**CLASSICAL RECORD :**

REPORT ZKALESHA\_CLASSIC\_REP NO STANDARD PAGE HEADING LINE-COUNT 20(3) LINE-SIZE 500.  
load-OF-PROGRAM.  
include ZKALESHA\_CLASSIC\_TD.  
include ZKALESHA\_CLASSIC\_EVENTS.  
  
PERFORM PRINT.  
  
\*&---------------------------------------------------------------------\*  
\*& Form DEFALUT  
  
FORM defalut .  
  
s\_vbeln-low = '1'.  
s\_vbeln-high = '100'.  
APPEND s\_vbeln.  
clear :  i\_tab,wa\_tab.  
ENDFORM.  
\*&---------------------------------------------------------------------\*  
\*& Form SELECTION\_SCREEN\_VALID  
  
FORM selection\_screen\_valid .  
SELECT SINGLE vbeln into lv\_vbeln from vbak where vbeln in s\_vbeln.  
    if sy-subrc eq 0.  
      MESSAGE S000(ZKALESHA\_MESSAGE).  
      else.  
        MESSAGE E001(ZKALESHA\_MESSAGE).  
       ENDIF.  
ENDFORM.  
\*&---------------------------------------------------------------------\*  
\*& Form SELECT\_QUERY  
  
FORM select\_query .  
SELECT VBELN ERDAT ERZET ERNAM ANGDT into TABLE i\_tab from vbak  
  WHERE VBELN in s\_vbeln.  
  
  if sy-subrc eq 0.  
    WRITE : / 'Records are found',sy-dbcnt.  
    else.  
      WRITE : / 'Records are not found'.  
      ENDIF.  
ENDFORM.  
\*&---------------------------------------------------------------------\*  
\*& Form PRINT  
  
FORM print .  
  FORMAT COLOR 1.  
loop at i\_tab into wa\_tAB.  
 uline.  
  write : / wa\_tab-vbeln,sy-vline,  
  20 wa\_tab-erdat,sy-vline,  
  40 wa\_tab-erzet,sy-vline,  
  60 wa\_tab-ernam,sy-vline,  
  100 wa\_tab-angdt,sy-vline.  
 ENDLOOP.  
ENDFORM.

**INTERACTIVE REPORTS :**

**AT LINE SELECTION :**

**Ex 1**

REPORT ZKALESHA\_INTERACTIVE\_R NO STANDARD PAGE HEADING LINE-COUNT 20(3) LINE-SIZE 500.  
include ZKALESHA\_CLASSIC\_TD.  
include ZKALESHA\_CLASSIC\_EVENTS.  
  
PERFORM PRINT.  
  
\*&---------------------------------------------------------------------\*  
\*& Form DEFALUT  
  
FORM defalut .  
s\_vbeln-low = '1'.  
s\_vbeln-high = '100'.  
APPEND s\_vbeln.  
clear :  i\_tab,wa\_tab.  
ENDFORM.  
\*&---------------------------------------------------------------------\*  
\*& Form SELECTION\_SCREEN\_VALID  
  
FORM selection\_screen\_valid .  
SELECT SINGLE vbeln into lv\_vbeln from vbak where vbeln in s\_vbeln.  
    if sy-subrc eq 0.  
      MESSAGE S000(ZKALESHA\_MESSAGE).  
      else.  
        MESSAGE E001(ZKALESHA\_MESSAGE).  
       ENDIF.  
ENDFORM.  
\*&---------------------------------------------------------------------\*  
\*& Form SELECT\_QUERY  
  
FORM select\_query .  
SELECT VBELN ERDAT ERZET ERNAM ANGDT into TABLE i\_tab from vbak  
  WHERE VBELN in s\_vbeln.  
  
  if sy-subrc eq 0.  
    WRITE : / 'Records are found',sy-dbcnt.  
    else.  
      WRITE : / 'Records are not found'.  
      ENDIF.  
ENDFORM.  
\*&---------------------------------------------------------------------\*  
\*& Form PRINT  
  
FORM print .  
  FORMAT COLOR 1.  
loop at i\_tab into wa\_tAB.  
 uline.  
  write : / wa\_tab-vbeln HOTSPOT,sy-vline,  
  20 wa\_tab-erdat,sy-vline,  
  40 wa\_tab-erzet,sy-vline,  
  60 wa\_tab-ernam,sy-vline,  
  100 wa\_tab-angdt,sy-vline.  
 HIDE wa\_tab-vbeln.  
  ENDLOOP.  
ENDFORM.  
  
at LINE-SELECTION.  
  CASE sy-lsind.  
    WHEN 1.  
      SELECT vbELN MATNR MATWA PMATN  into TABLE i\_tab1 from vbap  
        where vbeln eq wa\_tab-vbeln.  
        LOOP AT i\_tab1 into wa\_tab1.  
          write : / wa\_tab1-vbeln,  
          wa\_tab1-matnr,  
          wa\_tab1-matwa,  
          wa\_tab1-pmatn.  
  
          HIDE wa\_tab1-vbeln.  
          ENDLOOP.  
  
  ENDCASE. **Ex 2**

REPORT ZKALESHA\_INTERACTIVE\_R2 NO STANDARD PAGE HEADING LINE-COUNT 25(3) LINE-SIZE 500  
MESSAGE-ID ZKALESHA\_MESSAGE.  
load-OF-PROGRAM.  
TYPES : BEGIN OF s\_tab,  
  kunnr TYPE KUNNR,  
land TYPE LAND1\_GP,  
name type NAME1\_GP,  
  city TYPE ORT01\_GP,  
postal TYPE PSTLZ,  
region TYPE REGIO,  
  end of s\_tab.  
   DATA : itab type STANDARD TABLE OF s\_tab,  
         wa\_tab TYPE s\_tab.  
  
  
TYPES : BEGIN OF s1\_tab,  
  kunnr TYPE KUNNR,  
taxgr type VKORG,  
as type VTWEG,  
bj type SPART,  
  end of s1\_tab.  
   DATA : itab1 type STANDARD TABLE OF s1\_tab,  
         wa\_tab1 TYPE s1\_tab.  
   TYPES : BEGIN OF s2\_tab,  
  kunnr TYPE KUNNR,  
land TYPE LAND1\_GP,  
st type STCEG,  
  end of s2\_tab.  
   DATA : itab2 type STANDARD TABLE OF s2\_tab,  
         wa\_tab2 TYPE s2\_tab.  
  
   SELECTION-SCREEN  begin of BLOCK b WITH FRAME TITLE TEXT-001.  
     select-OPTIONS s\_kunnr for wa\_tab-kunnr.  
     SELECTION-SCREEN end of BLOCK b.  
  
     SELECTION-SCREEN BEGIN OF BLOCK b1 WITH FRAME TITLE TEXT-002.  
       PARAMETERS : R1 RADIOBUTTON GROUP RAD USER-COMMAND FLAG DEFAULT 'X',  
                    R2 RADIOBUTTON GROUP RAD,  
                    P\_city type ort01\_gp MODIF ID M1,  
                    p\_name type name1\_gp MODIF ID M2.  
       SELECTION-SCREEN ULINE /30(60).  
       SELECTION-SCREEN COMMENT /20(70) lv\_comm.  
       SELECTION-SCREEN skip 3.  
       SELECTION-SCREEN uline.  
       SELECTION-SCREEN end of block b1.  
       INITIALIZATION.  
       lv\_comm = 'KALESHA SCREEN'.  
  
       at SELECTION-SCREEN OUTPUT.  
         LOOP AT SCREEN.  
         IF R1 = 'X' AND SCREEN-GROUP1 = 'M1'.  
              SCREEN-ACTIVE = 0.  
              ELSEIF R2 = 'X' AND SCREEN-GROUP1 = 'M2'.  
                SCREEN-ACTIVE = 0.  
              ENDIF.  
              MODIFY SCREEN.  
              ENDLOOP.  
  
         START-OF-SELECTION.  
         if R1 = 'X'.  
         perform ort01.  
         else.  
         perform NAME1.  
         ENDIF.  
  
         TOP-OF-PAGE.  
         ULINE.  
         FORMAT COLOR 1.  
         write : /  
         'Customer Number',sy-vline,30'Country/Region Key',sy-vline,60'Name',sy-vline,  
         80'City',sy-vline,100'Postal Code',sy-vline,120'Region'.  
         FORMAT COLOR OFF.  
         uline.  
  
         end-of-PAGE.  
         write : /  
         'Time :',sy-uzeit,'Date :',sy-datum,'Page No :',sy-pagno.  
  
         end-of-SELECTION.  
              PERFORM body.  
  
         AT LINE-SELECTION.  
           perform case.  
  
  
  
\*&---------------------------------------------------------------------\*  
\*& Form ort01  
  
FORM ort01.  
 SELECT KUNNR LAND1 NAME1 ORT01 PSTLZ REGIO into TABLE itab from kna1  
           where kunnr in s\_kunnr AND ort01 eq p\_city.  
           IF  sy-subrc eq 0.  
              message s000.  
              else.  
                MESSAGE e001.  
                ENDIF.  
ENDFORM.  
\*&---------------------------------------------------------------------\*  
\*& Form NAME1  
  
FORM name1 .  
SELECT KUNNR LAND1 NAME1 ORT01 PSTLZ REGIO into TABLE itab from kna1  
           where kunnr in s\_kunnr AND NAME1 eq p\_name.  
           IF  sy-subrc eq 0.  
              message s000.  
              else.  
                MESSAGE e001.  
                ENDIF.  
ENDFORM.  
\*&---------------------------------------------------------------------\*  
\*& Form body  
  
FORM body .  
FORMAT COLOR COL\_GROUP.  
         loop at itab into wa\_tab.  
           write : /  
           sy-vline,  
           wa\_tab-kunnr COLOR 2,sy-vline,  
           30 wa\_tab-land,sy-vline,  
           60 wa\_tab-name,sy-vline,  
           80 wa\_tab-city,sy-vline,  
           100 wa\_tab-postal,sy-vline,  
           120 wa\_tab-region.  
  
           hide wa\_tab-kunnr.  
           ENDLOOP.  
           FORMAT COLOR OFF.  
ENDFORM.  
\*&---------------------------------------------------------------------\*  
\*& Form case  
  
FORM case .  
CASE sy-lsind.  
             WHEN 1.  
              select KUNNR VKORG VTWEG SPART into TABLE itab1 from kna1vv  
                where kunnr eq wa\_tab-kunnr.  
                FORMAT COLOR COL\_GROUP.  
         loop at itab1 into wa\_tab1.  
           write : /  
           sy-vline,  
           wa\_tab1-kunnr COLOR 2,sy-vline,  
           30 wa\_tab1-taxgr,sy-vline,  
           60 wa\_tab1-as,sy-vline,  
           80 wa\_tab1-bj,sy-vline.  
           hide wa\_tab1-kunnr.  
           ENDLOOP.  
           FORMAT COLOR OFF.  
  
             WHEN 2.  
               select KUNNR LAND1 STCEG into TABLE itab2 from knas  
                where kunnr eq wa\_tab1-kunnr.  
                 loop at itab2 into wa\_tab2.  
           write : /  
           sy-vline,  
           wa\_tab2-kunnr COLOR 2,sy-vline,  
           30 wa\_tab2-land,sy-vline,  
           60 wa\_tab2-st,sy-vline.  
           ENDLOOP.  
           ENDCASE.  
ENDFORM.

**AT USER COMMAND :**

REPORT ZKALESHA\_INTER\_UCOMM NO STANDARD PAGE HEADING  
LINE-COUNT 25(3) LINE-SIZE 500 MESSAGE-ID Zkalesha\_message.  
load-OF-PROGRAM.  
TYPES : BEGIN OF ls\_tab,  
  kunnr TYPE kunnr,  
  land TYPE land1\_gp,  
  name type NAME1\_GP,  
  city type ort01\_gp,  
  region type regio,  
  END OF ls\_tab.  
  
  DATA : itab TYPE STANDARD TABLE OF ls\_tab,  
        wa\_tab type ls\_tab,  
        lv\_kunnr TYPE kunnr.  
  
  select-OPTIONS s\_kunnr for wa\_tab-kunnr.  
  
  INITIALIZATION.  
  s\_kunnr-low = '1'.  
  s\_kunnr-high = '100'.  
  APPEND s\_kunnr.  
  clear : itab,wa\_tab.  
  
  at SELECTION-SCREEN on s\_kunnr.  
    SELECT SINGLE kunnr into lv\_kunnr from kna1  
      where kunnr in s\_kunnr.  
  
      IF sy-subrc eq 0.  
        MESSAGE s000.  
        else.  
          MESSAGE e001.  
  
      ENDIF.  
  
      START-OF-SELECTION.  
  
      set PF-STATUS 'ZKALESHA\_PFS'.  
  
      SELECT KUNNR LAND1 NAME1 ort01 REGIO into TABLE itab from kna1 WHERE  
        kunnr in s\_kunnr.  
  
        IF sy-subrc eq 0.  
          WRITE 'REcords are found'.  
          else.  
            WRITE 'Records Are not found'.  
             ENDIF.  
  
             TOP-OF-PAGE.  
             WRITE : / 'Kunnr','Land','Name','City','Region'.  
  
             end-of-PAGE.  
             write : / 'data :',sy-datum,'pageno:',sy-pagno.  
      end-of-SELECTION.  
      LOOP AT itab into wa\_tab.  
        WRITE : / wa\_tab-kunnr,  
        wa\_tab-land,  
        wa\_tab-name,  
        wa\_tab-city,  
        wa\_tab-region.  
    ENDLOOP.

**ALV :**

**LIST:**

REPORT ZKALESHA\_ALV\_LIST\_DISPLAY NO STANDARD PAGE HEADING MESSAGE-ID ZKALESHA\_MESSAGE.  
\*include ZKALESHA\_ALV\_LISTDIS\_TOP.  
  
TYPES : BEGIN OF s\_tab,  
  EBELN TYPE EBELN,  
  BYKRS TYPE BUKRS,  
  BSTYP TYPE EBSTYP,  
  BSART TYPE ESART,  
  END OF s\_tab.  
  
  DATA : itab type STANDARD TABLE OF s\_tab,  
        wa\_tab TYPE s\_tab,  
        lv\_ebeln TYPE EBELN,  
        wa\_fieldcat TYPE SLIS\_FIELDCAT\_ALV,  
        it\_fieldcat TYPE SLIS\_T\_FIELDCAT\_ALV.  
  
load-OF-PROGRAM.  
  
  
select-OPTIONS s\_ebeln for wa\_tab-ebeln.  
INITIALIZATION.  
s\_ebeln-low = '4500000001'.  
s\_ebeln-high = '4500000016'.  
append s\_ebeln.  
clear : itab,wa\_fieldcat,it\_fieldcat.  
  
at SELECTION-SCREEN on s\_ebeln.  
.  
  SELECT SINGLE ebeln into lv\_ebeln from ekko where ebeln in s\_ebeln.  
    IF sy-subrc eq 0.  
      MESSAGE s000.  
      else.  
        MESSAGE e001.  
    ENDIF.  
  
    START-OF-SELECTION.  
    return.  
  
    PERFORM ALV\_FIELD\_INSERT.  
    perform select\_query.  
    end-of-SELECTION.  
    PERFORM ALV\_DISPLAY\_FUNC.  
\*&---------------------------------------------------------------------\*  
\*& Form ALV\_FIELD\_INSERT  
  
FORM alv\_field\_insert .  
  
 " clear : wa\_fieldcat,it\_fieldcat.  
wa\_fieldcat-col\_pos = '1'.  
wa\_fieldcat-fieldname = 'EBELN'.  
wa\_fieldcat-seltext\_l = 'MAterial No'.  
wa\_fieldcat-tabname = 'ITAB'.  
APPEND wa\_fieldcat to it\_fieldcat.  
  
clear wa\_fieldcat.  
wa\_fieldcat-col\_pos = '2'.  
wa\_fieldcat-fieldname = 'BYKRS'.  
wa\_fieldcat-seltext\_l = 'MAterial Bukrs'.  
wa\_fieldcat-tabname = 'ITAB'.  
APPEND wa\_fieldcat to it\_fieldcat.  
  
clear wa\_fieldcat.  
wa\_fieldcat-col\_pos = '3'.  
wa\_fieldcat-fieldname = 'BSTYP'.  
wa\_fieldcat-seltext\_l = 'MAterial Type'.  
wa\_fieldcat-tabname = 'ITAB'.  
APPEND wa\_fieldcat to it\_fieldcat.  
  
clear wa\_fieldcat.  
wa\_fieldcat-col\_pos = '4'.  
wa\_fieldcat-fieldname = 'BSART'.  
wa\_fieldcat-seltext\_l = 'MAterial Art'.  
wa\_fieldcat-tabname = 'ITAB'.  
APPEND wa\_fieldcat to it\_fieldcat.  
  
ENDFORM.  
\*&---------------------------------------------------------------------\*  
\*& Form select\_query  
  
FORM select\_query .  
SELECT ebeln bukrs bstyp bsart into TABLE itab from ekko  
  where ebeln in s\_ebeln.  
  IF sy-subrc eq 0.  
    write : 'Records are found',sy-dbcnt.  
    else.  
      write 'Records are not found'.  
     ENDIF.  
ENDFORM.  
\*&---------------------------------------------------------------------\*  
\*& Form ALV\_DISPLAY\_FUNC  
\*&---------------------------------------------------------------------\*  
\*& text  
\*&---------------------------------------------------------------------\*  
\*& -->  p1        text  
\*& <--  p2        text  
\*&---------------------------------------------------------------------\*  
FORM alv\_display\_func .  
CALL FUNCTION 'REUSE\_ALV\_LIST\_DISPLAY'  
 EXPORTING  
\*   I\_INTERFACE\_CHECK              = ' '  
\*   I\_BYPASSING\_BUFFER             =  
\*   I\_BUFFER\_ACTIVE                = ' '  
  I\_CALLBACK\_PROGRAM               =  'ZKALESHA\_ALV\_LIST\_DISPLAY'  
\*   I\_CALLBACK\_PF\_STATUS\_SET       = ' '  
\*   I\_CALLBACK\_USER\_COMMAND        = ' '  
\*   I\_STRUCTURE\_NAME               =  
\*   IS\_LAYOUT                      =  
   IT\_FIELDCAT                    = IT\_FIELDCAT  
\*   IT\_EXCLUDING                   =  
\*   IT\_SPECIAL\_GROUPS              =  
\*   IT\_SORT                        =  
\*   IT\_FILTER                      =  
\*   IS\_SEL\_HIDE                    =  
\*   I\_DEFAULT                      = 'X'  
\*   I\_SAVE                         = ' '  
\*   IS\_VARIANT                     =  
\*   IT\_EVENTS                      =  
\*   IT\_EVENT\_EXIT                  =  
\*   IS\_PRINT                       =  
\*   IS\_REPREP\_ID                   =  
\*   I\_SCREEN\_START\_COLUMN          = 0  
\*   I\_SCREEN\_START\_LINE            = 0  
\*   I\_SCREEN\_END\_COLUMN            = 0  
\*   I\_SCREEN\_END\_LINE              = 0  
\*   IR\_SALV\_LIST\_ADAPTER           =  
\*   IT\_EXCEPT\_QINFO                =  
\*   I\_SUPPRESS\_EMPTY\_DATA          = ABAP\_FALSE  
\*   IO\_SALV\_ADAPTER                =  
\*   O\_COMMON\_HUB                   =  
\* IMPORTING  
\*   E\_EXIT\_CAUSED\_BY\_CALLER        =  
\*   ES\_EXIT\_CAUSED\_BY\_USER         =  
  TABLES  
    t\_outtab                       = ITAB  
\* EXCEPTIONS  
\*   PROGRAM\_ERROR                  = 1  
\*   OTHERS                         = 2  
          .  
IF sy-subrc <> 0.  
\* Implement suitable error handling here  
ENDIF.  
  
  
ENDFORM.

**GRID :**

**EX 1:**

REPORT ZKALESHA\_ALV\_GRID\_DISPLAY NO STANDARD PAGE HEADING MESSAGE-ID ZKALESHA\_MESSAGE.  
include ZKALESHA\_ALV\_LISTDIS\_TOP.  
  
  
  
load-OF-PROGRAM.  
  
  
select-OPTIONS s\_ebeln for wa\_tab-ebeln.  
INITIALIZATION.  
s\_ebeln-low = '4500000001'.  
s\_ebeln-high = '4500000016'.  
append s\_ebeln.  
clear : itab,wa\_fieldcat,it\_fieldcat.  
  
at SELECTION-SCREEN on s\_ebeln.  
  SELECT SINGLE ebeln into lv\_ebeln from ekko where ebeln in s\_ebeln.  
    IF sy-subrc eq 0.  
      MESSAGE s000.  
      else.  
        MESSAGE e001.  
    ENDIF.  
  
    START-OF-SELECTION.  
    PERFORM ALV\_FIELD\_INSERT.  
    perform select\_query.  
    end-of-SELECTION.  
    PERFORM ALV\_DISPLAY\_FUNC.  
\*&---------------------------------------------------------------------\*  
\*& Form ALV\_FIELD\_INSERT  
  
FORM alv\_field\_insert .  
  
 " clear : wa\_fieldcat,it\_fieldcat.  
wa\_fieldcat-col\_pos = '1'.  
wa\_fieldcat-fieldname = 'EBELN'.  
wa\_fieldcat-seltext\_l = Text-001."'MAterial No'.  
wa\_fieldcat-tabname = 'ITAB'.  
APPEND wa\_fieldcat to it\_fieldcat.  
  
clear wa\_fieldcat.  
wa\_fieldcat-col\_pos = '2'.  
wa\_fieldcat-fieldname = 'BYKRS'.  
wa\_fieldcat-seltext\_l = Text-002."'MAterial Bukrs'.  
wa\_fieldcat-tabname = 'ITAB'.  
APPEND wa\_fieldcat to it\_fieldcat.  
  
clear wa\_fieldcat.  
wa\_fieldcat-col\_pos = '3'.  
wa\_fieldcat-fieldname = 'BSTYP'.  
wa\_fieldcat-seltext\_l = Text-003."'MAterial Type'.  
wa\_fieldcat-tabname = 'ITAB'.  
APPEND wa\_fieldcat to it\_fieldcat.  
  
clear wa\_fieldcat.  
wa\_fieldcat-col\_pos = '4'.  
wa\_fieldcat-fieldname = 'BSART'.  
wa\_fieldcat-seltext\_l = Text-004."'MAterial Art'.  
wa\_fieldcat-tabname = 'ITAB'.  
APPEND wa\_fieldcat to it\_fieldcat.  
  
ENDFORM.  
\*&---------------------------------------------------------------------\*  
\*& Form select\_query  
  
FORM select\_query .  
SELECT ebeln bukrs bstyp bsart into TABLE itab from ekko  
  where ebeln in s\_ebeln.  
  IF sy-subrc eq 0.  
    write : 'Records are found',sy-dbcnt.  
    else.  
      write 'Records are not found'.  
     ENDIF.  
ENDFORM.  
\*&---------------------------------------------------------------------\*  
\*& Form ALV\_DISPLAY\_FUNC  
\*&---------------------------------------------------------------------\*  
\*& text  
\*&---------------------------------------------------------------------\*  
\*& -->  p1        text  
\*& <--  p2        text  
\*&---------------------------------------------------------------------\*  
FORM alv\_display\_func .  
CALL FUNCTION 'REUSE\_ALV\_GRID\_DISPLAY'  
 EXPORTING  
\*   I\_INTERFACE\_CHECK              = ' '  
\*   I\_BYPASSING\_BUFFER             =  
\*   I\_BUFFER\_ACTIVE                = ' '  
  I\_CALLBACK\_PROGRAM               =  'ZKALESHA\_ALV\_LIST\_DISPLAY'  
\*   I\_CALLBACK\_PF\_STATUS\_SET       = ' '  
\*   I\_CALLBACK\_USER\_COMMAND        = ' '  
\*   I\_STRUCTURE\_NAME               =  
\*   IS\_LAYOUT                      =  
   IT\_FIELDCAT                    = IT\_FIELDCAT  
\*   IT\_EXCLUDING                   =  
\*   IT\_SPECIAL\_GROUPS              =  
\*   IT\_SORT                        =  
\*   IT\_FILTER                      =  
\*   IS\_SEL\_HIDE                    =  
\*   I\_DEFAULT                      = 'X'  
\*   I\_SAVE                         = ' '  
\*   IS\_VARIANT                     =  
\*   IT\_EVENTS                      =  
\*   IT\_EVENT\_EXIT                  =  
\*   IS\_PRINT                       =  
\*   IS\_REPREP\_ID                   =  
\*   I\_SCREEN\_START\_COLUMN          = 0  
\*   I\_SCREEN\_START\_LINE            = 0  
\*   I\_SCREEN\_END\_COLUMN            = 0  
\*   I\_SCREEN\_END\_LINE              = 0  
\*   IR\_SALV\_LIST\_ADAPTER           =  
\*   IT\_EXCEPT\_QINFO                =  
\*   I\_SUPPRESS\_EMPTY\_DATA          = ABAP\_FALSE  
\*   IO\_SALV\_ADAPTER                =  
\*   O\_COMMON\_HUB                   =  
\* IMPORTING  
\*   E\_EXIT\_CAUSED\_BY\_CALLER        =  
\*   ES\_EXIT\_CAUSED\_BY\_USER         =  
  TABLES  
    t\_outtab                       = ITAB  
\* EXCEPTIONS  
\*   PROGRAM\_ERROR                  = 1  
\*   OTHERS                         = 2  
          .  
IF sy-subrc <> 0.  
\* Implement suitable error handling here  
ENDIF.  
  
  
ENDFORM.

**EX 2:**

**GRID WITH LOGO :**

REPORT ZKALESHA\_GRID\_LOGO NO STANDARD PAGE HEADING MESSAGE-ID ZKALESHA\_MESSAGE.  
include ZKALESHA\_ALV\_LISTDISLOGO\_TOP.  
  
  
  
load-OF-PROGRAM.  
  
  
select-OPTIONS s\_ebeln for wa\_tab-ebeln.  
INITIALIZATION.  
s\_ebeln-low = '4500000001'.  
s\_ebeln-high = '4500000016'.  
append s\_ebeln.  
clear : itab,wa\_fieldcat,it\_fieldcat.  
  
at SELECTION-SCREEN on s\_ebeln.  
  SELECT SINGLE ebeln into lv\_ebeln from ekko where ebeln in s\_ebeln.  
    IF sy-subrc eq 0.  
      MESSAGE s000.  
      else.  
        MESSAGE e001.  
    ENDIF.  
  
    START-OF-SELECTION.  
    PERFORM ALV\_FIELD\_INSERT.  
    perform select\_query.  
    end-of-SELECTION.  
    PERFORM ALV\_DISPLAY\_FUNC.  
\*&---------------------------------------------------------------------\*  
\*& Form ALV\_FIELD\_INSERT  
  
FORM alv\_field\_insert .  
  
 " clear : wa\_fieldcat,it\_fieldcat.  
wa\_fieldcat-col\_pos = '1'.  
wa\_fieldcat-fieldname = cs\_ebeln."'EBELN'.  
wa\_fieldcat-seltext\_l = Text-001."'MAterial No'.  
wa\_fieldcat-tabname = cs\_tab."'ITAB'.  
wa\_fieldcat-emphasize = 'C701'.  
APPEND wa\_fieldcat to it\_fieldcat.  
  
clear wa\_fieldcat.  
wa\_fieldcat-col\_pos = '2'.  
wa\_fieldcat-fieldname = cs\_bykrs."'BYKRS'.  
wa\_fieldcat-seltext\_l = Text-002."'MAterial Bukrs'.  
wa\_fieldcat-tabname = cs\_tab."'ITAB'.  
wa\_fieldcat-emphasize = 'C210'.  
APPEND wa\_fieldcat to it\_fieldcat.  
  
clear wa\_fieldcat.  
wa\_fieldcat-col\_pos = '3'.  
wa\_fieldcat-fieldname = cs\_bstyp."'BSTYP'.  
wa\_fieldcat-seltext\_l = Text-003."'MAterial Type'.  
wa\_fieldcat-tabname = cs\_tab."'ITAB'.  
wa\_fieldcat-emphasize = 'C311'.  
APPEND wa\_fieldcat to it\_fieldcat.  
  
clear wa\_fieldcat.  
wa\_fieldcat-col\_pos = '4'.  
wa\_fieldcat-fieldname = cs\_bsart."'BSART'.  
wa\_fieldcat-seltext\_l = Text-004."'MAterial Art'.  
wa\_fieldcat-tabname = cs\_tab."'ITAB'.  
wa\_fieldcat-emphasize = 'C110'.  
APPEND wa\_fieldcat to it\_fieldcat.  
  
ENDFORM.  
\*&---------------------------------------------------------------------\*  
\*& Form select\_query  
  
FORM select\_query .  
SELECT ebeln bukrs bstyp bsart into TABLE itab from ekko  
  where ebeln in s\_ebeln.  
  IF sy-subrc eq 0.  
    write : 'Records are found',sy-dbcnt.  
    else.  
      write 'Records are not found'.  
     ENDIF.  
ENDFORM.  
\*&---------------------------------------------------------------------\*  
\*& Form ALV\_DISPLAY\_FUNC  
\*&---------------------------------------------------------------------\*  
\*& text  
\*&---------------------------------------------------------------------\*  
\*& -->  p1        text  
\*& <--  p2        text  
\*&---------------------------------------------------------------------\*  
FORM alv\_display\_func .  
  "ls\_layout-no\_colhead = 'X'.  
  ls\_layout-no\_hline = 'X'.  
  "ls\_layout-zebra = 'X'.  
  ls\_layout-no\_vline = 'X'.  
CALL FUNCTION 'REUSE\_ALV\_GRID\_DISPLAY'  
 EXPORTING  
\*   I\_INTERFACE\_CHECK                 = ' '  
\*   I\_BYPASSING\_BUFFER                = ' '  
\*   I\_BUFFER\_ACTIVE                   = ' '  
  I\_CALLBACK\_PROGRAM                = sy-repid  
\*   I\_CALLBACK\_PF\_STATUS\_SET          = ' '  
\*   I\_CALLBACK\_USER\_COMMAND           = ' '  
  I\_CALLBACK\_TOP\_OF\_PAGE            = 'UST\_LOGO'  
\*   I\_CALLBACK\_HTML\_TOP\_OF\_PAGE       = ' '  
\*   I\_CALLBACK\_HTML\_END\_OF\_LIST       = ' '  
\*   I\_STRUCTURE\_NAME                  =  
\*   I\_BACKGROUND\_ID                   = ' '  
   I\_GRID\_TITLE                      = Text-005  
\*   I\_GRID\_SETTINGS                   =  
   IS\_LAYOUT                         = ls\_layout  
   IT\_FIELDCAT                       = it\_fieldcat  
\*   IT\_EXCLUDING                      =  
\*   IT\_SPECIAL\_GROUPS                 =  
\*   IT\_SORT                           =  
\*   IT\_FILTER                         =  
\*   IS\_SEL\_HIDE                       =  
\*   I\_DEFAULT                         = 'X'  
\*   I\_SAVE                            = ' '  
\*   IS\_VARIANT                        =  
\*   IT\_EVENTS                         =  
\*   IT\_EVENT\_EXIT                     =  
\*   IS\_PRINT                          =  
\*   IS\_REPREP\_ID                      =  
\*   I\_SCREEN\_START\_COLUMN             = 0  
\*   I\_SCREEN\_START\_LINE               = 0  
\*   I\_SCREEN\_END\_COLUMN               = 0  
\*   I\_SCREEN\_END\_LINE                 = 0  
\*   I\_HTML\_HEIGHT\_TOP                 = 0  
\*   I\_HTML\_HEIGHT\_END                 = 0  
\*   IT\_ALV\_GRAPHICS                   =  
\*   IT\_HYPERLINK                      =  
\*  IT\_ADD\_FIELDCAT                   =  
\*   IT\_EXCEPT\_QINFO                   =  
\*   IR\_SALV\_FULLSCREEN\_ADAPTER        =  
\*   O\_PREVIOUS\_SRAL\_HANDLER           =  
\*   O\_COMMON\_HUB                      =  
\* IMPORTING  
\*   E\_EXIT\_CAUSED\_BY\_CALLER           =  
\*   ES\_EXIT\_CAUSED\_BY\_USER            =  
  TABLES  
    t\_outtab                          = itab  
\* EXCEPTIONS  
\*   PROGRAM\_ERROR                     = 1  
\*   OTHERS                            = 2  
          .  
IF sy-subrc <> 0.  
\* Implement suitable error handling here  
ENDIF.  
  
  
ENDFORM.  
FORM UST\_LOGO.  
  data : wa\_header TYPE SLIS\_LISTHEADER,  
         it\_header TYPE SLIS\_T\_LISTHEADER.  
          wa\_header-key = 'Hyd'.  
          wa\_header-info = 'UST GLOBAL'.  
          wa\_header-typ = 'S'.  
          APPEND wa\_header to it\_header.  
  
          CALL FUNCTION 'REUSE\_ALV\_COMMENTARY\_WRITE'  
            EXPORTING  
              it\_list\_commentary       = it\_header  
            I\_LOGO                   = 'ZKALESHA\_LOGO'  
\*             I\_END\_OF\_LIST\_GRID       =  
\*             I\_ALV\_FORM               =  
                    .  
  
  
  ENDFORM.

**EX 2:**

**GRID WITH LOGO :**

REPORT ZKALESHA\_GRID\_LOGO\_LAYOUT2 NO STANDARD PAGE HEADING MESSAGE-ID ZKALESHA\_MESSAGE.  
TYPES : BEGIN OF S\_TAB,  
VBELN type VBELN\_VA,  
ERDAT TYPE ERDAT,  
ERZET TYPE ERZET,  
ERNAM TYPE ERNAM,  
ANGDT TYPE ANGDT\_V,  
  END OF S\_TAB.  
  DATA : ITAB TYPE STANDARD TABLE OF S\_TAB,  
        WA\_TAB TYPE S\_TAB,  
        LV\_VBELN TYPE VBELN,  
        IT\_FIELDCAT TYPE SLIS\_T\_FIELDCAT\_ALV,  
        WA\_FIELDCAT TYPE SLIS\_FIELDCAT\_ALV,  
        LS\_LAYOUT TYPE SLIS\_LAYOUT\_ALV.  
  CONSTANTS : CS\_TAB(30) TYPE C VALUE 'ITAB',  
              CS\_VBELN(10) TYPE C VALUE 'VBELN',  
              CS\_ERDAT(10) TYPE C VALUE 'ERDAT',  
              CS\_ERZET(10) TYPE C VALUE 'ERZET',  
              CS\_ERNAM(10) TYPE C VALUE 'ERNAM',  
              CS\_ANGDT(10) TYPE C VALUE 'ANGDT'.  
  
  SELECT-OPTIONS S\_VBELN FOR WA\_TAB-VBELN.  
  
  INITIALIZATION.  
  S\_VBELN-LOW = '1'.  
  S\_VBELN-HIGH = '50'.  
  APPEND S\_VBELN.  
  CLEAR : ITAB,WA\_FIELDCAT,IT\_FIELDCAT.  
  
  AT SELECTION-SCREEN ON S\_VBELN.  
    SELECT SINGLE VBELN INTO LV\_VBELN FROM VBAK WHERE VBELN IN S\_VBELN.  
      IF SY-SUBRC EQ 0.  
        MESSAGE S000.  
      ELSE.  
        MESSAGE E001.  
      ENDIF.  
  
      START-OF-SELECTION.  
      PERFORM INSERT\_DATA.  
      PERFORM SELECT\_QUERY.  
      END-OF-SELECTION.  
      PERFORM PRINT\_FUNC.  
\*&---------------------------------------------------------------------\*  
\*& Form INSERT\_DATA  
FORM insert\_data .  
  CLEAR : WA\_FIELDCAT,IT\_FIELDCAT.  
  WA\_FIELDCAT-col\_pos = SY-TABIX.  
  WA\_FIELDCAT-fieldname = CS\_VBELN.  
  WA\_FIELDCAT-seltext\_l = TEXT-001.  
  WA\_FIELDCAT-tabname = CS\_TAB.  
  WA\_FIELDCAT-emphasize = 'C410'.  
  WA\_FIELDCAT-no\_zero = 'X'.  
  APPEND WA\_FIELDCAT TO IT\_FIELDCAT.  
  
  CLEAR : WA\_FIELDCAT.  
  WA\_FIELDCAT-col\_pos = SY-TABIX.  
  WA\_FIELDCAT-fieldname = CS\_ERDAT.  
  WA\_FIELDCAT-seltext\_l = TEXT-002.  
  WA\_FIELDCAT-tabname = CS\_TAB.  
  WA\_FIELDCAT-emphasize = 'C501'.  
  APPEND WA\_FIELDCAT TO IT\_FIELDCAT.  
  
  CLEAR : WA\_FIELDCAT.  
  WA\_FIELDCAT-col\_pos = SY-TABIX.  
  WA\_FIELDCAT-fieldname = CS\_ERZET.  
  WA\_FIELDCAT-seltext\_l = TEXT-003.  
  WA\_FIELDCAT-tabname = CS\_TAB.  
  WA\_FIELDCAT-emphasize = 'C511'.  
  APPEND WA\_FIELDCAT TO IT\_FIELDCAT.  
  
  CLEAR : WA\_FIELDCAT.  
  WA\_FIELDCAT-col\_pos = SY-TABIX.  
  WA\_FIELDCAT-fieldname = CS\_ERNAM.  
  WA\_FIELDCAT-seltext\_l = TEXT-004.  
  WA\_FIELDCAT-tabname = CS\_TAB.  
  WA\_FIELDCAT-emphasize = 'C610'.  
  APPEND WA\_FIELDCAT TO IT\_FIELDCAT.  
  
  CLEAR : WA\_FIELDCAT.  
  WA\_FIELDCAT-col\_pos = SY-TABIX.  
  WA\_FIELDCAT-fieldname = CS\_ANGDT.  
  WA\_FIELDCAT-seltext\_l = TEXT-005.  
  WA\_FIELDCAT-tabname = CS\_TAB.  
  WA\_FIELDCAT-emphasize = 'C701'.  
  APPEND WA\_FIELDCAT TO IT\_FIELDCAT.  
  
ENDFORM.  
\*&---------------------------------------------------------------------\*  
\*& Form SELECT\_QUERY  
  
FORM select\_query .  
SELECT VBELN ERDAT ERZET ERNAM ANGDT INTO TABLE ITAB FROM VBAK  
  WHERE  VBELN IN S\_VBELN.  
      IF SY-SUBRC EQ 0.  
        WRITE : / 'RECORDS ARE FOUND',SY-DBCNT.  
      ELSE.  
        WRITE 'RECORDS ARE NOT FOUND'.  
      ENDIF.  
ENDFORM.  
\*&---------------------------------------------------------------------\*  
\*& Form PRINT\_FUNC  
  
FORM print\_func .  
  LS\_LAYOUT-no\_hline = 'X'.  
  LS\_LAYOUT-NO\_VLINE = 'X'.  
CALL FUNCTION 'REUSE\_ALV\_GRID\_DISPLAY'  
 EXPORTING  
   I\_CALLBACK\_PROGRAM                = SY-REPID  
   I\_CALLBACK\_TOP\_OF\_PAGE            = 'UST\_LOGO'  
   I\_GRID\_TITLE                      = TEXT-006  
   IS\_LAYOUT                         = LS\_LAYOUT  
   IT\_FIELDCAT                       = IT\_FIELDCAT  
  TABLES  
    t\_outtab                          = ITAB  
 EXCEPTIONS  
   PROGRAM\_ERROR                     = 1  
   OTHERS                            = 2  
          .  
IF sy-subrc <> 0.  
\* Implement suitable error handling here  
ENDIF.  
  
ENDFORM.  
  
FORM UST\_LOGO.  
  
  DATA : WA\_HEADER TYPE SLIS\_LISTHEADER,  
        IT\_HEADER TYPE SLIS\_T\_LISTHEADER.  
  
  WA\_HEADER-typ = 'S'.  
  WA\_HEADER-info = 'HYDERABAD'.  
  WA\_HEADER-key = 'UST GLOBAL'.  
  APPEND WA\_HEADER TO IT\_HEADER.  
  
  CALL FUNCTION 'REUSE\_ALV\_COMMENTARY\_WRITE'  
    EXPORTING  
      it\_list\_commentary       = IT\_HEADER  
     I\_LOGO                   = 'ZKALESHA\_LOGO'  
\*     I\_END\_OF\_LIST\_GRID       =  
\*     I\_ALV\_FORM               =  
            .  
  
  ENDFORM.

**HIERSEQ :**

REPORT zkalesha\_hierseq\_report NO STANDARD PAGE HEADING MESSAGE-ID zkalesha\_message.  
TYPES : BEGIN OF s\_tab,  
          ust(1),  
          vbeln  TYPE vbeln\_va,  
          erdat  TYPE erdat,  
          erzet  TYPE erzet,  
        END OF s\_tab.  
DATA : itab     TYPE STANDARD TABLE OF s\_tab,  
       wa\_tab   TYPE s\_tab,  
       lv\_vbeln TYPE vbeln.  
TYPES : BEGIN OF s1\_tab,  
          vbeln TYPE vbeln\_va,  
          posnr TYPE posnr\_va,  
          matnr TYPE matnr,  
        END OF s1\_tab.  
  
DATA :  
  itab2 TYPE STANDARD TABLE OF s1\_tab,  
  
  it\_fieldcat TYPE slis\_t\_fieldcat\_alv,  
  wa\_fieldcat LIKE LINE OF IT\_FIELDCAT,  
  ls\_layout   TYPE slis\_layout\_alv,  
  i\_key       TYPE slis\_keyinfo\_alv.  
  
SELECT-OPTIONS s\_vbeln FOR lv\_vbeln.  
  
INITIALIZATION.  
  s\_vbeln-low = '1'.  
  s\_vbeln-high = '100'.  
  APPEND s\_vbeln.  
  
  
AT SELECTION-SCREEN ON s\_vbeln.  
  SELECT SINGLE vbeln INTO lv\_vbeln FROM vbak WHERE vbeln IN s\_vbeln.  
  
  IF sy-subrc EQ 0.  
    MESSAGE s000.  
  ELSE.  
    MESSAGE e001.  
  
  ENDIF.  
  
START-OF-SELECTION.  
  PERFORM hierseq\_insert1 USING TEXT-001 TEXT-002 TEXT-003.  
  PERFORM hierseq\_insert1 USING TEXT-004 TEXT-002 TEXT-003.  
  PERFORM hierseq\_insert1 USING TEXT-005 TEXT-002 TEXT-003.  
  PERFORM hierseq\_insert1 USING TEXT-001 TEXT-008 TEXT-009.  
  PERFORM hierseq\_insert1 USING TEXT-006 TEXT-008 TEXT-009.  
  PERFORM hierseq\_insert1 USING TEXT-007 TEXT-008 TEXT-009.  
  
  PERFORM select\_query.  
  
END-OF-SELECTION.  
  PERFORM hierseq\_func.  
\*&---------------------------------------------------------------------\*  
\*& Form HIERSEQ\_INSERT  
  
FORM hierseq\_insert1 USING p\_text\_001 p\_text\_002 p\_text\_003 .  
  STATICS INDEX TYPE I VALUE 1.  
  IF INDEX EQ 4.  
    INDEX = 1.  
    ENDIF.  
  
  wa\_fieldcat-col\_pos = index.  
  wa\_fieldcat-fieldname = p\_text\_001.    "VBLEN  
  wa\_fieldcat-ref\_fieldname = p\_text\_001.    "VBLEN  
  wa\_fieldcat-ref\_tabname = p\_text\_002.     "VBAK  
  wa\_fieldcat-tabname = p\_text\_003.         "ITAB  
  
  index = index + 1.  
  APPEND wa\_fieldcat TO it\_fieldcat.  
  CLEAR : wa\_fieldcat.  
ENDFORM.  
\*&---------------------------------------------------------------------\*  
\*& Form HIERSEQ\_INSERT2  
  
"----\*  
\*& Form SELECT\_QUERY  
  
FORM select\_query .  
  SELECT vbeln erdat erzet INTO CORRESPONDING FIELDS OF TABLE itab FROM vbak WHERE vbeln IN s\_vbeln.  
  IF  sy-subrc EQ 0.  
    WRITE 'RECORDS ARE FOUND'.  
  ELSE.  
    WRITE 'RECORDS ARE NOT FOUND'.  
  
  ENDIF.  
  SELECT vbeln posnr matnr INTO TABLE itab2 FROM vbap WHERE vbeln IN s\_vbeln.  
ENDFORM.  
\*&---------------------------------------------------------------------\*  
\*& Form HIERSEQ\_INSERT6  
  
  
\*&---------------------------------------------------------------------\*  
\*& Form HIERSEQ\_FUNC  
\*&---------------------------------------------------------------------\*  
\*& text  
\*&---------------------------------------------------------------------\*  
\*& -->  p1        text  
\*& <--  p2        text  
\*&---------------------------------------------------------------------\*  
FORM hierseq\_func .  
  i\_key-header01 = 'VBELN'.  
  ls\_layout-expand\_fieldname = 'UST'.  
  CALL FUNCTION 'REUSE\_ALV\_HIERSEQ\_LIST\_DISPLAY'  
    EXPORTING  
\*     I\_INTERFACE\_CHECK  = ' '  
      i\_callback\_program = sy-repid  
\*     I\_CALLBACK\_PF\_STATUS\_SET       = ' '  
\*     I\_CALLBACK\_USER\_COMMAND        = ' '  
      is\_layout          = ls\_layout  
      it\_fieldcat        = it\_fieldcat  
\*     IT\_EXCLUDING       =  
\*     IT\_SPECIAL\_GROUPS  =  
\*     IT\_SORT            =  
\*     IT\_FILTER          =  
\*     IS\_SEL\_HIDE        =  
\*     I\_SCREEN\_START\_COLUMN          = 0  
\*     I\_SCREEN\_START\_LINE            = 0  
\*     I\_SCREEN\_END\_COLUMN            = 0  
\*     I\_SCREEN\_END\_LINE  = 0  
\*     I\_DEFAULT          = 'X'  
\*     I\_SAVE             = ' '  
\*     IS\_VARIANT         =  
\*     IT\_EVENTS          =  
\*     IT\_EVENT\_EXIT      =  
      i\_tabname\_header   = 'ITAB'  
      i\_tabname\_item     = 'ITAB2'  
\*     I\_STRUCTURE\_NAME\_HEADER        =  
\*     I\_STRUCTURE\_NAME\_ITEM          =  
      is\_keyinfo         = i\_key  
\*     IS\_PRINT           =  
\*     IS\_REPREP\_ID       =  
\*     IS\_HIERSEQ\_SETTINGS            =  
\*     I\_BYPASSING\_BUFFER =  
\*     I\_BUFFER\_ACTIVE    =  
\*     IR\_SALV\_HIERSEQ\_ADAPTER        =  
\*     IT\_EXCEPT\_QINFO    =  
\*     I\_SUPPRESS\_EMPTY\_DATA          = ABAP\_FALSE  
\*     I\_LIST\_DISPLAY\_ONLY            = ABAP\_FALSE  
\*     O\_COMMON\_HUB       =  
\* IMPORTING  
\*     E\_EXIT\_CAUSED\_BY\_CALLER        =  
\*     ES\_EXIT\_CAUSED\_BY\_USER         =  
    TABLES  
      t\_outtab\_header    = itab  
      t\_outtab\_item      = itab2  
\* EXCEPTIONS  
\*     PROGRAM\_ERROR      = 1  
\*     OTHERS             = 2  
    .  
  IF sy-subrc <> 0.  
\* Implement suitable error handling here  
  ENDIF.  
  
ENDFORM.

**LOCK OBJECTS :**

REPORT ZKALESHA\_LOCKOBJECT1.  
  
PARAMETERS lv\_matnr TYPE Matnr.  
CALL FUNCTION 'ENQUEUE\_EZKALESHA\_LOCK'  
 EXPORTING  
   MODE\_MARA            = 'E'  
   MANDT                = SY-MANDT  
   MATNR                = lv\_matnr  
\*   X\_MATNR              = ' '  
\*   \_SCOPE               = '2'  
\*   \_WAIT                = ' '  
\*   \_COLLECT             = ' '  
\* EXCEPTIONS  
\*   FOREIGN\_LOCK         = 1  
\*   SYSTEM\_FAILURE       = 2  
\*   OTHERS               = 3  
          .  
IF sy-subrc <> 0.  
\* Implement suitable error handling here  
ENDIF.  
  
\*BREAK-POINT.  
UPDATE mara SET matkl = '1' where matnr eq lv\_matnr.  
  
  
CALL FUNCTION 'DEQUEUE\_EZKALESHA\_LOCK'  
\* EXPORTING  
\*   MODE\_MARA       = 'E'  
\*   MANDT           = SY-MANDT  
\*   MATNR           =  
\*   X\_MATNR         = ' '  
\*   \_SCOPE          = '3'  
\*   \_SYNCHRON       = ' '  
\*   \_COLLECT        = ' '  
          .

**SELECTION SCREEN :**

REPORT zkalesha\_selscreen\_rad NO STANDARD PAGE HEADING LINE-COUNT 25(3) LINE-SIZE 500 MESSAGE-ID zkalesha\_message.  
  
TYPES : BEGIN OF ls\_tab,  
          vbeln TYPE vbeln\_va,  
          erdat TYPE erdat,  
          angdt TYPE angdt\_v,  
          auart TYPE auart,  
  
  
        END OF ls\_tab.  
  
DATA : i\_tab    TYPE STANDARD TABLE OF ls\_tab,  
       wa\_tab   TYPE ls\_tab,  
       lv\_vbeln TYPE vbeln\_va.  
  
SELECTION-SCREEN : BEGIN OF BLOCK b WITH FRAME TITLE TEXT-001.  
  SELECT-OPTIONS s\_vbeln FOR lv\_vbeln OBLIGATORY NO-EXTENSION.  
SELECTION-SCREEN : END OF BLOCK b.  
  
  
SELECTION-SCREEN : BEGIN OF BLOCK b2 WITH FRAME TITLE TEXT-002.  
  PARAMETERS : r1      RADIOBUTTON GROUP rad USER-COMMAND flag DEFAULT 'X',  
               r2      RADIOBUTTON GROUP rad,  
               p\_erdat TYPE erdat MODIF ID m1,  
               p\_angdt TYPE angdt MODIF ID m2.  
  SELECTION-SCREEN ULINE /20(60).  
  "SELECTION-SCREEN COMMENT /20(70) lv\_comm.  
  SELECTION-SCREEN SKIP 2.  
  SELECTION-SCREEN ULINE.  
SELECTION-SCREEN : END OF BLOCK b2.  
  
  
  
\*SELECTION-SCREEN : BEGIN OF BLOCK b1 WITH FRAME TITLE TEXT-003.  
\*  PARAMETERS : p\_erdat TYPE erdat MODIF ID M1,  
\*               p\_angdt TYPE angdt MODIF ID M2.  
\*  SELECTION-SCREEN ULINE /20(60).  
\*  "SELECTION-SCREEN COMMENT /20(70) lv\_comm.  
\*  SELECTION-SCREEN SKIP 2.  
\*  SELECTION-SCREEN ULINE.  
\*  
\*SELECTION-SCREEN : END OF BLOCK b1.  
  
  
  
AT SELECTION-SCREEN OUTPUT.  
  LOOP AT SCREEN.  
    IF r1 = 'X' AND screen-group1 = 'M2'.  
      screen-active  = 0.  
    ELSEIF r2 = 'X' AND screen-group1 = 'M1'.  
      screen-active = 0.  
    ENDIF.  
    MODIFY SCREEN.  
  ENDLOOP.  
  
  
  
\*INITIALIZATION.  
\*\*  s\_vbeln-low = '1'.  
\*\*  s\_vbeln-high = '100'.  
\*\*  APPEND s\_vbeln.  
\*\*  CLEAR : i\_tab,wa\_tab.  
\*  lv\_comm(30) = 'KALESHA SCREEN'.  
  
  
  
\*  SELECT SINGLE vbeln INTO lv\_vbeln FROM vbak  
\*    WHERE vbeln IN S\_VBELn.  
\*  IF sy-subrc EQ 0.  
\*    MESSAGE s000(zkalesha\_message).  
\*  ELSE.  
\*    MESSAGE e001(zkalesha\_message).  
\*  ENDIF.  
  
START-OF-SELECTION.  
  IF r1 = 'X'.  
    PERFORM angdt.  
  ELSE.  
    PERFORM erdat.  
  ENDIF.  
  
  
\*  IF sy-subrc EQ 0.  
\*    WRITE : 'RECORDS FOUND',sy-dbcnt.  
\*  ELSE.  
\*    WRITE 'not found'.  
\*  ENDIF.  
  
  
\*&---------------------------------------------------------------------\*  
\*& Form angdt  
\*  
FORM angdt .  
  SELECT vbeln erdat angdt auart  INTO TABLE i\_tab FROM vbak  
  
     WHERE vbeln IN s\_vbeln AND angdt EQ p\_angdt.  
\*     TOP-OF-PAGE.  
\*  ULINE.  
\*  FORMAT COLOR COL\_POSITIVE.  
\*  WRITE : 'VBELN',sy-vline,20 'ERDAT',sy-vline,50 'ANGDT',sy-vline,80 'AUART',sy-vline.  
\*  FORMAT COLOR OFF.  
\*  ULINE.  
\*  
\*END-OF-PAGE.  
\*  WRITE : / 'DATE :',sy-datum,'TIME :',sy-uzeit,'PAGENO :',sy-pagno.  
\*  
\*END-OF-SELECTION.  
  FORMAT COLOR 1.  
  LOOP AT i\_tab INTO wa\_tab.  
    WRITE : /  
    sy-vline,  
    wa\_tab-vbeln COLOR 4,sy-vline,  
    20 wa\_tab-erdat,sy-vline,  
    40 wa\_tab-angdt,sy-vline,  
    70 wa\_tab-auart,sy-vline,  
  
  
    sy-vline.  
  
  ENDLOOP.  
  FORMAT COLOR OFF.  
ENDFORM.  
\*&---------------------------------------------------------------------\*  
\*& Form erdat  
\*  
FORM erdat .  
  SELECT vbeln erdat angdt auart  INTO TABLE i\_tab FROM vbak  
  
      WHERE vbeln IN s\_vbeln AND erdat EQ p\_erdat.  
\*      TOP-OF-PAGE.  
\*  ULINE.  
\*  FORMAT COLOR COL\_POSITIVE.  
\*  WRITE : 'VBELN',sy-vline,20 'ERDAT',sy-vline,50 'ANGDT',sy-vline,80 'AUART',sy-vline.  
\*  FORMAT COLOR OFF.  
\*  ULINE.  
\*  
\*END-OF-PAGE.  
\*  WRITE : / 'DATE :',sy-datum,'TIME :',sy-uzeit,'PAGENO :',sy-pagno.  
\*  
\*END-OF-SELECTION.  
  FORMAT COLOR 1.  
  LOOP AT i\_tab INTO wa\_tab.  
    WRITE : /  
    sy-vline,  
    wa\_tab-vbeln COLOR 4,sy-vline,  
    20 wa\_tab-erdat,sy-vline,  
    40 wa\_tab-angdt,sy-vline,  
    70 wa\_tab-auart,sy-vline,  
  
  
    sy-vline.  
  
  ENDLOOP.  
  FORMAT COLOR OFF.  
ENDFORM.

**GET CURSOR :**

**EX 1:**

REPORT zkalesha\_getcourser.  
INCLUDE zkalesha\_simple\_getcourser.  
  
START-OF-SELECTION.  
  
  FORMAT COLOR 3.  
  WRITE 'BASIC LIST'.  
  FORMAT COLOR OFF.  
  
  FORMAT COLOR 7.  
  WRITE : / var1,var2,var3.  
  FORMAT COLOR OFF.  
  
  FORMAT COLOR 6.  
  WRITE /'UST HYD'.  
  FORMAT COLOR OFF.  
  
AT LINE-SELECTION.  
  GET CURSOR FIELD fname.  
  IF fname = 'VAR1'.  
    WRITE  'HAI'.  
  ELSEIF fname = 'VAR2'.  
    WRITE  'KALESHA'.  
  ELSEIF fname = 'VAR3'.  
    WRITE 'VALI'.  
  ENDIF.

**EX 2:**

REPORT zcl\_kalesha\_getcursor\_vbak.  
DATA lv\_vbeln TYPE vbeln\_va.  
  
TYPES : BEGIN OF ls\_tab,   " vbak  
          vbeln TYPE vbeln\_va,  
          kunnr TYPE kunnr,  
          erdat TYPE erdat,  
        END OF ls\_tab.  
  
DATA : it\_TAB TYPE TABLE OF ls\_tab,  
       wa\_TAB TYPE ls\_tab.  
  
SELECT-OPTIONS s\_vbeln FOR lv\_VBELN.  
  
  
START-OF-SELECTION.  
  SELECT vbeln  
  kunnr  
  erdat  
  INTO TABLE it\_TAB  
  FROM vbak  
  WHERE vbeln IN s\_vbeln.  
  
  
END-OF-SELECTION.  
  LOOP AT it\_TAB INTO wa\_TAB.  
  
    FORMAT HOTSPOT ON.  
    WRITE : / wa\_TAB-vbeln,  
              wa\_TAB-kunnr.  
    FORMAT HOTSPOT OFF.  
    WRITE     wa\_TAB-erdat.  
  ENDLOOP.  
  
TOP-OF-PAGE.  
  WRITE : / 'SaleS doc No',  
  'Customerno',  
  'Date'.  
  
AT LINE-SELECTION.  
  DATA : fname(20),  
         fval(20).  
  GET CURSOR FIELD fname VALUE fval.  
  IF fname = 'WA\_TAB-VBELN'.  
    SET PARAMETER ID 'AUN' FIELD fval.  
    CALL TRANSACTION 'VA03' AND SKIP FIRST SCREEN.  
  
  ELSEIF fname = 'WA\_TAB-KUNNR'.  
    SET PARAMETER ID 'KUN' FIELD fval.  
    CALL TRANSACTION 'XD02' AND SKIP FIRST SCREEN.  
  ENDIF.

**FOR ALL ENTRIES :**

REPORT zcl\_kalesha\_getcursor\_vbak.  
DATA lv\_vbeln TYPE vbeln\_va.  
  
TYPES : BEGIN OF ls\_tab,   " vbak  
          vbeln TYPE vbeln\_va,  
          kunnr TYPE kunnr,  
          erdat TYPE erdat,  
        END OF ls\_tab.  
  
DATA : it\_TAB TYPE TABLE OF ls\_tab,  
       wa\_TAB TYPE ls\_tab.  
  
SELECT-OPTIONS s\_vbeln FOR lv\_VBELN.  
  
  
START-OF-SELECTION.  
  SELECT vbeln  
  kunnr  
  erdat  
  INTO TABLE it\_TAB  
  FROM vbak  
  WHERE vbeln IN s\_vbeln.  
  
  
END-OF-SELECTION.  
  LOOP AT it\_TAB INTO wa\_TAB.  
  
    FORMAT HOTSPOT ON.  
    WRITE : / wa\_TAB-vbeln,  
              wa\_TAB-kunnr.  
    FORMAT HOTSPOT OFF.  
    WRITE     wa\_TAB-erdat.  
  ENDLOOP.  
  
TOP-OF-PAGE.  
  WRITE : / 'SaleS doc No',  
  'Customerno',  
  'Date'.  
  
AT LINE-SELECTION.  
  DATA : fname(20),  
         fval(20).  
  GET CURSOR FIELD fname VALUE fval.  
  IF fname = 'WA\_TAB-VBELN'.  
    SET PARAMETER ID 'AUN' FIELD fval.  
    CALL TRANSACTION 'VA03' AND SKIP FIRST SCREEN.  
  
  ELSEIF fname = 'WA\_TAB-KUNNR'.  
    SET PARAMETER ID 'KUN' FIELD fval.  
    CALL TRANSACTION 'XD02' AND SKIP FIRST SCREEN.  
  ENDIF.

**OOPS :**

**ATTRIBUTES :**

REPORT zcl\_kalesha\_oops\_attributes.  
  
\*\*\*\*\*\*\*\*ATTRIBUTES:  
  
  
\*DATA KAL\_OBJ TYPE REF TO ZCL\_KALESHA\_OOPS\_ATTRIBUTES.  
\*CREATE OBJECT KAL\_OBJ.  
\*  
\*START-OF-SELECTION.  
\*WRITE :/ KAL\_OBJ->i\_a,  
\*KAL\_OBJ->i\_c,  
\*ZCL\_KALESHA\_OOPS\_ATTRIBUTES=>s\_b.  
  
\*CLASS KAL DEFINITION.  
\*  PUBLIC SECTION.  
\*DATA I\_A TYPE I VALUE 10.  
\* CLASS-DATA   S\_B(10) TYPE C VALUE 'KALESHA'.  
\*ENDCLASS.  
\*  
\*DATA KAL\_OBJ TYPE REF TO KAL.  
\*CREATE OBJECT KAL\_OBJ.  
\*  
\*START-OF-SELECTION.  
\*WRITE : / KAL\_OBJ->i\_a,  
\*KAL=>s\_b.  
  
\*\*\*\*\*\*METHODS:  
  
\*DATA KAL\_OBJ TYPE REF TO ZCL\_KALESHA\_OOPS\_ATTRIBUTES.  
\*CREATE OBJECT KAL\_OBJ.  
\*  
\*START-OF-SELECTION.  
\*DATA(A) = KAL\_OBJ->i\_a.  
\*DATA(B) = KAL\_OBJ->i\_c.  
\*DATA(C) = A + B.  
\*WRITE : /  
\*KAL\_OBJ->i\_a,  
\*KAL\_OBJ->i\_c,  
\*C.  
\*  
\*KAL\_OBJ->i\_method( ).  
\*  
\*\*\*\*CALL METHOD KAL\_OBJ->i\_method.  
\*  
\*ZCL\_KALESHA\_OOPS\_ATTRIBUTES=>s\_method( ).  
  
  
CLASS kal DEFINITION.  
  PUBLIC SECTION.  
    DATA : i\_a TYPE i VALUE 10,  
           i\_b TYPE i VALUE 100.  
    CLASS-DATA s\_c TYPE i VALUE 20.  
  
    METHODS KALe.  
    METHODS vali.  
ENDCLASS.  
  
CLASS kal IMPLEMENTATION.  
  METHOD kale.  
    WRITE 'KALESHA'.  
  ENDMETHOD.  
  METHOD vali.  
    WRITE 'VALI'.  
  ENDMETHOD.  
ENDCLASS.  
  
DATA kal\_obj TYPE REF TO kal.  
CREATE OBJECT kal\_obj.  
  
START-OF-SELECTION.  
  DATA(a) = kal\_obj->i\_a.  
  DATA(b) = kal\_obj->i\_B.  
  DATA(c) = a + b.  
  WRITE : /  
  kal\_obj->i\_a,  
  kal\_obj->i\_B,  
  c.  
  
  kal\_obj->kale( ).  
  kal\_obj->vali( ).

**ITAB :**

REPORT zcl\_kalesha\_oops\_kna1.  
  
DATA : itab TYPE zkalesha\_kna\_tt.  
DATA lv\_kunnr TYPE kunnr.  
SELECT-OPTIONS s\_kunnr FOR lv\_kunnr.  
DATA kal\_obj TYPE REF TO zcl\_kalesha\_oops\_kna1\_itab.  
CREATE OBJECT kal\_obj.  
  
kal\_obj->itab\_kna1(  
  EXPORTING  
    s\_kunnr1         =      s\_kunnr-low            " KUNNR1  
    s\_kunnr2         =     s\_kunnr-high             " KUNNR2  
  IMPORTING  
    i\_tab            =     itab             " TT  
  EXCEPTIONS  
    customernotfound = 1                " RAISE THE TICKET  
    OTHERS           = 2  
).  
IF sy-subrc <> 0.  
\* MESSAGE ID SY-MSGID TYPE SY-MSGTY NUMBER SY-MSGNO  
\*   WITH SY-MSGV1 SY-MSGV2 SY-MSGV3 SY-MSGV4.  
ENDIF.  
  
cl\_demo\_output=>display( itab ).

**IMPORTING, EXPORTING, CHANGING, RETURNING :**

REPORT zkalesha\_oops\_impexpcharet.  
  
\*\*\*CLASS KAL DEFINITION.  
\*\*\*  PUBLIC SECTION.  
\*\*\* DATA : A  TYPE I VALUE 10,  
\*\*\* B TYPE I VALUE  20.  
\*\*\* CLASS-DATA S\_C TYPE I VALUE 30.  
\*\*\*  
\*\*\* ENDCLASS.  
\*\*\* DATA KAL\_OBJ TYPE REF TO KAL.  
\*\*\* CREATE OBJECT KAL\_OBJ.  
\*\*\* WRITE : KAL\_OBJ->a,  
\*\*\* KAL\_OBJ->b,  
\*\*\* KAL=>S\_C.  
  
  
  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*DIFFERENT WAYS TO CALL THE METHODS.  
  
\*\*PARAMETERS p\_a TYPE i.  
\*\*  
\*\*CLASS kal DEFINITION.  
\*\*  PUBLIC SECTION.  
\*\*DATA : a TYPE i,  
\*\*       b TYPE i.  
\*\*CLASS-DATA s\_c TYPE i VALUE 30.  
\*\*METHODS kal\_meth IMPORTING input TYPE i.  
\*\*ENDCLASS.  
\*\*  
\*\*CLASS KAL IMPLEMENTATION.  
\*\*  METHOD KAL\_METH.  
\*\*    B = INPUT \* S\_C.  
\*\*    WRITE B.  
\*\*    ENDMETHOD.  
\*\*    ENDCLASS.  
\*\*    START-OF-SELECTION.  
\*\*    DATA KAL\_OBJ1 TYPE REF TO KAL.  
\*\*     CREATE OBJECT KAL\_OBJ1.  
\*\*    CALL METHOD KAL\_OBJ1->kal\_meth EXPORTING INPUT = P\_A.  
\*\*     CALL METHOD KAL\_OBJ1->kal\_meth( INPUT = P\_A ).  
\*\*      CALL METHOD KAL\_OBJ1->kal\_meth(P\_A).  
\*\*      KAL\_OBJ1->kal\_meth( P\_A ).  
  
  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*WITH IN A STATIC METHOD CANNOT CONTAINS INSTANCE ATTRIBUTES.  
\*\*\*PARAMETERS P\_A TYPE I.  
\*\*\*CLASS KAL DEFINITION.  
\*\*\*  PUBLIC SECTION.  
\*\*\*  DATA : a TYPE i,  
\*\*\*       b TYPE i VALUE 40.  
\*\*\*CLASS-DATA s\_c TYPE i VALUE 30.  
\*\*\*CLASS-METHODS STAMETHOD IMPORTING INPUT TYPE I.  
\*\*\*METHODS INSMETHOD IMPORTING INPUT1 TYPE I.  
\*\*\*  ENDCLASS.  
\*\*\*  
\*\*\*  CLASS KAL IMPLEMENTATION.  
\*\*\*   METHOD STAMETHOD.  
\*\*\*     WRITE : INPUT,'STATIC:',S\_C.  
\*\*\*    """"""""""""""""""""""""""""" WRITE : A,B.  INSTANCE ATTRIBUTES ARE NOT ALLOWED  
\*\*\*     ENDMETHOD.  
\*\*\*  
\*\*\*     METHOD INSMETHOD.  
\*\*\*       A = INPUT1 + B .  
\*\*\*       WRITE A.  
\*\*\*       WRITE : 'STATIC:',S\_C.  
\*\*\*       ENDMETHOD.  
\*\*\*    ENDCLASS.  
\*\*\*    START-OF-SELECTION.  
\*\*\*    DATA KAL\_OBJ TYPE REF TO KAL.  
\*\*\*    CREATE OBJECT KAL\_OBJ.  
\*\*\*    KAL=>stamethod( KAL=>S\_C ).  
\*\*\*    KAL\_OBJ->insmethod( P\_A ).  
  
  
  
\*# 1 Use OF STATIC Attributes  
  
\*CLASS c1 DEFINITION .  
\*PUBLIC SECTION.  
\*CLASS-DATA : NUM TYPE I .   " STAT ATTR  
\*METHODS : M1. " INST  
\*  
\*ENDCLASS.  
\*  
\*CLASS c1 IMPLEMENTATION.  
\*  METHOD m1 .  
\*    num = num + 1.  
\*    WRITE:/5 num .  
\*  ENDMETHOD.  
\*ENDCLASS.  
\*  
\*START-OF-SELECTION.  
\*c1=>num = 3.  
\*WRITE:/5 c1=>num .  
\*  
\*DATA : OREF1 TYPE REF TO C1 ,  
\*       OREF2 TYPE REF TO C1 .  
\*CREATE OBJECT : OREF1 ,  
\*                OREF2 .  
\*CALL METHOD: OREF1->M1.  
\*CALL METHOD OREF2->M1.  
\*  
\*# 2 Different ways OF calling the METHOD WITH one IMPORT PARAMETER  
\*\*  
\*CLASS C1 DEFINITION.  
\*PUBLIC SECTION.  
\*DATA : NUM TYPE I VALUE 5. " INST ATTR  
\*METHODS : METH1  IMPORTING i\_INPUT1 TYPE I  .  " INST METHOD  
\*ENDCLASS.  
\*  
\*CLASS C1 IMPLEMENTATION.  
\*  METHOD : METH1.  
\*    num = NUM \* i\_INPUT1  .  
\*    WRITE:/5 NUM .  
\*    num = 5.  
\*  ENDMETHOD.  
\*ENDCLASS.  
\*  
\*\*\*\*\* prog  
\*START-OF-SELECTION.  
\*DATA : I\_OREF1 TYPE REF TO C1.  
\*CREATE OBJECT : I\_OREF1.  
\*  
\*CALL METHOD I\_OREF1->METH1 EXPORTING i\_INPUT1 = 4.   " # 1  
\*CALL METHOD I\_OREF1->METH1( i\_INPUT1 = 5 ).      " # 2  
\*CALL METHOD I\_OREF1->METH1( 6  ). " OBSLT        " # 3  
\*I\_OREF1->METH1( 6 ).       " final  # 4  
  
  
  
\*\*# 3  IMPORT PARAMETERS passed BY REF. can’t be changed inside the METHOD  
\*  
PARAMETERS p\_num TYPE i.  
\*DATA : num TYPE I .  
  
CLASS c1 DEFINITION.  
  
  PUBLIC SECTION.  
    METHODS : meth1 IMPORTING input1        TYPE i  " BY REF  
                              VALUE(input2) TYPE i. " BY VALUE  
ENDCLASS.  
  
CLASS c1 IMPLEMENTATION.  
  METHOD : meth1.  
    Input2 = 4.  
    WRITE:/5 input2.  
  ENDMETHOD.  
ENDCLASS.  
  
START-OF-SELECTION.  
  DATA : oref1 TYPE REF TO c1.  
  CREATE OBJECT : oref1.  
\*num = 3.  
  CALL METHOD oref1->meth1  
    EXPORTING  
      input1 = 5  
      input2 = p\_num.  
  
\*\*\*\*  
\*\*\*\*# 4 Use OF EXPORT AND CHANGING PARAMETERS OF a METHOD  
\*\*\*  
\*\*\*PARAMETERS p\_a(10) TYPE c.  " GRADE - A01 OR A02  --  ANY  
\*\*\*  
\*\*\*DATA : lv\_tax    TYPE P DECIMALS 2,  
\*\*\*       lv\_salary TYPE p DECIMALS 2.  
\*\*\*  
\*\*\*CLASS ctax DEFINITION.  
\*\*\*  PUBLIC SECTION.  
\*\*\*    METHODS : tax\_calc  IMPORTING grade  TYPE c  
\*\*\*                        EXPORTING itax   TYPE p  
\*\*\*                        CHANGING  salary TYPE p .  
\*\*\*ENDCLASS.  
\*\*\*  
\*\*\*CLASS ctax IMPLEMENTATION.  
\*\*\*  METHOD : tax\_calc.  
\*\*\*    CASE grade.  
\*\*\*      WHEN   'A01'.  
\*\*\*        itax = salary \* '0.2'.  
\*\*\*      WHEN 'A02'.  
\*\*\*        itax = salary \* '0.1'.  
\*\*\*      WHEN OTHERS.  
\*\*\*        itax = salary \* '0.15'.  
\*\*\*    ENDCASE.  
\*\*\*  
\*\*\*    salary = salary - itax.  "   24 = 30 - 6  
\*\*\*  
\*\*\*  ENDMETHOD.  
\*\*\*ENDCLASS.  
\*\*\*  
\*\*\*START-OF-SELECTION.  
\*\*\*  DATA : oref1 TYPE REF TO ctax.  
\*\*\*  CREATE OBJECT : oref1.  
\*\*\*  
\*\*\*  lv\_salary = 30000.  
\*\*\*  lv\_tax    = 0    .  
\*\*\*  
\*\*\*  WRITE:/5 'Before method call, salary and tax are' ,  
\*\*\*          lv\_salary ,  
\*\*\*          lv\_tax .  
\*\*\*  
\*\*\*  CALL METHOD oref1->tax\_calc  
\*\*\*    EXPORTING  
\*\*\*      grade  = p\_a  "'A01'  
\*\*\*    IMPORTING  
\*\*\*      itax   = lv\_tax  
\*\*\*    CHANGING  
\*\*\*      salary = lv\_salary.  
\*\*\*  
\*\*\*  WRITE:/5 'After method call, salary and tax are' ,  
\*\*\*                                      lv\_salary ,  
\*\*\*                                      lv\_tax .  
\*\*\*  
  
  
\*\*\*\*\*# 5:  Use OF RETURNING PARAMETERS IN METHOD ( FUNCTIONAL mETHOD)  
\*\*  
\*DATA : w\_num TYPE I.  
\*  
\*CLASS c1 DEFINITION .  
\*  PUBLIC SECTION.  
\*  METHODS : m1 IMPORTING input1        TYPE I  
\*                         input2        TYPE I  
\*               returning VALUE(result) TYPE I .  
\*ENDCLASS.  
\*  
\*CLASS c1 IMPLEMENTATION.  
\*  METHOD  : m1.  
\*    result = input1 \* 2 + input2.  
\*  ENDMETHOD.  
\*ENDCLASS.  
\*  
\*start-OF-selection.  
\*DATA : obj1 TYPE REF TO c1 .  
\*CREATE object obj1.  
\*\* Syntax 1  
\*CALL METHOD obj1->m1 EXPORTING input1 = 5  
\*                               input2 = 4  
\*                     RECEIVING result = w\_num.  
\*WRITE:/5 w\_num .  
\*\* Syntax 2  
\*w\_num = obj1->m1( input1 = 10 input2 = 20 ).  
\*WRITE:/5 w\_num .  
\*  
\*\* Syntax 3  
\*MOVE obj1->m1( input1 = 2 input2 = 3 ) TO w\_num .  
\*WRITE:/5 w\_num .  
\*\*\*\*  
\*\*\*\*  
  
\*\*\*\*# 6 STATIC METHODS can only use STATIC attributes, instance METHODS use both  
\*\*\*  
\*\*\*CLASS C1 DEFINITION.  
\*\*\*PUBLIC SECTION.  
\*\*\*CLASS-DATA : STNUM TYPE I VALUE 5.  " STAT ATTR  
\*\*\*DATA : INSTNUM TYPE I VALUE 6 .  " INST ATTR  
\*\*\*  
\*\*\*CLASS-METHODS : STMETH .   " STAT METH  
\*\*\*METHODS : INSTMETH .   " INST METY  
\*\*\*  
\*\*\*ENDCLASS.  
\*\*\*  
\*\*\*CLASS C1 IMPLEMENTATION.  
\*\*\*  METHOD : STMETH .  
\*\*\*    WRITE:/5 STNUM .  
\*\*\*\*\*\*    WRITE:/5 INSTNUM .  
\*\*\*  ENDMETHOD.  
\*\*\*  
\*\*\*  METHOD INSTMETH.  
\*\*\*    WRITE:/5 STNUM .  
\*\*\*    WRITE:/5 INSTNUM .  
\*\*\*  ENDMETHOD.  
\*\*\*ENDCLASS.  
\*\*\*  
\*\*\*START-OF-SELECTION.  
\*\*\*DATA : OREF1 TYPE REF TO C1.  
\*\*\*CALL METHOD c1=>stmeth .  
\*\*\*  
\*\*\*CREATE OBJECT OREF1.  
\*\*\*CALL METHOD oref1->instmeth.  
  
  
  
\*# 7: METHOD RAISING EXCEPTIONS  
\*\*  
\*CLASS c1 DEFINITION .  
\*PUBLIC SECTION.  
\*METHODS : m1  IMPORTING num1 TYPE I  
\*              EXPORTING num2 TYPE I  
\*  
\*  
\*              EXCEPTIONS e1.  
\*ENDCLASS.  
\*  
\*CLASS c1 IMPLEMENTATION.  
\*  METHOD  : m1.  
\*    IF num1 LT 5 .  
\*      MESSAGE i398(00) WITH 'Should be >=5' RAISING e1.  
\*    ELSE .  
\*      num2 = num1 \* 5 .  
\*    ENDIF.  
\*  ENDMETHOD.  
\*ENDCLASS.  
\*  
\*PARAMETERS : p\_no TYPE I .  
\*  
\*start-OF-selection.  
\*DATA : obj1 TYPE REF TO c1 .  
\*CREATE object obj1.  
\*  
\*CALL METHOD obj1->m1 EXPORTING num1 = p\_no  " 9  
\*                    IMPORTING num2 = p\_no  
\*                    EXCEPTIONS e1 = 4.  
\*IF sy-subrc = 4.  
\*  MESSAGE ID SY-MSGID TYPE SY-MSGTY NUMBER SY-MSGNO  
\*  WITH SY-MSGV1 SY-MSGV2 SY-MSGV3 SY-MSGV4.  
\*ELSE.  
\*  WRITE:/5 p\_no .  
\*ENDIF.  
\*  
\*  
  
\*  
  
  
\*\*\*\*\*\*\*8  CONSTRUCTOR  
\*  
\*CLASS C1 DEFINITION.  
\*  PUBLIC SECTION.  
\*  PROTECTED SECTION.  
\*\*\*\*   METHODS MAT\_METH.  
\*    METHODS : CONSTRUCTOR.  
\*ENDCLASS.  
\*  
\*CLASS C1 IMPLEMENTATION.  
\*  METHOD constructor.  
\*    WRITE:/5 'I am CALLING constructor'.  
\*    SKIP 2.  
\*  ENDMETHOD.  
\*ENDCLASS.  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* main program \*\*\*\*\*\*\*\*\*\*\*\*\*\*  
\*START-OF-SELECTION.  
\*  
\*DATA: I\_obj1 TYPE REF TO c1.  
\*CREATE OBJECT I\_obj1.  
  
\*\*\*\*call METHOD mat\_meth().  
\*\*\*\*\*\*\*\*CALL METHOD CONSTRUCTOR.  
  
  
\*  
\*\*\*8.2  Instance Constructors can have IMPORT PARAMETERS  
\*CLASS c1 DEFINITION.  
\*PUBLIC SECTION.  
\*\*PRIVATE SECTION.  
\*METHODS : CONSTRUCTOR IMPORTING I\_today TYPE d.  
\*  
\*ENDCLASS.  
\*\*  
\*CLASS C1 IMPLEMENTATION.  
\*  METHOD constructor.  
\*    WRITE:/5 'Today is : ' , I\_today dd/mm/yyyy.  
\*  ENDMETHOD.  
\*ENDCLASS.  
\*  
\*START-OF-SELECTION.  
\*DATA: obj1 TYPE REF TO c1.  
\*CREATE OBJECT: obj1 EXPORTING I\_today = sy-datum.  
  
  
  
\*\*\*\*\*\*\*\*\* \*8.2Instance Constructors can raise exceptions  
\*CLASS c1 DEFINITION.  
\*  PUBLIC SECTION.  
\*  METHODS : CONSTRUCTOR IMPORTING num TYPE I  
\*                        EXCEPTIONS e1 .  
\*ENDCLASS.  
\*  
\*CLASS C1 IMPLEMENTATION.  
\*  METHOD constructor.  
\*    IF num LT 7.  
\*      RAISE  e1.  
\*    ENDIF.  
\*  ENDMETHOD.  
\*ENDCLASS.  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* main program \*\*\*\*\*\*\*\*\*\*\*\*\*\*  
\*  
\*PARAMETERS p\_no type i .  
\*  
\*START-OF-SELECTION.  
\*DATA: obj1 TYPE REF TO c1.  
\*CREATE OBJECT: obj1 EXPORTING num = p\_no  
\*                    EXCEPTIONS e1 = 4.  
\*IF sy-subrc = 4.  
\*  WRITE:/5 'Exceptions raised'.  
\*ENDIF.  
\*  
  
  
  
\* 8.3. STATIC constructor can be triggered AT the beginning OF a processing BLOCK  
\*(FORM /event/block/procedure)  
\*\*  
\*CLASS c1 DEFINITION .  
\*PUBLIC SECTION.  
\*CLASS-DATA : NUM TYPE I VALUE 5.  
\*METHODs CONSTRUCTOR.  
\*CLASS-METHODS :CLASS\_CONSTRUCTOR .  
\*ENDCLASS.  
\*  
\*CLASS c1 IMPLEMENTATION.  
\*  
\*  METHOD CONSTRUCTOR.  
\*  WRITE:/5 'inst constructor'.  
\*  ENDMETHOD.  
\*  
\*  METHOD CLASS\_CONSTRUCTOR.  
\*    WRITE:/5 'I am class constructor'.  
\*  ENDMETHOD.  
\*ENDCLASS.  
\*  
\*START-OF-SELECTION.  
\*WRITE:/5 'Hello'.  
\*\*WRITE:/5 c1=>num.\*  
\*\*data i\_obj type REF TO c1.  
\*\*create OBJECT i\_obj.  
  
\*  
  
\*\*\*\*\*\*\*10)  Use OF ME keword (self REF var) IN METHODS  
\*  
\*CLASS testclass DEFINITION.  
\*PUBLIC SECTION.  
\*DATA : i\_num TYPE I VALUE 5.  
\*METHODS : testmethod .  
\*ENDCLASS.  
\*  
\*CLASS testclass IMPLEMENTATION.  
\*  METHOD :testmethod.  
\*    DATA : i\_num TYPE I VALUE 2.  
\*    WRITE:/5             i\_num ,   " access variable of the class  
\*          /5         me->i\_num .   " access variable of the method  
\*  ENDMETHOD.  
\*  
\*ENDCLASS.  
\*  
\*start-OF-selection.  
\*DATA : i\_num TYPE I.  
\*DATA : my\_obj TYPE REF TO testclass.  
\*CREATE object : my\_obj.  
\*CALL METHOD my\_obj->testmethod.  
  
\*  
\*\*11) METHOD can CALL itself  
\*\*  
\*CLASS c1 DEFINITION .  
\*PUBLIC SECTION.  
\*CLASS-DATA : statnum TYPE I .  
\*METHODS : m1 .  
\*ENDCLASS.  
\*  
\*CLASS c1 IMPLEMENTATION.  
\*  METHOD  : m1.  
\*    statnum = statnum + 10.  
\*    IF statnum GT 100.  
\*      EXIT.  
\*    ENDIF.  
\*    WRITE:/5 statnum .  
\*   CALL METHOD m1( ).  
\*  ENDMETHOD.  
\*ENDCLASS.  
\*  
\*start-OF-selection.  
\*DATA : obj1 TYPE REF TO c1 .  
\*CREATE object obj1.  
\*\*CALL METHOD  
“obj1->m1( ) .

**CONSTRUCTOR :**

REPORT zkalesha\_oops\_contructors.  
  
\*\*\*\*\*\*INSTANCE CONSTRUCTOR.  
  
\*\*CLASS KAL DEFINITION.  
\*\*  PUBLIC SECTION.  
\*\*  DATA : I\_A TYPE I VALUE 30,  
\*\*        I\_B TYPE I VALUE 50,  
\*\*        SUM TYPE I.  
\*\*  CLASS-DATA S\_C TYPE I VALUE 40.  
\*\*  METHODS : KALESHA IMPORTING INPUT TYPE I,  
\*\*            CONSTRUCTOR IMPORTING INPUT2 TYPE I.  
\*\*  ENDCLASS.  
\*\*  
\*\*  CLASS KAL IMPLEMENTATION.  
\*\*    METHOD KALESHA.  
\*\*      SUM = INPUT + I\_B.  
\*\*      WRITE : / 'SUM :',SUM.  
\*\*      ENDMETHOD.  
\*\*      METHOD CONSTRUCTOR.  
\*\*       " WRITE : / 'CONSTRUCTOR IS ON',I\_A.  
\*\*         WRITE : / 'CONSTRUCTOR IS ON',INPUT2.  
\*\*        ENDMETHOD.  
\*\*        ENDCLASS.  
\*\*  
\*\*        START-OF-SELECTION.  
\*\*        DATA KAL\_OBJ TYPE REF TO KAL.  
\*\*        CREATE OBJECT KAL\_OBJ EXPORTING INPUT2 = KAL\_OBJ->I\_A.  
\*\*        KAL\_OBJ->KALESHA( 30 ).  
  
  
  
\*\*\*\*\*\*\*\*\*\*\*STATIC CONTRUCTOR.  
\*\*CLASS KAL DEFINITION.  
\*\*  PUBLIC SECTION.  
\*\*  DATA : I\_A TYPE I VALUE 30,  
\*\*        I\_B TYPE I VALUE 50,  
\*\*        SUB TYPE I,  
\*\*        SUM TYPE I.  
\*\*  CLASS-DATA : S\_C TYPE I VALUE 40,  
\*\*               S\_D TYPE I VALUE 50.  
\*\*   METHODS : KALESHA1 IMPORTING INPUT TYPE I.  
\*\*  CLASS-METHODS : KALESHA,  
\*\*                CLASS\_CONSTRUCTOR.  
\*\*  ENDCLASS.  
\*\*  
\*\*  CLASS KAL IMPLEMENTATION.  
\*\*     METHOD KALESHA1.  
\*\*      SUB = INPUT - I\_B.  
\*\*      WRITE : / 'SUB :',SUB.  
\*\*      ENDMETHOD.  
\*\*    METHOD KALESHA.                   "STATIC METHOD  
\*\*  
\*\*      WRITE : / 'STATIC METHOD :',S\_D,S\_C.  
\*\*      ENDMETHOD.  
\*\*      METHOD CLASS\_CONSTRUCTOR.  
\*\*       " WRITE : / 'CONSTRUCTOR IS ON',I\_A.  
\*\*         WRITE : / 'STATIC CONSTRUCTOR IS ON',S\_C.  
\*\*        ENDMETHOD.  
\*\*        ENDCLASS.  
\*\*  
\*\*        START-OF-SELECTION.  
\*\*        DATA KAL\_OBJ TYPE REF TO KAL.  
\*\*        KAL=>KALESHA(  ).  
\*\*        CREATE OBJECT KAL\_OBJ .  
\*\*        KAL\_OBJ->KALESHA1( 100 ).  
\*\*  
  
  
\*\*\*\*\*\*\*\*\*\*\*\* "ME->" = SELF REFERENCE,BOTH STATIC AND INSTANCE CONSTRUCTOR IN A ONE ,  
PARAMETERS input TYPE i.  
CLASS kal DEFINITION.  
  PUBLIC SECTION.  
    DATA : i\_a TYPE i VALUE 50,  
           i\_b TYPE i VALUE 60.  
    CLASS-DATA s\_c TYPE i VALUE 70.  
    METHODS : kalesha IMPORTING input TYPE i.  
    constructor importing input1 type i exceptions e.  
    CLASS-METHODS class\_constructor.  
ENDCLASS.  
CLASS kal IMPLEMENTATION.  
  METHOD kalesha.  
    DATA i\_a TYPE i VALUE 78.  
    WRITE :/ input ,'METHOD :',i\_a,            " IT REFERS TO METHOD  
     'CLASS :', me->i\_a.                       " IT REFERS TO CLASS  
  ENDMETHOD.  
  
  METHOD constructor.  
    IF input1 > 20.  
      RAISE e.  
    ELSE.  
      WRITE :/ 'CONSTRUCTOR :',input1.  
    ENDIF.  
  ENDMETHOD.  
  METHOD class\_constructor.  
    WRITE :/ 'STATIC CONSTRUCTOR :',s\_c.  
  ENDMETHOD.  
ENDCLASS.  
  
START-OF-SELECTION.  
  DATA kal\_obj TYPE REF TO kal.  
  CREATE OBJECT kal\_obj EXPORTING input1 = input.  
  kal\_obj->kalesha( 40 ).