

abap- 9월 5일 4차시

9월 5일

N타입은 문자열인데 숫자가 들어가는 것입니다. 123이라고 하면 숫자 123이 아니라 문자열 123!

offset 사용하기

- str+3(3)하면 초기 문자에서 3칸 움직인 뒤 세자리 출력

Internal Table은 한 row씩 처리한다.

a b c d ← Header라 부름. 현재 row를 가리킴

b e d f // body

a b c d //body

- Internal table은 body에 있는 데이터를 header 부분으로 옮겨서 작업하고, 다시 저장할 때는 header 부분의 데이터를 body로 옮겨서 작성한다.

내가 만든 Structure: ZPSHS0010

DATA GT_DATA TYPE TABLE OF ZPSHS0010

→ 여기서 왜 GT라고 했을까? STRUCTURE인데?

- 이유: TYPE TABLE OF라고 해서, STRUCTURE을 TABLE로 만들어버림. 그래서 GS_DATA가 아니라 GT_DATA로 선언함.

ITAB TYPE S_TYPE OCCURS 0.

- 테이블 구조를 만들고, 메모리가 들어올 때마다 할당을 해주겠다. 0이 아니라 다른 수를 사용하면 EX 5 5크기 할당.

DATA GT_DATA2 TYPE TABLE OF ZCMMS0010 WITH HEADER LINE.
WITH HEADER LINE==OCCURS 0

DATA: ITAB LIKE TABLE OF SPFLI(여기는 데이터 타입이 아니라 오브젝트가 와야합니다! LIKE이기 때문이죠.)

DATA GS_123 LIKE LINE OF ITAB.

```
FORMAT COLOR 5 INTENSIFIED ON.
WRITE SY-VLINE, "가로선".
WRITE: 3 '이름', 10 SY-VLINE, 13 '파트', 23 SY-VLINE.
FORMAT RESET.
WRITE / SY-ULINE, "세로선".
WRITE :SY-VLINE, 3'김준호', 10 SY-VLINE, 13'영업지원', 23 SY-VLINE.
WRITE / SY-ULINE.
WRITE: SY-VLINE, 3'인세훈', 10 SY-VLINE, 13 '물류제조', 23 SY-VLINE.
WRITE SY-ULINE.

DATA GV_NAME(10) TYPE C.
GV_NAME = 'PARKSEONHYEONG'.
WRITE / GV_NAME+3(3).
| *DATA: ITAB LIKE TABLE OF SPFTL.
| *DATA: GS_123 LIKE LINE OF ITAB.

| DATA: BEGIN OF GS_DATA,
|   A(4) TYPE C,
|   B TYPE I,
|   END OF GS_DATA.
| DATA GT_DATA LIKE TABLE OF GS_DATA WITH HEADER LINE.

WRITE : 50.
```

위 코드 실행 결과

프로그램 Z 1

이름	파트
김준호	영업지원
인세훈	물류제조

KSE 50

ZTEST123에서 테이블 유형으로 만들면 STRUCTURE을 TABLE처럼 편하게 가져다 쓸 수 있다.

- 왜냐면 STRUCTURE을 TABLE유형으로 미리 만들어 놓았기 때문에!
- 근데 잘 안쓰대요...
- DICTIONARY를 쓴답니다!

HEADERLINE 없는 ITAB

```
DATA: ITAB LIKE TABLE OF SPFLI.
DATA: WA LIKE LINE OF ITAB.

SELECT * FROM SPFLI
      INTO CORRESPONDING FIELDS OF TABLE ITAB.

LOOP AT ITAB INTO WA.
  WRITE: / WA-CARRID, WA-CONNID, WA-COUNTRYFR.
ENDLOOP.
```

프로그램 ZPSH1000

```
AA 0017 US
AA 0064 US
AZ 0555 IT
AZ 0788 IT
AZ 0789 JP
AZ 0790 IT
DL 0106 US
DL 1699 US
DL 1984 US
JL 0407 JP
JL 0408 DE
LH 0400 DE
LH 0401 US
LH 0402 DE
LH 2402 DE
LH 2407 DE
QF 0005 SG
QF 0006 DE
SQ 0002 SG
SQ 0015 US
SQ 0158 SG
SQ 0988 SG
UA 0941 DE
UA 3504 US
UA 3516 US
UA 3517 DE
```

HEADERLINE 있는 ITAB.

```
DATA: ITAB LIKE TABLE OF SPFLI WITH HEADER LINE.
*DATA: WA LIKE LINE OF ITAB.
```

```
SELECT * FROM SPFLI
      INTO CORRESPONDING FIELDS OF TABLE ITAB.
```

```
⊞ LOOP AT ITAB.
  | WRITE: / ITAB-CARRID, ITAB-CONNID, ITAB-COUNTRYFR.
  | ENDLOOP.
```

프로그램 ZPSH1000

```
AA 0017 US
AA 0064 US
AZ 0555 IT
AZ 0788 IT
AZ 0789 JP
AZ 0790 IT
DL 0106 US
DL 1699 US
DL 1984 US
JL 0407 JP
JL 0408 DE
LH 0400 DE
LH 0401 US
LH 0402 DE
LH 2402 DE
LH 2407 DE
QF 0005 SG
QF 0006 DE
SQ 0002 SG
SQ 0015 US
SQ 0158 SG
SQ 0988 SG
UA 0941 DE
UA 3504 US
UA 3516 US
UA 3517 DE
```

ABAP의 반복문

- DO [N TIMES].
STATEMENTS ~~
ENDDO.
- WHILE [논리적인 조건].
STATEMENTS~~~

ENDWHILE.

- LOOP AT....옵션.

STATEMENTS~~~

ENDLOOP.

ABAP의 조건문

- IF ELSE문
- CASE문

REFRESH ITAB은 BODY만 삭제.

대부분 CLEAR을 씁니다.

INTER TABLE에는 APPEND.

DB TABLE에는 INSERT!

COLLECT WA INTO ITAB

금액, 수량 등 숫자에 관련된 것을 합산하는 함수 같아요.

SY-INDEX는 LOOP를 돌 때 몇번째 돌고 있는지 자동으로 CHECK해주는 변수이다.
LOOP를 처음에 돌 때 값이 1이고, 한번 돌 때 마다 1씩 증가함.

- EX) DO 5 TIMES면 1부터 5까지 증가하게됨.

COLLECT 사용 법 예제

```

DATA: BEGIN OF WA,
  ID(4) TYPE C,
  NAME(6) TYPE C,
  MATH(3) TYPE P,
  ENG(3) TYPE P,
END OF WA.
DATA: ITAB LIKE TABLE OF WA.

WA-ID = 'U20'.
WA-NAME = 'LEE'.
WA-MATH = 99.
WA-ENG = 90.
COLLECT WA INTO ITAB.
CLEAR: WA.

WA-ID = 'U21'.
WA-NAME = 'LEE'.
WA-MATH = 45.
WA-ENG = 60.
COLLECT WA INTO ITAB.
CLEAR: WA.

WA-ID = 'U20'.
WA-NAME = 'LEE'.
WA-MATH = 45.
WA-ENG = 60.
COLLECT WA INTO ITAB.
CLEAR: WA.

WA-ID = 'U20'.
WA-NAME = 'KIM'.
WA-MATH = 77.
WA-ENG = 88.
COLLECT WA INTO ITAB.
CLEAR: WA.

LOOP AT ITAB INTO WA.
  WRITE: / WA-ID, WA-NAME, WA-MATH, WA-ENG.
ENDLOOP.

```

프로그램 ZPSH1000			
U20	LEE	144	150
U21	LEE	45	60
U20	KIM	77	88

MODIFY ITAB FROM WA.

→ WA의 해당값 전체를 바꾼다.

MODIFY ITAB FROM WA TRANSPORTING MATH ENG.

→ 특정값만 바꾼다.

외부에서 바꾸고싶으면

MODIFY ITAB FROM WA INDEX 1 TRANSPORTING MATH ENG.

→ INDEX 설정해 주는 것이 굉장히 중요하다!!

DELETE ADJACENT DUPLICATES FROM GT_DATA COMPARING F1 F2.

→ F1과 F2를 비교해서 같은 값이 있으면 중복 제거! 단 밑에 있는 값으로 덮는다.

READ TABLE GT_DATA WITH KEY F1 = '500'.

- READ TABLE의 KEY 조건은 무조건 '=' 이것만 가능하다.

READ TABLE 이후에는 SY-SUBRC(읽으면 0 못읽으면 4)를 이용해서

IF

~~~~~

ELSE

~~~~~

ENDIF

형식을 취함.