

COBOL

程序设计与应用

刘治山 黄振雄 编著
张秀兰 刘俊英

宇航出版社

内 容 简 介

本书介绍的 COBOL 语言, 是一种开发较早、用途较广、应用较普遍的计算机高级语言。前八章介绍了 COBOL 语言的基本知识, 以后各章介绍了各种功能模块, 并附有应用实例。

本书对初学者是一本很好的入门书和工具书; 对从事计算机应用的技术人员、管理人员及大专院校师生是一本较好的参考书。

COBOL 程序设计与应用

刘治山 黄振雄 编著
张秀兰 刘俊英

*

宇航出版社出版
新华书店北京发行所发行
各地新华书店经售
岳各庄印刷厂印刷

开本787×1092 1/16 印张24 1/4 字数530千字

1985年12月 第一版第一次印刷

印 数: 00,001—20,000册

统一书号: 15244·0001

定 价: 2.40元

目 录

第一章 引论	1
1.1 数据处理简介	1
1.2 COBOL 语言的产生与发展	2
1.3 COBOL 语言的特点	3
1.4 COBOL 程序的编译与运行	5
第二章 COBOL 源程序	7
2.1 程序的词义	7
2.1.1 名词解释	7
2.1.2 格式中使用的符号	7
2.1.3 字符集	8
2.2 源程序的格式	8
2.3 程序结构	12
2.4 语言结构	13
2.4.1 分隔符	14
2.4.2 COBOL 字	15
2.4.3 常字	18
2.5 算术表达式	18
2.5.1 算术表达式的形成规则	18
2.5.2 表达式求值	20
2.6 条件表达式	22
2.6.1 简单条件	22
2.6.2 组合条件	24
第三章 数据描述的概念	27
3.1 数据的描述	27
3.1.1 记录与文件的概念	27
3.1.2 数据的层次	27
3.1.3 表格的操作	28
3.1.4 数据类型	30
3.1.5 字符表示和基数的选择	30
3.1.6 代数符号	31
3.2 数据项和表的引用	31
3.2.1 用户定义字的限定	31
3.2.2 表项的标识	33
3.2.3 标识符的定义	36

第四章 标识部	38
第五章 环境部	39
5.1 配置节	40
5.2 输入输出节	45
第六章 数据部	54
6.1 数据部结构	54
6.2 文件节	55
6.2.1 文件描述体	56
6.2.1.1 BLOCK CONTAINS 子句	56
6.2.1.2 CODE-SET 子句	57
6.2.1.3 RECORD CONTAINS 子句	58
6.2.1.4 LABEL RECORD 子句	58
6.2.1.5 VALUE OF 子句	59
6.2.1.6 DATA RECORD 子句	60
6.2.1.7 LINAGE 子句	61
6.2.2 记录描述体	62
6.2.2.1 BLANK WHEN ZERO 子句	65
6.2.2.2 JUSTIFIED 子句	65
6.2.2.3 OCCURS 子句	66
6.2.2.4 PICTURE 子句	69
6.2.2.5 REDEFINES 子句	76
6.2.2.6 SIGN 子句	79
6.2.2.7 SYNCHRONIZED 子句	80
6.2.2.8 USAGE 子句	82
6.2.2.9 VALUE 子句	84
6.2.2.10 RENAMES 子句	87
6.2.2.11 CONDITION-NAME VALUE 子句	89
6.3 工作存贮节	91
6.4 连接节	92
第七章 过程部	94
7.1 算术语句	95
7.1.1 ADD (加) 语句	96
7.1.2 SUBTRACT (减) 语句	98
7.1.3 MULTIPLY (乘) 语句	100
7.1.4 DIVIDE (除) 语句	101
7.1.5 COMPUTE (计算) 语句	103
7.2 数据传送语句	103
7.2.1 MOVE (传送) 语句	103
7.2.2 INSPECT (检验) 语句	109
7.2.3 STRING (联串) 语句	116

7.2.4 UNSTRING (拆串) 语句	118
7.3 控制语句	123
7.3.1 GOTO (转向) 语句	123
7.3.2 ALTER (变更) 语句	125
7.3.3 PERFORM (执行) 语句	125
7.3.4 STOP (停) 语句	131
7.3.5 IF (条件) 语句	133
第八章 输入输出语句与USE 语句	135
8.1 ACCEPT (接受) 语句	135
8.2 DISPLAY (显示) 语句	137
8.3 OPEN (打开) 语句	137
8.4 CLOSE (关闭) 语句	139
8.5 READ (读) 语句	141
8.6 WRITE (写) 语句	145
8.7 REWRITE (重写) 语句	150
8.8 START (起始) 语句	152
8.9 DELETE (删除) 语句	155
8.10 USE (使用) 语句	156
第九章 程序设计的策略和风格	159
9.1 结构程序设计	159
9.1.1 结构程序设计基础	156
9.1.2 结构程序设计在 COBOL 中的实现	163
9.2 程序设计的目标	164
9.3 COBOL 程序设计实践	166
9.4 程序设计实例——两个文件的合并	169
第十章 表处理	174
10.1 概述	174
10.2 表的定义	174
10.3 表的结构与描述	175
10.4 可变长表	181
10.5 下标与变址	181
10.6 SET 语句	185
10.7 表的检索	186
10.7.1 SEARCH 语句格式之一	187
10.7.2 SEARCH 语句格式之二	191
10.8 散列表	194
10.9 往表中读数	197
第十一章 磁带和磁盘顺序文件	206
11.1 磁带	206

11.2 程序实例——建立一个带文件	207
11.3 磁盘	210
11.4 程序实例——建立一个盘文件	211
11.5 变长记录	213
第十二章 分类与合并	216
12.1 分类、合并功能	216
12.2 分类、合并功能有关的语句形式	218
12.2 分类语句	219
12.4 合并语句	220
12.5 一个程序实例	222
第十三章 索引文件处理	225
13.1 索引顺序存取方法(ISAM)	225
13.2 设计一个索引文件	228
13.3 虚拟存贮存取方法(VSAM)	235
13.4 设计一个 VSAM 文件	237
第十四章 相对文件处理	243
14.1 相对文件的设计	243
第十五章 报表生成	250
15.1 报表文件	250
15.1.1 文件描述的结构	250
15.1.2 报表子句	251
15.2 专用寄存器	251
15.2.1 页计数器	251
15.2.2 行计数器	252
15.3 报表节	252
15.3.1 报表描述项	252
15.3.1.1 CODE 子句	253
15.3.1.2 CONTROL 子句	253
15.3.1.3 PAGE 子句	254
15.3.2 报表栏描述项	256
15.3.2.1 LINE NUMBER 子句	259
15.3.2.2 NEXT GROUP 子句	260
15.3.2.3 TYPE 子句	260
15.3.2.4 BLANK WHEN ZERO 子句	266
15.3.2.5 COLUMN NUMBER 子句	267
15.3.2.6 GROUP INDICATE 子句	267
15.3.2.7 JUSTIFIED RIGHT 子句	268
15.3.2.8 PICTURE 子句	268
15.3.2.9 SOURCE 子句	268

15.3.2.10 SUM 子句	268
15.3.2.11 VALUE 子句	270
15.3.2.12 USAGE 子句	270
15.4 过程部的语句	271
15.4.1 INITIATE 语句	271
15.4.2 GENERATE 语句	271
15.4.3 TERMINATE 语句	272
15.4.4 SUPPRESS 语句	273
15.4.5 USE 语句	273
15.5 二个实例	274
第十六章 内部程序通讯	287
16.1 被调用的 COBOL 程序	287
16.2 CALL (调用) 语句	288
16.3 ENTER (入口) 语句	289
16.4 EXIT (出口) 语句	290
16.5 CANCEL (取消) 语句	291
16.6 实例	291
第十七章 库功能	298
17.1 COPY (拷贝) 语句	298
17.2 实例	300
第十八章 分段功能	302
18.1 程序分段概述	302
18.2 程序分段的结构	302
18.3 段限	303
18.4 对程序流的限制	303
第十九章 调试模块	305
19.1 调试模块的功能	305
19.2 开关和调试行	305
19.3 调试语句	306
第二十章 通讯	309
20.1 通讯概念	309
20.1.1 通讯功能	309
20.1.2 通讯环境	309
20.1.3 调用 COBOL 目标程序的方法	310
20.1.4 信息和队的概念	311
20.2 通讯节	313
20.2.1 CD 输入区	314
20.2.2 CD 输出区	318
20.3 通讯功能语句	321

20.3.1	ACCEPT 语句	321
20.3.2	DISABLE 语句	321
20.3.3	ENABLE 语句	322
20.3.4	RECEIVE 语句	323
20.3.5	SEND 语句	324
附录一	事务处理程序实例	327
1.	编制银行本利对照表	327
2.	计算推销员提成费并打印报表	331
3.	处理银行各种业务	338
4.	为某百货公司结算代销服务费和欠款总额	346
5.	建立职工薪金文件	352
附录二	COBOL 语句格式	355
附录三	COBOL 保留字	372
	参考资料	375
图 2-1	程序纸的格式	10
图 2-2	程序行各区界划分	9
图 2-3	语句结构	14
图 6-1	编辑与不同数据的组合	71
图 6-2	使用 +, -, ', . 的例子	72
图 6-3	使用 B 和 / 及其它符号组合的例子	72
图 6-4	使用 . 的例子	73
图 6-5	使用 \$ 的例子	73
图 6-6	使用 CR 和 DB 的例子	73
图 6-7	不同字符的组合	73
图 7-1	PERFORM 语句流程图	132
图 9-1(a)	顺序结构	161
图 9-1(b)	选择结构	161
图 9-1(c)	重复结构	161
图 9-2(a)	一个选择结构, 一个重复结构和一个顺序结构	162
图 9-2(b)	三个选择结构和一个顺序结构	162
图 9-4	两个文件合并的主例行程序流程图	173
图 10-1	SEARCH 语句 (格式 1) 流程图	189
图 10-2	SEARCH 语句 (格式 2) 流程图	192
图 11-1	磁带上记录的安排	206
图 11-2	带记录不分块	206
图 11-3	带记录分块	207
图 11-4	磁盘读 / 写设备	210
图 11-5	盘片和盘道	211
图 13-1	分块和不分块关键字	226
图 13-2	柱面索引	226

图 13-3	柱面-120的磁道索引.....	226
图 13-4	磁道索引和记录关键字.....	227
图 13-5	增加三个记录的磁道索引.....	228
图 13-6	磁道索引和溢出链.....	228
图 13-7	VSAM 控制区道.....	236
图 13-8	顺序关键字数据集结构.....	236
图 13-9	控制区道的拼合.....	237
图 15-1	PAGE子句的区域示意图	256
图 20-1	COBOL 通讯环境	310
图 20-2	队的层次.....	312
图附录 1-1	银行本利对照程序框图.....	328
图附录 1-2	推销员提成费程序框图.....	332
图附录 1-2	推销员提成费程序框图(续).....	333
图附录 1-3	银行业务处理程序框图.....	340
图附录 1-3	银行业务处理程序框图(续).....	341
图附录 1-4	百货公司结算代销服务费和欠款总额程序框图.....	347
图附录 1-5	职工薪金文件程序框图.....	353
表 2-1	象征常数.....	17
表 2-2	算术表达式中符号的组合.....	20
表 2-3	关于运算量的计算.....	21
表 2-4	关于条件表达式的比较.....	23
表 2-5	条件逻辑运算符和括号组合.....	26
表 3-1	数据项的类和类型的关系.....	30
表 6-1	各种符号的组合规则.....	75
表 6-2	数据项的内部表示.....	84
表 7-1	MOVE 语句操作类型.....	109
表 20-1	INPUT CD 区通讯状态键字条件码	317
表 20-2	CD 输入区的用法	318
表 20-3	OUTPUT CD 区 STATUS KEY 代码	320
表 20-4	CD 输出区的用法	321

第一章 引 论

1.1 数据处理简介

当前，随着科学技术的发展和经济的繁荣，各行各业都相继利用电子计算机来处理日常事务中的各种数据信息，它既快速而又准确。随着我国四个现代化建设的进展，大量的数据处理任务必将由电子计算机来承担。

只要我们稍微留意于周围的社会活动，就会发现存在着大量的数据处理工作。例如银行、税务机关和保险机构内的职工，他们整天埋头在帐单、收据和卡片堆里找卡片、填表格、写票据，即使有大量精通业务的工作人员，也毕竟比不上每秒运算百万次的计算机；手脚再麻利，查表格的速度也有限，注意力再集中，也难免出差错。而电子计算机却能快速、准确，又能连续工作。就银行的存取业务而言，工作人员接收了某些单据，然后根据单据上的单位、住址、姓名或帐号，找出客户卡片，再根据卡片和单据的内容，经简单计算修改卡片留底存档，最后在客户单据上填写有关内容，如存取额和结余额等等。所有这些过程均可统称为数据处理。用电子计算机的术语来说：接受客户单据叫做读文件；找客户卡片叫做查找表格或检索；在输出设备上以报表的格式提供给用户单据叫做写文件。至于简单地计算利息和存款总额等都是在电子计算机的功能之内。这是仅就银行职工日常要处理的业务而言。其实各行各业都有大量类似的数据处理工作。

在仓库管理上，利用电子计算机可大大提高效率。各个部门都有仓库，各个仓库又存放着成千上万种不同规格、不同品种的物品。日常进货与发货，需要处理大量的票据，特别是查找存货位置和数量很费时间，而利用电子计算机很容易得到有关信息，便于调度管理。例如：根据发货情况进行统计计算，告诉仓库管理人员合理安排进货项目和数量。此外在人口调查统计、产品统计分析、财会工作、企业管理、行政管理、图书和档案的管理与检索、火车和飞机的调度、旅游业与工商业管理等等日常工作，都可以利用电子计算机来处理。总之，随着电子计算机的普及应用，在政府机关、商业、邮政、工矿企业、交通运输、文化教育团体，都有广泛应用的前景。

从上述事例可以看出，“数据处理”这个概念已具有广泛的含义，其主要工作不单是计算，而是对数据进行搜集、组织、排序、整理，它与工程、科学计算有着本质上的差别。一般地说，工程、科学计算是对为数不多的初始数据进行繁杂的、反复的大量计

算,所获得的最终结果也是不多的。但是,“数据处理”却是对大量的初始数据,有时仅作简单计算,有时根本不进行计算,只作某些逻辑判断,它的主要工作在于对初始数据进行严格组织、整理、传送;对大量的结果数据,也需要按所要求的格式进行组织处理。

因此,在“数据处理”这个概念里,“数据”已经不仅仅是“数”或者“数字”了,而是由各种各样的字母、符号组成的字符串和赋予一定含义的某些符号。所以,“数据”在这里是作为这些字符串、符号和数字的统称。在这种意义下,把这些数据称为信息更合适些。随着电子计算机的广泛应用,数据处理的概念已被信息处理所代替,所以信息处理就是对信息进行搜集、存贮、传送、计算、提供报表等的术语。

电子计算机由控制器、运算器和能存贮数据的内存贮器组成主机。由各种外部存贮器、输入、输出装置组成外部设备。它在程序控制下执行各种功能。控制器执行计算机程序,对内存贮器存取数据,运算器根据程序指定的操作进行运算。初始数据和程序是从输入装置上按一定格式要求输进机器,存放在指定的位置上,运算结果从输出装置上提供用户。

计算机程序是由人设计的一组能为机器接受、识别的命令的有序集合。操作人员先把程序输入机器内,并命令计算机启动、执行这个程序,计算机的各个部件在控制器的控制下,按照程序中指定的操作和顺序进行工作。程序的设计是很重要的,以前必须用计算机所能识别的数码形式的“机器指令”来编写。这给人们使用计算机带来很大的不便,用指令编排程序只有专业人员才能胜任,非专业人员不易掌握。幸运的是现在已经摆脱了这一局面,发展了用于不同领域的各种各样的高级语言,人们很容易掌握和使用它。利用高级语言编写程序使得电子计算机能为非专业人员所驾驭,给计算机的广泛应用创造了很好的条件。

我们所介绍的 COBOL 语言就是一种发展最早、应用最多、最普通的高级语言,它是一种便于事务处理的语言。随着我国国民经济的发展,用电子计算机进行信息处理已是大势所趋,推广和普及 COBOL 语言已经提到议事日程上来了。它将成为各部门、各行各业的技术人员和管理人员进行现代化管理和工作的有力工具。

1.2 COBOL 语言的产生和发展

五十年代中期,随着电子计算机应用领域的扩展,用于商业和事务处理的电子计算机数量猛增。最初,人们只利用电子计算机做一些簿记性的工作,到了五十年代末,随着计算机设计艺术的提高,数据处理领域中越来越广泛地应用电子计算机做更多的工作。

1959年5月,美国就开始酝酿建立一种通用商业语言并讨论种种要求和可能性,事后建立了数据系统语言协会,即 CODASYL (Conference on Data Systems Languages),负责 COBOL 语言的开发。到一九六〇年四月,由 CODASYL 中的短期委员会正式发表了 COBOL (Common Business Oriented Language)的说明书。到一九六一年六月又作了修改,重新发表。这就是我们通常说的 COBOL-61,它为 COBOL 语言的发展打

下了基础。此后,很多计算机厂在自己生产的计算机上实现了 COBOL 编译,并配备了 COBOL 编译程序。

廿多年来,COBOL语言得到广泛的应用,几经修改、扩充,出现了各种不同功能大小的 COBOL 语言。为了使 COBOL 语言具有一定的通用性,对 COBOL 语言进行了标准化,由国际标准化组织制定了COBOL语言标准化文本。ISO-COBOL-72就是该组织从一九六二年十月着手开发,直到一九七二年才制定出来的标准化文本,已被绝大多数用户所接受。为了实现和使用上的方便,把它分成八个功能模块,每个模块又按功能大小分成2—3级。用户可以根据自己的需要和具体情况,选取其中几个适当级别的模块,组成自己的 COBOL 语言。到一九七四年五月美国国家标准化协会发表了 ANSI-COBOL-74 文本。对某些功能模块作了必要的修改,还增加了一些新的模块。

COBOL 语言的历史是一个不断开发、不断修改和逐步完善的历史,同时也是不断应用、改进的历史。正是在不断的实际应用中,积累了大量的经验教训,成为开发、修改的最可靠依据。而标准化也是经过十多年的努力才完成的。标准化使得 COBOL 语言便于交流和移植,增加通用性和兼容性。

本书是根据 ANSI-COBOL-74 文本介绍 COBOL 语言的,而不针对某个具体机器。亦即介绍的语言是以美国国家标准化协会(ANSI)的X3.23任务组技术委员会发表的 ANSI-COBOL-74文本为依据。

到一九七六年,CODASYL 又提出 COBOL-76,又做了某些增删和修改,它是由以下十三种模块组成:

核心	(Nucleus)
表处理	(Table handling)
顺序 I/O	(Sequential I/O)
随机 I/O	(Random I/O)
索引 I/O	(Index I/O)
排序	(Sort)
报表打印	(Report writer)
分段	(Segmentation)
库	(Library)
内部程序通讯	(Inter-Program Communication)
调试	(Debugging)
通讯	(Communication)
数据库、数据操作	(Data base, Data manipulation)

COBOL 语言已经在我国应用,它必将促进我国国民经济的发展,并在实际应用中不断地得到扩充、完善。

1.3 COBOL 语言的特点

一种高级语言的生命力和应用的广泛程度,取决于它是否经济实惠,是否可以最大

限度地给用户方便。COBOL语言虽然有其弱点，但不可否认的事实是：它已成为应用最广泛、最普及的语言。目前它和 FORTRAN 语言一样成为世界上商品程序最多的两种高级语言之一。

COBOL 语言主要的特点有：类似英语口语；与具体机器基本上无关；具有功能模块化和强有力的数据描述能力；标准化工作做得较好。

1) 采用类似英语口语的形式

因为它允许用户用最简便、自然的表示手段与计算机交换信息，所以用户可以自由地选用各种符号串和语句。有此优点，用户使用 COBOL 语言就接近于用英语陈述计算和处理的过程，也提高了程序的易写性和易读性，便于交流检查，也较容易为非专业人员所了解和掌握，便于培训程序设计人员。

在我国随着汉字输入输出问题的解决，将进一步扩大 COBOL 语言的使用，使它扩展到我国国民经济各个领域，更好地为四个现代化服务。

2) 语言本身很少与具体机器有关

COBOL 源程序一般分四个部：标识部，环境部，数据部和过程部。这是 COBOL-61 就已经确定下来的，与机器有关的部份集中在环境部。因此，更换机器就不必重写程序，只需修改环境部中有关的内容，这便于算法的交流和程序的移植。

3) 功能模块化

ANSI-COBOL-74 是由一个核心及十一个功能处理模块组成，每个模块又按功能大小分成2—3级，具体实现可根据需要和机器的实际情况确定采用哪些模块，具有哪一级功能，使选用的COBOL语言更适合自己的实际需要。

ANSI-COBOL-74提供的模块如下：

ANSI-COBOL					
核 心 (nucleus)	功 能 处				
	表 处 理 (tablehandling)	顺 序 I/O (Sequential I/O)	相对 I/O (relative I/O)	索引 I/O (index I/O)	分类 / 合并 (Sort-merge)
二 级	二 级	二 级	二 级	二 级	二 级
			一 级	一 级	一 级
一 级	一 级	一 级	空	空	空

我们可以选取一级核心，一级表处理和一级顺序 I/O 组成最小规模的 COBOL 语言。很明显，如此处理给实现和应用提供很大的方便，又不影响整个语言的结构。

4) 强有力的数据描述功能。

COBOL 语言把数据描述和过程描述提到同样重要和明显的地位。把数据描述和过程描述并列。记录的有序集合——文件，可以在数据描述部进行描述，且对记录，即加工的数据单位，可用相应子句规定其各种数据内部形式，规定其信息表示和信息转换；把数据分为字母的、字母数字的、数值的、字母数字编辑的和数值编辑的五类；一个数据集合可以包含不同类型的数据，对处理、存放和加工数据提供了极大的灵活性；所有这些足以描述各行各业、各个领域的不同类型的数据。因此，COBOL 语言很适用于数据处理，它在这方面提供极大的方便。

5) 由于标准化工作做得好，程序格式比较定型，便于存档、查阅、相互交流。字符集一共只有 51 个字符，简明易记，便于推广。当然，COBOL 语言也存在一些缺点，主要是书写出的源程序一般比较冗长，语言比较庞大，难以记忆。相对于数据描述，它的过程部设施不够先进，随之而来的问题是目标质量相对地较差。

目前，任一种表达能力强，内容丰富的语言，都存在语言定义及其使用比较复杂的弱点。因此，带来了程序的可校验性较差，要检查书写的源程序的正确性比较困难。

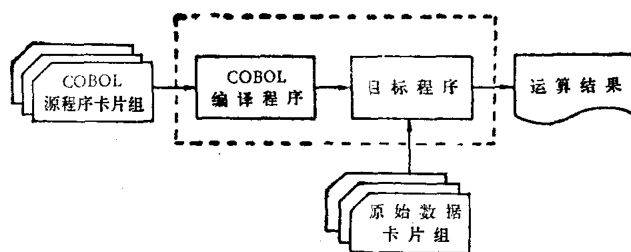
1.4 COBOL 程序的编译与运行

无论计算机多么复杂，它的执行动作都是由预先设计好的指令来指挥的，这些指令的全体我们称它为机器语言。电子计算机只能识别它自己所特有的机器语言。用 COBOL 语言写的程序，机器是不能够识别的。它必须通过一个叫做 COBOL 编译程序把它翻译成相应的机器语言程序才能供机器运行。

1974 模 块

理 模 块					
报表打印 (Report writer)	分 段 (Segmentation)	库 (Library)	调 试 (Debug)	内部程序通讯 (interprogram communication)	通 讯 (Communication)
一 级	二 级	二 级	二 级	二 级	二 级
	一 级	一 级	一 级	一 级	一 级
空	空	空	空	空	空

用 COBOL 语言写的用户程序叫做源程序。用户对源程序进行初步检查没有发现语法错误和逻辑错误后，把它们穿在一组卡片上。配置在机器里的 COBOL 编译程序就把所提供的源程序卡片组读入计算机内，并且编译处理它们，把源程序翻译成等价的称为目标程序的机器语言程序。计算机把目标程序和所要处理的数据一起执行运算。而这些运算的执行则是按照目标程序所规定的操作进行的，这样就可以得到 COBOL 源程序所要达到的结果。即：



在对 COBOL 源程序的编译过程中，如果源程序有语法错误。如：语句没有按规定的格式书写，保留字写错，缺分隔符等等。编译程序都能发现其语法错误，把它列入错误信息表。如果语法错误很严重或者数量较多，编译程序就会无法编译下去而停止工作。待源程序修改正确后再重新编译。如果没有发现语法错误，编译程序就能继续编译下去直到全部编译完毕。这样编译后的结果——目标程序就能在机器中运行或试运行。

但是，编译程序只能发现源程序中的语法错误。发现错误时能指出错误的位置和错误类型。而源程序中的逻辑错误在编译过程中是无法辨别的，只有在运行目标程序后，根据运行情况和打印出来的运算结果，由用户自己判别、分析、寻找。所以，一个没有语法错误的程序必须进行多次试运行，进行多方面考验、鉴定。通过运行情况来帮助发现错误，保证程序的正确性。

另外，语言本身也提供调试语句等手段，也可用来调试程序，确保程序的正确性。

只有通过认真地调试，严格地检查和考验，确保程序没有错误了，才能提供正式使用，进行正式运行。

在程序的编制，编译调试和运算的整个过程中，必须严格、认真。首先，必须对编制的程序认真地检查确认无错后才交付穿卡。上机编译和运算必须注意运行动态和结果是否正常，结果必须认真验证才能保证整个应用的正确性，提高应用效率。

第二章 COBOL 源程序

2.1 程序的词义

COBOL 所用的语言与英语相类似，程序设计人员在编写程序时，除需熟悉必要的英语外，更应了解 COBOL 语言中所用的名词、符号规则、表格形式和程序结构。

2.1.1 名词解释

源程序 (Source program)：指按照 COBOL 语言所编写的用户程序，该程序将被编译为某种计算机的机器语言程序。

目标程序 (Object program)：指由 COBOL 源程序所译成的可执行的机器语言程序，该程序将为作业控制语言所执行。

编译程序 (Compiler program)：指计算机厂商所提供的一种配备在机器系统中的系统软件，它可以将源程序编译为目标程序。

源程序计算机 (Source computer)：指将源程序编译成计算机机器语言所使用的计算机。

目标计算机 (Object computer)：指执行目标程序所使用的计算机。

2.1.2 格式中使用的符号

在语言表示格式中使用的符号有下列几种表示法必须说明一下：

大写体字是 COBOL 保留字 (Reserved word)，保留字包括连字号，他们的拼法不得有错。除了说明以外，它代表一定的处理功能。

横线字是关键字 (Key word)，它是程序中语句必须使用的保留字。

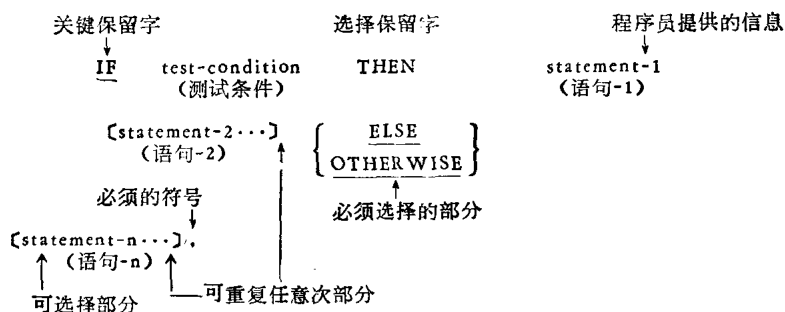
小写体字是程序员所拟定的字，该字称为指定字 (程序员提供的字)。通常可以填写在指定字的地方并且以小写英文字母表示。在以汉语句解释说明里，我们一般是以中文名称代替，便于说明。

方括号 []，在程序语句中，凡是用 [] 括着的语法内容表示可由程序设计人员选择或者省略。

大括号 { }，在程序语句中，凡是用大括号括着的语法符号可含有两种以上，程序设计人员必须选择其中之一。

省略号...，在程序语句中，有用省略号表示者，即表示其语法内容可重复若干次的意义。

在格式中所使用的符号如下图所示：



除了闭括号或作为选择的说明注解之外，要求在格式中给出终结符号。逗号和分号一般是可选择的，对于段、句子和数据部分描述体的终结要求用句号，在所有的终结符号之后必须至少跟一个空格。

2.1.3 字符集 (Character set)

COBOL 所使用的字符共有五十一一种：

字符	意义
0—9	数字
A—Z	字母
	空格
+	加号
-	减号或连字符
*	星号
/	斜竖
=	等号
\$	货币符号
,	逗号
;	分号
.	句号或小数点
"	引号
(	左括号
)	右括号
>	大于符号
<	小于符号

2.2 源程序的格式

COBOL 源程序是由一系列具有实际意义的语句行组成。编写源程序使用专用的程序纸。程序纸每一纵列均标明编号，它代表打穿孔卡片纵列的位置，每一横行须打印一张卡片，卡片的顺序必须由小到大排序。

COBOL 程序纸的格式如下：