

No

1	COBOL程序分为4个部分，格式是段名 + DIVISION:		
	(1) 标识部 (IDENTIFICATION	DIVISION.)	
	(2) 环境部 (ENVIRONMENT	DIVISION.)	
	(3) 数据部 (DATA	DIVISION.)	
	(4) 程序部 (PROCEDURE	DIVISION.)	
2	每个部中又有许多节，格式是节名 + SECTION.		
3	标识部：主要用来指定程序名，注意这个程序名是实际COBOL源程序文件名的前部分，不带扩展名，例如如果源程序文件名是COBIS47A.COB，则这里的程序名是COBIS47A 语法： IDENTIFICATION DIVISION. PROGRAM-ID. COBIS47A.		
4	环境部：定义程序用到的输入/输出设备，如数据文件或者打印机等的描述。 语法： ENVIRONMENT DIVISION. INPUT-OUTPUT SECTION. * FILE-CONTROL. SELECT 入力ファイル ASSIGN TO U01 FILE STATUS IS W－状態 ORGANIZATION IS LINE SEQUENTIAL. * SELECT 入力ファイル 1 ASSIGN TO U11 FILE STATUS IS W－状態 ORGANIZATION IS LINE SEQUENTIAL. * SELECT 出力ファイル 2 ASSIGN TO PRINTER FILE STATUS IS W－状態 ORGANIZATION IS SEQUENTIAL.		
	ASSIGN TO的文件名可以是磁盘上的实际文件名，也可以是逻辑文件名，然后在执行配置文件中指定实际磁盘文件，使用逻辑文件名的是改变文件名时不用修改源程序。		

5	<p>数据部：定义程序用到的全部变量，包括访问数据文件的纪录、访问数据库表的变量，程序用的内存变量等。</p> <p>文件节：定义输入输出文件用到的变量</p> <pre> FILE SECTION. FD 入力ファイル LABEL RECORD IS STANDARD BLOCK CONTAINS 0 RECORDS. * 01 入力レコード. COPY CISUF470 REPLACING ==()== BY ==入力1==. * FD 出力ファイル1 LABEL RECORD IS STANDARD BLOCK CONTAINS 0 RECORDS. * 01 出力1レコード. COPY CISUF470 REPLACING ==()== BY ==出力1==. * FD 出力ファイル2. * 01 行レコード PIC N(136). 01 注釈レコード PIC N(050). 01 制御レコード PIC X(100). </pre> <p>FD是文件描述的缩写。每个记录表示数据文件的一行数据，一个文件由多个记录组成。（文件→纪录→数据项的层次结构）</p>
6	<p>COPY句拷贝语句(COPY)</p> <pre> COPY CISUF470 REPLACING ==()== BY ==入力1==. </pre>
7	<p>文件打开操作</p> <pre> OPEN INPUT 文件1, 文件2, 文件3 OPEN OUTPUT 文件A, 文件B, 文件C </pre>

8	文件读入语句：(READ) READ 入力ファイル AT END MOVE "Y" TO W－終了－フラグ END-READ.
9	文件写语句：(WRITE) WRITE 出力－レコード.
10	文件关闭操作 CLOSE 文件1, 文件A
11	工作单元节：定义程序中用到的内存变量。 WORKING-STORAGE SECTION. * 01 ワークエリア. * *--< 件数エリア > 03 W－件数エリア. 05 W－持分以外－件数 PIC 9(09). 05 W－持分－件数 PIC 9(09). 05 W－金融展開－件数 PIC 9(09). * *--< ファイル状態 > 03 W－状態エリア. 05 W－状態 PIC X(02).

12	<p>变量的定义方法： 层号 变量名 PIC 数据类型（长度）。</p> <p>①数据类型：可以是数值型9型、S9型(带符号位，但是符号位不占一个字节，只是在最后一个字节放一个符号标志)，每一位表示0~9，表示形式：</p> <p style="text-align: center;">03 XX1 PIC 99999. 付给数字123，则内存中的表示为00123。 或者 03 XX1 PIC 9(05). 同上。 字符型X型，每一位可以时字母或者数字的字符。</p> <p>②小数： 03 XX1 PIC 999V99. 如果 付值12.345给XX1的话，则XX1的内存表示为00012。即不显示小数位。 定义小数位：999V99，付值12345.678给XX1的话，则XX1的内存表示为34567，写入文件也是34567，但是运算时是表时345.67。小数位是隐含的，不占内存空间。例如10.0+200.0，虽然表示成0010+0200，但是结果是0210(210.0)。</p> <p>③符号位： 03 XX1 PIC S9(05). 如果 付值-123给不带S的XX1的话，则XX1的内存表示为00123，即把符号位舍去了。如果要定义带符号的数值，则加S。</p> <p>④付初期值语句VALUE，例如 03 XX1 PIC 9(05) VALUE 12.</p>
13	<p>编辑型数据，仅仅用来进行显示或者打印，不用来进行计算。</p> <p>①小数位：99,999.99。</p> <p>②显示符号：‘+999’或者‘-999’。如果使用了‘+’，则正数据前面显示‘+’，负数据前面显示‘-’；如果使用了‘-’，则正数据前面显示空格，负数据前面显示‘-’。</p> <p>③取消高位的零。PIC Z(05)，付值12时候显示12而不是00012。</p>
14	<p>定义变量时的层号从01层开始，02层、03层等继续，但是为了维护时添加定义方便，可以采取间隔的方法，即01层、03层、05层，以后增加内容时可以添加02层或者添加04层。</p>
15	<p>变量名可以是半角字符，也可以是全角字符，但是不能全角和半角混排。变量名中不能有半角或者全角下滑线。</p> <p>变量名的长度在1~30之间，至少有一个字母。</p>

16	<p>重定义语句 (REDEFINE)</p> <pre> 01 AAA PIC 9(05). 01 BBB REDEFINES AAA PIC X(05). 01 AAA2. 03 AAA21 PIC 9(03). 03 AAA22 PIC 9(02). 01 BBB2. 03 BBB21 PIC 9(05).</pre>
17	<p>COBOL程序有固定的格式，每行有80个字符，其中</p> <ul style="list-style-type: none"> ① 1~6列是标号区，一般都用来写行号，这个区的内容对程序没有影响。 ② 第7列是续行注释区：‘-’表示这一行是接着上一行的内容；‘*’表示这一行的内容只是注释。 ③ 8~11列是A区域，各个部的名称、节名称、层号01、FD描述符等从这个区域开始书写。 ④ 12~72列是B区域，过程部的语句只能从B区开始。 ⑤ 73~80列是注释区域，写在这里的字符是注释，对程序的没有影响，这时第7列可以不是‘*’。
18	每个语句后都以逗点结束。
19	<p>定义数据时可以定义集团项目和子项目，他们是“从属”关系。访问子项目的格式是‘子项目名 OF 集团项目名’，也就是说在一个程序中子项目名可以重复，而集团项目名不能重复。如果子项目名不重复，则也可以直接访问‘子项目名’。例如</p> <pre> 01 AAAA1. 03 BBBB PIC 9(09). 01 AAAA2. 03 BBBB PIC 9(09). MOVE BBBB OF AAAA1 TO BBBB OF AAAA2.</pre>
20	集团项目不定义数据类型。虽然子项目数据有数值型数据，集团项目都是字符型数据。

21	<p>定义数组 (OCCURS)，带有OCCURS的变量不能是01层。</p> <pre> 01 集团项目 03 变量名1 OCCURS 5 TIMES. 05 变量名2 PIC X(09). MOVE "ABC" TO 变量名2(1).</pre>
22	<p>DISPLAY语句，可以显示数值变量和字符变量，一个DISPLAY语句显示在一行。</p> <pre> ①DISPLAY 变量1 "AAA" 变量2. ②DISPLAY 变量1. DISPLAY "AAA". DISPLAY 变量2.</pre>
23	<p>COBOL关键字：(关键字的单数形式和复数形式意义相同)</p> <pre> ①SPACE、SPACES (根据定义得变量长度) 表示一个或者多个空格，只能付给X型数据。 ②ZERO、ZEROS、ZEROES (根据定义得变量长度) 表示一个或者多个零，对于数值型和字符型数据都有效。9(05)型数据是 0000；于X(05)型数据是‘00000’、对于S9(05)型数据是0。 ③HIGH-VALUE、HIGH-VALUES 表示二进制的每一位都是1，对于数值型和字符型数据都有效。 ④LOW-VALUE、LOW-VALUES 表示二进制的每一位都是0，对于数值型和字符型数据都有效。 ⑤ALL ‘常量’ 由常量重复组成的字符串，例如 MOVE ALL "AB" TO 变量1.，变量1是X(05)，则付值后变量1得 内容是"ABABA"。</pre>
24	<p>字符串合并语句 (STRING)</p> <pre> ①STRING A,B,C DELIMITED BY SIZE INTO D. 按传送项的长度进行传递，即如果A="123 "，B="45 "，C="67 "，D的定义为X(12)的话，则D的内容为"123 45 67 "。 ②STRING A,B,C DELIMITED BY SPACE INTO D. 遇到空格时截止，即如果A="123 "，B="45 "，C="67 "，D的定义为X(12)的话，则D的内容为"1234567"。</pre>
25	<p>字符串分解语句 (UNSTRING)</p> <pre> ①UNSTRING D INTO A,B,C. 把字符串D的内容，按A、B、C的长度分解。 ②UNSTRING D DELIMITED BY "A" OR "B" INTO A,B,C. 把字符串D的内容，分解给A、B、C，"A"或者"B"作为定届符。</pre>

26	加法语句: (ADD) ①ADD A TO B. (B=A+B) ②ADD A, B TO C. (C=A+B+C) ③ADD A, B GIVING C. (C=A+B) ④ADD A, B TO C, D. (C=A+B+C, D=A+B+D)
27	减法语句: (SUBTRACT) ①SUBTRACT B FROM A. (A=A-B) ②SUBTRACT B, C FROM A. (A=A-B-C) ③SUBTRACT B, C FROM A, T. (A=A-B-C, T=T-B-C) ④SUBTRACT B, C FROM A GIVING X. (X=A-B-C)
28	乘法语句: (MULTIPLY) ①MULTIPLY A BY B. (B=A*B) ②MULTIPLY A BY B GIVING C. (C=A*B) ③MULTIPLY A BY B, C. (B=A*B, C=A*C)
29	除法语句: (DIVIDE) ①DIVIDE A INTO B. (B=A/B) ②DIVIDE A INTO B GIVING C. (C=B/A) ③DIVIDE A BY B GIVING C. (C=A/B) ④DIVIDE A, B TO C, D. (C=A+B+C, D=A+B+D) ⑤DIVIDE A INTO B GIVING C REMAINDER D. (C=B/A, 余数付给D。)
30	四舍五入处理语句 (变量名后+ROUNDED) ADD A, B TO C ROUNDED 如果A+B+C的值是186.7851, 而C的定义是9(03), 则C最后的值是187。 COMPUTE C ROUNDED = B + C.
31	算术运算语句 (COMPUTE) COMPUTE A = B * C + D.

32	付值语句 MOVE 源数据 TO 目的数据. ①付值时注意9型数据可以付值给X型数据，但是X型数据不能付值给9型数据； ②长的数据付值给段的数据，则会取舍： 对于字符型数据会左对齐截取多出的字符； 数值型数据付给字符型数据时也是左对齐； 而数值型数据会对应小数位向左或者向右进行截取； 带符号位的数据传送给字符型数据时符号位不传送。
33	跳转语句 (GO TO) GO TO 段名
34	停止语句 (STOP RUN) STOP RUN.
35	退出程序语句 (EXIT PROGRAM) EXIT PROGRAM.
36	CONTINUE语句 CONTINUE
37	判断条件 > (大于) = (等于) < (小于) >= (大于等于) <= (小于等于) AND (与) 条件1 AND 条件2 ，只有当条件1和条件2都为真时整个条件才为真，否则为假。 OR (或) 条件1 OR 条件2 ，只有当条件1和条件2都为假时整个条件才为假，否则为真。 NOT (非) NOT 条件1 ，如果条件1为假，则NOT 条件1为真。

38	IF语句 IF 条件1 IF 条件 内容… END-IF ELSE IF 条件 内容… END-IF END-IF.
39	IF 变量 = (值1 OR 值2) 内容… END-IF.
40	多分支语句(EVALUATE) ①EVALUATE 变量 WHEN 值A WHEN 值1 THRU 值2 WHEN OTHER END-EVALUATE. ②EVALUATE 变量1, 变量2, 变量3 WHEN 值1, ANY, 值2 WHEN ANY, ANY, 值2 WHEN OTHER END-EVALUATE. ③EVALUATE TRUE WHEN 表达式1 WHEN 表达式2 WHEN OTHER END-EVALUATE.

41	<p>PERFORM语句。</p> <p>①PERFORM 过程名1 THRU 过程名2. (从过程名1执行到过程名2)</p> <p>②PERFORM 过程名. (执行节名为“过程名”的部分)</p> <p>③PERFORM 过程名 3 TIMES. (执行3遍节名为“过程名”的部分)</p> <p>④PERFORM 过程名 UNTIL AA1 = HIGH-VALUE. (执行节名为“过程名”的部分，直到表达式为真才停止)</p> <p>⑤PERFORM 过程名 VARYING AA1 FROM 1 BY 1 UNTIL AA1 > 5. (执行节名为“过程名”的部分，AA1变量从1开始每次算1，直到AA1 > 5时才停止，AA1需要在数据部中定义)</p>
42	<p>模块节的定义</p> <p>模块名 SECTION.</p> <p>模块名-START.</p> <p>模块名-EXIT.</p> <p>EXIT.</p>
43	<p>接收语句：(ACCEPT)</p> <p>ACCEPT 变量名. (从键盘输入中读入数据)</p> <p>ACCEPT 变量名 FROM DATE. (读入系统日期数据)</p> <p>ACCEPT 变量名 FROM TIME. (读入系统时间数据)</p>
44	<p>显示语句：(DISPLAY)</p> <p>DISPLAY 变量名. (向屏幕输出数据)</p> <p>ACCEPT 变量名 FROM DATE. (读入系统日期数据)</p> <p>ACCEPT 变量名 FROM TIME. (读入系统时间数据)</p>
45	<p>初期化语句(INITIALIZE)</p> <p>INITIALIZE 变量名.</p>
46	<p>嵌套SQL文语句(EXEC SQL END-EXEC.)</p> <p>EXEC SQL</p> <p>具体的SQL语句</p> <p>END-EXEC.</p>
47	<p>调用函数语句(CALL)</p> <p>CALL 函数名 USING 参数.</p>