,程序所预定的功能主要是靠这部分来完成的。前面三个部分是对过程部中用到的各文件、数据项和程序执行时的环境等作必要的描述和声明。例如,在过程部中指定要进行 A + B 的操作,结果放在 B 中。则应在数据部中说明 A 和 B 是数值型的数据项,并说明它在计算机内存中的存储形式和所占内存的情况(例 1.2 中 A 和 B 在内存中各占三个字节。一个字节为八位(bit),这里“位”指的是二进制位)。在执行到过程部中加法语句“ADD A TO B”时,就从数据部中所定义的数据项 A 和 B(它们代表内存中一定的存储单元)中把数值取出来进行相加,把结果再送回内存中名为 B 的存储单元中。

从这两个程序中可以看出:一个程序中四大部分缺一不可。即使有的部(例如例 1.1 中环境部)的下面并无具体内容,也要与上“部头”(或称“部首”),如 ENVIRONMENT DIVISION。

1.4.2 节和段

除标识部以外,在每一个部的“部头”的下面,可以有若干个节(SECTION),每一个节以“节头”作标识。每一节下面又可包括若干段(PARAGRAPH)。每一个段都有自己的名字(即段名)。如上节例 1.2 中,数据部下面有一个节,WORKING-STORAGE SECTION(工作单元节)。在标识部下面不设节,直接设段,如上节例中,PROGRAM-ID。就是“程序标识段”的段头。在它的后面写上由程序设计者确定的程序名(如 EXAM1 或 EXAM2),以便与其它程序相区别。过程部下面可以设节,下面再设段,也可以直接设段。例 1.1 和例 1.2 中过程部下面都没有设节,它直接由一个段构成。段名是“S”。在一般简单的程序中,过程部内可以不设节,直接由段构成。只有复杂的程序才在过程部下设节,节下分段。环境部

(ENVIRONMENT DIVISION)和数据部(DATA DIVISION)下面是设节的。

程序结构可以示意如下:

程序:

```
IDENTIFICATION DIVISION .
  段
  …
  …
ENVIRONMENT DIVISION .
  节
    段
    …
  节
  …
DATA DIVISION .
  节
    描述体
    …
  …
  …
PROCEDURE DIVISION .
 (节)
  段
  …
  …
```

1.4.3 句子、语句和子句

在过程部中,每一段由若干个句子(Sentence)组成。一个句子是以句号加一个以上的空格来结束的。例如例 1.1 中 S 段由两个句子组成。第一个句子是:DISPLAY‘ THIS IS A COBOL PROGRAM .’.”,第二个句子是:“ STOP RUN .”。注意每个句子最后都有一个句点和一个以上的空格。

图 1.1

句子又由语句(Statement)组成。例 1.2 中 S 段由两个句子组成。其中第一个句子包括四个语句(占四行),在第四行的最后才有句点和空格(想要知道有几个句子只需数一下有几个后面跟空格的句点即可)。每一语句都是一条完整的指令,它们各自有一个相应的动词(Verb),表示该语句指定计算机要进行的操作,如 ACCEPT(从指定的设备上接收某些值),ADD(进行加法),DISPLAY(在指定的外部设备上显示信息)。

一个句子可以只由一个语句组成,如例 1.2 的第二个句子 STOP RUN ., 一个语句加一个句点后跟一个(多个)空格就成为一个句子。

在一个语句中又可以包含若干个子句(clause),每一子句也有一个动词(但这个动词往往是可以省略的),它指定某一方面特定的功能。

过程部中程序的结构如下:

```
部(Division)          一部可包括若干节
  节(Section)         一节可包括若干段
    段(Paragraph)     一段可包括若干句子
      句子(Sentence)  一句子可包括若干语句
        语句(Statement)指定计算机完成一定的操作
```

子句(Clause)指定完成某一方面的特定功能

除了过程部的语句可以包含子句外,在环境部和数据部中也可以出现子句,如 SELECT 子句,文件描述子句等。见 1.5.4 段。

1.4.4 描述体

在数据部中有若干节,每个节中有若干个描述体(Description entry,亦译作描述款目或描述款),每个描述体又由若干个子句构成。

例如,例 1.2 中数据部(DATA DIVISION)中的工作单元节(WORKING-STORAGE SECTION)下面,有二行(以 77 打头的),它们就是数据描述体,分别描述 A 和 B 的数据形式,说明 A 和 B 都是数值型的数据,而且是三位整数的数。以后我们还可以看到其它形式的描述体(如文件描述,记录描述等)。有关描述体的概念我们将在第四章中作详细介绍。

整个 COBOL 程序的结构可用图 1.2 表示:

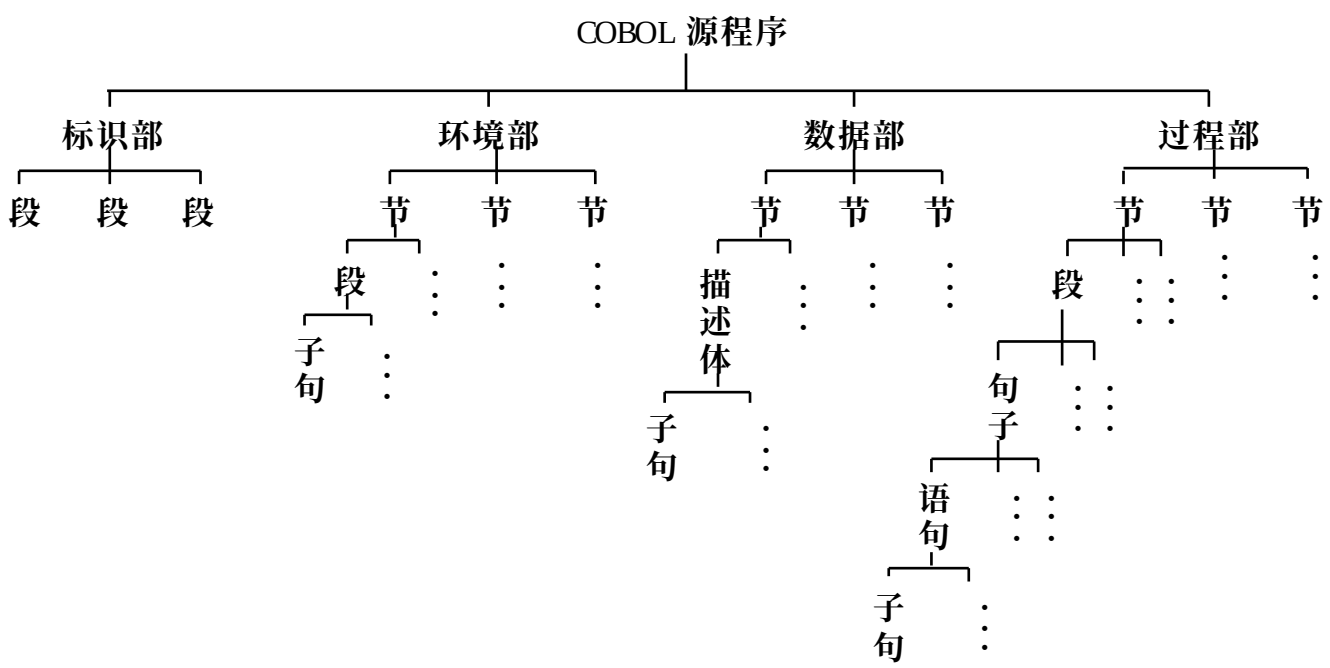

图 1.2

§1.5 COBOL 源程序的书写格式

COBOL 源程序必须严格地按规定的格式书写。其书写格式通常采用 ANSI 格式(American National Standard COBOL reference format)。有些计算机系统除了可以采用 ANSI 格式书写和输入源程序外,还可以采用终端格式(Terminal format)。如 VAX COBOL,在编译时使用不同命令以分别对这两种格式的源程序进行编译。

1.5.1 ANSI 书写格式

ANSI 源程序书写格式见图 1.3。

COBOL 程序纸每行有 80 列(每张程序纸包括多少行不作规定,一般为 2025 行)。每一行分为几个区:

标 号 区	续行区	A 区	B 区 (正文区)	注 解
1 6	7	8 11	12 72	73 80

图 1 3

1 .16 列,为‘ 标号区 ”。可以填写 6 个数字。标号由程序编写者自定,一般用前 3 列表示页号,后 3 列表示在本页中的行号。如用 001015 表示程序的第一页第 15 行。标号应按由小到大的顺序,但不一定连续。标号区内可以写标号也可以不写标号(标号区空白)。标号对源程序的执行结果没有任何影响。标号只是为程序人员查阅程序时方便而设的。例如在一个长的程序中要找某一行,显然按标号查找是比较方便的。在程序编译时是按程序书写的顺序进行的,而不是按标号大小顺序进行的。

2 .第 7 列,是‘ 续行标志区 ”。如果在第 7 列上写上连接符‘ - ”,则表示本行是紧接在上一行的后面的。如上一行中有一个字 MOVE 未写完,只写了 MO 二个字符,在下一行的第 7 列上写‘ - ”,然后在 12 列开始写 VE 二个字符,则表示 VE 是紧连在上一行 MO 之后的,等效于在上一行中写 MOVE,图 1 .4 中①和②两种写法是等效的。但如一个字已写完,本应空一格再写下一个字,因上一行写不下而要求在下一行续写,此时不必在第七列上写“ - ”,而是使第七列空白,则下一行 12 列后的内容自动连在上一行之后,中间插入一个空格。见图 1 .4③。

以上①②③④四种写法是等效的。第①种是将 VE 紧接在 MO 之后,二行连起来,组成

图 1 .4

MOVE WANG TO ZHANG。第③种是把 WANG 接在 MOVE 之后,二者间自动插入空格,MOVE 和 WANG 分别是两个字。第④种写法中由于在输入“ MO ”后就按回车键结束了本行,而在继续行的第七列中有‘ - ”,因此,将 MO 与 VE 紧连,成 MOVE。因此,只有在上下

二行“紧连”时才在下一行的第七列上加“-”。否则不要加“-”。但应尽可能避免将一个字拆成二行,以免产生错误。

如果有一个用引号括起来的字符串(如例 1.1 中 DISPLAY 语句中的‘THIS IS A COBOL PROGRAM.’),上一行没写完,需下一行继续时,应在续行的第 7 列上写上“-”,然后在 12 列后写继续的部分,并在其第一个字符前加一个引号。如图 1.5 中的①和②是等效的。

图 1.5

注意如果在上一行的最后一个字符后面有一个或多个空格,如图 1.5 的③中所示,在“PROGR”后面有多个空格,这个字符串中 PROGR 和 AM 之间就插入了许多个空格。因此③和①或②是不等效的。这是在使用字符串时所应注意的。

如果在第 7 列中不写“-”而写“*”,则表示此行是注解行,即此行可由程序员任意写上自己所需的内容,以对程序(或程序的一部分)作说明。它对程序的执行不起任何影响,仅起一个注释作用(只给程序员自己作备忘之用,或使别人能看懂此程序而作必要的说明)。计算机在编译时不理会此行。只是在打印源程序清单时把此行原样印出。

3. 第 8~11 列,称为“A 区”,第 8 列称“A 区边界”。COBOL 规定,程序中有些内容,例如部头,节头,段头,层号 01,层号 77 以及文件描述符 FD 等应从 A 区开始书写(有关这些内容将在后面叙述)。从 A 区开始写,并不一定要从第 8 列开始,也可以从第 9 列,或第 10 列、第 11 列开始写。

4. 12~72 列,称“B 区”。写程序中的正文部分,例如过程部中的句子只能从 B 区开始写,而不能写到 A 区去。哪些内容应从 B 区开始,以后会陆续介绍的。

5. 73~80 列,为“注释区”。程序员如想对源程序的某些行作些简单说明,可写在这 8 列中,例如编号,或注明其作用。计算机在编译源程序时不理会这 8 列。它对程序的执行不起作用。因此写源程序时注意不应超过第 72 列,超过的部分在编译时将被舍弃。

注意,以上对 80 列的划分区,只是对源程序来说的,如果程序纸上写的不是源程序,而是输入的数据,则不受以上分区的限制(输入数据时可以从第 1 列用到第 80 列)。这一点有的初学者常易搞混。我们只是在这里预先指出这一点。关于输入数据的方法将在以后介绍。

图 1.6 表示一个 COBOL 源程序的书写格式的例子。

关于这个程序将在第四章中解释。

图 1.6

在写程序时,应注意以下几点:

(1) 每个字符占程序纸中的一个格。

(2) 较早的 COBOL 版本规定所有字母都应大写,但现在使用的 COBOL 编译系统允许使用大写或小写字母,二者等价,但用引号括起来的字符串中的大写和小写是有区别的,二者不等价。

(3) 相邻的两个字(如 COBOL 的保留字或用户自己定义的名字)之间必须留一个以上的空格。

(4) 运算符(如加、减、乘、除、乘方)和等号左右两边必须各留一个空格。在过程部中左括号的左侧和右括号的右侧要留一空格,而内侧不必留空格。如:

```
A + (B + C) / D.
```

(5) 逗号、句号、分号的左边不能留空格,而右边应有空格。

(6)一个空格和多个空格作用相同,如:

```
MOVE A TO B
和 MOVE A    TO    B
和 MOVE A
  TO    B 等价。
```

但被引号括起来的字符串中的每一个空格都作为一个字符,是字符串中的一部分,不能任意变。

COBOL 允许一行内写几个语句(语句间应以一个或多个空格分隔),也允许一个语句写在多行上。

1.5.2 终端格式

为了方便用户书写和输入源程序,有的计算机系统允许使用终端格式。终端格式不设标号区,把续行标志区和 A 区结合在一起,除以下说明之外,它和 ANSI 格式使用规则一样。

(1) 终端格式的每一行可用到 256 列,即源程序每行最大长度为 256 个字符,在输入字符占满屏幕上一行(80 列)后不要按回车键,继续输入,直到输入完本行(最多 256 个字符)全部字符为止。

(2) A 区占用 14 列。第一列可以作为续行标志区。当第一列上写上连接符“ - ”,则表示本行是上一行的续行,而当第一列写“ * ”,表示此行是注释行。需要说明的是,当第一列作为续行标志区时,A 区将占用 25 列,而 B 区相应为第 6256 列。

(3) B 区占用第 5256 列。

注意虽然编译系统可以处理每行 256 个字符,但是在列源程序清单时只能列出前 125 个字符,而用“·”代替未列出的字符。

§1.6 COBOL 字符和 COBOL 字

1.6.1 COBOL 字符

每一种计算机系统都规定了所允许使用的字符。并非所用的字符都能在 COBOL 程序中使用。例如在 COBOL 中,就不能用数学上的乘号“ × ”和除号“ ÷ ”,而要用“ * ”和“/ ”代替。

有两种字符集：系统字符集和 COBOL 字符集。所谓系统字符集指的是在输入输出操作中允许出现的字符的集合。例如，有的计算机系统采用 ASCII 字符集，可以输出 ASCII 字符集中的多个字符。

COBOL 字符集指的是：在 COBOL 程序中允许出现的字符(用引号括起来的字符串中的字符除外)。每一种语言都规定了它自己的字符集，例如，BASIC 字符集、FORTRAN 字符集、PASCAL 字符集、C 字符集等，它们各不相同。例如在 BASIC 程序中要用到字符“ # ”、“ !”等，而在 COBOL 程序中却不使用这些字符(用引号括起来的字符串中使用的字符不在此例)。每一种语言的字符集都比“ 系统字符集 ”所包含的字符少。它是源程序中所用到的字符的最小集合。

COBOL 字符集包括以下字符。

数字：09

大写字母：AZ

小写字母：az(旧版本 COBOL 字符集不包括小写字母)

专用字符共 15 个：

+	加号
-	减号或连接号
*	乘号或星号
/	除号
=	等号
,	逗号
.	句号或小数点
;	分号
'	引号(有的用双引号，注意不分起始引号和终止引号，都用同一个字符)
(	左括号
)	右括号
<	小于号
>	大于号
	空格(即空白)。在本书中有时为说明某位置上有空格，以表示。
	货币符号(美元符号)

在具体实现时，有的 COBOL 编译系统在上述 COBOL 字符集的基础上作少量调整，例如将美元符号“ ”改成“ 元 ”的符号“ ¥ ”等。

“ # ”、“ !”、“ ?”、“ % ”等字符属于系统字符集而不属于 COBOL 字符集，这些字符只能在 COBOL 程序中的字符串(用引号括起来)中出现，而不能在程序中其它地方出现。

1.6.2 COBOL 字

COBOL 字是由 COBOL 字符组成的，如同英文字母组成英文单词一样。COBOL 字是为了表示一定的意思，由字符组合而成的最小单位。如前面已见到过的，DIVISION，SECTION，MOVE，ADD…等都是 COBOL 字。

COBOL 规定，每一个 COBOL 字不允许超过 30 个字符。

COBOL 字分为两类：保留字和用户字(非保留字)。保留字指在 COBOL 中已规定作专门用途的字，它们代表特定的含义。例如 MOVE，在 COBOL 中它代表“ 传送 ”这一特定含义。同样，ADD 代表进行“ 加 ”的操作。这些保留字不能另作它用。附录Ⅵ中列出 COBOL 的全部保留字。