1. OVERVIEW

Field-symbol이란?

- 필드 심볼은 abap 프로그램 내에서 data object에 동적인 접근이 가능하게 한다.

- 필드 심볼은 주소를 가지고 있어서, 주소값에 해당하는 메모리의 값으로 작동한다.
  즉, 필드심볼을 선언하여 사용한다고 하여서 필드심볼의 공간(메모리)을 가지는 것이 아니라,
  단지 연결된 필드의 메모리 주소만을 가지고 있다.

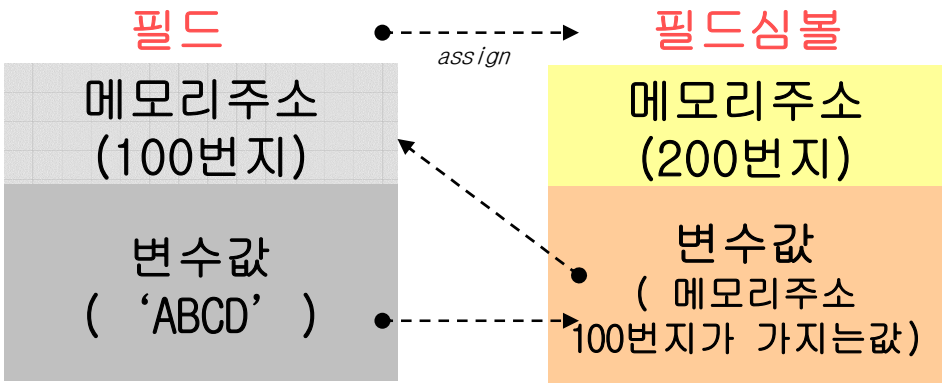
- C언어의 포인터(de-refrenced) 개념과 유사하다

- 필드 심볼의 데이터 이름과 속성은 실행시에 결정이 된다.

- 필드 심볼은 모든 데이터 오브젝트에 지정될수 있다.

- 일단 필드 심볼이 할당되면 필드와 필드 심볼 간에는 차이가 없다.
  MOVE와 같은 abap 명령어도 동일하게 사용할수 있다.

- 필드심볼은 타입을 명시하여 선언할수도 있으며, 타입없이 생성하여 된다.타입이 명시되지
  않으면 할당되는 필드(오브젝트의)의 타입을 그대로 상속받는다.



▲그림1-1. 필드 심볼 메커니즘

프로그램을 실행하여 필드심볼이 필드에 할당되는순간
메모리 주소를 가지게 되고 필드심볼의 변수값은
Link된 필드의 메모리 주소값을 가지게 된다.

2. 필드심볼 선언

```
FIELD-SYMBOLS <fs> [typing].
```

필드 심볼을 선언하는 구문이다. 필드 심볼 변수는 <>를 사용하여 선언한다.
[typing] 구문에는 타입을 선언하는 구문이며, 옵션 사항이다.

```
FIELD-SYMBOLS <fs1>.
FIELD-SYMBOLS <fs2> type any [TABLE].
FIELD-SYMBOLS <fs3> like gt_tab.
```

필드 심볼 선언법 예이다. FIELD-SYMBOLS <fs2> type any table. 을 사용하기 위해서는
ASSIGN구문에서 할당할 인터널 테이블이 table 타입으로 선언되어야 한다.
이렇게 선언된 필드심볼은 그 자체가 인터널 테이블이 되어 READ 와 같은 구문을 사용할수 있다.

```
TYPES: BEGIN OF line,
         col1 TYPE c,
         col2 TYPE c,
       END OF line.
DATA: wa TYPE line,
      itab TYPE HASHED TABLE OF line WITH UNIQUE KEY col1,
      key(4) TYPE c VALUE 'COL1'.
FIELD-SYMBOLS <fs> TYPE ANY TABLE.
ASSIGN itab TO <fs>.
```

필드심을 선언하여 프로그램 내에서 사용하기 위해서는 반드시 assign 구문을 이용하여 오브젝트를 할당 하여야 한다.

ASSIGN 의 이해

필드 심볼에 오브젝트를 assign 하기 위해서는 ASSIGN 구문을 활용한다.
ASSIGN 구문은 여러 개의 variant와 parameter를 가진다.

1. Basic Forms of the ASSIGN Statement

1-1. Static ASSIGN

필드명을 이미 알고 있는 경우라면 다음 예문을 이용하면 된다.

```
ASSIGN dobj TO <fs>.
```

1-2. Static ASSIGN with Offset Specification

필드의 일부분만을 필드심볼에 assign을 할 경우가 있다.

```
ASSIGN dobj[+off][(len)] TO <fs>.
```

ABAP에서 위 예문의 off를 offset이라하고, len을 lenth라고 부른다.
Len 을 명시적으로 선언하지 않으면 ABAP은 dobj의 길이와 동일하게 간주한다.
Data : string1(10) VALUE '0123456789 '.
string1+5. 0
이 구문은 실제 string+5(10)와 동일한 의미로 string10이 할당된 메모리의 영역을 벗어나게 된다.
이렇게 사용해도 문제는 없으나 field symbols의 data area를 넘어서는 안된다.

[예제13-2-1-] 통하여 이해를 돕자.
WRITE / line-string1+5. 구문은 수행되나,
ASSIGN line-string1+5 TO <fs>. 구문에는 에러가 발생하여 컴파일이 되지 않는다.

예제13-2-1

```
REPORT  Z13_001                                                    .
FIELD-SYMBOLS <fs> TYPE ANY.

DATA: BEGIN OF line,
        string1(10) VALUE '0123456789',
        string2(10) VALUE 'abcdefghij',
      END OF line.

WRITE / line-string1+5.

*ASSIGN line-string1+5 TO <fs>.  "error occured
*WRITE / <fs>.

*ASSIGN line-string1+5(10) TO <fs>. "error occured
*WRITE / <fs>.

ASSIGN line-string1+5(*) TO <fs>.
WRITE / <fs>.

ASSIGN line-string1+5(5) TO <fs>.
WRITE / <fs>.

ASSIGN line-string1+5(3) TO <fs>.
WRITE / <fs>.
```

결과13-2-1

```
56789
56789
56789
567
```

ASSIGN line-string1+5(*) TO <fs>. 에서와 같이 offset이 0보다 큰 값이 할당 된 경우에 필드심볼 assign을 사용하기 위해서는 * 문자를 사용하여야 한다. 이것은 필드심볼이 오브젝트의 길이를 넘어서는 것을 방지하게 해준다.

1-3. Dynamic ASSIGN

필드명을 알수 없을 경우(프로그램내에서 동적으로 할당되는 경우)에는 동적 assign 구문을 이용 한다.

```
ASSIGN (dobj) TO <fs>.
```

테이블 필드를 ASSIGN 할 경우에는 다음 예문을 이용하면 도니다.

```
ASSIGN TABLE FIELD (dobj) TO <fs>.
```

예제13-2-2

```
REPORT  Z13_012  .
TABLES sbook.

DATA: name1(20) TYPE c VALUE 'SBOOK-
FLDATE',
      name2(20) TYPE c VALUE 'NAME1'.

FIELD-SYMBOLS <fs> TYPE ANY.

ASSIGN TABLE FIELD (name1) TO <fs>.
WRITE: / 'sy-subrc:', sy-subrc.

ASSIGN TABLE FIELD (name2) TO <fs>.
WRITE: / 'sy-subrc:', sy-subrc.
```

결과13-2-2

```
sy-subrc:      0
sy-subrc:      4
```

ASSIGN TABLE FIELD (name1) TO <fs>. 구문에서 SBOOK-FLDATE 라는 필드가 존재하므로 SY-SUBRC 0 값을 반환하게 된다.

2. Assigning Components of Structures to a Field Symbol

스트럭쳐의 필드를 필드 심볼에 ASSIGN 할 수 있다.

```
ASSIGN COMPONENT comp OF STRUCTURE struc TO <fs>.
```

예제13-2-3

```
REPORT  Z13_014                                                    .

DATA: BEGIN OF line,
        col1 TYPE i VALUE 11,
        col2 TYPE i VALUE 22,
        col3 TYPE i VALUE 33,
      END OF line.

DATA comp(5) TYPE c VALUE 'COL3'.

FIELD-SYMBOLS: <f1> TYPE ANY, <f2> TYPE ANY, <f3> TYPE ANY.

ASSIGN line TO <f1>.
ASSIGN comp TO <f2>.

  DO 3 TIMES.
  ASSIGN COMPONENT sy-index OF STRUCTURE <f1> TO <f3>.
  WRITE <f3>.
  ENDDO.

ASSIGN COMPONENT <f2> OF STRUCTURE <f1> TO <f3>.
WRITE / <f3>.
```

결과13-2-3

```
11          22          33
33
```

3. Casting Data Objects

필드 심볼의 데이터 타입에 관계없이 오브젝트를 할당하고자 할경운에는 CASTING 구문을 이요하면 된다.

```
ASSIGN ... TO <fs> CASTING.
```

예제13-2-4

```
REPORT  Z13_015                                                    .
TYPES: BEGIN OF t_date,
          year(4)  TYPE n,
          month(2) TYPE n,
          day(2)   TYPE n,
       END OF t_date.

FIELD-SYMBOLS <fs> TYPE t_date.

ASSIGN sy-datum TO <fs> CASTING.

WRITE / sy-datum.
SKIP.
WRITE: / <fs>-year , / <fs>-month, / <fs>-day.
```

결과13-2-4

```
2007.01.08

2007
01
08
```

ASSIGN sy-datum TO <fs> CASTING. 구문에서 CASTING을 없애면 데이터 타입이 일치 하지 않는다는 에러가 발생한다.

4. Data Areas for Field Symbols

필드 심볼의 데이터 타입에 관계없이 오브젝트를 할당하고자 할경운에는 CASTING 구문을 이요하면 된다.

```
ASSIGN ... TO <fs> CASTING.
```

예제13-2-4

```
REPORT  Z13_015                                                    .
TYPES: BEGIN OF t_date,
          year(4)  TYPE n,
          month(2) TYPE n,
          day(2)   TYPE n,
       END OF t_date.

FIELD-SYMBOLS <fs> TYPE t_date.

ASSIGN sy-datum TO <fs> CASTING.

WRITE / sy-datum.
SKIP.
WRITE: / <fs>-year , / <fs>-month, / <fs>-day.
```

결과13-2-4

```
2007.01.08

2007
01
08
```

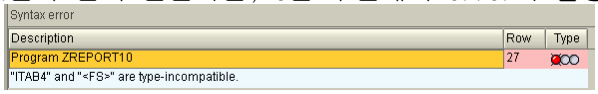
ASSIGN sy-datum TO <fs> CASTING. 구문에서 CASTING을 없애면 데이터 타입이 일치 하지 않는다는 에러가 발생한다.

3. 필드심볼 사용

예제13-2-1

```
REPORT  Z13_002                                                    .
TYPES: BEGIN OF line,
         col1 TYPE c,
         col2 TYPE c,
       END OF line.

DATA : wa type line.

DATA : fname(4) TYPE c VALUE 'COL1'.

1      DATA: itab TYPE sorted TABLE OF line  WITH UNIQUE KEY col1.

2      DATA: itab2 TYPE hashed TABLE OF line  WITH UNIQUE KEY col1.

3      DATA: itab3 TYPE TABLE OF line.

4      DATA: itab4 TYPE LINE OCCURS 0 WITH HEADER LINE.


5      FIELD-SYMBOLS <fs> TYPE ANY TABLE.

6      ASSIGN itab TO <fs>.

7      READ TABLE <fs> WITH TABLE KEY (fname) = 'X' INTO wa.
```

1. 인터널 테이블을 SORTED TABLE로 선언
2. 인터널 테이블을 HASHED TABLE로 선언
3. 인터널 테이블을 일반 테이블 타입으로 선언
4. 인터널 테이블 선언(구식방법)
   1, 2, 3번과 같이 선언하여 5번 구문을 사용하면 6, 7번 구문이 실행되나,
   4번과 같이 선언하면, 6번 구문에서 error가 발생한다



7. 필드 심볼을 READ 하기 위해서는 5번 구문처럼 TYPE ANY TABLE. 로 선언하여야
   필드 심볼이 인터널 테이블 역할을 수행하게 된다. (fname)을 사용하여 동적구문을 이용한다.
   READ TABLE <fs> with table key col1 = X' . 와 같이 사용하면 에러가 발생한다.

4.필드심볼과 Structure

```
DATA: BEGIN OF line,
        col1(1) TYPE c,
        col2(1) TYPE c VALUE 'X',
      END OF line.

FIELD-SYMBOLS <fs>.

ASSIGN line TO <fs>.

MOVE <fs>-col2 TO <fs>-col1.
```

MOVE <fs>-col2 TO <fs>-col1. 구문에서 에러가 발생한다. 필드심볼을 타입을 선언하지 않은 경우에는 필드심볼 내부에 structure를 모두 가지고 있다. 즉, 프로그램 소스코드에서 <fs>-col1 처럼 명시적으로 구조체의 필드를 선언하여 사용하고자 할경우에는 필드심볼을 타입을 지정하여 선언하여야 한다는 의미이다. 왜냐하면 필드심볼을 타입을 참고하지 않고 선언할 경우에는 프로그램이 실행되어 assing 구문이 수행되는 순간 타입이 할당되기 때문이다. 아래 예문을 참고하자

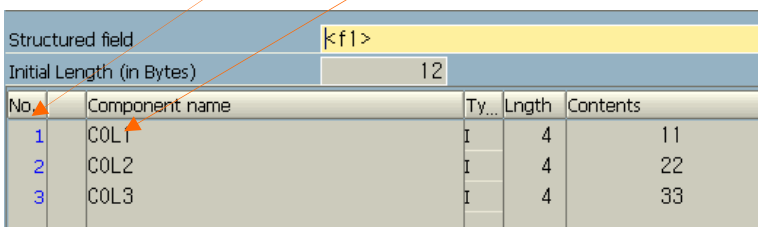
```
DATA: BEGIN OF line,
        col1(1) TYPE c,
        col2(1) TYPE c VALUE 'X',
      END OF line.

FIELD-SYMBOLS <fs>  LIKE LINE.

ASSIGN line TO <fs>.

MOVE <fs>-col2 TO <fs>-col1.
```

필드심볼을 structure처럼 사용하고자 할경우에는 필드심볼을 선언시에 like, 또는 Type을 선언하여야 한다

5.필드심볼 Structure

```
ASSIGN COMPONENT comp OF STRUCTURE struc TO <fs>.
```

필드심볼 타입을 구조체로 선언한 경우나 TYPE ANY 로 선언 한 경우에는 필드심볼을 Structure 구조로 할당할 수 있다. 예문은 구조체 struc의 comp(필드)를 필드심볼 <fs>에 할당하는 것을 보여준다. comp 항목에 올수 있는 것은 line의 **숫번**이나 **컬럼명**이다.



▲그림1-5-1. 필드 심볼 구조체

예제13-5-1

```
REPORT  Z12_004                                                    .
DATA: BEGIN OF line,
        col1 TYPE i VALUE 11,
        col2 TYPE i VALUE 22,
        col3 TYPE i VALUE 33,
      END OF line.
DATA comp(5) TYPE c VALUE 'COL3'.
FIELD-SYMBOLS: <f1> TYPE ANY,
                          <f2> TYPE ANY,
                          <f3> TYPE ANY.
ASSIGN line TO <f1>.

ASSIGN comp TO <f2>.
```

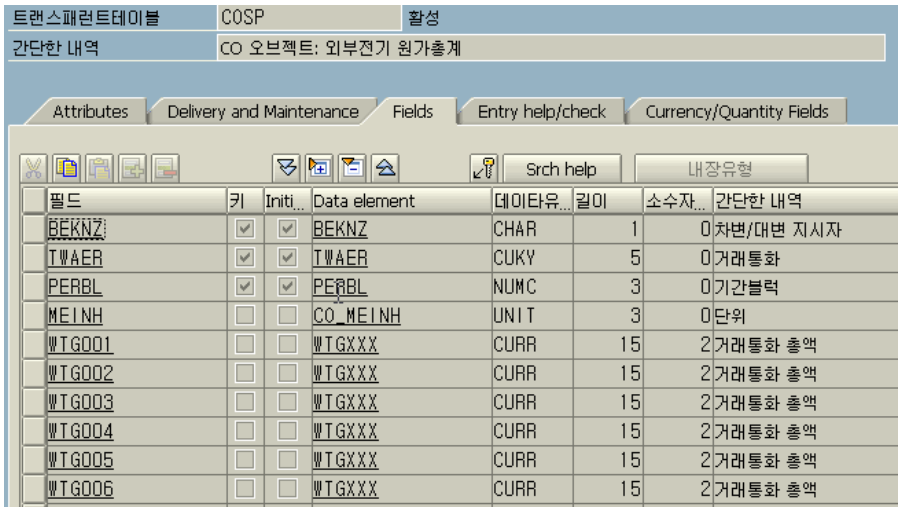
```
DO 3 TIMES.
  ASSIGN COMPONENT sy-index OF STRUCTURE <f1> TO <f3>.
  WRITE <f3>.
ENDDO.
ASSIGN COMPONENT <f2> OF STRUCTURE <f1> TO <f3>.
WRITE / <f3>.
```

결과13-5-1

```
11      22      33
33
```

필드심볼 <fs1>의 구조는 line과 동일하다.
DO 3 TIMES는 LINE 타입의 필드가 3개이기때문이며, LOOP 횟수를 가지고 있는 시스템 변수 SY-IDEX를 통해 값을 할당하고 있다.

6. 필드 심볼 활용



트랜스패런트테이블 COSP 활성

간단한 내역 CO 오브젝트: 외부전기 원가총계

Attributes | Delivery and Maintenance | Fields | Entry help/check | Currency/Quantity Fields

Srch help | 내장유형

필드	키	Initi...	Data element	데이타유...	길이	소수자...	간단한 내역
BEKNZ	☑	☑	BEKNZ	CHAR	1	0	차변/대변 지시자
TWAER	☑	☑	TWAER	CUKY	5	0	거래통화
PERBL	☑	☑	PERBL	NUMC	3	0	기간블럭
MEINH	☐	☐	CO_MEINH	UNIT	3	0	단위
WTG001	☐	☐	WTGXXX	CURR	15	2	거래통화 총액
WTG002	☐	☐	WTGXXX	CURR	15	2	거래통화 총액
WTG003	☐	☐	WTGXXX	CURR	15	2	거래통화 총액
WTG004	☐	☐	WTGXXX	CURR	15	2	거래통화 총액
WTG005	☐	☐	WTGXXX	CURR	15	2	거래통화 총액
WTG006	☐	☐	WTGXXX	CURR	15	2	거래통화 총액

▲그림1-6-1. 월별비용을 관리하는 테이블의 구조

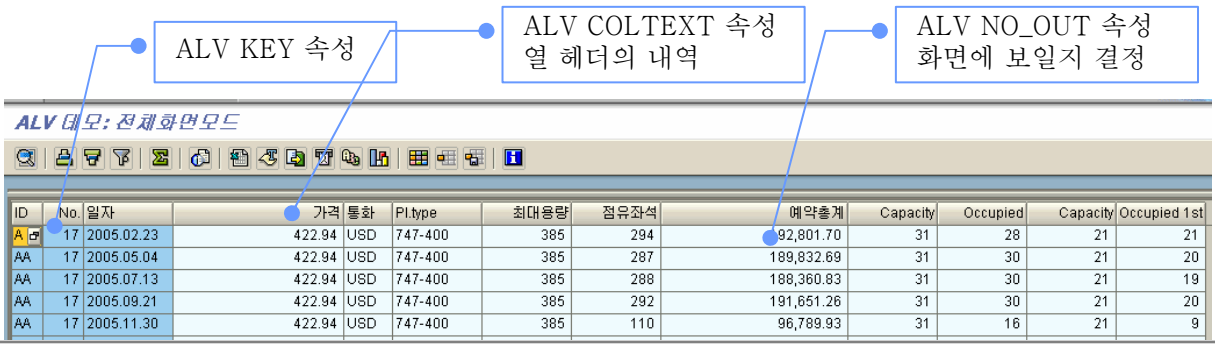
SAP CO 모듈에서는 월별 비용을 관리할 경우 필드명 + 월(MONO1)과 같은 구조를 많이 취한다.
WTG001 ~ WTG012의 필드의 SUM 을 구하고자 할경우 12번을 합을 해야 하는 경우가 발생한다.
이 경우에는 필드심볼을 사용하여 필드합을 12번 코딩하는 수고를 줄일 수 있다.
실제 업무에서 많이 사용되고 있는 소스 코드이므로 사용법을 숙지하기 바란다.

```
FIELD-SYMBOLS <FIELD>.
DATA :  FNAME(10),
            SUM LIKE COSP-WTG001.

DO 12 TIMES.
  CC = SY-INDEX.
  CONCATENATE  'COSP-WGTO' CC INTO FNAME.

 ASSIGN (FNAME) TO <FIELD>.
  SUM = SUM + <FIELD>.
  CLEAR : FNAME, <FIELD>.
ENDDO.
```

예제13-5-1

```
FORM set_field_catalogs_grid  USING   lt_fieldcat TYPE lvc_t_fcat.

  DATA: ls_field TYPE lvc_s_fcat.

  FIELD-SYMBOLS: <ls_fcat> TYPE lvc_s_fcat.

  LOOP AT lt_fieldcat ASSIGNING <ls_fcat>.
    IF <ls_fcat>-fieldname  = 'CUSTM'.
      <ls_fcat>-coltext   = '고객명'.
    ENDIF.

    IF <ls_fcat>-fieldname NE 'WERK'.
      <ls_fcat>-no_out = 'X'.
    ELSE.
      <ls_fcat>-no_out = ' '.
    ENDIF.

    IF <ls_fcat>-fieldname EQ 'MATNR'.
      <ls_fcat>-key = 'X'.
      <ls_fcat>-hotspot = 'X'.
    ENDIF.

  ENDLOOP.

ENDFORM.                    " SET_FIELD_CATALOGS_GRID
```

[예제13-5-1], [예제13-5-2]는 ALV 필드 카타로그에서 필드 심볼을 이용하여 자주쓰는 스크립트이다.
[예제13-5-1]필드 심볼을 사용하게 되면 [예제13-5-2]와 같이 MODIFY 구문으로 인터널 테이블을 변경 해줄 필요가 없다. 필드 심볼은 주소에 있는 값을 바로 변경하기때문이다.



ID	No.	일자	가격	통화	Pl.type	최대용량	점유좌석	예약총계	Capacity	Occupied	Capacity	Occupied 1st
A	17	2005.02.23	422.94	USD	747-400	385	294	92,801.70	31	28	21	21
AA	17	2005.05.04	422.94	USD	747-400	385	287	189,832.69	31	30	21	20
AA	17	2005.07.13	422.94	USD	747-400	385	288	188,360.83	31	30	21	19
AA	17	2005.09.21	422.94	USD	747-400	385	292	191,651.26	31	30	21	20
AA	17	2005.11.30	422.94	USD	747-400	385	110	96,789.93	31	16	21	9

예제13-5-2

```
FORM set_field_catalogs_grid  USING   lt_fieldcat TYPE lvc_t_fcat.

  DATA: ls_fcat          TYPE lvc_s_fcat.
  LOOP AT lt_fieldcat INTO ls_fcat .
    IF ls_fcat-fieldname  = 'CUSTM'.
      ls_fcat-coltext   = '고객명'.
    ENDIF.

    IF ls_fcat-fieldname NE 'WERK'.
      ls_fcat-no_out = 'X'.
    ELSE.
      ls_fcat-no_out = ' '.
    ENDIF.

    IF ls_fcat-fieldname EQ 'MATNR'.
      ls_fcat-key = 'X'.
      ls_fcat-hotspot = 'X'.
    ENDIF.
    MODIFY lt_fieldcat FROM ls_fcat INDEX sy-tabix.
  ENDLOOP.

ENDFORM.                    " SET_FIELD_CATALOGS_GRID
```

6. 필드 심볼 활용

Data reference는 data objet에 대한 포인터(pointer)이다. Data object를 동적으로 생성하기 위해 Data reference를 사용할 수 있다. 또는 이미 존재하는 data object를 참고하여 생성할 수 있다.

1. Reference variable

Reference variable은 reference를 포함하고 있다. Reference의 실제 항목ㄷ르은 ABAP 프로그램에 보이지 않는다. ABAP은 data reference와 object reference를 가지고 있을 뿐이다.
두가지의 reference variable이 존재한다. 이것은 data reference variables과 object reference Variable 이다.

다음 예문을 통하여 data reference 타입을 생성한다.

```
TYPES t_dref TYPE REF TO DATA.
```

선언된 data ference 타입을 이용하여 data refernce vaiable을 생성한다. Reference variable은 Elemetay data type과 같은 오브젝트돠 동일하게 사용된다. 즉, refernce vaiable은 구조체, 인터널 테이블, 필드의 항목(component)으로 사용될 수 있다.

```
DATA dref TYPE REF TO DATA.
```

위 구문까지 선언된 dref는 아직 오브젝트에 연결(point)되지 않은 상태이므로 사용할수가 없다. CREATE DATA dref TYPE SFLIGHT 과 같은 구문을 선언하여, reference vaiale이 data object에 연결되도록 해야 한다.

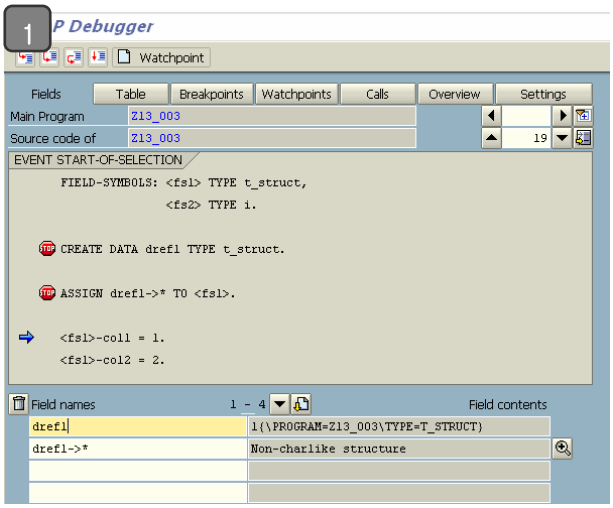
```
CREATE DATA dref {TYPE type}|{LIKE dobj}}.
GET REFERENCE OF dobj INTO dref.
DREF2 = DREF2.
```

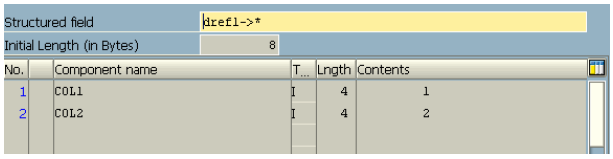
예제13-7-1

```
REPORT  Z13_003                                                    .

TYPES: BEGIN OF t_struct,
         col1 TYPE i,
         col2 TYPE i,
       END OF t_struct.

DATA: dref1 TYPE REF TO data,
      dref2 TYPE REF TO data.

FIELD-SYMBOLS: <fs1> TYPE t_struct,
               <fs2> TYPE i.

CREATE DATA dref1 TYPE t_struct.

ASSIGN dref1->* TO <fs1>.

<fs1>-col1 = 1.
<fs1>-col2 = 2.

dref2 = dref1.

ASSIGN dref2->* TO <fs2> CASTING.
WRITE / <fs2>.

GET REFERENCE OF <fs1>-col2 INTO dref2.

ASSIGN dref2->* TO <fs2>.
WRITE / <fs2>.
```

결과13-7-1

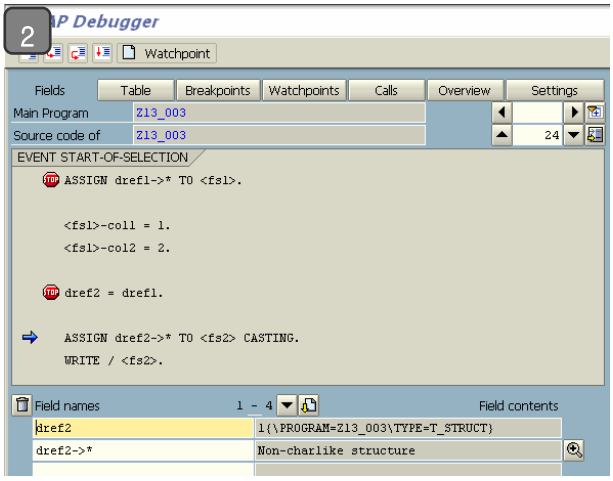
```
1
2
```



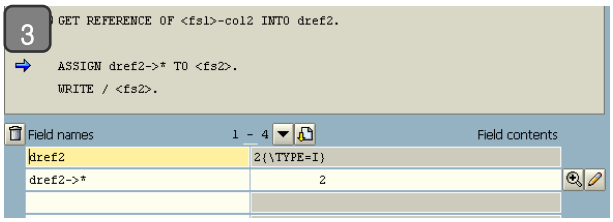


[예제13-7-1]의 프로그램을 디버깅하면서 이해를 돕자.

1. CREATE DATA dref TYPE t_struct 구문이 수행되면, t_struct를 가리키는 referece Variable이 생성된다. 이 시점에서 dref1변수에는 프로그램명과 type 정보를 가지게 되고, 실제 구조와 데이터는 dref->* 를 사용하여 assign 하게 된다.



2. dref2= drffe1. 구문을 수행하게 되면 Dref2 reference vaiable에 dref1과 동일한 구조와 데이터가 할당이 된다.



3. GETE REFERENCE OF <fs1>-col2 INTO dref2. 구문이 수행되면 dref2 변수에 <fs1>-col2 타입이 연결된 reference varaible이 생성된다.