Modularization techniques:

Breaking of large programs into small piece of units is known as modularization.

Purpose:1)reusability,

2)To improve the readability,

3)prevention of redundancies,

4)encapsulation of the data.

Types : A)Includes,

B)Sub-routines,

c)Function modules.

INCLUDES : They aren't self executable programs mainly these programs are required to make the code which is reusable.These are called include programs.

Subroutines : It is a block of code introduced by FORM and concluded by ENDFORM and they are used for local modularization.

Calling of sub-routines : a)Call by value,

b) Call by Reference

c)Call by Value & Return.

```
*&---------------------------------------------------------------------*
*& Report ZPROGRAM_DAY6
*&---------------------------------------------------------------------*
*&
*&---------------------------------------------------------------------*
REPORT zprogram_day6.

*INCLUDE zsb_pro1_top.
*SELECT-OPTIONS s_kunnr FOR lv_kunnr.
*PARAMETERS p_land1 TYPE kna1-land1.
*PERFORM get_cust_info.
*PERFORM get_dist_info.
*
*FORM get_cust_info.
*  SELECT kunnr name1 ort01 land1 INTO TABLE lt_tab FROM kna1
*    WHERE kunnr IN s_kunnr and land1 eq p_land1.
*ENDFORM.
*
```

```
*FORM get_dist_info.
*  LOOP AT lt_tab INTO wa_tab.
*    WRITE /: wa_tab-kunnr,
*             wa_tab-name1,
*             wa_tab-ort01,
*             wa_tab-land1.
*    ENDLOOP.
*ENDFORM.


***SUBROUTINES***
**CALL BY VALUE**
*DATA B TYPE I VALUE 20.
*PERFORM SUB1 USING B.
*      WRITE B.
*FORM SUB1 USING VALUE(P_B)
*      P_B = 50.
*ENDFORM.


**CALL BY REFERENCE AND IT IS BY DEFAULT and no value keyword is used in the form statemnt **
*DATA A TYPE I VALUE 20.
*PERFORM SUB2 USING A.
*      WRITE A .
*FORM SUB CHANGING P_A
*      P_A = 50.
*  ENDFORM.


**CALL BY Value and return**

*  DATA A TYPE I VALUE 20.
*PERFORM SUB CHANGING A.
*      WRITE A .
*FORM SUB CHANGING VALUE(P_A)
*      P_A = 50.
*  ENDFORM.
*&---------------------------------------------------------------------*
*& Form SUB2
*&---------------------------------------------------------------------*
*& text
*&---------------------------------------------------------------------*
*&      --> A
*&---------------------------------------------------------------------*
*FORM sub2  USING    p_a.
*
*
*ENDFORM.
```

```
*PARAMETERS:LV_ONE TYPE I,
*          LV_TWO TYPE I.
*DATA SUM_OK TYPE I.
*
*CALL FUNCTION 'Z280_FM1'
*EXPORTING
*     INPUT_ONE = LV_ONE
*     INPUT_TWO = LV_TWO
*IMPORTING
*      SUM = SUM_OK
*.
*WRITE : SUM_OK.




*DATA : lt_tab TYPE ZST_P,
*       wa_tab TYPE ZST_F.
*
*
*select-options s_vbeln for wa_tab-vbeln.
*
*start-of-selection.
*
*CALL FUNCTION 'ZFM_TABLES'
*EXPORTING
*INPUT_VBELEN1       = s_vbeln-low
*INPUT_VBELEN2       = s_vbeln-high
*  TABLES
*    gt_tab             = lt_tab
*          .
*
*loop at lt_tab into wa_tab.
*    write : / wa_tab-vbeln,
*              wa_tab-ernam,
*              wa_tab-erzet,
*              wa_tab-erdat,
*              wa_tab-bzirk.
*  endloop.
```

```
*DATA : it_tab type   ZST_P1,
*      wt_tab type   ZST_F1.
*
*SELECT-OPTIONS s_vbeln for wt_tab-vbeln.
*
*START-OF-SELECTION.
*
*CALL FUNCTION 'ZFM_TABLES_JOINS1'
* EXPORTING
*   INPUT_VBELEN1      = s_vbeln-low
*   INPUT_VBELEN2      = s_vbeln-high
*  TABLES
*    gt_tab           =  it_tab
*         .
*loop at it_tab into wt_tab.
*    write : / wt_tab-vbeln,
*            wt_tab-ernam,
*            wt_tab-matnr,
*            wt_tab-matwa.
*  endloop.



*&---------------------------------------------------------------------*
*& Report ZPROGRAM_DAY9
*&---------------------------------------------------------------------*
*&
*&---------------------------------------------------------------------*
*report ZPROGRAM_DAY9 no standard page heading line-count 20(3)
*line-size 500 message-id ztrupen_ust_msg.
*
*load-of-program.
*
*INCLUDE ZPROGRAM_DAY_TOP.
*
*select-options S_VBELN FOR wa_tab-VBELN obligatory.
*
*initialization.
*  clear: wa_Tab, lt_Tab.
*  S_VBELN-low  = '80000000'.
*  S_VBELN-high  = '80000009'.
*  append S_VBELN.
*
*at selection-screen on S_VBELN.
*
*  select single VBELN
*    into LV_VBELN
```

```
*    from LIKP
*     where VBELN in S_VBELN.
*
*  if sy-subrc eq 0.
*    message s000.  " se91 - tcode.
*  else.
*    message e001.  " se91 - tcode.
*  endif.
*
*start-of-selection.
*  perform get_mat_data.
*
*end-of-selection.
*  perform disp_mat_data.
*
*
*  data lv_count type i.
*  lv_count = sy-linct - sy-linno.
*  skip lv_count.
*
*top-of-page.
*  uline.
*  write :/ 'Records are found in LIKP table', 'the curent list index is :  ', sy-lsind.
*  write :/ 'Material No',
*         30 'Mat Ind',
*         40 'Mat Type',
*         55 'Mat UOM',
*         70 'Gross Weight',
*         90 'Net weight'.
*  uline.
*
*end-of-page.
*  write :/ 'current list page no:' , sy-pagno,
*          'Date: ', sy-datum,
*          'Time:', sy-uzeit.
*
*
*at LINE-SELECTION.
*
*CASE sy-lsind.
*  WHEN 1.
*
*  SELECT VBELN POSNR PSTYV
*         into TABLE lt_Tab2
*        from LIPS
*        WHERE VBELN eq wa_tab-VBELN.    " hide stmt
*
*    loop at lt_tab2 INTO wa_tab2.
```

```
*     WRITE : /3 wa_tab2-VBELN HOTSPOT,
*             30 wa_tab2-POSNR,
*             40 wa_tab2-PSTYV.
*
*
*     hide   wa_tab2-VBELN.
*
*     ENDLOOP.
*
*
*
*ENDCASE.
*
*
*TOP-OF-PAGE DURING LINE-SELECTION.
*
*CASE sy-lsind.
*  WHEN 1.
*WRITE : 'the curent list index is :  ', sy-lsind.
*    uline.
*  write :/3 'Material No',
*         30 'Mat plant',
*         40 'Mat stat',
*         55 'Mat flag'.
*     uline.
*ENDCASE.
*
*
**&---------------------------------------------------------------------*
**& Form get_mat_data
**&---------------------------------------------------------------------*
*form get_mat_data .
*
*  select VBELN ERNAM ERZET ERDAT BZIRK
*      into table lt_Tab
*      from LIKP
*      where VBELN in  S_VBELN.
*  if sy-subrc eq 0.
*    write :/ 'Records are found in LIKP table'.
*  else.
*    write : / 'Records are not found in LIKP table'.
*  endif.
*
*endform.
**&---------------------------------------------------------------------*
**& Form disp_mat_data
**&---------------------------------------------------------------------*
*
```

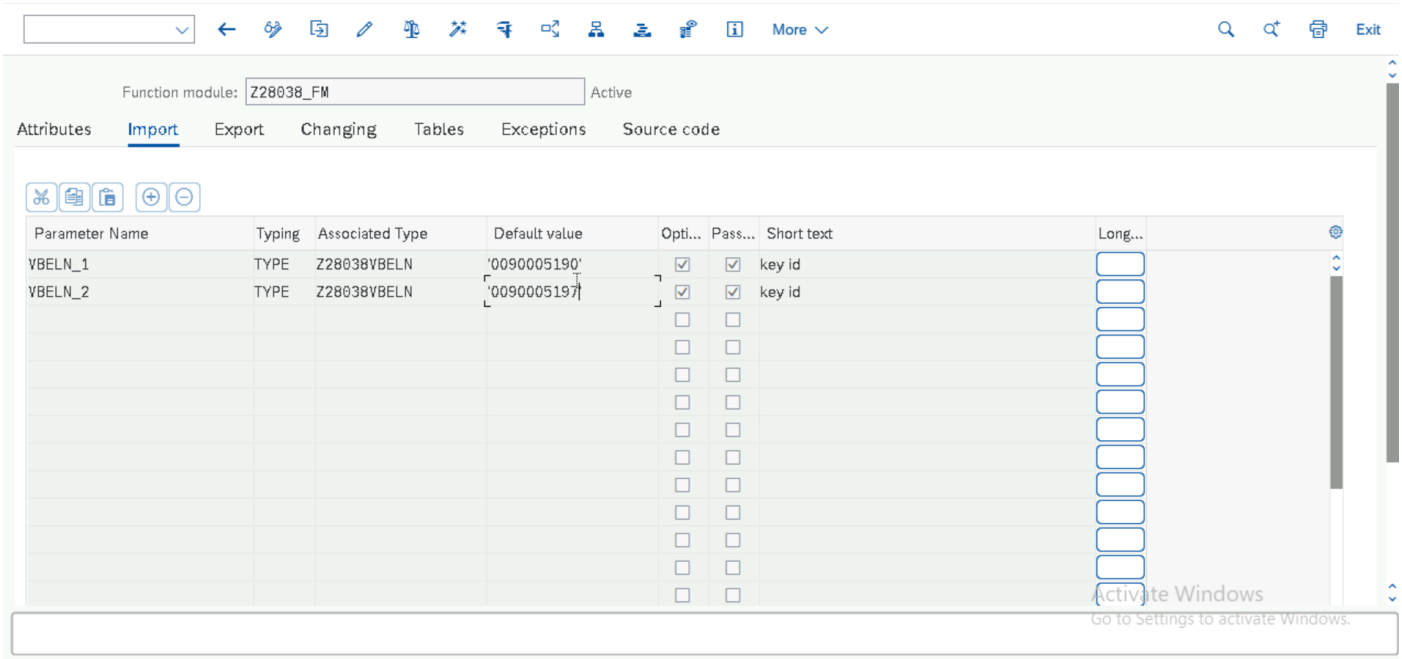
```
*form disp_mat_data .  " basic list disp
*  loop at lt_tab into wa_tab.
*    write : / wa_tab-VBELN  HOTSPOT,
*            30 wa_tab-ERNAM,
*            40 wa_tab-ERZET,
*            55 wa_tab-ERDAT,
*            70 wa_tab-BZIRK.
*
*    hide wa_tab-VBELN.  " for next where
*
*  endloop.
*endform.
```

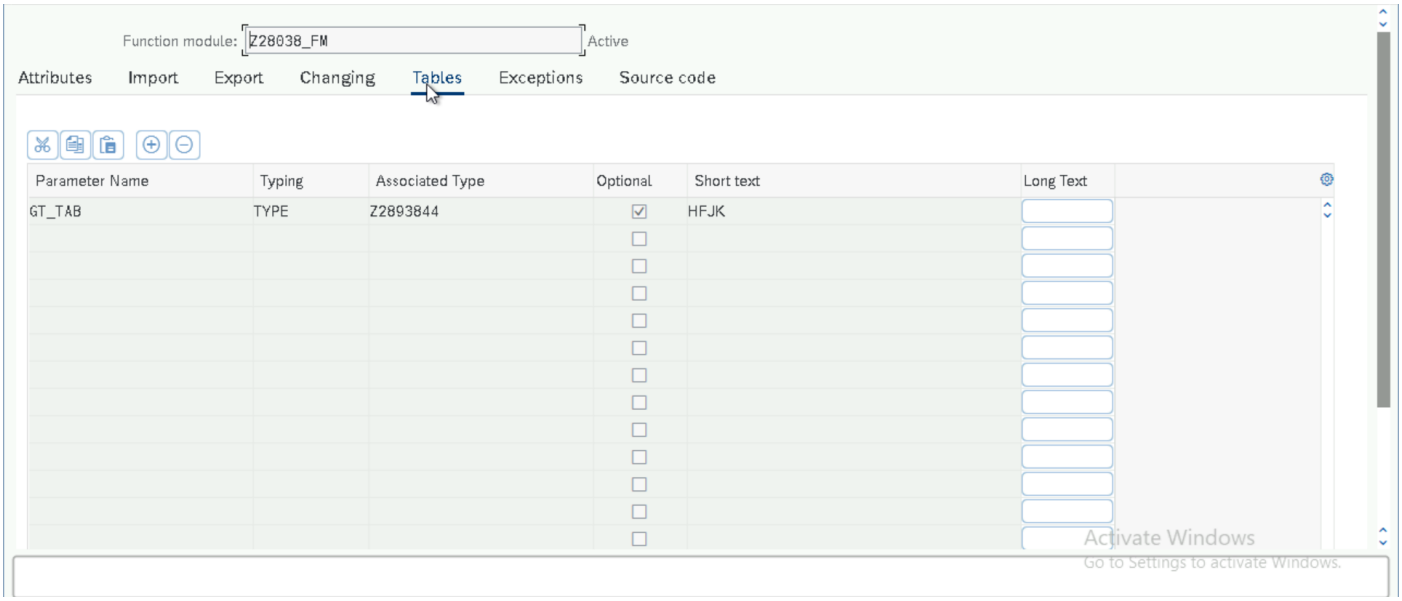
Function Modules :

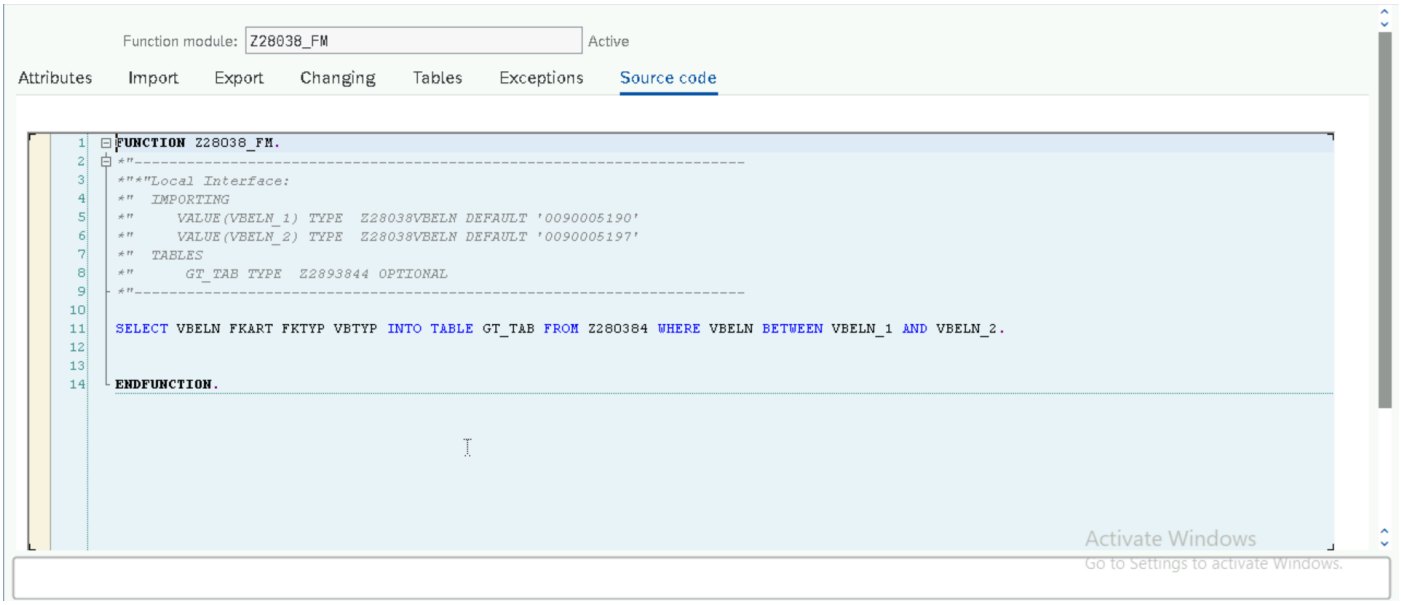
They are of 3 types a)Normal b)Remote c)Update respectively.

```
*&---------------------------------------------------------------------*
*& Report Z280384_QUESFMM
*&---------------------------------------------------------------------*
*&
*&---------------------------------------------------------------------*
REPORT Z280384_QUESFMM.

DATA : LT_TAB TYPE Z2893844, "tabletype
      WA_AREA TYPE Z289384, "linetype
      lv_vbeln TYPE Z28038VBELN.

SELECT-OPTIONS S_vbeln for lv_vbeln.

CALL FUNCTION 'Z28038_FM'
 EXPORTING
   VBELN_1      = S_vbeln-low
   VBELN_2      = S_vbeln-high
 TABLES
    gt_tab       = LT_TAB
         .

LOOP AT LT_TAB INTO WA_AREA.
  WRITE : /
     WA_AREA-VBELN,
     WA_AREA-FKART,
```

```
    WA_AREA-FKTYP,
    WA_AREA-VBTYP.
ENDLOOP.
```







REPORTS: Reports are the one of the 'receif' components of the SAP.

They are mainly of 3 types : 1)Classial Reports,

2)Interactive Reports,

3)ALV Reports(Abap list viewer).

Classical Reports : These are the most simplest reports and no interaction can be done from the user after the interacion.

Events : 1)Load of Program,

2)Initialization,

3)At-Selection-Screen,

4)At-Selection-Screen on output,

5)Start-of-selection,

6)Top-of-page,

7)End-of-page,

8)End-of-selection.

```
REPORT zguru_reports_ci NO STANDARD PAGE HEADING LINE-COUNT 20(3)
LINE-SIZE 450 MESSAGE-ID z280_msg1.

LOAD-OF-PROGRAM.

INCLUDE ZCLASSICAL_MATNR_TOP.
```

```
SELECT-OPTIONS S_MATNR FOR WA_AREA-MATNR OBLIGATORY.

INITIALIZATION.
     CLEAR : WA_AREA, IT_TAB.

     S_MATNR-low = '1'.
     S_MATNR-high = '100'.
     APPEND S_MATNR.


AT SELECTION-SCREEN ON S_MATNR.
  SELECT SINGLE MATNR INTO LV_MATNR FROM MARA WHERE MATNR IN S_MATNR.

  IF SY-subrc EQ 0.
    MESSAGE S000.
    ELSE.
    MESSAGE E000.
  ENDIF.

START-OF-SELECTION.
      PERFORM GET_MARA_DATA.
    END-OF-SELECTION.

    PERFORM DIS_MARA_DATA.

    DATA LV_COUNT TYPE I.
    LV_COUNT = SY-linct - SY-linno.
    SKIP LV_COUNT.

TOP-OF-PAGE.
    ULINE.
    WRITE :/ 'Material No',
          30 'Mat Ind',
          40 'Mat Type',
          55 'Mat UOM',
          70 'Gross Weight',
          90 'Net weight'.

END-OF-PAGE.
    ULINE.
    WRITE :/ 'current list page no:' , sy-pagno,
           'Date: ', sy-datum,
           'Time:', sy-uzeit.
*&---------------------------------------------------------------------*
*& Form GET_MARA_DATA
*&---------------------------------------------------------------------*
FORM get_mara_data .

  SELECT MATNR MBRSH MTART MEINS BRGEW NTGEW INTO TABLE IT_TAB FROM MARA WHERE MATNR IN
 S_MATNR.

    IF SY-subrc EQ 0.
      WRITE : 'rECORDS ARE FOUND IN MARA'.
      ELSE.
        WRITE : 'RECORDS ARE NOT FOUND'.
    ENDIF.

ENDFORM.
*&---------------------------------------------------------------------*
*& Form DIS_MARA_DATA
*&---------------------------------------------------------------------*
*& text
FORM dis_mara_data .
```

```
  LOOP AT IT_TAB INTO WA_AREA.
    WRITE : WA_AREA-MATNR,
            WA_AREA-MBRSH,
            WA_AREA-MTART,
            WA_AREA-MEINS,
            WA_AREA-BRGEW,
            WA_AREA-NTGEW.
    ENDLOOP.


ENDFORM.
```

Material No	Mat Ind	Mat Type	Mat UOM	Gross Weight	Net weight
Records are found in mara table					
22	M	FERT	PC	280.000	250.000
23	M	FERT	PC	280.000	250.000
24	M	FERT	PC	290.000	260.000
25	M	FERT	PC	280.000	250.000
26	M	FERT	PC	1.000	0.000
27	M	FERT	PC	1.000	0.000
28	M	FERT	PC	1.000	0.000
30	M	HIBE	PC	280.000	0.000
31	M	HIBE	PC	280.000	0.000
32	M	FERT	PC	270.000	250.000
33	M	FERT	PC	280.000	250.000
34	M	FERT	PC	280.000	250.000
35	M	FERT	PC	280.000	250.000

current list page no: 1 Date: 05/14/2024 Time: 08:01:17

INTERACTIVE-REPORTS:

The lists are produced by class report and can be interactive with the user.An interactive report list user can be participate and retrieving the data at any point during the session.

We can have 1 basic report and 20 interactive reports.

Events :

1)Load of program

2)Start of selection

3)End of selection

4)Screen selection

5)Top of page

6)end of page

7)At line selection

8)At user command

9)At pf-status

10)end of screen selection

11)initialization.

They are 2 types : a)At line user,

b)At user command.

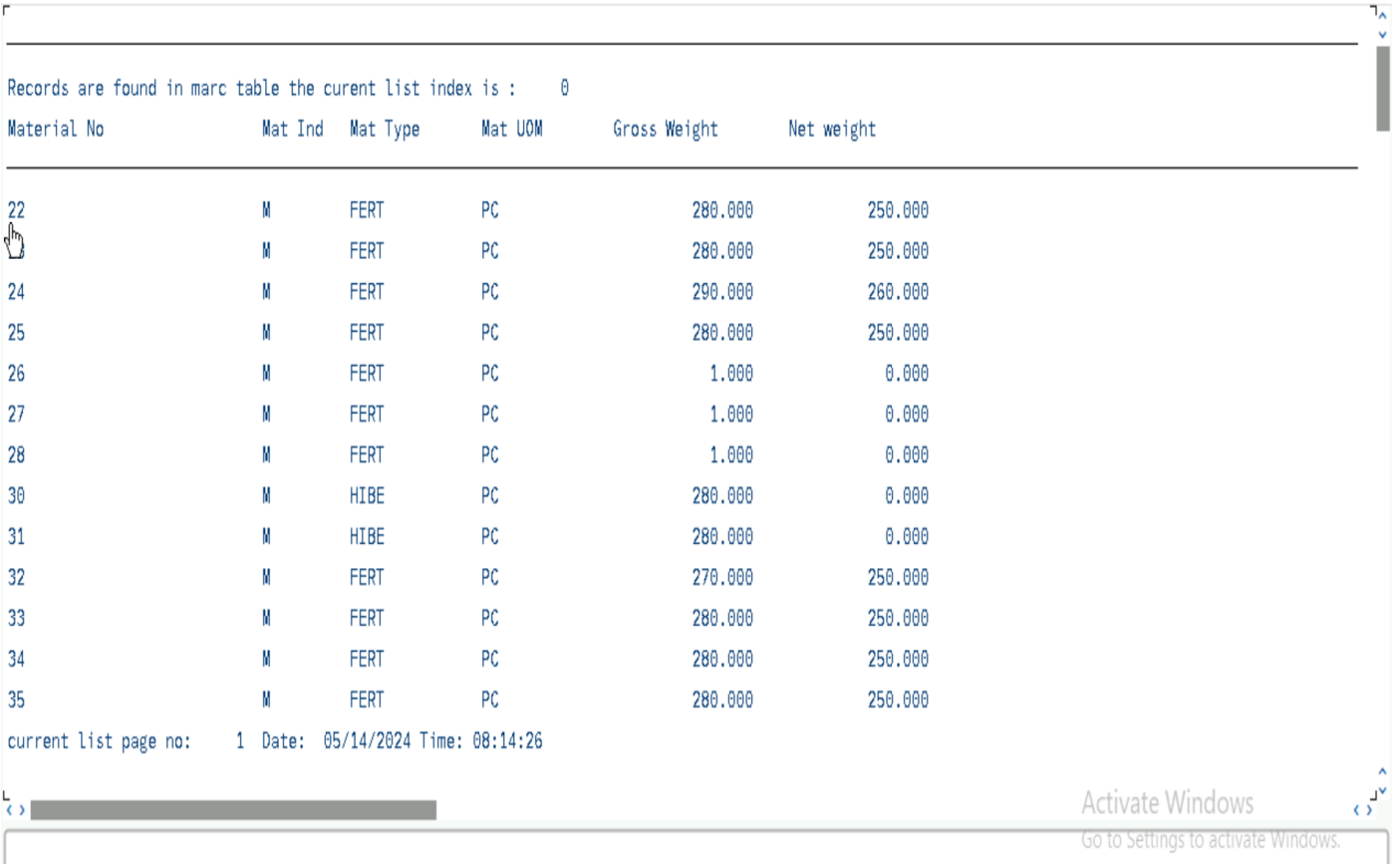
```
*********INTERACTIVE - AT LINE************

LOAD-OF-PROGRAM.

  INCLUDE zinclude_inter_atline.


  SELECT-OPTIONS s_matnr FOR wa_1-matnr OBLIGATORY.

INITIALIZATION.

  CLEAR: wa_1,it_tab1.
  s_matnr-low = '1'.
  s_matnr-high = '100'.
  APPEND s_matnr.


AT SELECTION-SCREEN ON s_matnr.
  SELECT SINGLE matnr INTO lv_matnr FROM mara WHERE matnr IN s_matnr.

  IF sy-subrc EQ 0.
    MESSAGE s000.
  ELSE.
    MESSAGE e001.
  ENDIF.


START-OF-SELECTION.
  PERFORM get_mat_data.

END-OF-SELECTION.

  PERFORM dis_mat_data.


  lv_count = sy-linct - sy-linno.
  SKIP lv_count.

TOP-OF-PAGE.
  ULINE.
  WRITE :/ 'Records are found in marc table', 'the curent list index is :  ', sy-lsind.
  WRITE :/ 'Material No',
         30 'Mat Ind',
         40 'Mat Type',
         55 'Mat UOM',
         70 'Gross Weight',
         90 'Net weight'.
  ULINE.

END-OF-PAGE.
  WRITE :/ 'current list page no:' , sy-pagno,
           'Date: ', sy-datum,
           'Time:', sy-uzeit.


******EXTRA FROM THAT OF THE CLASSICAL REPORTS*****

  CASE sy-lsind.
    WHEN 1.
```

```
      SELECT matnr werks pstat lvorm INTO TABLE it_tab2
        FROM marc
        WHERE matnr EQ wa_1-matnr.

      LOOP AT it_tab2 INTO wa_2.
        WRITE : /3 wa_2-matnr HOTSPOT,
                  30 wa_2-werks,
                  40 wa_2-pstat,
                  55 wa_2-lvrom.
        HIDE wa_2-matnr.
      ENDLOOP.
    WHEN 2.


      SELECT matnr spras maktx maktg INTO TABLE it_tab3
        FROM makt
        WHERE matnr EQ wa_2-matnr.

      LOOP AT it_tab3 INTO wa_3.
        WRITE : /3 wa_3-matnr,
                  30 wa_3-spras,
                  40 wa_3-maktx,
                  55 wa_3-maktg.

      ENDLOOP.
  ENDCASE.

TOP-OF-PAGE DURING LINE-SELECTION.
  CASE sy-lsind.
    WHEN 1.
      WRITE :/ 'Records are found in marc -
plant  table', 'the curent list index is :  ', sy-lsind.
      ULINE.
      WRITE :/3 'Material No',
             30 'Mat plant',
             40 'Mat stat',
             55 'Mat flag'.

      ULINE.
    WHEN 2.
      WRITE : / 'Records are found in makt- text table', 'the curent list index is :  '
, sy-lsind.
      ULINE.
      WRITE :/3 'Material No',
             30 'Mat language',
             40 'Mat description',
             55 'Mat grp'.
      ULINE.
  ENDCASE.




&---------------------------------------------------------------------*
& Form GET_MAT_DATA

FORM get_mat_data .


  SELECT matnr mbrsh mtart meins brgew ntgew
    INTO TABLE it_tab1 FROM mara
```

```
    WHERE matnr IN s_matnr.



ENDFORM.
&---------------------------------------------------------------------*
& Form DIS_MAT_DATA
&---------------------------------------------------------------------*
FORM dis_mat_data .

  LOOP AT it_tab1 INTO wa_1.
    WRITE : / wa_1-matnr HOTSPOT,
              wa_1-mbrsh,
              wa_1-mtart,
              wa_1-meins,
              wa_1-brgew,
              wa_1-ntgew.
    HIDE wa_1-matnr.
  ENDLOOP.

ENDFORM.



********at user command ***************


             report ztrupen_ust_classical_rep no standard page heading line-count 20(3)
line-size 500 message-id z280_ust_msg.

load-of-program.

  include ztrupen_ust_cmd_top.


  select-options s_matnr for wa_tab-matnr obligatory.

initialization.
  clear: wa_Tab, lt_Tab.
  s_matnr-low  = '1'.
  s_matnr-high  = '100'.
  append s_matnr.

at selection-screen on s_matnr.    " 1 to 100

  select single matnr
    into lv_matnr
     from mara
     where matnr in s_matnr.

  if sy-subrc eq 0.
    message s000.   " se91 - tcode.
  else.
    message e001.   " se91 - tcode.
  endif.

start-of-selection.

  set pf-status 'ZTRUPEN_PFS' EXCLUDING 'INFO' .   " SE41 - CODE
```

```
  perform get_mat_data.

end-of-selection.
  perform disp_mat_data.


  data lv_count type i.
  lv_count = sy-linct - sy-linno.
  skip lv_count.

top-of-page.
  uline.
  write :/ 'Records are found in marc table', 'the curent list index is :  ', sy-lsind.
  write :/ 'Material No',
         30 'Mat Ind',
         40 'Mat Type',
         55 'Mat UOM',
         70 'Gross Weight',
         90 'Net weight'.
  uline.

end-of-page.
  write :/ 'current list page no:' , sy-pagno,
          'Date: ', sy-datum,
          'Time:', sy-uzeit.

at user-command.

  case sy-ucomm.
    when 'DISP'.
      write 'UST INFO '.
    when 'INFO'.
      write / : 'the current date:', sy-datum,
                'TimeL ', sy-uzeit.

    when 'TCODE'.
      call transaction 'ZTRUPEN_UST_MATMAST'.

when 'SAP'.
    write 'SAP LABS BNGL '.

when 'ABAP'.
    write 'SAP ABAP HYD '.

  endcase.




*&---------------------------------------------------------------------*
*& Form get_mat_data
*&---------------------------------------------------------------------*
form get_mat_data .

  select matnr mbrsh mtart meins brgew ntgew
      into table lt_Tab
      from mara
      where matnr in  s_matnr.
  if sy-subrc eq 0.
    write :/ 'Records are found in mara table'.
  else.
```

```
    write : / 'Records are not found in mara table'.
  endif.

endform.
*&---------------------------------------------------------------------*
*& Form disp_mat_data
*&---------------------------------------------------------------------*

form disp_mat_data .   " basic list disp
  loop at lt_tab into wa_tab.
    write : / wa_tab-matnr  hotspot,
              30 wa_tab-mbrsh,
              40 wa_tab-mtart,
              55 wa_tab-meins,
              70 wa_tab-brgew,
              90 wa_tab-ntgew.

    hide wa_tab-matnr.   " for next where

  endloop.
endform
```



Records are found in marc table the curent list index is : 0

Material No	Mat Ind	Mat Type	Mat UOM	Gross Weight	Net weight
22	M	FERT	PC	280.000	250.000
	M	FERT	PC	280.000	250.000
24	M	FERT	PC	290.000	260.000
25	M	FERT	PC	280.000	250.000
26	M	FERT	PC	1.000	0.000
27	M	FERT	PC	1.000	0.000
28	M	FERT	PC	1.000	0.000
30	M	HIBE	PC	280.000	0.000
31	M	HIBE	PC	280.000	0.000
32	M	FERT	PC	270.000	250.000
33	M	FERT	PC	280.000	250.000
34	M	FERT	PC	280.000	250.000
35	M	FERT	PC	280.000	250.000

current list page no: 1 Date: 05/14/2024 Time: 08:14:26

.

**********At-User-Command**********

Records are found in marc table the curent list index is : 0

Material No	Mat Ind	Mat Type	Mat UOM	Gross Weight	Net weight
Records are found in mara table					
22	M	FERT	PC	280.000	250.000
23	M	FERT	PC	280.000	250.000
24	M	FERT	PC	290.000	260.000
25	M	FERT	PC	280.000	250.000
26	M	FERT	PC	1.000	0.000
27	M	FERT	PC	1.000	0.000
28	M	FERT	PC	1.000	0.000
30	M	HIBE	PC	280.000	0.000
31	M	HIBE	PC	280.000	0.000
32	M	FERT	PC	270.000	250.000
33	M	FERT	PC	280.000	250.000
34	M	FERT	PC	280.000	250.000

current list page no: 1 Date: 05/14/2024 Time: 09:11:01


Display UST info
Transaction info
More
Exit
Records are found in marc table the curent list index is : 0
Material No Mat Ind Mat Type Mat UOM Gross Weight Net weight
Records are found in mara table
22 M FERT PC 280.000 250.000
23 M FERT PC 280.000 250.000
24 M FERT PC 290.000 260.000
25 M FERT PC 280.000 250.000
26 M FERT PC 1.000 0.000
27 M FERT PC 1.000 0.000
28 M FERT PC 1.000 0.000
30 M HIBE PC 280.000 0.000
31 M HIBE PC 280.000 0.000
32 M FERT PC 270.000 250.000
33 M FERT PC 280.000 250.000
34 M FERT PC 280.000 250.000
current list page no: 1 Date: 05/14/2024 Time: 09:11:01
Activate Windows
Go to Settings to activate Windows.
Save
Cancel