

数据部初步

1. 数据部的作用

数据部 (DATA DIVISION) 是整个 COBOL 源程序中唯一描述数据的部分。它是四个部分中的第三部分,是任何一个源程序不可缺少的部分。凡在程序中涉及的全部数据(包括输出的数据,输入的数据,中间数据)都要在数据部中加以说明(描述)。

数据有两种:一种是独立的数据,没有内在联系,他们各自占内存中的一个域,是具有独立逻辑意义的项;另一种是组合的数据项,它们是相互关联的,存在内在的联系。

程序过程部中出现的所有数据项都应当在数据部中对他们的属性进行说明。

包括:

- 每个数据项的类型,它们在内存中的存储形式,它们的长度。
- 数据相间的相互关系。
- 描述记录与文件的关系,即内存中的输入输出记录区是与哪个文件有关的。
- 文件的属性。

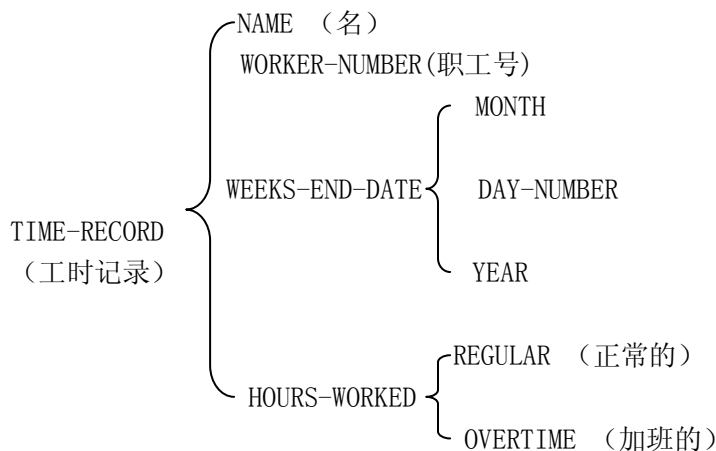
总之,数据部的作用是定义数据项属性,描述数据结构,准备被加工的“原料”。而过程部则是对这些“原料”进行加工。COBOL 把数据部和过程部分开,使每一部分的任务都比较单纯,清晰,便于书写和修改。

2. 数据的层次和层号

COBOL 中把有从属关系的数据用层次 (LEVEL) 来描述。数据的层次结构是:记录→组合项→初等项。

逻辑上不能再分的项目,称为初等项。包含若干个初等项的项目成为组合项。数据项的最高层次是记录。在一个记录中可以分为若干层次。

比如“工人的每周工时记录”,其数据关系如下:



TIME-RECORD 是记录名, WEEKS-END-DATE 和 HOURS-WORKED 是组合项。我们可以在数据部中用下面的层次关系来表示:

```
01 TIME-RECORD。  
    03 NAME PIC X (20)。  
    03 WORKER-NUMBER PIC 9 (6)。  
    03 WEEKS-END-DATE。  
        05 MONTH PIC 99。  
        05 DAY-NUMBER PIC 99。  
        05 YEAR PIC 99。  
    03 HOURS-WORKED。
```

06 REGULAR PIC 99。

06 OVERTIME PIC 99。

层次规定如下：• 用来描述层次结构的层号从 01 开始，到 49，即可用的层号为 01~49。

- 从属项的层号比其上属项的层号高，即层号小的组合项包含层号大的组合项，层号不必是连续的。
- 如果几个数据项都从属于同一个组合项但互不从属，则这几个数据项应具有相同的层号。

3. 数据部的结构

数据部中通常用到下面几个节：

- ① 文件节 (FILE SECTION)：用来描述程序中用到的输入文件和输出文件及其记录中各数据项的属性。
- ② 工作单元节 (WORKING-STORAGE SECTION)：用来描述程序中用到的中间数据项。
- ③ 联接节 (LINKAGE SECTION)：用来描述调用程序间发生数据传递的数据项。
- ④ 报表节 (REPORT SECTION)：为了完成报表编制功能，此节用来规定欲输出的报表的“体裁”，设计各报表栏的打印形式和方法等。

4. 文件节 (FILE SECTION)

- ① **文件节的作用**：程序中每个输入或输出文件都要在文件节中加以描述。描述的内容包括：
 - 文件名和文件属性。
 - 文件中包括的记录的名字。
 - 每个记录中数据的层次关系。
 - 记录中各数据项的数据形式和占内存的大小。

例：3-1-1

```
DATA    DIVISION.
FILE    SECTION.
FD      ACCOUNTS-RECEIVABLE
        LABEL RECORD IS STANDARD.
        DATA RECORD IS RECEIVALE. ---可以不写
01 RECEIVALE.
02 ACCOUNT PIC 9(6).
02 FILLER  PIC X(3).
02 AMOUNT  PIC 9(6).
02 NAME    PIC X(65).
```

- ② **文件描述**：从上面的例子可以看到，文件描述体不是用层号开始，而是用 FD (FILE DESCRIPTION) 开始，FD 称为文件描述符或层指示符。FD 后面跟文件名。上面例子中的 ACCOUNTS-RECEIVABLE (是由程序员自选的)。它是在程序中用到的内部文件，是在环境部中定义的。

最简单的文件描述体的一般格式为：

FD 文件名

$$\text{LABEL} \left\{ \begin{array}{l} \text{RECORD IS} \\ \text{RECORDS ARE} \end{array} \right\} \left\{ \begin{array}{l} \text{STANDARD} \\ \text{OMITTED} \end{array} \right\}$$
$$\left[\text{DATA} \left\{ \begin{array}{l} \text{RECORD IS} \\ \text{RECORDS ARE} \end{array} \right\} \text{数据名} \right]$$

- ③ 记录描述：记录描述体由 01 层号开头，后跟记录名。
- ④ 数据项描述：每个初等项（上例中 02 层）的名字后跟一个 PIC 子句。用它来描述数据的类型和长度
- ⑤ 文件节的书写格式：FD 从 A 区开始书写，01 层号也从 A 区开始，其他层号（如 02）可以从 A 区也可以从 B 区开始。为了看起来层次清楚，最好从 B 区开始，并按层次关系写成锯齿形状。

5. 工作单元节 (WORKING-STORAGE SECTION)

程序中用到的数据项分两部分，一部分是属于输入或输出文件的，这些数据项都应在数据部的文件节中加以描述。例如运算过程的中间结果，或用作累计数的数据项等，这些数据项则在工作单元节中描述。此外，还可利用工作单元节为某些数据赋初值(用 VALUE 子句)。

工作单元节中描述的数据项也有两种形式：一种是孤立的数据项，他们是初等项。一种是组合项。COBOL 规定，孤立的数据项的描述体以层号 77 开头，组合项描述体以 01 到 49 之间的一个数做层号。在次序上常先写 77 层，再写 01 到 49 层。

① 赋初值子句 (VALUE 子句)

COBOL 允许对工作单元节中的数据赋以初值。

- 说明：
- 只有对工作单元节中的数据项可以赋初值。不能对文件节中输入输出文件中的数据项赋初值
 - 如果在组合项的描述体中使用 VALUE 子句，初值只能是表意常量或非数值型常量。
 - 当用一个带符号的数据做初值时，相应的 PIC 子句中应该有“S”描述符。
 - 赋值时应注意类型的一致性。
 - VALUE 子句给出的值应适合 PIC 子句描述的范围，否则会出现截断或产生错误。

6. 字型子句 (PIC 子句)

PIC 是 Picture 的缩写。PIC 子句用来描述每个初等数据项。

(1) 数值型数据的描述

- “9”描述符：表示在该位置上放置一个 0—9 之间的数字。

例：01 A PIC 9(3).

注意：(1) 在数值型数据项中只能存放数字，不能存放空格。

(2) 用“9”描述符只能表示整数。如果输入小数部分会被舍去。

(3) 如果输入负号，负号会被舍去。

- “V”描述符：指出在数值数据结构中隐含的小数点的位置。

例：03 M PIC 999V99.

注意：(1) V 在描述符的最后时则等于无小数点。

(2) 一个数据的描述符中只能出现一个 V。

(3) 运算时，按隐含的小数点位置对准进行运算。

(4) 显示时，只将内存中各字节的内容显示出来，由于 V 不占内存，因此小数点不会显示出来。

- P 描述符：表示前零或后零。

例：01 A PIC 9PPPPP. 表示一位数字和 5 个 0

注意：(1) P 必须出现在全部 9 之前或之后。

(2) 用“P”则表示小数点的位置已隐含决定了。

(3) 如果在“9”后有 n 个 P，表示内存中的数应乘以 10 的 n 次方。

(4) 显示或打印时 P 是不出现的。

➤ S 描述符

如果想在数据项中放入一个带符号的数，可以使用“S”描述符

例：02 D PIC S99。

注意：(1) S 必须是最左边一个描述符。

(2) 在内存中，S 不占位，在运算中起作用。

(3) 根据数据项的不同描述决定符号在内存中的实际位置。

(2) 字母型数据的描述

字母型数据项用 A 描述符。在这种类型的数据项中只允许存放字母或空格。

例：02 T PIC AAAA。

(3) 字符型数据的描述

可以用 X 描述符来描述字符型数据。

例：03 A PIC X(3)

说明：(1) 字符型数据可以用 X 描述符来描述，也可用 9 和 A 描述符来描述。

(2) 字母型数据既可用 A 描述，也可用 X 描述。

(3) 字符型数据中可以存放数值。

(4) 编辑型描述符

编辑型数据项仅仅是为了输出的需要，没有其他的作用，它不能用来运算。

编辑描述符：(1) 插入小数点“.”用“.”描述符。

例：77 T PIC 99V99. —————> 77 N PIC 99.99

(2) 插入逗号“,”作分位号，用“,”描述符。

例：77 A PIC 9(6)V99. —————> 77 B PIC 999,999.99

(3) 插入零，用“0”描述符。

例：77 R PIC 999PP. —————> 77 T PIC 99900.

(4) 插入空格，用“B”描述符。

例：77 R PIC 9(2). —————> 77 T PIC B9(2)B.

(5) 插入正负号，用“+”或“-”描述符。

(6) 插入“\$”。

(7) 浮动插入正负号和“\$”。

(8) 取消高位零，用“Z”和“*”描述符。

(5) PIC 子句小结

每一种类型数据可以使用的描述字符如下：

数据类型	在 PIC 子句中允许使用的描述字符
数值数据项	9 V S P
字母数据项	A
字符数据项	9 A X
编辑数值数据项	9PV . , BZ+ - \$ * 0 CR DB
编辑字符数据项	A X 9 B 0

例 3-1-2: IDENTIFICATION DIVISION.

PROGRAM-ID. EXAM3-1-2.

ENVIRONMENT DIVISION.

CONFIGURATION

SECTION.

```

SOURCE-COMPUTER.                                E5800.
OBJECT-COMPUTER.                                E5800.
SPECIAL-NAMES.                                  SYSOUT IS OUT.
INPUT-OUTPUT SECTION.
FILE-CONTROL.
    SELECT PRINT-FILE    ASSIGN TO    OU-MSD
    FILE STATUS    SYS-FILE-STATUS.
DATA DIVISION.
FILE SECTION.
FD  PRINT-FILE
    LABEL RECORD IS STANDARD
    VALUE OF IDENTIFICATION IS "PRINT".
01  T.
    02  FILLER    PIC X.
    02  T1        PIC X(5).
    02  T2        PIC X(3).
    02  T3        PIC X(2).
    02  T4        PIC X(5).
WORKING-STORAGE SECTION.
01  SYS-FILE-STATUS PIC X(2) VALUE "00".
01  A VALUE ' 123456'.
    02  A1        PIC X(2).
    02  A2        PIC X(2).
    02  A3        PIC X(2).
PROCEDURE DIVISION.
MAIN.
    OPEN OUTPUT PRINT-FILE.
    MOVE "THIS" TO T1.
    MOVE "IS"   TO T2.
    MOVE "A"    TO T3.
    MOVE "BOOK" TO T4.
    DISPLAY T UPON OUT.
    WRITE T AFTER 1.
    DISPLAY A UPON OUT.
    CLOSE PRINT-FILE.
    STOP RUN.

```