

COBOL

程序设计

候 炳 辉

清 华 大 学 出 版 社

COBOL 程 序 设 计

侯 炳 辉 编

清华大学出版社
1982

内 容 简 介

本书是作者在清华大学经济管理工程系、计算机科学与工程系大学生、研究生的教材及应用学习班、干部进修班的讲稿基础上写成的，全书共分九章，一至四章为核心部分，五章为表处理，六章文件处理，七章排序，八章应用程序，九章上机操作说明。本书内容充实，通俗易懂，使读者不感到因 COBOL 内容繁杂而畏难止步。可供读者自学。

本书有大量应用程序供阅读、模仿或直接使用。既适合于大中型计算机，也适合于微型计算机。

读者对象：大专院校师生、从事计算机软、硬件工作者、各经济部门、规划设计及企业管理部门、工交财贸、人事档案和情报检索部门的工作人员。

COBOL 程序设计

侯炳辉 编

俞鑫祥 安树兰 校

☆

清华大学出版社出版

北京 海淀 清华园

北京景山学校印刷厂印刷

新华书店北京发行所发行·各地新华书店经售

☆

开本：787×1092 1/16 印张：15 3/4 字数：401 千字

1982年1月第一版 1982年1月第一次印刷

印数：1~20,000

统一书号：15235·27

定价：1.75 元

前 言

计算机最广阔的应用领域是事务处理。企业管理、工交财贸、银行会计、人事档案、情报检索……凡是需要处理数据和文件的地方都可用计算机管理。COBOL是面向事务处理的语言，由于它具有极强的数据处理能力，已无可争辩地成为国际上应用最广泛的一种高级语言。因此，推广计算机在管理中的应用，就首先必须大力推广、普及COBOL语言。

近年来，我国在推广应用COBOL语言已做了大量的工作，但总的说来还不如其它高级语言那样被人熟悉。高等院校以及社会上迫切需要一本适合于教学及自学的教材。教研组以及其他同志建议我将我给经济管理工程系、计算机科学与工程系讲课的讲义整理成书，以急读者所需。

毫无疑问，学习一种计算机语言的根本目的是学会程序设计，作者在教学和编写教材时始终遵循这个原则，当然也应严格遵循COBOL本身的语法规则。本书不是一本纯粹的计算机语言的书，也不是纯粹的一本手册，而是一本“COBOL程序设计”的教材。无论如何，本书首先强调的是应用性。

全书共分九章，一至四章为COBOL核心，第五章为表处理，第六章文件处理，第七章COBOL的排序，第八章应用程序，第九章上机操作说明，还有三个附录。讲授时不一定按上述次序，最好把第六章的前两节移到第二章前面讲，也可根据需要，对内容有所增减，一般学时可在40—50之间。

本书曾得到清华大学计算机科学与工程系软件教研组俞盘祥同志以及清华大学图书馆计算机室安树兰同志的详细审阅和修改；在教学和写作过程中，还得到教师王锡清同志、研究生汪授泓、王立国同志的大力协助，教研组负责人薛华成副教授为本书的写作和出版做了许多工作，系资料室及系办公室的同志提供了许多方便条件，对所有这些同志，作者谨致深深的谢意。

由于本人学识浅薄，书中肯定有许多缺点错误，敬希从事COBOL的专家、学者以及一切读者指正。

编 者

1981年8月于清华园

目 录

概述

第一章 COBOL 的基本概念	5
§ 1.1 COBOL 的程序结构	5
§ 1.2 COBOL 字符集	6
§ 1.3 直接量 (或引用字 Literal)	7
1.3.1 数字直接量 (数字常数)	7
1.3.2 非数字直接量 (非数字常数)	8
1.3.3 特定直接量 (或赋形常数 Figurative Constant)	8
§ 1.4 COBOL 字	9
1.4.1 COBOL 字的组成	9
1.4.2 保留字 (Reserved words)	9
1.4.3 用户字 (User-defined words)	10
§ 1.5 COBOL 源程序书写格式	10
§ 1.6 几点必要的约定	11
习题	12
第二章 标识部及设备部	13
§ 2.1 标识部 (IDENTIFICATION DIVISION)	13
§ 2.2 设备部 (ENVIRONMENT DIVISION)	14
2.2.1 外形节 (CONFIGURATION SECTION)	14
2.2.2 输入-输出节 (INPUT-OUTPUT SECTION)	15
第三章 数据部 (DATA DIVISION)	18
§ 3.1 几个与数据有关的概念	18
3.1.1 记录和数据项的概念	18
3.1.2 数据记录与文件的概念	20
3.1.3 数据在内存中的表示	20
§ 3.2 数据部的结构	22
§ 3.3 文件节	22
3.3.1 文件描述体	22
3.3.1.1 LABEL RECORD 与 VALUE OF 子句	23
3.3.1.2 RECORDING MODE 子句	23
3.3.1.3 BLOCK CONTAINS 子句	23
3.3.1.4 RECORD CONTAINS 子句	24
3.3.1.5 DATA RECORD 子句	24
3.3.2 记录描述体	24
3.3.3 数据描述体	25

§ 3.4 与数据描述体有关的子句.....	26
3.4.1 字型子句 (PICTURE 子句)	26
3.4.1.1 描述数字数据的字型字符.....	27
3.4.1.2 描述字母 (文字) 数据的字型字符.....	28
3.4.1.3 描述字符数据的字型字符.....	28
3.4.1.4 描述编辑数据的字型字符.....	28
习题	31
3.4.2 用法子句 (USAGE子句)	32
3.4.2.1 USAGE IS DISPLAY.....	32
3.4.2.2 USAGE IS { COMPUTATIONAL COMP }	34
3.4.2.3 USAGE IS { COMPUTATIONAL-1 COMP-1 }	34
3.4.2.4 USAGE IS { COMPUTATIONAL-2 COMP-2 }	35
3.4.2.5 USAGE IS { COMPUTATIONAL-3 COMP-3 }	35
3.4.2.6 USAGE IS INDEX	35
3.4.3 赋初值子句 (VALUE 子句)	35
3.4.4 对位子句 (JUSTIFIED 子句).....	37
3.4.5 重复子句 (OCCURS 子句).....	37
3.4.6 重定义子句 (REDEFINES 子句).....	39
3.4.7 重命名子句 (RENAMES 子句).....	41
3.4.8 遇零置空 (BLANK WHEN ZERO)及同步安置 (SYNCHRONIZED) 子句.....	42
3.4.8.1 遇零置空子句.....	42
3.4.8.2 同步安置子句.....	42
§ 3.5 工作存储节 (WORKING-STORAGE SECTION)	44
§ 3.6 连接节 (LINKAGESECTION)	46
§ 3.7 标识符 (IDENTIFIER).....	46
3.7.1 数据名的不唯一性	46
3.7.2 用下标法或指标法使数据名单义	46
3.7.3 对数据名进行限定	47
3.7.4 标识符的格式	47
§ 3.8 一个完整的COBOL 程序——实例 1	48
习题	49
第四章 过程部 (PROCEDURE DIVISION)	51
§ 4.1 过程部的作用及其结构	51
§ 4.2 输入/输出语句 (I/O 语句)	52
4.2.1 打开语句 (OPEN Statement)	52

4.2.2	读语句 (READ Statement)	52
4.2.3	写语句 (WRITE Statement)	53
4.2.4	关闭语句 (CLOSE Statement)	54
4.2.5	接收语句 (ACCEPT Statement)	54
4.2.6	显示语句 (DISPLAY Statement)	56
4.2.7	重写语句 (REWRITE Statement)	57
4.2.8	删除语句 (DELETE Statement)	57
§ 4.3	停止与转移语句	58
4.3.1	停止语句 (STOP Statement)	58
4.3.2	转移语句 (GO TO Statement)	58
§ 4.4	传送语句 (MOVE Statement)	60
4.4.1	格式 1 —— 标识符传送到标识符	60
4.4.2	格式 2 —— 直接量传送到标识符	65
4.4.3	格式 3 —— 对应传送	65
4.4.4	各种数据的传送规则小结	66
§ 4.5	算术语句	68
4.5.1	舍入短语 (ROUNDED Phrase)	68
4.5.2	容量溢出短语 (ON SIZE ERROR Phrase)	68
4.5.3	留余短语 (REMAINDER Phrase)	68
4.5.4	加法语句 (ADD Statement)	69
4.5.5	减法语句 (SUBTRACT Statement)	72
4.5.6	乘法语句 (MULTIPLY Statement)	73
4.5.7	除法语句 (DIVIDE Statement)	74
4.5.8	计算语句 (COMPUTE Statemet)	75
	习题	76
§ 4.6	简单程序设计实例——实例 2	77
§ 4.7	条件和分支	79
4.7.1	条件及其种类	79
4.7.1.1	关系条件	79
4.7.1.2	符号条件	80
4.7.1.3	类型条件	80
4.7.1.4	条件名条件	81
4.7.2	条件语句与分支程序	84
4.7.3	嵌套条件与复合条件	86
4.7.3.1	条件语句的嵌套	86
4.7.3.2	复合条件	88
4.7.3.3	复杂分支程序设计实例——实例 3	88
	习题	90
§ 4.8	循环与子程序	92

4.8.1 执行语句 (PERFORM Statement)	92
4.8.2 出口语句 (EXIT Statement)	99
4.8.3 调用语句 (CALL Statement)	99
4.8.4 子程序的两种调用方法	100
4.8.5 子程序应用实例——实例 4	104
§ 4.9 字符串操作语句——检验和转换语句	108
4.9.1 检验语句 (EXAMINE Statement)	108
4.9.2 转换语句 (TRANSFORM Statement)	110
第五章 表处理	112
§ 5.1 引言	112
§ 5.2 表的建立及引用	113
5.2.1 一维表的建立及引用	113
5.2.2 二维表的建立及引用	113
5.2.3 三维表的建立及引用	114
5.2.4 表元素引用时注意事项	114
5.2.5 可变长表的建立	118
§ 5.3 表处理实例——实例 5	119
第六章 文件的处理	123
§ 6.1 文件的基本概念	123
6.1.1 信息、数据、记录、文件	123
6.1.2 文件的种类	123
§ 6.2 文件的外部介质	125
§ 6.3 与文件有关的描述和语句	126
6.3.1 顺序组织文件	126
6.3.2 索引顺序组织文件	130
6.3.3 直接组织文件	132
§ 6.4 CROMEMCO 微型机中磁盘文件的有关描述及语句	137
6.4.1 磁盘顺序文件的描述和语句	137
6.4.2 磁盘索引文件的描述和语句	138
§ 6.5 关于相对文件的概念	141
第七章 COBOL 的排序	143
本章提要	143
§ 7.1 排序过程与排序中间文件	143
7.1.1 排序过程	143
7.1.2 排序中间文件的文件描述与记录描述	144
§ 7.2 排序语句 (SORT Statement)	145
7.2.1 排序语句的一般格式	145
7.2.2 对排序语句的说明	146
§ 7.3 自动方式的排序	147

§ 7.4 排序程序实例: 将各国英文计算机期刊名按字母为序进行排序——实例 6	148
第八章 经济与管理应用程序	153
引言	153
§ 8.1 一个纯计算程序——用 COBOL 语言计算两矩阵相乘	153
§ 8.2 银行业务——银行本利和计算	158
§ 8.3 统计问题——统计和打印一年中各产品的最高月产量	162
§ 8.4 库存问题——建立一个库存文件并打印出来	164
§ 8.5 商业管理——建立商品购、销、及利润的磁盘文件	167
§ 8.6 生产管理——建立在制品台账的程序	169
§ 8.7 工资管理——计算并打印工资表	172
§ 8.8 人事管理——建立人事文件、统计和打印各项人事数据	178
§ 8.9 成本管理——产品的成本、工时、计划文件的建立及成本、工时的汇总	192
§ 8.10 磁盘文件的建立、存取及修改	219
8.10.1 磁盘索引文件的建立	219
8.10.2 读(取)磁盘文件的程序	222
8.10.3 建立调试程序(包括读记录、插入记录、重写记录、删文件记录)	222
第九章 COBOL 程序的上机操作	227
§ 9.1 在 FELIX C-256 机上的上机操作	227
9.1.1 源程序的书写	227
9.1.2 如何检查穿孔卡片	227
9.1.3 控制卡片简介	227
§ 9.2 在 CROMEMCO 微型机上的上机操作简介	230
9.2.1 起动	230
9.2.2 建立文件	231
9.2.3 编译源程序	231
9.2.4 连接运行	231
9.2.5 文件修改	232
9.2.6 删除磁盘文件	233
9.2.7 在 CDOS 状态下打印文件清单	233
9.2.8 文件易名和“拷贝”	233
9.2.9 COBOL 需要的系统软件及存贮量	233
9.2.10 注意事项	234
§ 9.3 关于程序编制的步骤及几个注意事项	234
9.3.1 编制源程序的步骤	234
9.3.2 编制程序时几个注意事项	235
附录 1 FELIX C-256 COBOL 保留字表	236
附录 2 CROMEMCO COBOL 保留字表	239
附录 3 常用字符的 EBCDIC 码	242
主要参考书	243

概 述

一、COBOL 的历史简述

COBOL 为 Common Business Oriented Language 的缩写, 意为面向企业的通用语言, 或直译为通用商业语言, 这个语言是美国政府与工商业者通力合作所建立起来的一种标准商务资料处理的程序语言。1959 年在美国国防部的协助下, 成立一个委员会, 要发展一种可以应用于各种型号计算机的程序语言。1960 年 4 月资料系统语言会议提出 COBOL 初稿, 1960 年完成 COBOL 规范, 并由 CODASYL (Conference on Data Systems Language) 于 1961 年公布, 称 “COBOL-60”。次年, 出版修改版本 “COBOL-61”。1965 年编辑了更为精细的 COBOL, 称 “COBOL-Edition 1965”, 这个文本最后于 1968 年 8 月美国国家标准化协会 (ANSI) 通过为美国国家标准文本—ANSI COBOL X3.23-1968 (简称 ANSI COBOL-68), 1972 年国际标准化组织 (ISO) 通过了国际标准 COBOL 文本—ISO COBOL-72。ISO COBOL-72 实际上和 ANSI COBOL-68 大同小异, 该文本受到美、英、法、日、苏等 21 个会员国承认。与此同时, COBOL 继续发展, 新的文本接连在 1968、1969、1970 和 1973 年发表于 COBOL 发展杂志上。1974 年 ANSI 在 ANSI COBOL X3.23-1968 的基础上发展为 ANSI COBOL X 3.23-1974 文本, 1978 年 ISO 宣布 ANSI COBOL X3.23-1974 文本即为 ISO COBOL-78。

ISO COBOL-72 由一个核心和七个功能处理模块组成 (如表 1 实线部份所示), ISO COBOL-78 由一个核心和十一个功能处理模块组成 (见表 1)。核心和功能处理模块再进行分级: 1 级低于 2 级, 2 级低于 3 级。COBOL 语言的实施与不同的机器有关, 不同的机器配置不同的 COBOL 标准。最小的 COBOL 标准由一级核心、一级表处理及一级顺序存取模块组成, 最大的 COBOL 标准由核心及各功能处理模块的最高级组成。

本书作为一般教科书, 具有入门性质, 不可能介绍最高档 COBOL, 本书所举的程序实例大都是在 FELIX 及 CROMEMCO 微型机上实现的, 上述两种机器也没有全部实现 ANSI-COBOL 的标准。读者如果需要实施更高档的 COBOL, 可参考具体机器的 COBOL 文本及操作手册。

表 1 ISO COBOL 的语言结构图

核 心	功 能 处 理 模 块										
	表处理	顺序存取	随机存取	排 序	报表打印	程序分段	程序库	索引存取	调 试	通 信	程序间通信
2 级	3 级	2 级	2 级	2 级	2 级	2 级	2 级	2 级	2 级	2 级	2 级
	2 级		1 级	1 级	1 级	1 级	1 级	1 级	1 级	1 级	1 级
1 级	1 级	1 级	空	空	空	空	空	空	空	空	空

二、COBOL 所解决的对象——数据处理概述

COBOL 是面向数据处理的语言。什么叫数据？过去我们对数据的概念是一些数字，其实数字只是数据的一种类型，是数值化了的数据。而在数据处理中的数据包括下列内容：

数据 { 数值化了的数据，如实验数据，统计数据等。
文字数据。
图象数据。
声音数据。

所以在数据处理中的数据概念已大大扩展了，文字、图象、声音等都叫数据，都是数据处理的对象，不过这里讨论的对象不包括图象及声音的数据。

数据处理的任务是对各种类型的数据进行收集、存贮、传送、分类、排序、计算以及打印表报、输出图象等。由此可见，数据处理的内容很多，工作量很大，而 COBOL 语言正是由于数据处理的需要而产生的，所以它是世界上应用最广泛的一种语言。以日本为例，各种语言的使用情况如表 2 所示。

表 2

年 分 \ 语 言 使 用 百 分 比	汇 编	ALGOL	FORTRAN	COBOL	PL/1	其 它
1973	33.0	0.2	7.5	44.6	1.7	12.7
1974	27.0	0.3	8.0	44.8	2.4	17.5
1975	2.64	0.1	7.5	50.6	2.7	13.7
1976	21.7	0.1	8.2	53.8	3.9	12.3
1977	19.6	0.3	7.3	56.3	3.1	13.4

三、COBOL 程序设计的教学方法

学习好 COBOL 语言程序设计的关键之一是弄清数据处理中有关文件、记录、数据项等一系列概念。一般 COBOL 语言的书先讲 COBOL 核心部分，而把文件的概念放在后面讲，这样就会对学习语言本身带来很大困难，所以本书在讲授时应把第六章的有关文件处理的概念放在前面，当第一章基本概念及第二章的标识部及设备部讲完后，应立即讲第六章文件处理的前两节文件的基本概念以及文件的外部介质，同时再重复一下设备部的输入输出节文件控制段的内容，使能对设备部的这一部分搞得更清楚一些。

由于 COBOL 的内容很多，所以讲授一开始先介绍一下 COBOL 的特点、学习的方法，以及 COBOL 的全貌。本书把 COBOL 源程序的结构比作一棵树（实际上 COBOL 源程序的结构就是树形结构），我们暂且叫它“COBOL 树”（见图 1）。

在“COBOL 树”上，靠近树根的是标识部，依次向树梢上去是设备部、数据部、过程部。

COBOL 和 BASIC、FORTRAN 等语言有很大的不同，它既严格（有许多繁琐的规定）又灵活（有很多任选要求），而且并不象 BASIC 及 FORTRAN 那样，学习几条语句就可以进行简单的程序设计，可以边学习边编制程序边上机练习。而 COBOL 非要在 4 个部以及其

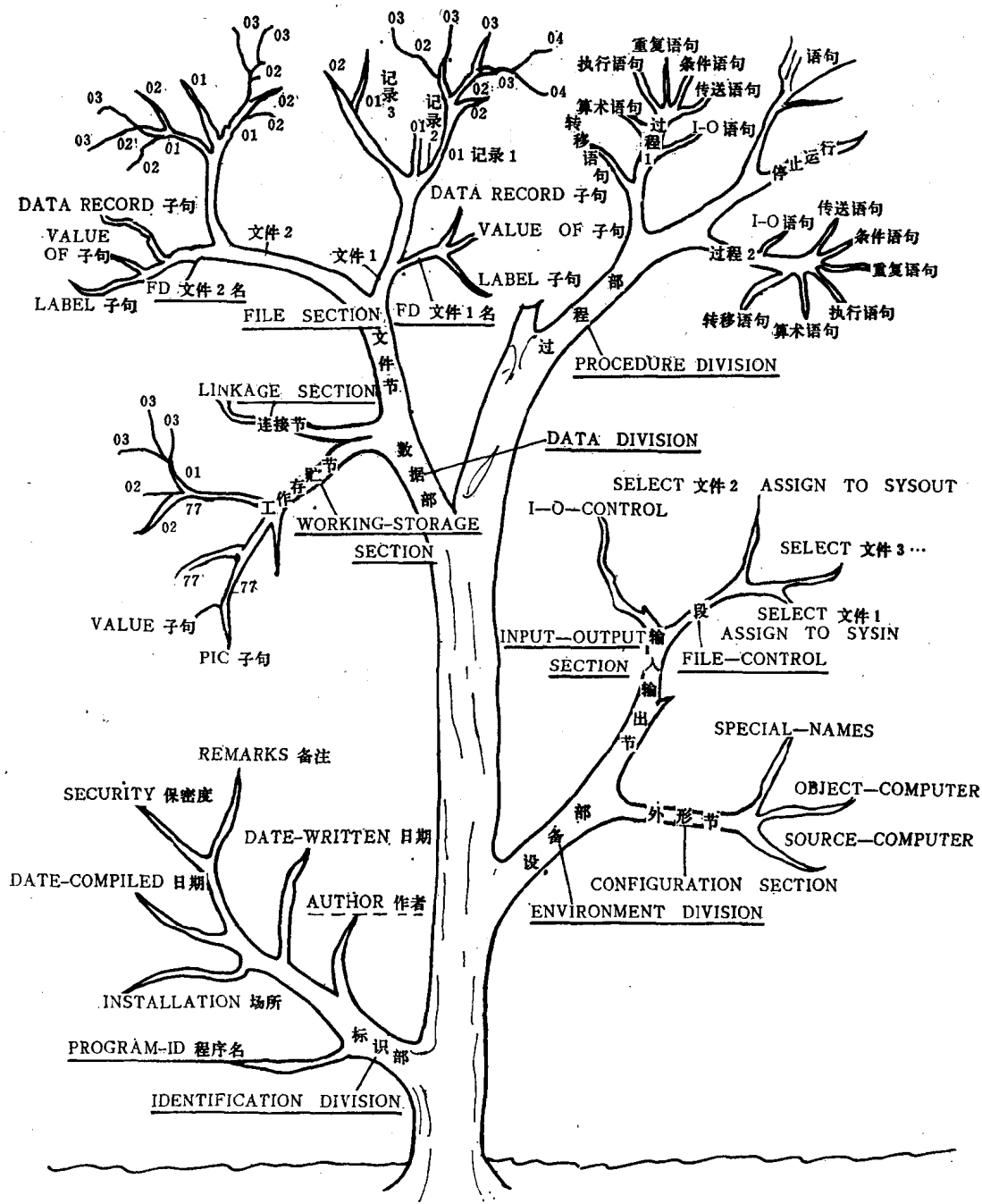


图1 COBOL 程序结构示意图

它内容学完以后才能编程序，所以学习 COBOL 就不可能很快见到效果，故也不易引起学生的兴趣。为了使学习 COBOL 的人确实能掌握内容，就必须化一定篇幅讲授程序实例以及上机操作，所以本书第八章列举了大量应用实例及其程序调试时遇到的各种问题的分析。正因为学习 COBOL 的最后目的是学会程序设计、解决问题，所以每个学生必须上机操作，故第九章关于 COBOL 的上机操作也是很实用和很重要的。

本书的上机操作主要以FELIX C-256(或FELIX C-512)以及微处理机 CROMEMCO 系统 2 和系统 3 作标准，所有应用程序的例子都已在上列两种机器上通过。对于其它机器的 COBOL 程序操作，可能稍有出入，读者可参考具体机器的使用手册。

第一章 COBOL 的基本概念

§ 1.1 COBOL 的程序结构

COBOL 和其它语言最大的区别是它更接近于英语语言。一个 COBOL 程序也确实如象一篇文章，有章有节，有段有句子。但是 COBOL 程序规定了必须有四个部 (DIVISION) 组成，这四个部是：

标识部 IDENTIFICATION DIVISION.

设备部 ENVIRONMENT DIVISION.

数据部 DATA DIVISION.

过程部 PROCEDURE DIVISION.

这四个部必须按上列次序出现，各部的下面分节 (SECTION)，节下面分段 (PARAGRAPH)，段下面有句子 (SENTENCE)，句子由语句 (STATEMENT) 或子句 (CLAUSE) 组成。而句子、语句、子句都由 COBOL 字 (WORDS) 组成，COBOL 字由 COBOL 字符 (CHARACTER) 组成。由此可见 COBOL 程序的结构是树形结构，它正象一棵树一样 (图 1—1)。

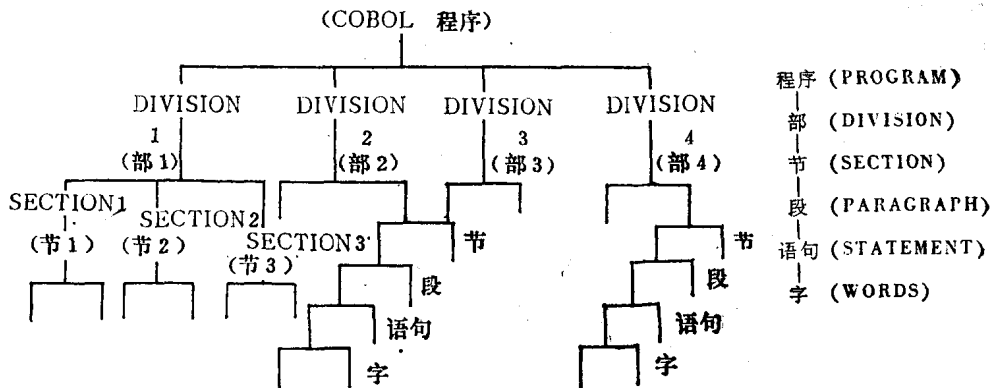


图1—1 COBOL 程序结构

COBOL 的标识部用于说明程序的名字、作者的姓名、编写与编译源程序的日期、设置计算机的场所、使用资料人员的限制以及其它需要说明的备注性内容，这象一棵树的根部枝叉上挂一块牌子，上写树的名字，栽种日期等等。标识部最重要的是程序说明段 (PROGRAM-ID 程序名)，每个程序是必写的。

第二个部是设备部，这个部是程序中唯一出现计算机硬件描述的地方。它标明程序中应用的计算机主机及外设的名称、文件及输入输出控制等。

第三个部叫数据部，这个部是用来描述一切数据 (输入输出数据及中间数据) 的数据结构。在整个 COBOL 程序中这一部分是既复杂又很繁琐的。

第四个部叫过程部，这是进行一系列处理和计算的部，也是最重要的一个部。

COBOL 规定，每个程序必须写上这四个部，部下面的一些节和段有些也是必写的。下面再详细介绍。COBOL 程序的分层结构也可以由图1—1来表示，这种结构图我们就叫它树形结构，它有四个部，部下面为节，节下面为段，段下面为语句（或子句），语句（或子句）又根据彼此间的逻辑关系构成句子（或描述体），语句和子句又由 COBOL 字或分隔符组成。

一个最简单的 COBOL 源程序

```
IDENTIFICATION DIVISION.  
PROGRAM-ID. X.  
ENVIRONMENT DIVISION.  
DATA DIVISION.  
PROCEDURE DIVISION.  
A. DISPLAY 'OK, IT IS GOOD'.  
STOP RUN.
```

这是一个最简单的源程序，程序名为 X 即使这样一个简单程序也要写上四个部 (DIVISION)，在过程部中仅有一个段 (A 段)，它的操作就只显示一句话“OK, IT IS GOOD”。

§ 1.2 COBOL 字符集

COBOL 字符共51个，即：

0—9	10个数字
A—Z	26个英文字母（大写）
+	加号
-	减号，连接号
*	乘号，星号
/	除号（左斜杠）
=	等号
,	逗号（特殊规定下可作小数点用）
.	句号，小数点
;	分号
'	单引号
(	左圆括号
)	右圆括号
<	小于号
>	大于号
□	空格（下面描述中用□作空格代号）
\$	钱币号

在FELIX C-256中还提供下列13个字符供打印输出使用，它们是：

： 冒号

!	惊叹号
[	左方括号
]	右方括号
←	左向箭头
↑	上向箭头
"	双引号
%	百分号
?	问号
&	和号
#	井号
\	右斜杠
@	

COBOL 源程序由字符串和分隔符组成，字符串是一系列连续字符，分别构成 COBOL 字、直接量、字形串或注释项。字符串与字符串之间用分隔符定界。

分隔符可分标点符、算术运算符、关系运算符。

标点符：□，；．'（）

算术运算符：+ - * / **（乘方）

关系运算符：< = >

在书写 COBOL 源程序时，必须遵守下列关于标点符号的使用法则：

（1）两个相邻字符串之间必须至少用一个空格分开。

（2）逗号与分号可以用来分隔一串子句或语句，但也可用空格分开。句号必须紧跟在一个句子或描述体之后。逗号、分号、句号作为分隔符使用时，其后必须至少有一空格，其前不得有空格。

（3）单边运算符（+，-）当出现数据名之前时，其后必须跟空格；在其出现在数字常数之前时，其后不得有空格，单边运算符之前必须有空格，而左括号可直接置于单边运算符之前，

例：	-□ABC	- 1	正确的
	-ABC	-□ 1	错误的

§ 1.3 直接量（或引用字 Literal）

直接量可以是数字常数、非数字常数及特定常数，把这些量放在内存单元中，以便直接引用，故叫它们为直接量或引用字。

1.3.1 数字直接量（数字常数）

数字直接量的有关规定：

- （1）必须为数字，长度不能超过18位（包括小数点及正、负号）。
- （2）负号在数字的最前端，负号与数字之间不得有空，小数点不得在数字的最右端。
- （3）不能包含引号。
- （4）可以作算术运算。

数字直接量的例子：4.93, +42.57, -15, 0, 256.2

1.3.2 非数字直接量（非数字常数）

非数字直接量是一组处于两个单引号之间，除去单引号本身外的字符组成的串。非数字直接量的有关规定如下：

（1）必须使用引号（FELIX 机为单引号，CROMEMCO机为双引号）。

（2）不能作计算用，只作显示（DISPLAY）用。

（3）长度不超过120位。

（4）除可以用任何 COBOL 字符外，还可使用保留字。

例：'77.50', 'COBOL-74', "F-CHB"

如果要把引号也作为非数字直接量的一部分，则在字符串（包括引号）的两边加上“QUOTE”。

例：QUOTE' TOTAL AMOUNT'QUOTE 此时非数字直接量的值为：

' TOTAL AMOUNT'

非数字直接量不能参加运算，即使由纯数字组成的非数字直接量也不能参加运算。例如 90.45 可以参加运算，而 '90.45' 不能参加运算。

1.3.3 特定直接量（或赋形常数Figurative Constant）

特定直接量又叫赋形常数或象征常数，是一种特殊的常数，这些常数有固定的名称及含义。特定直接量的单数和复数等价。

特定直接量共有 6 个：

（1）ZERO 或ZEROES、ZEROS,代表一个或若干个零。

（2）SPACE 或SPACES,代表一个或若干个空格。

（3）QUOTE 或QUOTES,代表一个或若干个单引号。

（4）HIGH-VALUE 或 HIGH-VALUES 代表在所有二进制位为“1”（FELIX 机），在 IBM360中代表一个或若干个最高数值“FF”（16进制位），也可能有些机器上为“99”等。

（5）LOW-VALUE 或LOW-VALUES 代表在所有二进制位上皆为“0”（FELIX机），在 IBM 360中为“00”（16进制位），也可能在有些机器上为“□□”等。

（6）ALL 直接量，代表一个单一的非数字直接量出现一次或一次以上。

例：假如数据A的长度为 5，则将特定直接量传输给A时，特定直接量的长度被认为是 5。即：

执行的语句	语句执行后A的值
MOVE ZERO TO A	0 0 0 0 0
MOVE SPACE TO A	□ □ □ □ □
MOVE QUOTE TO A	' ' ' ' '
MOVE LOW-VALUE TO A	所有二进制位皆为零
MOVE HIGH-VALUE TO A	所有二进制位皆为 1
MOVE ALL 'P' TO A	P P P P P