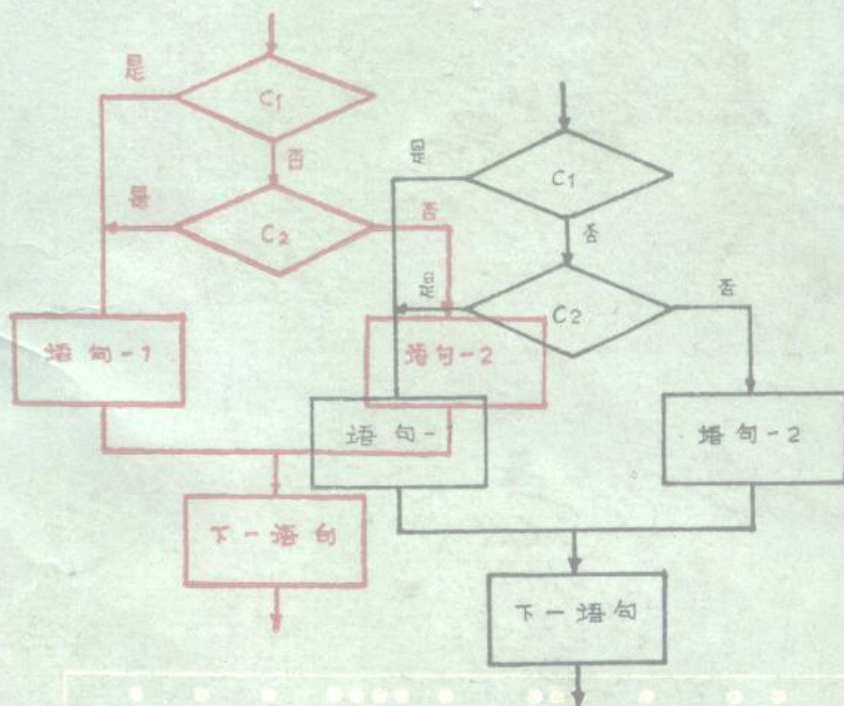


COBOL 程序设计初步

● 钱 树 人 史 九 林 等 编 著

● 人 民 邮 电 出 版 社



TP 312
Q 33

· 200650

COBOL程序设计初步

钱树人 史九林 等编著



人民邮电出版社

JS194/61

内 容 提 要

本书以通俗的语言,较系统介绍广泛应用于数据处理的数据语言。
内容包括 COBOL 语言的基本概念、基本功能、基本内容和基本使用方
法,涉及了核心、表处理、顺序 I-O、排序功能模块。最后简单介绍了结
构程序设计方法。

本书可供有一定计算机知识的科技人员、管理人员阅读。

COBOL 程序设计初步

钱树人 史九林 等 编著

人民邮电出版社出版
北京东长安街22号
河北省邮电印刷厂印刷
新华书店北京发行所发行
各地新华书店经售

开本: 787×1092 1/32 1984年7月第 一 版
印张: 11 16/32 页数: 184 1984年7月河北第一次印刷
字数: 262 千字 印数: 1—17,000 册

统一书号: 15045·总2861—无6279

定价: 1.50元

前 言

数据处理是计算机的重要应用领域。随着大容量存贮器等的发展和利用计算机进行数据处理的能力也越来越大。这对企业管理、情报检索、科技计算等各方面带来很大的影响。目前已经有许多部门采用计算机来处理日常的统计管理等工作。今后,随着软硬件的发展,终端的大量使用,网络的实现,计算机在数据处理领域中的应用必将更为广泛。

在数据处理用的程序设计语言中,最通用的是COBOL (Common Business Oriented Language) 语言。COBOL语言也在不断地发展,其功能仍在不断地扩大。它也是标准化工作开展得最早的一种语言。COBOL语言有许多特点,例如源程序类似于英语的表示形式,阅读较方便;功能比较强;对数据的描述和组织十分细致和详尽。但语法语义的描述不如某些程序设计语言精确和简洁,其语言文本比较冗长。

本书通俗地介绍COBOL语言。内容包括COBOL语言中的基本概念、基本功能、基本内容和基本的使用方法。主要涉及其核心,表处理,顺序I-O,排序等功能模块。这几个部分也是数据处理中必须的和常用的功能。处理比较简单的问题基本上是够用的。通过对这些功能的理解、使用,加上一定时间的上机实践,初学者就能初步掌握COBOL语言的一般的程序编制技术。但是要编制一个高质量的COBOL程序和迅速正确地解决数据处理问题,当然需要了解更多的内容,掌握更多的技巧。为帮助读者进行整个算法的构思和编出正确且结构清晰的

源程序，对程序设计方法学方面的“结构程序设计”作了简单介绍。

全书共分十一章和两个附录。第一章为绪论，第二章介绍基本概念，基本元素，标识部和环境部；第三、四、七章介绍数据部；第五、八章介绍过程部；第六章通过一个例子来说明用COBOL语言解决数据处理问题的过程；第九章简单介绍排序；第十章介绍表处理；第十一章简单介绍结构程序设计。学完前六章，希望使读者具有初步编制COBOL程序的能力，如果结合简单题目上机实践将更为有益。学习后五章的内容，可使读者对COBOL语言有进一步的了解，能编出较好的程序。有些COBOL语言的功能本书并未涉及，如果想了解更深的內容，可参阅其他有关的书籍和资料。

本书由南京大学计算机科学系数据处理小组的钱树人、史九林、承子才、宋满英、陈本林、朱树春、潘金贵等同志集体编写。其中，第一、二、六章由钱树人同志编写，第三、四、七、十章由史九林同志编写，第五章由钱树人、承子才、宋满英、陈本林同志编写，第八章由钱树人、承子才同志编写，第九章由朱树春同志编写，第十一章由潘金贵同志编写。全书由钱树人、史九林同志审订。

在此书编写过程中得到了有关单位和领导的支持。特别是南京大学计算机科学系、软件研究所及有关教研室的领导和同志们的支持和指导。陈胜凡同志审阅了本书的初稿，并提出宝贵意见；修订稿又经萨师煊、刘惠芳同志审阅和修改。在此一并表示感谢。

作 者

目 录

第一章 结 论	(1)
第一节 数据处理和计算机	(1)
1.1.1 数据处理	(1)
1.1.2 电子计算机的数据加工	(3)
1.1.3 COBOL语言的发展历史	(5)
1.1.4 国际标准COBOL的结构	(6)
第二节 计算机	(10)
1.2.1 计算机和程序设计语言	(10)
1.2.2 原始输入介质	(15)
1.2.3 常用的打印输出介质	(21)
1.2.4 磁性存贮介质	(23)
第三节 程序设计语言的一些基本概念	(24)
1.3.1 基本字符, 基本字符集, 基本符号	(24)
1.3.2 源程序, 目标程序, 编译程序	(25)
1.3.3 几个编译概念	(28)
第二章 COBOL程序及其基本元素	(30)
第一节 COBOL程序	(30)
2.1.1 执行流程	(30)
2.1.2 COBOL样本程序	(32)
第二节 COBOL语言的基本元素	(36)
2.2.1 基本字符	(36)
2.2.2 基本符号	(37)
2.2.3 用户定义名	(38)
2.2.4 字值	(39)
2.2.5 分隔符号	(39)

第三节	COBOL语言的描述	(40)
第四节	标识部	(43)
第五节	环境部	(44)
2.5.1	配置节	(45)
2.5.2	输入输出节	(47)
第三章	数据和数据的组织	(49)
第一节	数据	(49)
第二节	数据的表示法	(51)
3.2.1	变量	(51)
3.2.2	常量	(53)
第三节	数据在计算机内的表示	(59)
第四节	数据的组织	(62)
3.4.1	数据的层次结构	(63)
3.4.2	文件	(67)
3.4.3	顺序文件及批量处理	(72)
第四章	简单的数据描述——数据部(I)	(77)
第一节	数据部的组成	(77)
第二节	文件描述	(79)
4.2.1	块长子句(BLOCK)	(80)
4.2.2	记录长子句(RECORD CONTAINS)	(82)
4.2.3	数据记录子句(DATA RECORD)	(84)
4.2.4	标号子句(LABEL RECORD)	(84)
4.2.5	标号值子句(VALUE OF)	(85)
第三节	记录描述	(87)
4.3.1	层号及数据名	(90)
4.3.2	形象子句(PICTURE)	(92)
第四节	工作存区的指定	(101)
4.4.1	独立项描述	(102)

4.4.2	初值子句 (<i>VALUE</i>)	(102)
第五章	过程部 (I)	(106)
第一节	概述	(106)
5.1.1	节、段、句子、语句	(106)
5.1.2	语句的类别	(110)
5.1.3	例子	(111)
5.1.4	标识符, 受限	(113)
第二节	传送语句 (<i>MOVE</i> 语句)	(118)
5.2.1	传送语句的简单形式	(118)
5.2.2	发送项和接收项的匹配及其传送规则	(119)
5.2.3	对应操作的传送语句	(127)
5.2.4	具有多重结果的传送语句	(130)
第三节	算术运算语句	(131)
5.3.1	加语句 (<i>ADD</i> 语句)	(132)
5.3.2	减语句 (<i>SUBTRACT</i> 语句)	(134)
5.3.3	乘语句 (<i>MULTIPLY</i> 语句)	(135)
5.3.4	除语句 (<i>DIVIDE</i> 语句)	(137)
5.3.5	截断和舍入	(138)
5.3.6	溢出的检测	(140)
5.3.7	计算语句 (<i>COMPUTE</i> 语句)	(141)
5.3.8	算术表达式的求值	(143)
5.3.9	算术语句中的对应操作	(145)
5.3.10	多重结果的算术语句	(147)
第四节	条件和如果语句 (<i>IF</i> 语句)	(148)
5.4.1	如果语句的一般格式	(149)
5.4.2	条件	(151)
5.4.3	条件语句的嵌套	(161)
5.4.4	<i>IF</i> 语句中的条件的缩写	(164)
第五节	顺序控制语句	(167)

5.5.1	转向语句	(167)
5.5.2	更改语句 (ALTER语句)	(169)
5.5.3	停语句 (STOP语句)	(171)
5.5.4	简单的执行语句 (PERFORM语句)	(172)
5.5.5	出口语句 (EXIT语句)	(180)
第六节	输入输出语句	(181)
5.6.1	输入输出和文件	(181)
5.6.2	打开语句 (OPEN语句)	(183)
5.6.3	关闭语句 (CLOSE语句)	(185)
5.6.4	读语句 (READ语句)	(185)
5.6.5	写语句 (WRITE语句)	(188)
第六章	COBOL程序的编制	(194)
第一节	框图	(194)
第二节	应用举例	(199)
6.2.1	例子及算法	(200)
6.2.2	源程序	(201)
6.2.3	输入数据和打印结果	(207)
6.2.4	错误信息的处理和操作命令	(209)
第七章	复杂的数据描述——数据部 (I)	(213)
第一节	更完整的数据描述	(213)
第二节	数据的编辑	(214)
7.2.1	形象子句的编辑功能	(215)
7.2.2	零置空子句 (BLANK) 的编辑功能	(230)
7.2.3	对数据进行编辑的几种方法	(231)
第三节	数据形式及存贮定位	(232)
7.3.1	几种数据形式	(233)
7.3.2	数据的存贮定位	(240)
第四节	数据的对界	(244)
7.4.1	实现对界的办法	(245)

7.4.2	对界子句 (<i>SYNCHRONIZED</i>)	(247)
第五节	内存区的重叠使用	(249)
7.5.1	记录域的重叠使用	(250)
7.5.2	项域的重叠-重定义子句 (<i>REDEFINES</i>)	(251)
第八章	过程部 (<i>I</i>)	(256)
第一节	执行语句	(256)
第二节	待收语句和显示语句	(265)
8.2.1	待收语句 (<i>ACCEPT</i> 语句)	(266)
8.2.2	显示语句 (<i>DISPLAY</i> 语句)	(268)
第三节	字符串运算语句	(270)
8.3.1	检测语句 (<i>INSPECT</i> 语句)	(271)
8.3.2	并串语句 (<i>STRING</i> 语句)	(276)
8.3.3	拆串语句 (<i>UNSTRING</i> 语句)	(279)
第九章	排序	(283)
第一节	概述	(283)
9.1.1	什么叫排序	(283)
9.1.2	排序程序的组织	(284)
9.1.3	输入/输出过程	(288)
9.1.4	排序的操作过程	(288)
第二节	排序的语言成分	(290)
9.2.1	环境部	(291)
9.2.2	数据部 (<i>DATA DIVISION</i>)	(293)
9.2.3	过程部 (<i>PROCEDURE DIVISION</i>)	(293)
9.2.4	释放语句 (<i>RELEASE</i> 语句)	(297)
9.2.5	回收语句 (<i>RETURN</i> 语句)	(298)
第十章	表处理	(299)
第一节	什么是表	(299)
第二节	表的描述	(301)

10.2.1	出现子句 (<i>OCCURS</i>)	(302)
10.2.2	表的维	(304)
10.2.3	表的种类	(306)
第三节	表元的引用	(309)
10.3.1	下标法	(309)
10.3.2	位标法	(312)
第四节	表的检索	(319)
10.4.1	检索语句 (<i>SEARCH</i> 语句) 的检索功能	(319)
10.4.2	执行语句 (<i>PERFORM</i> 语句) 的检索功能	(327)
第十一章	COBOL 结构程序设计	(330)
第一节	引言	(330)
第二节	结构程序设计的控制结构及其在 <i>COBOL</i> 语言中的实现	(331)
11.2.1	结构程序设计使用的三种基本控制结构	(331)
11.2.2	基本控制结构在 <i>COBOL</i> 语言中的实现	(333)
第三节	结构化 <i>COBOL</i> 程序的组织 and 例子	(339)
附录 1	七单位字符编码表	(349)
附录 2	<i>ISO COBOL 78</i> 保留字表	(351)

第一章 绪 论

第一节 数据处理和计算机

1.1.1 数 据 处 理

自从人们有生产活动和社会活动以来，就产生了数据和对数据的处理。数据是客观事实的记录，它包括数值式的和非数值式的。数值式的有实验数据、观测数据、统计数据等，非数值式的有文字形式、图象形式等。对数值式数据大家已很熟悉。这里着重介绍非数值型数据。

例如，有一张登记表格，上面填着“张伟”、“男”、“上海”、“学生”……或者“面粉”、“大米”、“煤”、“青菜”……。再如图书馆的目录卡上，有各种摘要，等等。这类数据就是“非数值型”数据，在某些领域中，它是十分重要的。例如打印表格时，出现的标题就是这类非数值型数据。在COBOL语言中没有汉字，因此这些用汉字表示的非数值型数据应转换为由计算机所允许的字符组成的值。非数值型数据也有相应的运算，如对字符串的并串与拆串，某些情况下的计数和替换等。当然，这只是对个别的数据或者若干个数据进行某些运算和加工。

另外，常见的例子是对大批数据进行加工。例如把一叠帐单（发票）或者按编号大小排好次序，或者按照品种型号分门别类的排好次序，或者按照物品名称依字典顺序排好次序等

等。这种数据加工的职能很重要，其排序功能的实现与许多因素有关。例如，数据量的多寡，排序方法的不同等。因此，排序的效果也有不同。总之，数据处理是从各种形式的数据中提取出有用的资料、文件、数字等的加工过程。

随着社会的发展，计算工具也在发展，例如由笔、墨、算盘、手摇计算机发展到电子计算机。这种发展对数据处理领域来说是十分重要的。凭着人类的智力，辅之以简单的工具只能进行简单的、数量很少、速度要求很低的数据处理工作。一旦需要对某些数据进行成千上万次运算，或者需对大量的数据进行并不复杂的计算，这在电子计算机出现前是很困难的，甚至是不可想象的。电子计算机的出现，才使得对数据进行复杂加工或者对大量数据进行加工成为可能。它为某些学科和某些部门提供了十分有力的工具。反过来又推进了社会的发展和科学技术的进步。

数据量的多寡及其组织形式，数据加工的复杂程度对于数据加工来说是非常重要的。一般地讲，科学技术方面的问题，往往原始数据较少，而加工的算法复杂；商用（或称事务）方面的问题却刚好相反，数据量很大而加工方式简单。这种特点在早期的计算机处理中即已反映出来。为了便于解决这两方面的问题，分别发展了各具特点的程序设计语言（算法语言和数据处理语言），它们之间有共性，也有许多个性。随着应用领域的发展、数据量的增加、数据加工的复杂程度增加，大量数据之间的内在联系也愈来愈得到重视。随着计算技术的飞速发展，出现了各式各样的计算机系统，如计算机网络和数据库。在某些应用领域内，出现了数据量大且算法也复杂的问题，这也推动了计算机软件的发展，推动了程序设计语言和操作系统的发展，大量的应用程序也应运而生。

此外，加工对象初始数据的来源也是很复杂的。例如，存贮原始数据的外部介质可以为磁带，磁盘，纸带，卡片等等；也可能直接由模数转换器实时地获得数据；等等。同样地，存储加工后的数据的介质可能是能长期保存的介质，如行式打印纸，控制台打印机纸，磁带，磁盘，绘图纸和卡片等；也可能是临时供人们交换信息用的，如控制台输出的信息，显示器等等；也可能是用作实时控制用的，如数模转换器等。同样，数据的存贮设备及其容量也在日新月异的更新和扩大。这些都向数据处理提出了新的课题，各类设备的发展提供了物质保证，从而有力地促进了信息加工的发展。

总之，随着计算机的产生和发展，计算机的应用领域的扩大和深入。电子计算机的高速处理能力和大容量快速外存储设备等都为发展数据处理提供了广阔的前景，使信息加工进入新的时代。

1.1.2 电子计算机的数据加工

简单地讲，数据加工的过程可用图1.1表示。

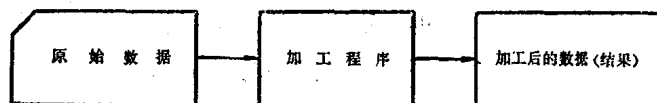


图 1.1 数据加工过程

在计算机上数据处理问题的程序设计可以分成若干步：

(1) 将所处理的问题抽象成一个计算机上能解的问题，并确定一个合适的算法。

(2) 用计算机上适用的程序设计语言描述该算法。

(3) 将描述该算法的源程序穿孔并上机进行编译，准备好数据并进行运行，计算出结果来。如果不能运行或不能正确编

译则需修改，再编译运行，直到能正确运行为止。

(4)分析计算结果的正确与否。如果不正确，应当检查源程序，数据带等。因为算法也可能有错误，所以必要时还须检查算法。然后再编译运行分析结果，如此重复，直到完全正确为止。

一般来说，算法是由用户根据实际问题用自己熟悉的知识 and 技巧抽象和概括起来的，并且该算法对特定的计算机必须是合适的，这不是本书涉及的范围。这个算法要用恰当的程序设计语言来描述，程序设计语言是多种多样的，应采用比较合适的一种。例如，国际上普遍使用的商用通用程序设计语言COBOL即是适用于数据处理的一种。其他还有RPG（报表生成程序），APL等等。此外，有些语言兼有数据处理的功能，如PL/1等。本书中叙述的是COBOL语言中最基本的最常见的成分，作为熟悉和使用COBOL语言的入门。希望使读者初步具有将COBOL语言作为工具来描述并解算数据处理问题的能力。

当然，将算法描述成源程序，并且用相应的数据结构把加工用的数据组织好，在计算机上进行计算，这整个过程的实现须通过上机实践才能掌握。在某种程度上讲，从有一个正确的算法，到算出正确的结果，这中间的差距还相当大。整个过程是必须一步步进行，任何疏忽可能会导致大的错误出现，从而可能无法计算出结果来，也可能计算出来的结果是错误的。另外，这个过程比较繁琐，机械的重复的劳动甚多。这个过程不管让谁做，都是不可跨越的，至少目前仍是如此。

用计算机对大量的数据进行管理和加工，需要计算机的硬件和软件的支持才能实现。硬件是由各种元件器件（如集成电路，各种外部设备等等）组成的，它是计算机发挥作用的物质基础。另一部分是软件。软件的功能和质量对使用计算机是非常重要的，软件只是一个总名称，它至少可以分成系统软件和

非系统软件两种。所谓系统软件是指操作系统，程序设计语言的编译程序，数据管理系统以及维护检测用的程序，由许多公用程序组成的程序库（包）等等。非系统软件通常包括应用软件等。COBOL语言的编译程序是属于系统软件的。

1.1.3 COBOL语言的发展历史

在数据处理方面使用广泛、历史较长且已标准化的高级程序设计语言是COBOL语言(*Common Business Oriented Language*)。它从开始定型到现在，随着计算技术和软件的发展以及应用范围的扩大，一直在不断地发展和完善。它已经广泛应用于各个领域，有许多问题是用它来描述的；如对它进行修改必须特别注意其兼容性，即用原来的语言文本描述的算法无须进行修改也能被新的语言文本所允许。换句话说，修改过的语言文本的功能应当包含原有文本的语言功能。

最早的COBOL语言文本在1959年9月由数据系统语言联合会(CODASYL—*Conference on Data Systems Languages*)提出的，经过修改于1960年4月正式发表，称为COBOL60。

此后，CODASYL执行委员会成立了一个特别任务组，收集了对COBOL60的使用意见，经修改后又提出了一个文本，称为COBOL61，于1961年中期发表。

下一个COBOL版本是由维护委员会提出的，并于1963年中期发表，称为扩充的COBOL61。

1964年1月COBOL维护委员会改组。独立的用户小组和制造商小组合并成COBOL委员会，该委员会编辑了COBOL65。

1968年7月，CODASYL进行改组，它改为由4个固定的委员会组成，即执行委员会，程序设计语言委员会，计划委员会以及系统委员会。程序设计语言委员会(PLC)的目的任务比

以前的COBOL委员会有所扩大, 且先后发表了COBOL68, COBOL69 (1970年发表), COBOL70 (1971年发表), COBOL73. 1974年CODASYL又进行了改组, 仍由程序设计语言委员会发表了COBOL76.

上面叙述的是COBOL语言开发的情况, 至于标准COBOL语言则有所不同. 除了各国国家标准外, 国际上通用的标准COBOL主要有两个, 即推荐的标准版本ISO COBOL72和标准版本ISO COBOL78. 国际标准化组织(ISO)决定接受美国标准COBOL74作为国际标准ISO COBOL78. 国际标准COBOL允许的描述语言规定为英文, 俄文和法文.

1.1.4 国际标准COBOL的结构

ISO COBOL72和ISO COBOL78都是由一个核心和若干个功能模块组成的. 某个特定计算机上配置的某个COBOL语言, 将根据可能和需要由挑选出来的核心的某一级和若干功能模块的某一级组成.

ISO COBOL72由核心和七个功能模块组成. 其结构图见图1.2. 七个功能模块分别是表处理模块, 顺序存取模块, 随机存取模块, 排序(分类)模块, 报表编制模块, 程序分段模块以及库模块. 为了叙述方便起见, 简写为:

核心(NUC, NUCLEUS)

表处理(TBL, TABLE HANDLING)

顺序存取(SEQ, SEQUENTIAL)

随机存取(RAC, RANDOM ACCESS)

排序(SRT, SORT)

报表编制(RPW, REPORT WRITER)

程序分段(SEG, SEGMENTATION)