第三章 ABAP 语法示例

3.1 FIELD-SYMBOLS

FIELD-SYMBOLS 是字段符号，类似于 FOXPRO 的宏，FOXPRO 能把一段程序存入一个字符串中，再通过某种方式使该字符串运行。SAP 的 FIELD-SYMBOLS，把一个结构定义分配给字段符号，通过字段符号来引用结构中的成员。

【例 3.1】

```
REPORT    YTEST001.

DATA:BEGIN OF MAN,
          NAME(30) TYPE C,
          HIGH TYPE P DECIMALS 2,
          WEIGHT TYPE P DECIMALS 2,
        END OF MAN.

FIELD-SYMBOLS <FSA> LIKE MAN.
DATA MAN1 LIKE MAN.

MAN1-NAME = '张林'.
MAN1-HIGH = '1.78'.
MAN1-WEIGHT = 140.

ASSIGN MAN1 TO <FSA>.
WRITE: / <FSA>-NAME,
               <FSA>-HIGH,
               <FSA>-WEIGHT.
```

输出结果如图 3-1 所示。

图 3-1

3.2 字符串处理

3.2.1 合并字符串

【例 3.2】

```
REPORT    YTEST001.
```

```
DATA: C1(2) TYPE C,
      C2(2) TYPE C,
      C3(2) TYPE C,
      C4(2) TYPE C,
      C5(20) TYPE C,
      C9(2) TYPE C.

C1 = 'AB'.
C2 = 'CD'.
C3 = 'EF'.
C4 = 'GH'.
C9 = '+'.
CONCATENATE C1 C2 C3 C4 INTO C5.
WRITE C5.
CONCATENATE C1 C2 C3 C4 INTO C5 SEPARATED BY C9.
WRITE / C5.
```

输出结果如图 3-2 所示。

```
ABCDEFGH
AB+ CD+ EF+ GH
```

图 3-2

本例中，先将字符串 C1、C2、C3、C4 合并到字符串 C5，再将字符串 C1、C2、C3、C4 用间隔符字符串 C9 合并到字符串 C5。

3.2.2 拆分字符串

【例 3.3】

```
REPORT    YTEST001.
DATA: C1(2) TYPE C,
      C2(2) TYPE C,
      C3(2) TYPE C,
      C4(2) TYPE C,
      C5(20) TYPE C VALUE '11 * 22 * 33 * 44',
      C9(2) TYPE C.

C9 = '*'.
WRITE C5.
SPLIT C5 AT C9 INTO C1 C2    C3    C4.
WRITE: / C1, C2, C3, C4.
```

输出结果如图 3-3 所示。

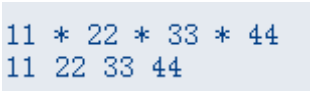

图 3-3

3.3 内表带有标题行

【例 3.4】

```
REPORT    YTEST001.
DATA: BEGIN OF MAN,
NAME(20) TYPE C,
HIGH TYPE P DECIMALS 2,
WEIGHT TYPE P DECIMALS 2,
                END OF MAN.

DATA:    MAN1 LIKE TABLE OF MAN WITH HEADER LINE,
MAN2 LIKE TABLE OF MAN.

MAN-NAME = '张参'.
MAN-HIGH = '1.68'.
MAN-WEIGHT = 120.
APPEND MAN TO MAN1.

MAN-NAME = '刘志'.
MAN-HIGH = '1.78'.
MAN-WEIGHT = 160.
APPEND MAN TO MAN1.

MOVE MAN1[] TO MAN2.

LOOP AT MAN2 INTO MAN.
   WRITE: / MAN-NAME,MAN-HIGH, MAN-WEIGHT.
ENDLOOP.

LOOP AT MAN1.
   WRITE: / MAN1-NAME,MAN1-HIGH,MAN1-WEIGHT.
ENDLOOP.
```

本例中内表 MAN1 有表头，内表 MAN2 无表头，输出结果一样，如图 3-4 所示。

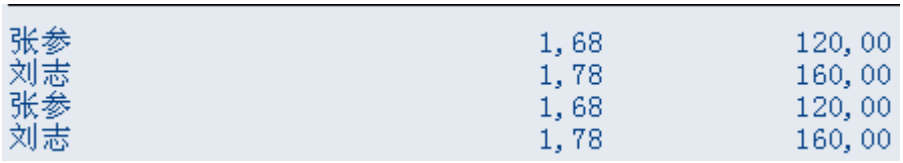

图 3-4

3.4 内表排序

【例 3.5】

```
REPORT    YTEST001.

DATA: BEGIN OF MAN,
            NAME(20) TYPE C,
            HIGH TYPE P DECIMALS 2,
            WEIGHT TYPE P DECIMALS 2,
          END OF MAN.
DATA: MAN1 LIKE HASHED TABLE OF MAN WITH UNIQUE KEY NAME.

MAN-NAME = '张参'.
MAN-HIGH = '1.68'.
MAN-WEIGHT = 120.
INSERT MAN INTO TABLE MAN1.

MAN-NAME = '刘志'.
MAN-HIGH = '1.78'.
MAN-WEIGHT = 160.
INSERT MAN INTO TABLE MAN1.

MAN-NAME = '朱强'.
MAN-HIGH = '1.72'.
MAN-WEIGHT = 130.
INSERT MAN INTO TABLE MAN1..

LOOP AT MAN1 INTO MAN.
   WRITE: / MAN-NAME,MAN-HIGH, MAN-WEIGHT.
ENDLOOP.

SORT MAN1 DESCENDING BY WEIGHT ASCENDING.
SKIP.
ULINE.
LOOP AT MAN1 INTO MAN.
   WRITE: / MAN-NAME,MAN-HIGH,MAN-WEIGHT.
ENDLOOP.
```

输出结果如图 3-5 所示。

张参	1,68	120,00
刘志	1,78	160,00
朱强	1,72	130,00

张参	1,68	120,00
朱强	1,72	130,00
刘志	1,78	160,00

图 3-5

3.5 修改内表数据

【例 3.6】

```
REPORT    YTEST001.

DATA: BEGIN OF MAN,
            NAME(20) TYPE C,
            HIGH TYPE P DECIMALS 2,
            WEIGHT TYPE P DECIMALS 2,
          END OF MAN.
DATA: MAN1 LIKE HASHED TABLE OF MAN WITH UNIQUE KEY NAME.

MAN-NAME = '张参'.
MAN-HIGH = '1.68'.
MAN-WEIGHT = 120.
INSERT MAN INTO TABLE MAN1.

MAN-NAME = '刘志'.
MAN-HIGH = '1.78'.
MAN-WEIGHT = 160.
INSERT MAN INTO TABLE MAN1.

LOOP AT MAN1 INTO MAN.
   WRITE: / MAN-NAME,MAN-HIGH, MAN-WEIGHT.
ENDLOOP.

MAN-NAME = '张参'.
MAN-HIGH = '2.22'.
MAN-WEIGHT = 220.
MODIFY TABLE MAN1 FROM MAN.

LOOP AT MAN1 INTO MAN.
   WRITE: / MAN-NAME,MAN-HIGH,MAN-WEIGHT.
ENDLOOP.
```

本例根据关键字修改内表，输出结果如图 3-6 所示。

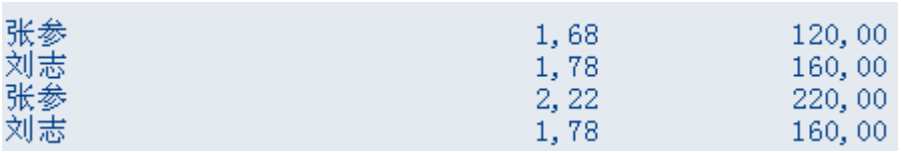

图 3-6

3.6 删除内表记录

【例 3.7】

```
REPORT    YTEST001.

DATA: BEGIN OF MAN,
            NAME(20) TYPE C,
            HIGH TYPE P DECIMALS 2,
            WEIGHT TYPE P DECIMALS 2,
          END OF MAN.
DATA: MAN1 LIKE HASHED TABLE OF MAN WITH UNIQUE KEY NAME.

MAN-NAME = '张参'.
MAN-HIGH = '1.68'.
MAN-WEIGHT = 120.
INSERT MAN INTO TABLE MAN1.

MAN-NAME = '刘志'.
MAN-HIGH = '1.78'.
MAN-WEIGHT = 160.
INSERT MAN INTO TABLE MAN1.

LOOP AT MAN1 INTO MAN.
   WRITE: / MAN-NAME,MAN-HIGH, MAN-WEIGHT.
ENDLOOP.

DELETE MAN1 WHERE NAME = '张参'.
SKIP.
ULINE.
LOOP AT MAN1 INTO MAN.
   WRITE: / MAN-NAME,MAN-HIGH,MAN-WEIGHT.
ENDLOOP.
```

输出结果如图 3-7 所示。

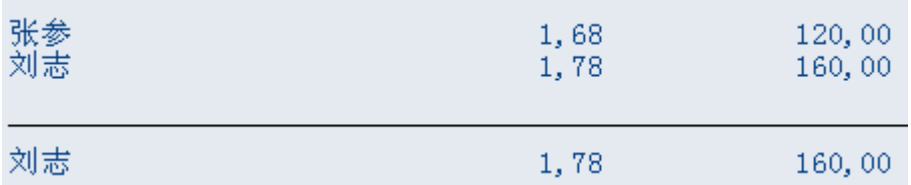

图 3-7

3.7 使用索引插入内表行

【例 3.8】

```
REPORT    YTEST001.

DATA: BEGIN OF MAN,
            NAME(20) TYPE C,
            HIGH TYPE P DECIMALS 2,
            WEIGHT TYPE P DECIMALS 2,
          END OF MAN.
DATA: MAN1 LIKE TABLE OF MAN.

MAN-NAME = '张参'.
MAN-HIGH = '1.68'.
MAN-WEIGHT = 120.
INSERT MAN INTO TABLE MAN1.

MAN-NAME = '刘志'.
MAN-HIGH = '1.78'.
MAN-WEIGHT = 160.
INSERT MAN INTO TABLE MAN1.

LOOP AT MAN1 INTO MAN.
   WRITE: / MAN-NAME,MAN-HIGH, MAN-WEIGHT.
ENDLOOP.

MAN-NAME = '李志'.
MAN-HIGH = '1.58'.
MAN-WEIGHT = 110.
INSERT MAN INTO MAN1 INDEX 2.
ULINE.

LOOP AT MAN1 INTO MAN.
   WRITE: / MAN-NAME,MAN-HIGH,MAN-WEIGHT.
ENDLOOP.
```

输出结果如图 3-8 所示。

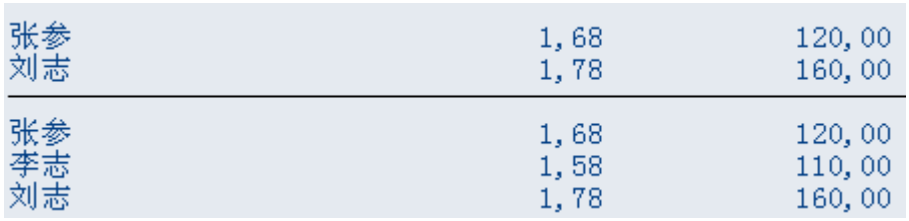

图 3-8

3.8 格式化输出

【例 3.9】

```
REPORT    YTEST001.
DATA WA LIKE SPFLI.
WRITE: /.

WRITE: 10'航班承运人',40'航班连接',60'国家代码',80'起飞城市',100'起飞机场'.

SELECT * INTO WA FROM SPFLI.
   WRITE: / WA-CARRID UNDER '航班承运人',
            WA-CONNID UNDER '航班连接',
            WA-COUNTRYFR UNDER '国家代码',
            WA-CITYFROM UNDER '起飞城市',
            WA-AIRPFROM UNDER '起飞机场'.

ENDSELECT.
```

本例通过 UNDER 定位输出位置，输出结果如图 3-9 所示。

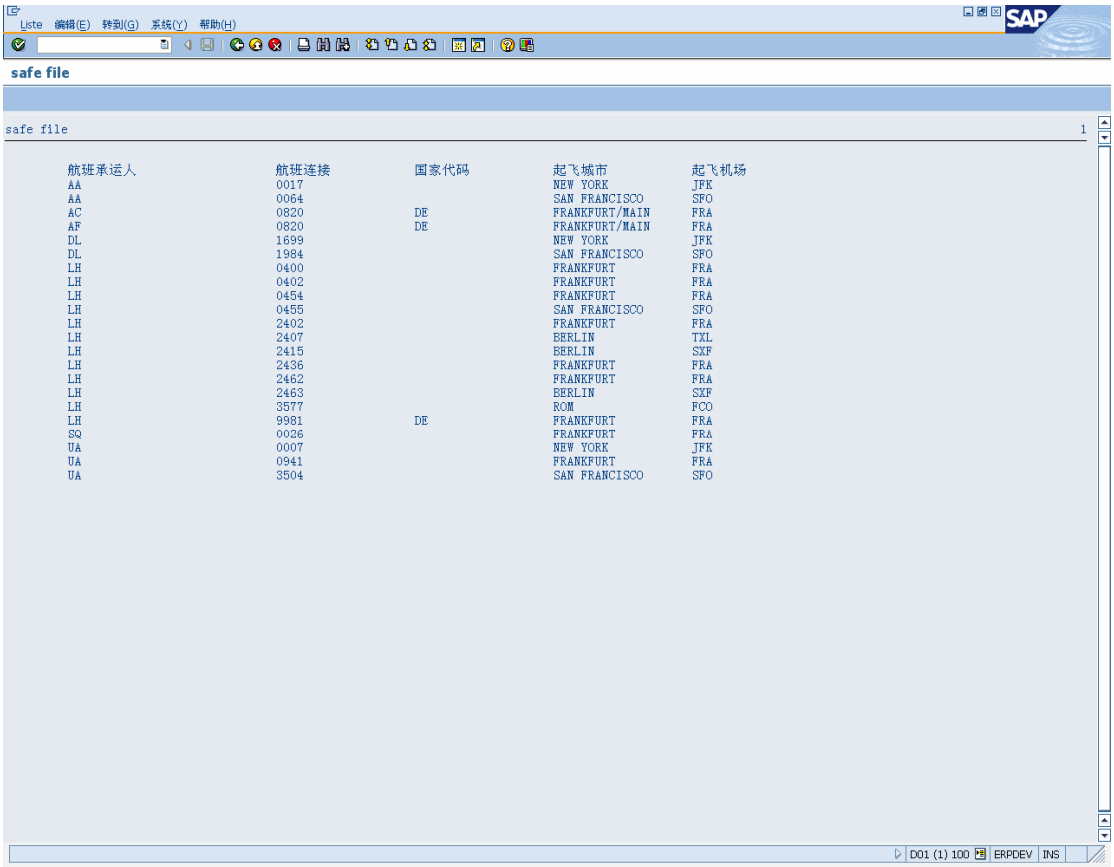

图 3-9

3.9 内部数据存为文件

【例 3.10】

```
REPORT  YTEST001.

DATA: BEGIN OF MAN,
        NAME(20) TYPE C,
        HIGH TYPE P DECIMALS 2,
        WEIGHT TYPE P DECIMALS 2,
      END OF MAN.
DATA: MAN1 LIKE TABLE OF MAN.
DATA: NAME TYPE RLGRAP-FILENAME, TYPA TYPE RLGRAP-FILETYPE.

MAN-NAME = '张参'.
MAN-HIGH = '1.68'.
MAN-WEIGHT = 120.
INSERT MAN INTO TABLE MAN1.

MAN-NAME = '刘志'.
MAN-HIGH = '1.78'.
MAN-WEIGHT = 160.
```

```
INSERT MAN INTO TABLE MAN1.

MAN-NAME = '李志'.
MAN-HIGH = '1.58'.
MAN-WEIGHT = 110.
INSERT MAN INTO MAN1 INDEX 2.

NAME = 'C:\TEMP\TESTA.TXT'.
TYPA = 'DAT'.

CALL FUNCTION 'DOWNLOAD'
  EXPORTING
    CODEPAGE                = 'TESTA'
    FILENAME                = NAME
    FILETYPE                = TYPA
    ITEM                    = '文件测试'
  TABLES
    DATA_TAB                = MAN1
  EXCEPTIONS
    INVALID_FILESIZE        = 1
    INVALID_TABLE_WIDTH     = 2
    INVALID_TYPE            = 3
    NO_BATCH                = 4
    UNKNOWN_ERROR           = 5
    GUI_REFUSE_FILETRANSFER = 6
    OTHERS                  = 7.
IF SY-SUBRC <> 0.
* MESSAGE ID SY-MSGID TYPE SY-MSGTY NUMBER SY-MSGNO
*         WITH SY-MSGV1 SY-MSGV2 SY-MSGV3 SY-MSGV4.
ENDIF.
```

打开运行实例存储文件，有以下内容：

```
张参    1,68 120,00
李志    1,58 110,00
刘志    1,78 160,00
```

3.10 直接存入文件

【例 3.10】

```
REPORT  YTEST001.

DATA: BEGIN OF MAN,
        NAME(20) TYPE C,
        HIGH TYPE P DECIMALS 2,
```

```
        WEIGHT TYPE P DECIMALS 2,
      END OF MAN.
DATA: MAN1 LIKE TABLE OF MAN.

CALL FUNCTION 'UPLOAD'
  EXPORTING
    CODEPAGE                = 'TEST'
    FILENAME                = 'C:\TEMP\TESTA.TXT'
    FILETYPE                = 'DAT'
    ITEM                    = '读放文件'
  TABLES
    DATA_TAB                = MAN1
  EXCEPTIONS
    CONVERSION_ERROR        = 1
    INVALID_TABLE_WIDTH     = 2
    INVALID_TYPE            = 3
    NO_BATCH                = 4
    UNKNOWN_ERROR           = 5
    GUI_REFUSE_FILETRANSFER = 6
    OTHERS                  = 7.
IF SY-SUBRC <> 0.
* MESSAGE ID SY-MSGID TYPE SY-MSGTY NUMBER SY-MSGNO
*         WITH SY-MSGV1 SY-MSGV2 SY-MSGV3 SY-MSGV4.
ENDIF.

LOOP AT MAN1 INTO MAN.
  WRITE: / MAN-NAME,MAN-HIGH, MAN-WEIGHT.
ENDLOOP.
```

输出结果如图 3-10 所示。

```
张参                    1,68                120,00
李志                    1,58                110,00
刘志                    1,78                160,00
```

图 3-10

3.12 不使用提示框提示直接读入文件

在例 3.11 中，程序运行时弹出输入读入文件名的提示框（如图 3-10 示），将程序中“CALL FUNCTION ‘UPLOAD’”改为“CALL FUNCTION ‘WS——UPLOAD’”，程序运行时将不再提示直接读入文件。

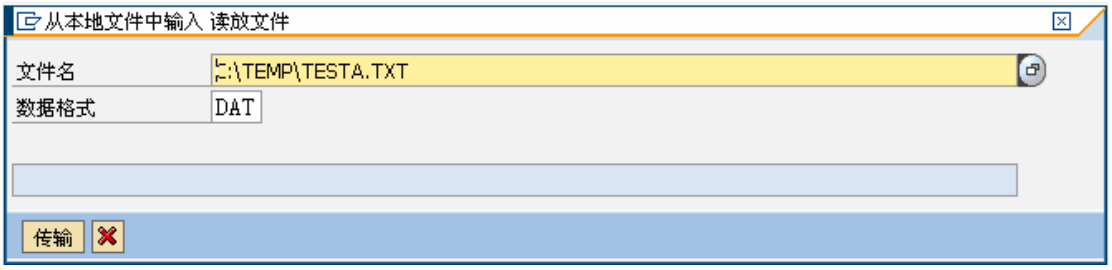

图 3-11

3.13 列表输出

WRITE 语法

WRITE AT [/] [<pos>] [(<len>)] 文本

“/”： 下一行

“pos”： 列

“(len)”：长度

【例 3.10】

```
REPORT  YTEST001.
DATA: STR1(10) TYPE C VALUE 'AABBCCDDEE'.
WRITE '接着的字符串_____将被替换.'.
WRITE AT 14(10) STR1.
```

输出结果如图 3-11 所示。

接着的字符串_AABBCCDDEE_将被替换.

图 3-12