**Sales Order Report Using OOPS ALV.**

\*&---------------------------------------------------------------------\*  
\*& Report ZSALES\_ORDER  
\*&---------------------------------------------------------------------\*  
\*&  
\*&---------------------------------------------------------------------\*  
REPORT zsales\_order\_oop NO STANDARD PAGE HEADING.  
TYPES: BEGIN OF ty\_first,  
         kunnr TYPE kna1-kunnr,  
         name1 TYPE kna1-kunnr,  
         vbeln TYPE vbak-vbeln,  
         erdat TYPE vbak-erdat,  
         netwr TYPE vbak-netwr,  
       END OF ty\_first.  
  
TYPES: BEGIN OF ty\_second,  
         vbeln TYPE vbak-vbeln,  
         posnr TYPE vbap-posnr,  
         matnr TYPE vbap-matnr,  
         arktx TYPE vbap-arktx,  
         zmeng TYPE vbap-zmeng,  
         zieme TYPE vbap-zieme,  
         netwr TYPE vbak-netwr,  
       END OF ty\_second.  
  
DATA: t\_first  TYPE TABLE OF ty\_first,  
      t\_second TYPE TABLE OF ty\_second WITH HEADER LINE,  
      t\_third  TYPE TABLE OF ty\_second WITH HEADER LINE,  
      w\_first  TYPE ty\_first,  
      w\_second TYPE ty\_second,  
      w\_third  TYPE ty\_second.  
  
DATA: g\_vbeln TYPE vbak-vbeln,  
      g\_erdat TYPE vbak-erdat,  
      g\_kunnr TYPE kna1-kunnr.  
  
DATA: ok\_code LIKE sy-ucomm,  
      line    TYPE i.  
  
DATA: r\_alv         TYPE REF TO cl\_salv\_table,  
      gr\_functions  TYPE REF TO cl\_salv\_functions\_list,  
      gr\_columns    TYPE REF TO cl\_salv\_columns\_table,  
      gr\_column     TYPE REF TO cl\_salv\_column\_table,  
      gr\_display    TYPE REF TO cl\_salv\_display\_settings,  
      gr\_content TYPE REF TO cl\_salv\_form\_element,  
      gr\_selections TYPE REF TO cl\_salv\_selections.

DATA: lv\_title      TYPE lvc\_title,  
      lt\_column\_ref TYPE salv\_t\_column\_ref,  
      ls\_column\_ref TYPE salv\_s\_column\_ref.  
  
DATA : ty\_toolbar TYPE stb\_button.  
  
CLASS : lcl\_handle\_events DEFINITION DEFERRED.  
DATA: gr\_events TYPE REF TO lcl\_handle\_events,  
      lr\_events TYPE REF TO cl\_salv\_events\_table.  
  
CLASS lcl\_handle\_events DEFINITION.  
  PUBLIC SECTION.  
    METHODS : on\_double\_click FOR EVENT double\_click OF cl\_salv\_events\_table  
      IMPORTING row column.  
    METHODS : user\_command FOR EVENT added\_function OF cl\_salv\_events  
      IMPORTING e\_salv\_function.  
ENDCLASS.  
\*----------------------------------------------------------------------\*  
\*       CLASS lcl\_handle\_events IMPLEMENTATION  
\*----------------------------------------------------------------------\*  
CLASS lcl\_handle\_events IMPLEMENTATION.  
  METHOD on\_double\_click.  
    PERFORM get\_ord\_info USING row column.  
  ENDMETHOD.  
  
  METHOD user\_command.  
    PERFORM user\_cmd USING e\_salv\_function.  
  ENDMETHOD.  
ENDCLASS.  
  
SELECTION-SCREEN: BEGIN OF BLOCK b1.  
SELECT-OPTIONS: s\_vbeln FOR g\_vbeln DEFAULT '18400' TO '19000',  
                s\_erdat FOR g\_erdat,  
                s\_kunnr FOR g\_kunnr.  
SELECTION-SCREEN: END OF BLOCK b1.  
  
START-OF-SELECTION.  
  PERFORM get\_order\_data.  
  PERFORM get\_order\_details.  
  PERFORM display\_alv.  
\*&---------------------------------------------------------------------\*  
\*& Form GET\_ORD\_INFO  
\*&---------------------------------------------------------------------\*  
\*& text  
\*&---------------------------------------------------------------------\*  
\*      -->P\_ROW  text  
\*      -->P\_COLUMN  text  
\*&---------------------------------------------------------------------\*  
FORM get\_ord\_info USING row TYPE salv\_de\_row  
                        column TYPE salv\_de\_column.  
  IF column EQ 'VBELN'.  
    REFRESH t\_third.  
    CLEAR t\_third.  
    GET CURSOR LINE line.  
    READ TABLE t\_first INTO w\_first INDEX row.  
    IF sy-subrc = 0.  
      LOOP AT t\_second WHERE vbeln = w\_first-vbeln.  
        MOVE t\_second TO t\_third.  
        APPEND t\_third.  
      ENDLOOP.  
    ENDIF.  
    PERFORM display\_alv2.  
  ENDIF.  
ENDFORM.                    "get\_ord\_info  
\*&---------------------------------------------------------------------\*  
\*& Form TOP\_OF\_PAGE  
\*&---------------------------------------------------------------------\*  
\*& text  
\*&---------------------------------------------------------------------\*  
\*      <--P\_GR\_CONTENT  text  
\*&---------------------------------------------------------------------\*  
FORM top\_of\_page CHANGING lr\_content TYPE REF TO cl\_salv\_form\_element.  
  DATA : lr\_grid  TYPE REF TO cl\_salv\_form\_layout\_grid,  
         lr\_text  TYPE REF TO cl\_salv\_form\_text,  
         lr\_label TYPE REF TO cl\_salv\_form\_label,  
         lr\_head  TYPE string.  
  
  MOVE 'Sales Order List' TO lr\_head.  
  CREATE OBJECT lr\_grid.  
  
  lr\_grid->create\_header\_information( row     = 1  
                                      column  = 1  
                                      text    = lr\_head  
                                      tooltip = lr\_head ).  
  lr\_grid->add\_row( ).  
  lr\_label = lr\_grid->create\_label( row = 2  
                                    column = 1  
                                    text = 'No of Records').  
  DATA l\_rec(5) TYPE n.  
  DESCRIBE TABLE t\_first LINES l\_rec.  
  lr\_text = lr\_grid->create\_text( row = 2  
                                  column = 2  
                                  text = l\_rec ).  
  lr\_label->set\_label\_for( lr\_text ).  
ENDFORM.                    "top\_of\_page

\*&---------------------------------------------------------------------\*  
\*& Form DISPLAY\_ALV2  
\*&---------------------------------------------------------------------\*  
\*& text  
\*&---------------------------------------------------------------------\*  
\*& -->  p1        text  
\*& <--  p2        text  
\*&---------------------------------------------------------------------\*  
FORM display\_alv2.  
\*  TRY.  
  CALL METHOD cl\_salv\_table=>factory  
    EXPORTING  
      list\_display = if\_salv\_c\_bool\_sap=>true  
\*     r\_container  =  
\*     container\_name =  
    IMPORTING  
      r\_salv\_table = r\_alv  
    CHANGING  
      t\_table      = t\_third[].  
\*   CATCH cx\_salv\_msg .  
\*  ENDTRY.  
  r\_alv->display( ).  
ENDFORM.  
\*&---------------------------------------------------------------------\*  
\*& Form USER\_CMD  
\*&---------------------------------------------------------------------\*  
\*& text  
\*&---------------------------------------------------------------------\*  
\*      -->P\_E\_SALV\_FUNCTION  text  
\*&---------------------------------------------------------------------\*  
FORM user\_cmd  USING    p\_e\_salv\_function.  
  DATA: lt\_rows TYPE salv\_t\_row,  
        ls\_rows TYPE i.  
  
  CASE p\_e\_salv\_function.  
    WHEN '&DETAIL'.  
      DATA sel\_fld TYPE slis\_selfield.  
      gr\_selections = r\_alv->get\_selections( ).  
      lt\_rows = gr\_selections->get\_selected\_rows( ).  
      READ TABLE lt\_rows INTO ls\_rows INDEX 1.  
      READ TABLE t\_first INTO w\_first INDEX ls\_rows.  
      SET PARAMETER ID 'AUN' FIELD w\_first-vbeln.  
      CALL TRANSACTION 'VA03' AND SKIP FIRST SCREEN.  
  ENDCASE.  
ENDFORM.  
\*&---------------------------------------------------------------------\*  
\*& Form GET\_ORDER\_DATA  
\*&---------------------------------------------------------------------\*  
\*& text  
\*&---------------------------------------------------------------------\*  
\*& -->  p1        text  
\*& <--  p2        text  
\*&---------------------------------------------------------------------\*  
FORM get\_order\_data .  
  SELECT a~kunnr  
          a~name1  
          b~vbeln  
          b~erdat  
          b~netwr  
     FROM kna1 AS a INNER JOIN vbak AS b  
     ON a~kunnr = b~kunnr  
     INTO TABLE t\_first  
     WHERE a~kunnr IN s\_kunnr  
     AND b~vbeln IN s\_vbeln  
     AND b~erdat IN s\_erdat.  
  
ENDFORM.  
\*&---------------------------------------------------------------------\*  
\*& Form GET\_ORDER\_DETAILS  
\*&---------------------------------------------------------------------\*  
\*& text  
\*&---------------------------------------------------------------------\*  
\*& -->  p1        text  
\*& <--  p2        text  
\*&---------------------------------------------------------------------\*  
FORM get\_order\_details .  
  SELECT a~vbeln  
          b~posnr  
          b~matnr  
          b~arktx  
          b~zmeng  
          b~zieme  
          a~netwr  
     FROM vbak AS a INNER JOIN vbap AS  b  
     ON a~vbeln = b~vbeln  
     INTO TABLE t\_second  
     WHERE a~vbeln IN s\_vbeln.  
  
  SORT t\_second BY vbeln.  
ENDFORM.  
\*&---------------------------------------------------------------------\*  
\*& Form DISPLAY\_ALV  
\*&---------------------------------------------------------------------\*  
\*& text  
\*&---------------------------------------------------------------------\*  
\*& -->  p1        text  
\*& <--  p2        text  
\*&---------------------------------------------------------------------\*  
FORM display\_alv .  
\*TRY.  
  CALL METHOD cl\_salv\_table=>factory  
    EXPORTING  
      list\_display = if\_salv\_c\_bool\_sap=>true  
\*     r\_container  =  
\*     container\_name =  
    IMPORTING  
      r\_salv\_table = r\_alv  
    CHANGING  
      t\_table      = t\_first.  
  
\* CATCH cx\_salv\_msg .  
\*ENDTRY.  
  
  gr\_functions = r\_alv->get\_functions( ).  
  CALL METHOD r\_alv->set\_screen\_status(  
    EXPORTING  
      report        = sy-repid   " ABAP Program: Current Master Program  
      pfstatus      = 'SALV\_TABLE\_STANDARD'    " Screens, Current GUI Status  
      set\_functions = r\_alv->c\_functions\_all ).  " ALV: Data Element for Constants  
  
  CLEAR : gr\_display.  
  MOVE 'Sales Order Details' TO lv\_title.  
  gr\_display = r\_alv->get\_display\_settings( ).  
  gr\_display->set\_striped\_pattern( if\_salv\_c\_bool\_sap=>true ).  
  gr\_display->set\_list\_header( lv\_title ).  
  
  PERFORM top\_of\_page CHANGING gr\_content.  
  r\_alv->set\_top\_of\_list( gr\_content ).  
  lr\_events = r\_alv->get\_event( ).

  CREATE OBJECT gr\_events.  
  SET HANDLER gr\_events->on\_double\_click FOR lr\_events.  
  SET HANDLER gr\_events->user\_command FOR lr\_events.

  gr\_selections = r\_alv->get\_selections( ).  
  gr\_selections->set\_selection\_mode( 1 ).  
  gr\_columns = r\_alv->get\_columns( ).

  IF gr\_columns IS NOT INITIAL.  
    REFRESH : lt\_column\_ref.  
    CLEAR   : ls\_column\_ref.  
    lt\_column\_ref = gr\_columns->get( ).  
    gr\_columns->set\_optimize( if\_salv\_c\_bool\_sap=>true ).  
    gr\_columns->set\_key\_fixation( if\_salv\_c\_bool\_sap=>true ).  
    TRY.  
        gr\_columns->set\_color\_column( 'T\_COLOR' ).  
      CATCH cx\_salv\_data\_error .  
    ENDTRY.  
  ENDIF.  
  r\_alv->display( ). "display grid  
ENDFORM.