

高等学校试用教材

COBOL

程序设计

刘惠芳



高等教育出版社

高等学校试用教材

COBOL 程序设计

刘 惠 芳

高等教育出版社

内 容 提 要

本书根据 ISO COBOL-78 标准文本,详细介绍了 COBOL 语言的内容及程序设计技巧。全书从介绍基本简单的 COBOL 程序所需的语句入手,由浅入深地安排内容,循序渐近地扩充其功能。这样,既符合一般读者学习程序设计语言的规律,也会大大地增加他们学习的兴趣。通常在学完 8~10 个学时后,即可编制简单程序上机实习。为了配合上述的内容安排,书中把一个完整的、较全面的实用程序简化为若干个独立的相对完整的由简单到复杂的程序系列,由浅入深。按每章教材扩充的功能相应地扩充该程序实例的功能,使学生既能巩固前章又能练习本章新学内容,给上机实习带来了很大方便。书中提供的三十多个完整的程序实例和若干实例片断(虽以 IBMPC 微机和 HP 3000 计算机为背景,但并不失其一般性),是作者和他的同事们在多年教学中积累起来的第一手材料。全书贯穿了结构化程序设计思想。书的最后一章,对汉字 COBOL 作了一定的介绍。本书学完之后,读者就可以采用结构化程序设计方法,编制出能处理汉字信息的实用 COBOL 程序。

本书可作为高等院校计算机专业、财经管理专业的教材或参考书。对企事业单位管理人员、从事数据处理的工作人员及其他程序设计人员也是一本很好的参考书和自学用书。

责任编辑:刘建元

高等学校试用教材 COBOL 程序设计

刘惠芳

*

高等教育出版社出版
新华书店北京发行所发行
北京印刷一厂印装

*

开本 787×1092 1/16 印张 26.25 字数 600,000

1986 年 5 月第 1 版 1987 年 2 月第 2 次印刷

印数 10,151—14,160

书号 13010·01194 定价 4.00 元

前 言

当前,我国正在进行四个现代化的建设,国民经济管理的现代化是刻不容缓的当务之急。而电子计算机是实现管理现代化最重要的手段之一。

在计算机上使用的各种语言中,COBOL是最适合于数据处理的程序设计语言,它可以对各种形式的数据进行人们希望的加工和处理,在国内外都得到普遍的应用。

本书较全面地介绍了国际标准 ISO COBOL-78 文本的主要内容,详细地叙述了使用此种语言编写程序的基本方法和在 IBM PC 微机及 HP 3000 小型计算机上的操作要点。根据我校多年的经验,本书在内容编排上作了如下改进:

首先,鉴于 COBOL 语言格式严谨,规定较繁琐的特点,采用了按编制 COBOL 简单程序到编制 COBOL 高级复杂程序的要求,循序渐进地安排本课程内容。即先将四个部中编制简单程序必需的内容抽出来,先用很短的时间介绍,使初学者从中可以概略地了解 COBOL 语言的全貌,并具有编制最简单程序的能力。在后面各章节中再逐步深化有关各部的内容,扩充其功能,逐步使学生能全面地掌握编制 COBOL 实用程序的各项技能。这就改变了原来学完四个部的主要内容后,才着手编制程序的老方式,从而可以消除初学时的枯燥乏味感觉。

为了配合上述的内容安排,本书把一个完整的、全面的实用程序简化为若干个独立的、相对完整的、由简单到复杂的程序系列,由浅入深地贯穿全书。即按教材扩充的功能相应地扩充程序实例和作业的内容,使学生既能巩固前章的内容,又能练习本章新学的内容,大大节省了上机时间,且每部分都有上机练习的机会。

根据人民大学、北京工业大学二分校及北京教师进修学院等使用情况看,此种讲授方式是一种大胆的尝试,已获得一定成功,并得到国内一些专家和同行的肯定。一般学生在学完 8~10 个学时后,即可以着手编写简单的程序上机实习。

其次,为了加深学生对课文的理解,本书结合有关章节,提供约三十余个完整的程序实例和若干实例片断,这是我校在多年教学中积累起来的第一手材料。虽然主要以 HP 3000 计算机和 IBM PC 微机为背景,但并不失其一般性,在实际应用中,参照具体的机器,将环境部稍加修改便可以应用。这部分内容和每章后面所附的习题正是许多翻译的课本所欠缺的。

最后,本书对结构程序设计方法及其在 COBOL 中的应用作了简单的介绍。考虑到近年来 COBOL 在汉字信息处理上取得很大进展,用一章的篇幅对汉字 COBOL 作了一定的介绍,可以帮助读者了解汉字 COBOL 在国内的发展动态。

本书共分 16 章,第一章为绪言,第二至六章介绍 COBOL 的基本内容,第七章介绍结构程序设计,第八章介绍强功能的数据操纵,第九至十二章分别介绍顺序文件,相对文件,索引文件和排序-合并,第十三至十五章分别介绍 COBOL 中的表处理,程序间通信和程序分段等功能模块,

第十六章是 COBOL 功能的最新扩充——汉字 COBOL。

本书是为有关高等院校经济信息管理专业学生编写的教材，也可供从事数据处理工作的科技人员和教学工作的同志参考。

南京大学计算机科学系史九林同志和清华大学经济管理学院侯炳辉同志详细阅读了全稿，并提出许多有益的意见，在此谨向他们致以衷心的感谢。

在编写过程中，受到中国人民大学萨师燧教授的指导和鼓励，此外本书部分程序实例，由北京师范大学图书管理系方波同志协助键入 IBM PC 微机完成的，在此表示感谢。

由于水平有限，时间仓促，错误疏漏之处难免，欢迎读者批评指正。

刘惠芳

1985 年 3 月于中国人民大学

目 录

第一章 绪言.....1	§ 3.1 执行(PERFORM)语句(一).....25
§ 1.1 数据处理简介.....1	§ 3.2 程序实例 2.....25
§ 1.2 COBOL 的历史.....1	§ 3.3 字型(PICTURE)子句.....27
1.2.1 COBOL 的产生.....1	3.3.1 字母型数据.....28
1.2.2 COBOL 文本.....2	3.3.2 数值型数据.....28
§ 1.3 COBOL 的功能结构.....2	3.3.3 字符型数据.....29
第二章 COBOL 初步.....4	3.3.4 数值编辑型数据.....29
§ 2.1 COBOL 程序的结构.....4	3.3.5 字符编辑型数据.....32
§ 2.2 COBOL 程序的基本元素.....4	§ 3.4 符号(SIGN)子句.....32
2.2.1 字符集.....4	§ 3.5 遇零置空(BLANK WHEN ZERO)
2.2.2 字符串.....5	子句.....33
2.2.3 常量.....5	§ 3.6 对齐(JUSTIFIED)子句.....33
2.2.4 COBOL 字.....6	§ 3.7 重命名(RENAMES)子句.....34
§ 2.3 COBOL 程序的书写格式.....7	§ 3.8 传送(MOVE)语句的传送规则.....35
§ 2.4 COBOL 语言格式中的几点约定.....9	3.8.1 初等项传送(由初等项传送到初等项).....35
§ 2.5 标识部.....9	3.8.2 组合项传送.....36
§ 2.6 环境部.....10	§ 3.9 程序实例 3.....37
§ 2.7 数据部.....11	练习题和作业.....39
2.7.1 COBOL 程序的数据结构.....11	第四章 算术运算和逻辑测试(一).....41
2.7.2 数据部的结构.....12	§ 4.1 算术运算.....41
2.7.3 工作存储节.....12	4.1.1 加(ADD)语句.....41
2.7.4 值(VALUE)子句.....13	4.1.2 减(SUBTRACT)语句.....42
§ 2.8 过程部.....14	4.1.3 乘(MULTIPLY)语句.....43
2.8.1 过程部的结构.....14	4.1.4 除(DIVIDE)语句.....44
2.8.2 COBOL 语句的分类.....15	4.1.5 计算(COMPUTE)语句.....45
2.8.3 待收(ACCEPT)语句和显示(DISPLAY)	§ 4.2 用法(USAGE)子句.....46
语句.....16	§ 4.3 程序实例 4.....48
2.8.4 传送(MOVE)语句.....17	§ 4.4 如果(IF)语句(一).....51
2.8.5 停(STOP)语句.....18	4.4.1 如果语句的功能和格式.....51
§ 2.9 程序实例 1.....18	4.4.2 条件(符号条件,类别条件,关系条件).....53
§ 2.10 设计一个COBOL 程序的步骤.....20	§ 4.5 程序实例 5.....55
练习题和作业.....23	练习题和作业.....58
第三章 重复过程和初等项的定义.....25	第五章 文件的输入输出.....61

§ 5.1 程序实例 6	61	8.2.2 对应加 (ADD CORRESPONDING)	99
§ 5.2 输入输出 (INPUT-OUTPUT) 节	64	8.2.3 对应减 (SUBTRACT CORRESPONDING)	100
§ 5.3 文件 (FILE) 节	65	§ 8.3 程序实例 9	101
§ 5.4 打开 (OPEN) 语句	66	§ 8.4 检测 (INSPECT) 语句	104
§ 5.5 关闭 (CLOSE) 语句	66	§ 8.5 合串 (STRING) 语句	108
§ 5.6 读 (READ) 语句	67	§ 8.6 程序实例 10	110
§ 5.7 写 (WRITE) 语句	68	§ 8.7 拆串 (UNSTRING) 语句	115
§ 5.8 缓冲区 (BUFFERS)	70	§ 8.8 程序实例 12	119
练习题和作业	71	练习题和作业	123
第六章 如果语句 (二) 和执行语句 (二)	72	第九章 顺序文件及其磁性存储设备	125
§ 6.1 条件名 (CONDITION-NAME) 条件	72	§ 9.1 磁带	125
§ 6.2 复合 (COMPOUND) 条件	74	§ 9.2 磁盘	126
§ 6.3 执行 (PERFORM) 语句 (二)	75	§ 9.3 文件的结构	126
§ 6.4 程序实例 7	79	9.3.1 逻辑记录	126
§ 6.5 转向 (GO TO) 语句	82	9.3.2 物理记录	128
§ 6.6 出口 (EXIT) 语句	83	9.3.3 盘区	129
§ 6.7 预防程序出错的措施 (ERROR PROVENTION)	84	9.3.4 文件标号	129
§ 6.8 程序实例 8	84	§ 9.4 顺序文件的处理	130
练习题和作业	87	9.4.1 什么是文件处理	130
第七章 结构程序设计	90	9.4.2 建立顺序文件	130
§ 7.1 结构程序设计基础	90	9.4.3 读顺序文件	130
§ 7.2 程序控制 (PROGRAM CONTROL)	92	9.4.4 修改记录	130
§ 7.3 程序设计的目标 (PROGRAMMING OBJECTIVES)	93	9.4.5 顺序文件的文件控制段的一般格式	131
§ 7.4 COBOL 程序设计实践 (PROGRAMMING PRACTICES)	94	§ 9.5 程序实例 13	131
§ 7.5 程序的效率 (PROGRAM EFFICIENCY)	95	练习题和作业	135
练习题	95	第十章 相对文件	137
第八章 强功能的数据操纵	97	§ 10.1 相对文件的组织	137
§ 8.1 限定名 (QUALIFIED names)	97	§ 10.2 相对文件的文件控制段的一般格式	137
§ 8.2 对应操作 (CORRESPONDING operations)	98	§ 10.3 同步安置 (SYNCHRONIZED) 子句	138
8.2.1 对应传送 (MOVE CORRESPONDING)	98	§ 10.4 相对文件的处理	139
		10.4.1 建立文件	139
		10.4.2 随机读文件	140
		10.4.3 修改记录	140
		10.4.4 插入记录	141
		10.4.5 删除记录	141

10.4.6 顺序读文件·····	141	12.2.1 排序语句格式·····	236
§ 10.5 程序实例 15·····	142	12.2.2 排序语句的使用·····	237
练习题和作业·····	148	12.2.3 自动方式·····	238
第十一章 索引文件 ·····	149	12.2.4 输入过程·····	238
§ 11.1 索引顺序文件的结构·····	149	12.2.5 输出过程·····	239
11.1.1 索引·····	150	§ 12.3 程序实例 22~24·····	240
11.1.2 溢出区·····	151	§ 12.4 合并(MERGE)语句·····	247
§ 11.2 索引顺序文件的处理·····	153	§ 12.5 程序实例 25~26·····	248
11.2.1 索引文件的文件控制段的一般格式·····	153	练习题和作业·····	252
11.2.2 建立索引顺序文件·····	153	第十三章 表处理 ·····	253
11.2.3 随机读索引顺序文件·····	154	§ 13.1 表的确定·····	253
11.2.4 更新索引顺序文件·····	154	§ 13.2 给表赋值·····	255
§ 11.3 程序实例 16~18·····	154	13.2.1 重定义(REDEFINES)子句·····	255
§ 11.4 键顺序索引文件(KSAM文件)·····	162	13.2.2 建立定值表·····	256
11.4.1 键顺序索引文件的结构·····	162	13.2.3 程序实例 27·····	257
11.4.2 键文件和数据文件之间的关系·····	164	13.2.4 用输入语句给表赋初值·····	258
11.4.3 键顺序索引文件的容量·····	165	§ 13.3 表元的引用·····	258
11.4.4 键顺序索引文件的额外数据段·····	167	13.3.1 下标法·····	258
§ 11.5 键顺序索引文件的处理·····	169	13.3.2 位标法·····	259
11.5.1 索引文件的文件控制段的一般格式·····	169	13.3.3 置(SET)语句·····	259
11.5.2 顺序存取方式的索引文件·····	169	13.3.4 用法是位标(USAGE IS INDEX) 子句·····	260
11.5.3 动态存取方式的索引文件·····	170	13.3.5 下标法与位标法的比较·····	261
§ 11.6 程序实例 19~21 (1)——图书管理 和检索·····	171	13.3.6 使用下标和位标的规则·····	261
§ 11.7 程序实例 21(2)——库存 管理·····	193	§ 13.4 程序实例 28~29·····	261
§ 11.8 程序实例 21(3)——职工 统计·····	204	§ 13.5 二维表和三维表·····	263
§ 11.9 程序实例 21(4)——工资 管理·····	213	13.5.1 二维表和三维表的建立·····	263
§ 11.10 程序实例 21(5)——人事 管理·····	222	13.5.2 给二维表和三维表赋值及引用 表元·····	265
§ 11.11 盘文件组织(摘要)·····	234	§ 13.6 程序实例 30~32·····	269
练习题和作业·····	234	§ 13.7 变长表·····	272
第十二章 排序-合并 ·····	235	§ 13.8 表的检索·····	273
§ 12.1 排序步骤和排序文件·····	235	13.8.1 用执行语句(三)顺序查找·····	273
12.1.1 排序步骤·····	235	13.8.2 顺序检索语句·····	274
12.1.2 排序文件·····	235	13.8.3 二分检索语句·····	278
§ 12.2 排序(SORT)语句·····	236	§ 13.9 程序实例 33~35·····	282
		13.9.1 为银行编制本利和对照表(单利)·····	282
		13.9.2 处理银行存款业务·····	285
		13.9.3 计算银行存款的本利和·····	290
		13.9.4 程序实例 36——职工统计通用	

报表程序.....	293
13.9.5 程序实例 37——库存管理通用 报表程序.....	303
练习题和作业.....	312
第十四章 程序间通信	314
§ 14.1 调用(CALL)语句.....	314
14.1.1 调用语句的格式.....	314
14.1.2 调用程序(CALLING PROGRAMS)	314
14.1.3 用(USING)短语.....	315
14.1.4 溢出(OVERFLOW)短语.....	315
14.1.5 调用语句的执行过程.....	315
§ 14.2 COBOL 被调程序.....	316
14.2.1 连接节(LINKAGE SECTION).....	316
14.2.2 过程部的部首.....	316
14.2.3 程序出口(EXIT PROGRAM) 语句.....	317
14.2.4 返回(GOBACK)语句.....	317
14.2.5 撤消(CANCEL)语句.....	317
§ 14.3 HP3000 计算机的 COBOL II 扩充的部分功能.....	317
14.3.1 入口(ENTRY)语句.....	317
14.3.2 扩大的调用语句格式.....	318
14.3.3 调用内部过程(CALLING INTRINSICS).....	318
14.3.4 程序实例片断.....	318
作业.....	324
第十五章 程序分段	325
§ 15.1 目标程序的结构.....	325
§ 15.2 程序分段的方法.....	325
15.2.1 程序块限于句.....	325
15.2.2 段号.....	325
第十六章 COBOL 功能的最新扩 充——汉字 COBOL	327
§ 16.1 汉字 COBOL 语言的设计思想.....	327
16.1.1 创建崭新的汉字 COBOL 编译 程序.....	327

16.1.2 用文字转换技术来实现汉字 COBOL.....	327
§ 16.2 汉字 COBOL 符号的转换技术.....	328
16.2.1 汉字终端.....	328
16.2.2 带有汉字功能的操作系统.....	329
§ 16.3 在 COBOL 中汉字输入的编 码方案.....	330
16.3.1 综述.....	330
16.3.2 汉字编码方案的现状.....	331
16.3.3 IBM PC 微机上的汉字编码.....	332
§ 16.4 COBOL 汉字库技术.....	338
16.4.1 汉字的存储方式.....	338
16.4.2 汉字的存储技术.....	340
§ 16.5 COBOL 汉字的输入输出设备.....	341
16.5.1 输入设备.....	341
16.5.2 输出设备.....	341
§ 16.6 汉字 COBOL 源程序的书写 方式.....	341
16.6.1 全汉字方式.....	341
16.6.2 数据汉化方式.....	345
附录.....	346
附录 I IBM PC 微机(支持汉字)的 COBOL 程序上机操作及有 关命令.....	346
附录 II HP3000 计算机的上机操作及有 关命令.....	360
附录 II-1 HP3000 计算机的上机操作.....	360
附录 II-2 HP3000 计算机上的 BUILD 命令和系统文件名中的参数.....	368
附录 II-3 KSAM 文件的有关命令.....	369
附录 II-4 状态键值的含义表.....	371
附录 III COBOL 保留字表.....	374
附录 IV IBM PC L/II COBOL 的格 式(83 年版).....	377
附录 V HP3000 COBOL II 的格式.....	393
参考文献.....	410

第一章 绪 言

COBOL 是 Common Business Oriented Language 的缩写,即面向公共事务的语言,是一种高级程序设计语言。它在工业、商业和国家企事业的数据处理中,例如在银行业务、情报检索(包括图书文献、人事、档案、库存、工资等检索…),火车调度,气象预报,飞机、车、船票的预订,生产、质量和销售等方面的管理中,都得到广泛使用。

§ 1.1 数据处理简介

什么叫数据?我们这里将它定义为事实的记录。记录的形式有数值式和非数值式两种。数值式有实验数据、观测数据和统计数据等;非数值式有文字数据、图象数据等。

数据处理是将各种形式的数据转换成信息的过程。信息是有用的资料、文件、数值等。信息用数据来表示。

数据处理包括对数据的收集、组织、传输、存储、排序和计算等环节。本门课将陆续介绍怎样用 COBOL 来实现这些环节。

数据处理的主要特点是处理的数据量大,并伴有大量的文字数据。

数据处理是计算机应用最广泛的一个领域。据先进的工业国家统计,它占计算机用量的 75~80%。在我国,目前绝大多数计算机主要用于数据处理方面。

§ 1.2 COBOL 的历史

1.2.1 COBOL 的产生

在 50 年代中期,用于企业和事业数据处理的计算机数量不断增长。用户迫切希望有一种合适的程序设计语言来解决这方面的问题,使得编写程序方便,提高易读性。

1956 年,在美国的富兰克林学院召开了一次重要的会议,后人称它为里程碑式的会议,会议的中心议题是讨论高级程序设计语言。会上提出了多种语言,其中包括当时在 UNIVAC-I 机上实现的 FLOWMATIC 语言,它是事务数据处理问题方面第一个类似于英语的语言,这种语言的实现和使用为 COBOL 的产生奠定了基础。

1959 年 5 月末,在美国五角大楼召开了一次会议,讨论了建立一种面向公共事务的程序设计语言的种种要求和可能性。出席这次会议的代表有工业和政府部门的用户,计算机制造厂商和其它有关人士。会议认为,建立这样一种语言不仅十分必要,而且完全可能,会议确定了建立这种语言应该强调的方针和这种语言应该具有的特点。会后,成立了数据系统语言委员会 CODASYL (Conference on Data Systems Languages 的缩写),委员会下属的短期委员会专

门负责 COBOL 的开发。

由于数据处理应用领域的迫切需要和已有语言的实践经验，加上组织机构的建立，导致 COBOL 很快就诞生了。

数据系统语言委员会的工作重点是对 COBOL 的开发、维护和完善。另外一项重要任务是 COBOL 的标准化工作，这主要由国际标准化组织 ISO(International Standards Organization 的缩写)和美国国家标准化协会 ANSI(American National Standard Institute 的缩写)承担。

1.2.2 COBOL 文本

CODASYL 于 1960 年发表了 COBOL 的第一个正式文本，称为 COBOL-60。经过应用，不断地对早期文本的功能进行修改、完善和扩充，相继发表了 COBOL-61, COBOL-61 扩充, COBOL-65, COBOL-68, COBOL-69, COBOL-70, COBOL-73, COBOL-76, COBOL-81 等更新文本。

美国国家标准协会批准并发表的国家标准文本有 ANSI COBOL X3.23-68 和 ANSI COBOL X3.23-74。

国际标准化组织发表的国际标准 COBOL 文本有 ISO COBOL-72 (同 ANSI COBOL X3.23-68 基本相似)和 ISO COBOL-78(它直接引用 ANSI COBOL X3.23-74)。

由 COBOL 的发展历史可以看出，它是一种适用于数据处理的程序设计语言。

§ 1.3 COBOL 的功能结构

COBOL 的功能由核心和若干功能模块组成。因为 ISO COBOL-78 是较新、较完善的文本，所以本书主要介绍这个文本。它由核心和 11 个功能模块组成。其结构如下表：

核心	表处理	顺序 I-O	相对 I-O	索引 I-O	排序合并	报表编制	程序分段	库	排错	程序间通信	通信
二级	二级	二级	二级	二级	二级	一级	二级	二级	二级	二级	二级
			一级	一级	一级		一级	一级	一级	一级	一级
一级	一级	一级	空	空	空	空	空	空	空	空	空

由上表可以看出，每一个功能模块又按功能大小分成 2~3 个等级，每个等级之间的关系是低一级的功能总是包含在较高级之中。在实现时，可以根据需要和机器的实际配置来确定采用哪些功能模块，以及哪一级功能，从而使选用的 COBOL 语言功能更适合自己的实际需要和机器的支撑能力。最简功能的 COBOL 由核心一级，表处理一级和顺序 I-O 一级等三个模块组成。最全功能的 COBOL 由最高级的核心和所有功能模块的最高级组成。

下面对 ISO COBOL-78 的各功能模块作简单介绍：

核心(NUCLEUS)：提供语言的基本成份。

表处理(TABLE HANDLING)：提供确定表的结构和存取表元素。

顺序 I-O (SEQUENTIAL I-O): 提供确定顺序文件结构和存取记录的功能。

相对 I-O (RELATIVE I-O): 提供海量存储文件, 按相对记录号顺序地或随机地存取文件中记录的功能。

索引 I-O (INDEXED I-O): 提供海量存储文件, 按记录键顺序地或随机地存取文件中记录的功能。

排序-合并 (SORT-MERGE): 提供对一个或多个文件中的记录按照某种顺序排序, 以及将有序的两个或多个文件合并成一个文件的功能。

报表编制 (REPORT WRITER): 提供报表格式定义, 建立数据的逻辑关系, 用简单的报表语句生成报表的功能。

程序分段 (SEGMENTATION): 提供目标程序执行期间的覆盖技术, 从而实现用小的内存空间处理较大程序的功能。

库 (LIBRARY): 提供在海量存储设备上建立源库或目标库, 以及在源程序或目标程序中复制库中程序段的功能, 以节省书写程序的工作量。

排错 (DEBUG): 提供用户在程序中规定排错算法, 使执行程序期间能监视数据和流程的功能。

程序间通信 (INTER-PROGRAM COMMUNICATION): 提供调用其它程序 (包括用不同程序设计语言编写的程序) 并传递调用参数的功能。

通信 (COMMUNICATION): 提供在本地和远地通信设备之间直接对话, 存取, 处理和产生报文或部分报文的功能。

ISO COBOL-72 由核心和七个功能模块组成, 它没有上述的索引 I-O, 相对 I-O, 程序间通信, 通信和排错五个功能模块。另外配有随机存取模块 (它的功能是确定海量存储文件, 用键随机地存取文件中记录)。而 ISO COBOL-78 与 ISO COBOL-72 相比主要有如下几点差别:

1. 将 1972 标准的 REMARKS 段和 NOTE 语句, 改用程序纸第 7 列的星号 “*” 代替。
2. 1972 的 EXAMINE 语句由新的 INSPECT 语句代替。
3. 1972 标准中的某些句法规定放宽, 层号 77 不需要在 01 级之前; 逗号、句号和分号之前可以有空; 变量和它的下标之间不需要留空。
4. 增加 STRING 和 UNSTRING 语句, 加强了字符处理能力。
5. 程序员能动态地确定对比序列。
6. 扩大随机存取 I-O 的功能成索引 I-O。索引文件既能随机存取又能顺序存取。
7. 排序功能模块重新命名为排序-合并功能模块, 扩充的 MERGE 语句能将两个以上有序的文件合并为一个文件。
8. 编制报表的能力有所扩充, 并有更清楚的结构。
9. 增加了调试功能模块。
10. 增加了被调程序的连接功能 (CALL, EXIT, PROGRAM, LINKAGE 等语句)。
11. 增加了通信功能模块。

第二章 COBOL 初步

这一章介绍 COBOL 程序的一般结构,源程序的格式,COBOL 程序中最简单的成份和设计 COBOL 程序的步骤.学完本章后,读者可了解 COBOL 程序设计的基本方法,编写最简单的程序.

§ 2.1 COBOL 程序的结构

COBOL 程序由四个部组成.这四个部的逻辑顺序是:

IDENTIFICATION DIVISION.	标识部
ENVIRONMENT DIVISION.	环境部
DATA DIVISION.	数据部
PROCEDURE DIVISION.	过程部

标识部主要规定源程序的名称及其注释信息.

环境部主要确定程序中使用的数据文件(包括文件名,文件所用的存储设备及组织方法).

数据部确定源程序中使用的所有数据项.

过程部规定程序必须执行的操作,使能得到希望的结果.

一个完整的 COBOL 程序必须由这四个部组成,也必须严格按照上述顺序书写.

COBOL 程序是一种层次结构.在每个部内依次是所选的节 (SECTION), 段 (PARAGRAPH)、句子 (SENTENCE), 语句 (STATEMENT), 子句 (CLAUSE) 和短语 (PHRASE). 每个部的结构都不相同,这将在有关部中作具体介绍.

§ 2.2 COBOL 程序的基本元素

2.2.1 字符集 (CHARACTER SET)

字符集是所有 COBOL 字符的总称. COBOL 字符是组成 COBOL 程序的最小单位. ISO COBOL 使用的字符有 51 个,它们是英文字母 A~Z, 数字 0~9 和 15 个专用字符. 这些专用字符如下:

+	加号
-	减号或连字符
*	乘号或星号
/	除号或斜杠
=	等号

\$	货币号
,	逗号
;	分号
.	句号或小数点
"	双引号
'	单引号
(	左括号
)	右括号
>	大于号
<	小于号
□	空格(显示时没有任何图形的符号)

不同的计算机可能扩充某些字符(如 HP3000 计算机增加%, @, \ 等三个符号), 在此不一一作介绍。

2.2.2 字符串(character strings)

字符串是 COBOL 字符集中的一串连续字符(包括一个字符)。它组成 COBOL 常量, COBOL 字等。字符串之间用分隔符隔开, 分隔符可以是逗号、分号、句号或空格。

2.2.3 常量(Literal)*

常量的值是它本身, 它不在数据部中描述就可直接在过程部中使用。它有三种形式: 数值常量, 非数值常量和象征常量。

1. 数值常量(numeric literal)

数值常量即通常所说的数值。它由数字 0~9, 正号, 负号和/或小数点组成。其组成规则如下:

- 1) 长度不能超过 18 位数字;
- 2) 至少有一位是数字;
- 3) 正负号不能多于一个, 如果使用正负号, 则必须放在字符串的左端;
- 4) 小数点不能多于一个, 使用时不能放在字符串的右端。

2. 非数值常量(non-numeric literals)

非数值常量由一对引号中的任意字符串或汉字串(若计算机支持使用汉字)组成, 引号只作为定界用, 不能作为常量的值。非数值常量不能参加算术运算, 如将数值常量 12.345 括在一对引号中, "12.345" 就成为非数值常量, 若用它参加算术运算就是错误的。

* 有的书将 literal 译成字值。

** 由于计算机上所用引号可为 " 或 ', 故本书未作统一。

非数值常量的长度随计算机不同而异, HP3000 计算机可以在 1~132 个字符范围内, FELIX 计算机是在 1~229 个字符范围内, IBM PC 计算机是在 1~128 字符范围内。

例 DISPLAY "工作人员表"

或 DISPLAY "STAFF AND WORKERS LIST"

执行该语句后, 在屏幕上显示出引号中的数据。

3. 象征常量(figurative constant)

象征常量是以字的含义为其值的 COBOL 字, 有下面六种:

{HIGH-VALUE	表示一个或多个计算机编码对比序列中有最高值的字符。
{HIGH-VALUES	
{LOW-VALUE	表示一个或多个计算机编码对比序列中有最低值的字符。
{LOW-VALUES	
{QUOTE	表示一个或多个引号。
{QUOTES	
{SPACE	表示一个或多个空格。
{SPACES	
{ZERO	表示一个或多个零。
{ZEROS	
{ZEROES	

ALL 常量 表示一个或一串指定的常量, 必须是非数值常量。

象征常量的单数和复数形式是等价的, 它们的长度按照下面的规则确定:

a) 象征常量同另外的数据项没有关系时, 它的长度是一个字符(ALL 除外)。

例 执行 DISPLAY ZERO 则在终端上显示出一个零。

b) 象征常量同另外的数据项有关时, 它的长度同有关数据项的长度相等。

例 77 WORK PIC 9(5) VALUE IS ZERO.

上面描述说明, 数据名 WORK 的长度是 5 位数值, 则 ZERO 表示 5 个零。

c) 使用 ALL 常量时该常量被重复直到相关数据项的存储空间被填满为止。

例 77 A PIC X(10) VALUE ALL '*'.
MOVE ALL '123' TO A.

这里第一行中 A 的值是 10 个星号, 执行 MOVE 语句后, A 的值是 1231231231.

2.2.4 COBOL 字

COBOL 字代表某种实体的名字, 如数据项, 常量等。COBOL 字通常又分为用户定义字和保留字两种。

1. 用户定义字(user-defined words)

用户定义字是为了满足 COBOL 语句或子句的格式要求, 由用户定义的字。其定义规则如

下:

- a) 由字母, 数字和连字符组成;
- b) 至少应包括一个字母;
- c) 字的长度不能超过 30 个字符;
- d) 连字符不能出现在字的开始和结尾处.

用户定义字有文件名, 数据名, 条件名, 记录名, 过程名等 16 种. 有的用户定义字还有特殊要求, 这将在它们出现的地方再给予具体说明.

2. 保留字(reserved words)

COBOL 语言为自己保留的字约有 300 个, 在程序中它们有固定的含义, 必须正确拼写. 如 ACCEPT, ADD, READ 等. 程序员不能将它们作为用户定义字使用, 但非数值常量中可以使用它们, 因加上引号后便失去了保留字本来的含义. 保留字分为三类:

a) 基字(KEY WORDS)

它在 COBOL 语句或子句中是必写的字, 否则不能表达相关子句或语句的含义.

例 ACCESS MODE IS RANDOM

该子句中的 ACCESS 和 RANDOM 是基字, 缺少任意一个都将使该子句失去原来的意义.

b) 任选字(Optional words)

在语句或子句中可写可不写的字是任选字, 写上它能增加程序的可读性, 如上列的 MODE 和 IS 都是任选字.

c) 象征常量

前面已介绍过(共有六种).

COBOL 保留字均列在书后附录 III 中.

§ 2.3 COBOL 程序的书写格式

COBOL 程序不管是写在程序纸上, 还是直接从终端键入计算机, 每一行最多可以写 80 个字符, 即 80 列. 根据作用不同将它们划分为五个区. 见图 2-1.

1~6 列为行号区, 专门用来书写程序的行号. 从终端直接键入程序时, 系统会自动地提供行的顺序号. 若程序写在程序纸上, 供穿孔卡片用, 也往往空着不写.

8~72 列写源程序的正文. 它又分为 A 区和 B 区.

8~11 列称为 A 区, 所有的部首, 节首, 段首, 段名, 层指示符 FD 和 SD, 层号 01, 77 必须从这个区开始, 如果需要, 上述的项目可以继续进入 B 区.

12~72 列称为 B 区, 除了必须从 A 区开始的项目外, 所有其余的 COBOL 正文都应写在这个区域内. 如所有的子句, 过程部的语句等.

73~80 列是标识区, 一般空着不用.

第 7 列是续行和注释行标志区. 一个句子用一行写不完, 需要在下一行继续写, 则后续的行称为续行. 一个字或一个数值常量被分割写在两行上, 则必须在第 7 列写上连字符, 这个连字符

行顺序	C	A	B	标识区
1 2 3 4 5 6 7 8		12		7272-80
		IDENTIFICATION DIVISION.		
		PROGRAM-ID TEXTWNN.		
		ENVIRONMENT DIVISION.		
		DATA DIVISION.		
		WORKING-STORAGE SECTION.		
01		IN-RECORD.		
05		UNIT PIC X(30) VALUE " 中国人民大学 经济信息		
		" 管理系 ".		
05		STUDENT PIC 9(5) VALUE 114		
		75.		
05		TEACHER PIC 9(3) VALUE 532.		
05		STAFF PIC 9(3) VALUE 234.		
05		SUM-NUMBER PIC 9(5) VALUE 0.		
		PROCEDURE DIVISION.		
ST.		ACCEPT IN-RECORD.		
		COMPUTE SUM-NUMBER = STUDENT + TEACHER + ST		
		AFF.		
		DISPLAY IN-RECORD.		
		STOP RUN.		

2-1