



● 高等学校试用教材

● 钱树人 编

COBOL 导论

高等学校试用教材

COBOL 导 论

钱树人 编著

高等教育出版社

内 容 提 要

本书是为高等学校学生学习计算机语言课程而编写的一本教材, 主要介绍我国国家标准 COBOL 程序设计语言, 它等效于国际标准 ISO COBOL₇₈。本书共十一章, 第一章到第五章介绍了 COBOL 语言的发展概况及其基本结构以及 COBOL 语言的标识部、环境部、数据部和过程部, 学完第五章, 读者便可以自己动手编制具有一般水平的程序。第六章到第十一章介绍了 COBOL 语言的表处理、排序、合并、文件组织、报表编制和通信等功能。这一部分内容彼此之间有一定的独立性, 教学中可以自行调整顺序, 也可以根据需要进行取舍。

本书可作为计算机软件专业、管理专业、财经专业以及其他有关专业的学生学习 COBOL 语言课程的教材或教学参考书, 也可供从事这方面工作的科技人员和管理人员阅读参考。

责任编辑 鲍涌

JS461/32
06

高等学校试用教材

COBOL 导 论

钱树人 编著

*

高等教育出版社出版

新华书店北京发行所发行

北京印刷二厂印刷

*

开本 787×1092 1/16 印张 21 字数 480,000

1985 年 10 月第 1 版 1985 年 10 月第 1 次印刷

印数 00,001—9,800

书号 13010·01114 定价 4.05 元

前 言

COBOL 程序设计语言,是数据处理领域中使用最为广泛的程序设计语言,也是最重要的通用程序设计语言之一。它广泛用于商业、银行、财会、统计、事务管理、企业管理等各个方面,同时与数据库有着密切的联系。

COBOL 语言是从五十年代末开始发展起来的。它不但开发早,而且开发工作也进展得很快,开发版本不断地出现,随之标准化工作也开展得十分迅速。**COBOL** 程序设计语言至今仍是“活”的程序设计语言。

COBOL 语言包含的功能较丰富,但由于计算机系统的发展和多样性,**COBOL** 语言及其实现的发展历史,以及 **COBOL** 语言本身有些部分需由实现系统规定或者允许由实现系统作进一步的补充规定等原因,使得各计算机系统上配置的 **COBOL** 语言系统会有某些差异或者需由实现系统另行指明和规定的地方,但绝大部分还是相同的。

本书除考虑适用于计算机系软件专业 **COBOL** 课程的教学要求外,同时还考虑到其它系科专业和各方面对 **COBOL** 课程的要求。因此在组织全书章节时,除了考虑到系统性以外,还考虑到某种程度的灵活性。在全书十一章中,第一章到第五章是最基本的内容(作为第一层次),第六章到第八章是很重要的内容(作为第二层次),第九章到第十一章是较深入的内容(作为第三层次)。从第六章开始,章节之间有一定的独立性。教材章节分开层次后,便于教师根据教学的需要进行取舍,以适应不同专业的要求。

为了使 **COBOL** 语言课讲授的内容具有“标准性”和“通用性”,故取材于我国新近制订的国家标准 **COBOL** 语言文本。我国国家标准 **COBOL** 语言等效于 **ISO COBOL78**,该国际标准是当前国际上 **COBOL** 语言的标准。在教材的取材和叙述上,尽量不涉及具体的计算机系统和语言的实现系统,并从使用者的角度上尽量系统、全面同时又有重点地介绍国家标准 **COBOL** 语言。

掌握 **COBOL** 程序设计语言和掌握其它的程序设计语言一样,需要编制一些程序和在上计算机上解算。在实习过程中,需要介绍有关计算机系统和 **COBOL** 语言实现系统的一些概况,并说明其实现系统中的一些规定和约定等,以便能使读者在 **COBOL** 语言方面具有较全面较系统的知识和具有相当水平的解题能力。

各章的内容分别简述如下:第一章叙述 **COBOL** 语言和 **COBOL** 源程序的概貌。第二章叙述 **COBOL** 源程序中“词”一级的成分, **COBOL** 源程序的处理对象及其组织的概述。这两章基本上属于轮廓性和概念性的介绍,但对掌握 **COBOL** 语言却是有帮助的。第三章到第五章分别叙述了 **COBOL** 源程序中必须出现的四个部(标识部、环境部、数据部和过程部)的基本描述。这五章是最基本的内容。从第五章开始就能编制出完整的 **COBOL** 源程序和上机解题了。第六章叙述处理功能。开始介绍下标方法,以后介绍位标方法的定义及其有关的引用方法和操作。第七章介绍排序合并功能。第八章叙述在大容量存储设备(以磁盘为背景外设)上的文件的

描述及其操作,它的各节对应于顺序 I-O,相对 I-O 和索引 I-O 功能模块的内容。第六章到第八章是很重要的内容。第九章从如何组织 COBOL 源程序的角度上叙述了程序间的通信功能,库功能,程序分段功能以及插入功能等,此外还简单地介绍了结构程序设计思想及其在 COBOL 程序设计中的应用。第十章介绍了报表编制功能。第十一章扼要地叙述了对非数值型数据进行操作的语句以及 COBOL 语言中的其他功能。第九章到第十一章属于较深入的内容。附录 A(国家标准 COBOL 语言中的保留字表)、附录 B(国家标准 COBOL 语言中的英中名词对照表)和附录 D(国家标准 COBOL 语言中的格式表)供读者备查。附录 C(例)供读者参考,有些例子不但用于提供完整的源程序,而且还对正文部分进行了某些补充。这些例子主要用于加深对内容的理解,因而例子是较简单的。

本书的基本对象是高等院校计算机专业的学生。通常用于第二程序设计语言课,但也可用于第一程序设计语言课。由于本书的组织采用分层和积木式结构,因此也可供管理、财经以及其他专业的学生和技术人员阅读参考。

本书由上海计算技术研究所朱三元副研究员主审,参加审阅的还有吉林大学周长林副教授,中国人民大学刘惠芳同志,北京航空学院吴清秀同志和谢佩同志以及北京大学何宇才同志,他们对全书进行了认真的审阅,提出了许多宝贵的意见和建议,在此谨表示衷心的感谢。

在本书稿的最后修改过程中,南京大学计算机科学系宋满英同志做了不少工作,在此表示衷心的感谢。

在本书编写过程中,得到了南京大学计算机科学系徐家福教授的支持和指导,还得到了南京大学计算机科学系及软件研究所、教研室的领导以及同志们的支持和帮助,并且还得到其他兄弟单位及其他同志们的支持和帮助。在此一并表示衷心的感谢。鉴于作者的水平和条件所限,缺点和错误在所难免,敬请读者批评指正。

作 者

1983 年 12 月于南京大学

目 录

第一章 概论	1
1.1. 数据处理和数据处理语言.....	1
1.2. COBOL 语言发展概况	2
1.3. 我国国家标准 COBOL 语言概述	3
1.4. 样本程序	6
1.5. 书写 COBOL 源程序的基准格式.....	10
1.6. COBOL 语言文本的描述提要.....	11
习题.....	14
第二章 COBOL 语言中的基本元素和处理对象	15
2.1 符号.....	15
2.1.1. COBOL 字符集和计算机字符集	15
2.1.2. 基本符号的构成	15
2.1.3. 保留字	16
2.1.4. 系统名	16
2.1.5. 用户定义名	16
2.1.6. 分隔符	17
2.2. 数据	18
2.2.1. 数据的类型	18
2.2.2. 常量在 COBOL 源程序中的表示	19
2.2.3. 变量在 COBOL 源程序中的表示——数据名和标识符	21
2.2.4. 数据在计算机内存中的存储表示	21
2.3. 文件与外设的简述	23
2.3.1. 外设的基本类别	23
2.3.2. 记录(逻辑记录)和块(物理记录)的关系	24
2.3.3. 文件的类型及其操作简述	25
习题.....	26
第三章 标识部和环境部	27
3.1. 标识部	27
3.2. 环境部	28
3.2.1. 配置节.....	28
3.2.2. 输入输出节.....	30
3.2.2.1. FILE—CONTROL 段(文件控制段).....	30
3.2.2.2. I—O—CONTROL 段(I—O 控制段).....	32
习题.....	34

第四章 数据部的基本描述	35
4.1. FD 文件描述款	36
4.1.1. BLOCK CONTAINS 子句(块长子句).....	37
4.1.2. RECORD CONTAINS 子句(记录长子句).....	38
4.1.3. LABEL 子句(标号子句)	38
4.1.4. VALUE OF 子句(标号值子句).....	40
4.1.5. DATA RECORD 子句(数据记录子句)	41
4.2. 记录描述和数据项描述	42
4.2.1. 树形结构和层号	43
4.2.2. PICTURE 子句(形象子句)和 BLANK WHEN ZERO 子句(遇零置空 白子句).....	45
4.2.2.1. 初等数据项的类型及其描述.....	46
4.2.2.2. 形象字符和编辑功能.....	49
4.2.3. USAGE 子句(用法子句)	57
4.2.4. SIGN 子句(正负号子句)	59
4.3. 工作存储节和 VALUE 子句.....	60
4.3.1. 工作存储节	60
4.3.2. VALUE 子句和 88 层号(变量条件名)	61
4.4. 其他的数据项描述子句	64
4.4.1. JUSTIFIED 子句(对齐子句)	64
4.4.2. SYNCHRONIZED 子句(对边界子句)	64
4.4.3. 66 层号和 RENAME 子句(重命名子句)	67
4.4.4. REDEFINES 子句(重定义子句).....	69
4.5. 例	71
习题	75
第五章 过程部的基本描述	78
5.1. 过程部的概貌	78
5.1.1. 过程部的框架	78
5.1.2. 句子、语句和执行流程.....	79
5.1.3. 定义及同名定义, 引用及限定引用.....	82
5.2. 与文件无关的基本传输语句	86
5.2.1. ACCEPT 语句(待收语句)	86
5.2.2. DISPLAY 语句(显示语句)	89
5.3. MOVE 语句(传送语句).....	90
5.3.1. 基本格式和功能	90
5.3.2. 发送项和接收项的匹配传送	92
5.3.3. CORR 和带有 CORR 的传送语句	96
5.4. 算术表达式和算术语句	98

5.4.1. 算术表达式	98
5.4.2. 截断,舍入和长度错误条件.....	99
5.4.3. ADD 语句(加法语句)	100
5.4.4. SUBTRACT 语句(减法语句).....	105
5.4.5. MULTIPLY 语句(乘法语句).....	108
5.4.6. DIVIDE 语句(除法语句).....	109
5.4.7. COMPUTE 语句(计算语句)	112
5.5. 条件和 IF 语句(如果语句).....	113
5.5.1. 简单条件和否定的简单条件.....	114
5.5.2. 组合条件和否定的组合条件.....	118
5.5.3. 组合关系条件的省写.....	118
5.5.4. IF 语句(如果语句).....	119
5.6. 流向控制语句	121
5.6.1. GO TO 语句(转向语句)	121
5.6.2. STOP 语句(停语句)	122
5.6.3. PERFORM 语句(执行语句)和 EXIT 语句(出口语句)	123
5.6.3.1. PERFORM 语句的基本功能和调用域	123
5.6.3.2. EXIT 语句.....	127
5.6.3.3. 格式 2 和格式 3 的 PERFORM 语句	128
5.6.3.4. 格式 4 的 PERFORM 语句.....	131
5.7. 与顺序文件有关的最基本的传输语句.....	134
5.7.1. OPEN 语句(打开语句)	134
5.7.2. CLOSE 语句(关闭语句).....	135
5.7.3. READ 语句(读语句).....	136
5.7.4. WRITE 语句(写语句).....	138
习题.....	143
• 第六章 表处理.....	152
6.1. 表的基本描述和 OCCURS 子句(出现子句)	152
6.2. 表元的表示.....	157
6.2.1. 下标法.....	158
6.2.2. 位标法.....	161
6.2.3. 表元的有序表示.....	163
6.3. 位标型量的运算.....	165
6.3.1. 位标型量的比较.....	165
6.3.2. SET 语句(置语句)	167
6.4. 表元的查找.....	169
6.4.1. 用 PERFORM 语句查找.....	170
6.4.2. 用格式 1 的 SEARCH 语句(查找语句)进行顺序查找.....	171

6.4.3. 用格式 2 的 SEARCH 语句进行快速查找	173
习题	175
7 第七章 排序和合并	179
7.1. 排序操作和 SORT 语句(排序语句)简述	179
7.2. 排序合并功能在环境部和数据部中所需的成分	184
7.2.1. 环境部中的 FILE—CONTROL 段	184
7.2.2. 环境部中的 I—O—CONTROL 段	184
7.2.3. SD 文件描述	185
7.3. 排序功能的使用	185
7.3.1. 不用输入输出程序段的排序语句	185
7.3.2. 输入程序段和 RELEASE 语句(释放语句)	186
7.3.3. 输出程序段和 RETURN 语句(回收语句)	187
7.3.4. 使用输入输出程序段的排序语句	187
7.4. MERGE 语句(合并语句)	189
习题	190
第八章 大容量存储设备上的文件及其处理	192
8.1. 概述	192
8.2. 顺序文件的补充介绍	193
8.2.1. OPEN 语句	193
8.2.2. CLOSE 语句	194
8.2.3. REWRITE 语句(重写语句)	195
8.3. 相对文件	196
8.3.1. 相对文件在环境部中的描述	196
8.3.2. 相对文件在数据部中的描述	197
8.3.3. 相对文件在过程部中的描述	197
8.3.3.1. OPEN 语句和 CLOSE 语句	197
8.3.3.2. READ 语句	198
8.3.3.3. WRITE 语句	200
8.3.3.4. REWRITE 语句(重写语句)和 DELETE 语句(删除语句)	201
8.3.3.5. START 语句(开始语句)	203
8.4. 索引文件	205
8.4.1. 索引文件在环境部中的描述	205
8.4.2. 索引文件在数据部中的描述	206
8.4.3. 索引文件在过程部中的描述	207
8.4.3.1. OPEN 语句和 CLOSE 语句	207
8.4.3.2. READ 语句	207
8.4.3.3. WRITE 语句	210
8.4.3.4. REWRITE 语句和 DELETE 语句	211

8.4.3.5. START 语句	211
习题	214
第九章 COBOL程序的构造和组织	217
9.1. 程序间的通信功能	217
9.1.1. 被调程序中的语言元素——LINKAGE SECTION(连接节)和 EXIT PROGRAM 语句(程序出口语句)	217
9.1.2. 调用程序中的语言元素——CALL 语句(调用语句)和 CANCEL 语句(撤消语句)	218
9.2. 库功能	220
9.3. 程序段的插入功能	222
9.4. 用结构程序设计思想编写 COBOL 源程序	223
9.5. 程序分段功能和内存使用的优化措施	225
9.6. ALTER 语句(更向语句)	228
第十章 报表编制功能	231
10.1. 概述	231
10.2. 报表编制功能在数据部中的描述	233
10.2.1. 文件描述款	233
10.2.2. 报表描述款	234
10.2.3. 报表栏描述款	236
10.3. 报表编制功能在过程部中的描述	242
10.3.1. INITIATE 语句(初化语句)	242
10.3.2. GENERATE 语句(生成语句)	242
10.3.3. TERMINATE 语句(终止语句)	243
第十一章 COBOL语言的其余功能的简述	247
11.1. 非数值数据及其操作	247
11.1.1. INSPECT 语句(检测语句)	248
11.1.2. STRING 语句(合串语句)	251
11.1.3. UNSTRING 语句(拆串语句)	252
11.2. 通信功能的简述	254
附录 A 国家标准 COBOL 语言中的保留字表	259
附录 B 国家标准 COBOL 语言中的英中名词对照表	263
附录 C 例	277
附录 D 国家标准 COBOL 语言中的格式表	308

第一章 概 论

1.1. 数据处理和数据处理语言

人类很早就开始和数据产生了联系。首先是自然数,以后是整数、小数和分数,随之从有理数发展到无理数,又从实数发展到复数等等。而且,不但引进了十进制数,还引进了二进制数(包括八进制、十六进制等)。除了数值型数据外,还引进了非数值型数据(这些数据通常也称为字符串数据,例如,“NAME”,“名字”等等)。为了表示数据,引进了常量和变量等概念;为了加工处理(包括计算)数据分别引进了相应的运算,例如,对数值型数据可以施行加、减、乘、除、乘幂以及三角函数等各种运算,对非数值型数据可以施行并置,抽取,计数,替代等各种运算。随着发展,还会有新的数据类型以及相应运算出现。

这里的“数据处理”是狭义理解的,是针对数据处理领域(即事务处理领域)而言的。因为除了数据处理领域外,还有科技计算领域,软件生成领域,图象处理领域,人工智能领域等等,它们也是对“数据”进行“加工、处理和转换”的。计算机是加工数据的重要工具,为了能够针对不同领域的特点而更有效、更方便地使用计算机,人们在不同的领域中常常使用不同的程序设计语言。这些程序设计语言可能是通用的(如科技计算用的 FORTRAN, ALGOL, 数据处理用的 COBOL 等等),也可能是专用的,当然,所谓“通用”和“专用”是相对的。

相对地讲,科技计算方面的问题往往原始数据较少,但算法复杂,而数据处理方面的问题却常常需要对大量的初始数据进行加工,加工过程是一些极其简单的运算(有时甚至不执行计算)或逻辑判断,但需要重复许多次,最后输出大量的结果数据,这些数据往往还是一些表格。这种数据的大量和可组织性及加工过程的简单和重复性、构成了这类问题的主要特性。

在数据处理领域中,对数据的加工流程可能是成批的,也可能是零星的和随机的。数据通常以记录为处理单位和以块为传输单位构成文件形式。文件和记录在数据处理中具有很重要的地位。

针对不同领域发展起来的程序设计语言,尽管有其不同的特点,但也有许多共同点。而且随着用户要求的不断提出,计算机体系和系统的发展,应用领域的相互渗透和拓宽,软件技术的发展等因素,程序设计语言也在随之发展和相互渗透。常用程序设计语言的发展,明显地反映出版本的兼容性和功能的汇集型,同时要求有更有力的支撑系统和环境来满足其发展要求。

COBOL 程序设计语言(COMmon Business Oriented Language)是国际上普遍使用的数据处理中通用的程序设计语言。它在数据处理领域中应用最为广泛,发展历史最早,标准化也是最早的,而且至今仍是在“积极”发展中的程序设计语言。由于数据处理领域在使用计算机的领域中占的比例很大,因此 COBOL 语言的使用范围是很广泛的。COBOL 语言具有“国际性”,当然在不同的国家和不同的计算机实现系统中可能有某些不同。例如,扩充其计算机字符集,扩充

某些功能,对标准文本中的某些规定进行补充说明以及增加某些限制等等.除 COBOL 语言外,还有一些用于数据处理的程序设计语言(例如 RPG),但都远不如它重要和使用广泛.

本书对我国国家标准 COBOL 程序设计语言(它等效采用于 ISO COBOL₇₈)的内容进行较系统和较全面地介绍,同时根据实用要求和难易程度进行了不同深度的叙述.

1.2. COBOL 语言发展概况

COBOL 语言从开始形成到现在一直在发展中,它是“活”的程序设计语言之一.

最早的 COBOL 语言文本是在 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.

随后,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,经过修改后又提出了一个文本,称为 COBOL₆₁,于 1961 年发表.

下一个 COBOL 版本是由维护委员会提出的,并于 1963 年发表,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,计划委员会和系统委员会.程序设计语言委员会(PLC)的目的与任务,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,且由程序设计语言委员会发表了 COBOL₇₆.此外,最近又有新的 COBOL 版本出现.

上述各种版本是发展版本,是 COBOL 语言的开发情况.标准 COBOL 语言与之有所不同.国际标准 COBOL 语言有两个,即原有的推荐标准版本 ISO COBOL₇₂ 和当前(代替 ISO COBOL₇₂)的标准版本 ISO COBOL₇₈. ISO COBOL₇₂ 是 ANSI COBOL₆₈,而 ISO COBOL₇₈ 是 ANSI COBOL₇₄.随着发展,必将还会有新的国际标准 COBOL 版本出现.

各国国家标准 COBOL 语言大多数是国际标准 COBOL 语言的等效采用.我国的国家标准 COBOL 语言的第一个版本是 ISO COBOL₇₈ 的等效采用版本.

有的国家在 COBOL 语言的计算机字符集中增加了一些其他的符号(如日本的标准版本中有片假名等).由于我国普遍使用的汉字有较大的特殊性以及其他各种因素,在目前的我国国家标准 COBOL 中尚未引入新符号,这有待于今后当时机成熟时再进行考虑.

从 COBOL 语言的各种版本中可以看到 COBOL 语言在发展中的某些特点.例如,在文本上是兼容的,即功能在不断地扩大而删去的极少,而且原有的 COBOL 程序仍可在新的版本下运行.在功能上是汇集的,即语言中提供各种各样的功能供用户选用.这些特点几乎在其它通用的程序设计语言的发展中也同样存在.

其修改版本的原因看来至少有下列几点:

计算机系统的发展;

外设种类的增加;
程序设计方法学等方面的发展;
各种系统软件的相互影响;
支撑系统 and 环境的发展;
应用领域的扩大和用户的需要;
各版本的深入研究等等.

目前, COBOL 语言文本已经相当大, 例如, ISO COBOL₇₈ 全文有近 600 页, 而且新的版本有愈来愈大的趋势。

1.3. 我国国家标准 COBOL 语言概述

ISO COBOL₇₈ 是 COBOL 语言当前的国家标准. 我国国家标准 COBOL 就是它的等效采用版本. 因此首先对它进行概貌介绍, 以便对其有一些全面的当然也还是很粗浅的认识.

ISO COBOL₇₂ 和 ISO COBOL₇₈ 均是由一个核心和若干个功能模块组成. 某一特定计算机上配置的某个 COBOL 语言将根据可能和需要挑选核心的某一级和各功能模块的某一级组成.

ISO COBOL₇₂ 是由核心和七个功能模块组成的. 这七个功能模块分别是表处理, 顺序存取, 随机存取, 排序, 报表编制, 程序分段和库.

ISO COBOL₇₈ 是替代 ISO COBOL₇₂ 的国际标准, 并且有相当大的改动. 它由一个核心和十一个功能模块组成. 这十一个功能模块分别为:

核心(NUC, NUCLEUS)

表处理(TBL, TABLE HANDLING)

顺序 I—O(SEQ, SEQUENTIAL I—O)

相对 I—O(REL, RELATIVE I—O)

索引 I—O(INX, INDEXED I—O)

排序合并(SRT, SORT—MERGE)

报表编制(RPW, REPORT WRITER)

程序分段(SEG, SEGMENTATION)

库(LIB, LIBRARY)

排错(DEB, DEBUG)

程序间的通讯(IPC, INTER—PROGRAM COMMUNICATION)

通讯(COM, COMMUNICATION)

COBOL 的最小子集是由 1 级核心, 1 级表处理, 1 级顺序 I—O 组成的. 而 COBOL 的全集是由 2 级核心, 2 级表处理, 2 级顺序 I—O, 2 级相对 I—O, 2 级索引 I—O, 2 级排序合并, 1 级报表编制, 2 级程序分段, 2 级库, 2 级排错, 2 级程序间的通讯, 2 级通讯组成.

国家标准 COBOL 语言的结构图见表 1.1 (因为是等效采用), 因此也是 ISO COBOL₇₈ 的

功 能 模 块											
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	程序间的 通信	通 信
2NUC 1, 2	2TBL 1, 2	2SEQ 1, 2	2REL 0, 2	2INX 0, 2	2SRT 0, 2	1RPW 0, 2	2SEG 0, 2	2LIB 0, 2	2DEB 0, 2	2IPC 0, 2	2COM 0, 2
			1REL 0, 2	1INX 0, 2	1SRT 0, 2		1SEG 0, 2	1LIB 0, 2	1DEB 0, 2	1IPC 0, 2	1COM 0, 2
1NUC 1, 2	1TBL 1, 2	1SEQ 1, 2	空	空	空	空	空	空	空	空	空

表中的符号说明：中间的三个字母表示模块名称；

字母左边的数字表示级号，下面的两个数字表示最低级和最高级的级号。

例如：2NUC 1, 2 表示对应的模块为核心，该模块由 1 级或 2 级组成，该级是核心 2 级。

“空”表示该级为空级。

表 1.1 国家标准 COBOL 语言的结构图

结构图)。

核心模块分成两级，它是 COBOL 语言不可缺少的部分。它主要用于完成在中央处理器和内存中进行加工处理的工作。例如可以完成对一个数据项或一个记录的传送、加工和修改；有加减乘除等数值运算功能，还有非数值运算功能和比较运算功能；另外还含有各种控制功能(如 IF 语句, PERFORM 语句, STOP 语句, GO TO 语句)等等。1 级的功能较小，并且是 2 级功能的真子集。

表处理模块分成两级，它也是不可缺少的部分。表从某种意义上讲相当于“数组”，即由相同类型的数据项所组成的量。对表需要定义，指明大小及其他的属性。对表的引用可以是表元素，也可以是表。为了标识表元素，不但可以使用如通常所理解的下标，还可以使用位标(它对应于表元素的物理位置的相对始址)。因此，可以理解为：下标对应于“按名”编位，而“位标”对应于“按址”编位。此外还提供了查找语句来完成查找表元素的功能。

顺序 I—O 模块分成两级，它也是 COBOL 语言中不可缺少的部分。在 COBOL 中，无论是输入数据或是输出数据，除了极少数例外以外，总是组织成“文件-记录-组项-初等项”的形式。无论是顺序存取的外设，或者是能随机存取的外设，均可以以顺序方式组织数据。例如，行式打印机是最常见的外设，它能逐行打印，还具有走纸格式等功能，对应于它的文件是顺序文件。对于每一文件，需要对文件及其记录等进行定义和描述，同时需要采用打开，读写(可能还有重写)关闭的操作流程。

相对 I—O 模块分成三级。最小一级为 0 级，它是空集。凡出现空集的功能模块，则表示该模块的功能可以不被选取。相对文件必须建立在大容量存储设备上(亦称海量存储器或大规模存储器，例如磁盘即属此种存储设备)。每一个记录用一个相对号来标识，该相对号从 1 开始，依次加 1。对这种文件可以使用顺序存取方式、随机存取方式及动态存取方式。存取时根据其相对

号进行。

索引 I—O 模块分成三级, 最小一级为空集。索引文件也是常见的很有用的文件, 它必须建立在大容量存储设备上, 其记录需由记录中的索引部分来标识。索引是记录键数据项的值, 且根据它来对索引文件的记录进行存取修改等各种操作。索引文件的存取方法通常是随机的, 如果索引文件的记录已是有序的, 则还可以是顺序的和动态的。

排序合并模块分成三级, 最小一级为空集。这是很有用的功能模块。实际上, 这是采用通过语言中的语句来实现用户所要的标准功能。通过排序语句, 可以将一个或几个文件按照用户指定的排序要求, 将所有记录整理成一个有序的文件。还允许用户在排序前以及排序后进行某些处理。通过合并语句可将相同性质和相同排序要求的已有序的若干个文件合并成仍是原来排序要求的一个文件。排序要求是由用户指定的一串排序键来指明的。

报表编制模块分成两级, 最小一级为空集。编制表格是数据处理中经常需要做的工作, 有时需要的报表比较简单, 有时却需要十分复杂的报表。这个模块提供的功能使用户能方便地编制复杂的报表, 因为许多原来需要由用户描述的细节将由系统来完成。不过值得一提的是没有报表编制模块仍然可以编制报表, 只是用户需要考虑打印报表的所有细节(包括, 编排格式, 走纸形式等等)。通常对于简单的报表往往用户不使用报表编制功能, 而是由用户自己考虑全部细节。

程序分段模块分成三级, 最小一级为空集。COBOL 语言提供了将目标程序分成不可覆盖的程序段和可覆盖的程序段的表示方法和用于覆盖功能的设施。用户通过段号来提供覆盖的信息, 而由系统来实现程序段的覆盖。这样有利于在较小的内存中解算较大的问题。由于目标程序可以允许覆盖, 因此对相应的源程序需要有一定的限制。

库模块分成三级, 最小一级为空集。在 COBOL 语言中“库”设施提供了这样一种功能, 即在编译时可以从正文库把指定的原文复写到源程序中。不过需要有一个(或几个)COBOL 正文库来支撑, 以备由用户用 COPY 语句来引用, 从而实现在源程序一级上的替换插入。由于 COBOL 语言的使用范围很广泛, 对某些用户建立一个或若干个正文库是有利的, 从而可以减少设计和调试的时间, 有利于用户编制源程序。

排错模块分成三级, 最小一级为空集。排错有时也称为调试, 在解题过程中, 从问题的抽象到得到最后结果之间的各个阶段均可能出现错误, 排除各种错误当然需要有不同的方法。对程序而言, 排错主要指编译时的排错和运行时的排错。运行时的排错涉及的方面较多, 致错的原因更是多种多样的。COBOL 的排错模块为用户提供了在目标程序执行时能包含监督数据项和过程在内的排错算法的工具, 它提供了编译时开关和目标时开关, 用于排错的语句以及排错行、排错项等设施。

程序间的通信模块分成三级, 最小一级为空集。程序间的通信是一个很有用的功能模块, 它涉及到程序的结构。它通过划分调用程序和被调程序, 以及引进 USING 短语和连接节 LINKAGE SECTION 来完成两者之间数据的沟通共享, 通过调用程序中的 CALL 语句转入被调程序, 在被调程序中遇见 EXIT PROGRAM 语句返回调用程序的方式来完成流程的连接。

通信模块分成三级, 最小一级为空集。该模块是为了由 COBOL 程序使用网(本地的或

远程的)而设置,它需要有其他软件的支撑。该模块对接口、消息、消息传输动作加以描述。

1.4. 样 本 程 序

COBOL 语言文本或者相应的参考手册给用户提供了写出正确源程序的依据。当然除了需有正确的源程序外,尚需要有正确的而且合乎源程序描述和组织的数据,才有可能算得正确的结果。

COBOL 程序的书写有一定的格式,而且比较严格。

COBOL 程序由固定次序的四个部组成,即不能少也不能多,也不可调动次序。这四个部是

标识部 (IDENTIFICATION DIVISION)

环境部 (ENVIRONMENT DIVISION)

数据部 (DATA DIVISION)

过程部 (PROCEDURE DIVISION)

在每一个部中有节和段,还有句子和语句,子句和短语,最基本的元素是符号,符号又是由字符组成的。

源程序的表示方法类似于英语,因此对英语作为自然语言的用户有着易读易记的明显优点,但对于我们以汉语为自然语言的用户也许有些冗长和难记。整个 COBOL 语言有 300 多个保留字,虽然其表示接近于自然语言,但其规定还是相当严格的,用户是不能随意改动的。

下面举二个例子。第一个例子非常简单,目的仅用来说明 COBOL 源程序的形式。第二个例子介绍得较为详细些,目的在于使读者对用 COBOL 语言解题和 COBOL 源程序的结构有一个较为清晰的概貌,因此如果开始不能全部接受的话,可以留待以后适当时候再来参阅。

例 1

```
IDENTIFICATION DIVISION. } 标识部
PROGRAM-ID. EXAMPLE1.

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.      } 环境部
SOURCE-COMPUTER. DJSXXX.
OBJECT-COMPUTER. DJSXXX.

DATA DIVISION.
WORKING-STORAGE SECTION.    } 数据部
77 XY PIC 999.

PROCEDURE DIVISION.

    ACCEPT XY.
    ADD 10 TO XY.
    DISPLAY "XY= ", XY.      } 过程部
    STOP RUN.
```


这个例子中引进的变量为 XY, 它的值是具有三位十进制数字的正整数。通过 ACCEPT 语句, 使 XY 从键盘上获得值, 然后加上 10, 再从显示设备上显示出来。整个程序极为简单, 不再多说明了。

例 2 在实际应用中经常会遇见如计算工资一类问题, 这种问题具有一定的典型意义, 这里所举的例子是其简化情况。

每个职工的工资情况由一张卡片来表明, 因此所有职工的工资情况由一叠卡片来表明, 这一叠卡片构成一个卡片文件。换句话讲, 输入数据组成一个文件, 一张卡片构成该文件的一个记录。每张卡片有 80 列, 其数据安排如表 1. 2 (其中左边 40 列已被使用, 而右边 40 列未使用)。

1 列—6 列 职工号 6 位数字 9(6)	7 列—26 列 姓名 20 位字母数字 X(20)	27 列—31 列 基本工资 5 位数字 9(5)	32 列—35 列 附加工资 4 位数字 9(4)	36 列—40 列 扣除 5 位数字 9(5)	41 列—80 列 空白 40 位字母数字 X(40)
---------------------------------	-------------------------------------	------------------------------------	------------------------------------	----------------------------------	--------------------------------------

表 1. 2

通过上述每张卡片, 即可算得每个职工的支付额。其计算公式为

支付额 = 基本工资 + 附加工资 - 扣除

如果要求将每个职工的支付额连同该职工的职工号和姓名组成一个记录, 则所有这些记录构成一个文件。该文件存储在磁带上, 因此它是一个磁带文件。这个记录的数据安排如表 1. 3。

1 位—6 位 职工号 6 位数字 9(6)	7 位—26 位 姓名 20 位字母数字 X(20)	27 位—32 位 支付额 6 位数字 9(6)
---------------------------------	-------------------------------------	-----------------------------------

表 1. 3

其中, 9(4), 9(5), 9(6), X(20), X(40) 是相应数据项的 PICTURE 描述表示。

支付额的“6 位数字”是根据极端情况计算出来的。因为从允许情况看, 设基本工资为 99999, 附加工资为 9999, 而扣除为 00000, 则支付额为 $99999 + 9999 - 00000$, 即为 199998, 故支付额需要有 6 位数字位才会不产生“溢出”现象。

解法的流程见图 1. 1。

用 COBOL 语言编制的源程序为:

001010 IDENTIFICATION DIVISION.
001020 PROGRAM-ID. SAMPLE.
001030 ENVIRONMENT DIVISION.
001040 CONFIGURATION SECTION.
001050 SOURCE-COMPUTER. DJS × × ×.
001060 OBJECT-COMPUTER. DJS × × ×.
001070 INPUT-OUTPUT SECTION.

identification