REPORT: ZJANA\_DISPLAY

A screenshot of a computer

Description automatically generated

*\*&---------------------------------------------------------------------\**  
*\*& Report ZJANA\_DISPLAY*  
*\*&---------------------------------------------------------------------\**  
*\*&*  
*\*&---------------------------------------------------------------------\**  
REPORT ZJANA\_DISPLAY.  
  
  
CLASS lcl\_report DEFINITION.  
  
 PUBLIC SECTION.  
  
  
 TYPES: BEGIN OF ty\_vbak,  
 vbeln TYPE vbak-vbeln,  
 erdat TYPE erdat,  
 auart TYPE auart,  
 kunnr TYPE kunnr,  
 END OF ty\_vbak.  
  
 DATA: t\_vbak TYPE STANDARD TABLE OF ty\_vbak.  
  
  
 DATA: o\_alv TYPE REF TO cl\_salv\_table.  
  
  
 METHODS:  
  
 get\_data,  
  
  
 generate\_output.  
  
 PRIVATE SECTION.  
  
 METHODS:  
 set\_pf\_status  
 CHANGING  
 co\_alv TYPE REF TO cl\_salv\_table.  
*\**  
 METHODS:  
 set\_layout  
 CHANGING  
 co\_alv TYPE REF TO cl\_salv\_table.  
  
 METHODS:  
 set\_top\_of\_page  
 CHANGING  
 co\_alv TYPE REF TO cl\_salv\_table.  
 METHODS:  
 set\_end\_of\_page  
 CHANGING  
 co\_alv TYPE REF TO cl\_salv\_table.  
  
 METHODS:  
 set\_display\_setting  
 CHANGING  
 co\_alv TYPE REF TO cl\_salv\_table.  
  
  
  
  
  
ENDCLASS.  
  
START-OF-SELECTION.  
 DATA: lo\_report TYPE REF TO lcl\_report.  
  
 CREATE OBJECT lo\_report.  
  
 lo\_report->get\_data( ).  
  
 lo\_report->generate\_output( ).  
  
CLASS lcl\_report IMPLEMENTATION.  
*\**  
 METHOD get\_data.  
  
 SELECT vbeln erdat auart kunnr  
 INTO TABLE t\_vbak  
 FROM vbak  
 UP TO 20 ROWS.  
  
 ENDMETHOD.  
  
 METHOD generate\_output.  
  
 DATA: lx\_msg TYPE REF TO cx\_salv\_msg.  
 TRY.  
 cl\_salv\_table=>factory(  
 IMPORTING  
 r\_salv\_table = o\_alv  
 CHANGING  
 t\_table = t\_vbak ).  
 CATCH cx\_salv\_msg INTO lx\_msg.  
 ENDTRY.  
  
 DATA: lc\_sort TYPE REF TO cl\_salv\_sorts.  
try.  
 lc\_sort = O\_alv->get\_sorts( ).  
  
 lc\_sort->add\_sort( columnname = 'VBELN'  
 subtotal = abap\_true  
 sequence = if\_salv\_c\_sort=>sort\_up ).  
  
 CATCH:cx\_salv\_not\_found cx\_salv\_data\_error cx\_salv\_existing.  
 endtry.  
  
  
 CALL METHOD set\_pf\_status  
 CHANGING  
 co\_alv = o\_alv.  
  
 CALL METHOD set\_layout  
 CHANGING  
 co\_alv = o\_alv.  
  
 CALL METHOD me->set\_top\_of\_page  
 CHANGING  
 co\_alv = o\_alv.  
  
 CALL METHOD me->set\_end\_of\_page  
 CHANGING  
 co\_alv = o\_alv.  
  
 CALL METHOD set\_display\_setting  
 CHANGING  
 co\_alv = o\_alv.  
  
  
 DATA: lo\_functions TYPE REF TO cl\_salv\_functions\_list.  
  
 lo\_functions = o\_alv->get\_functions( ).  
 lo\_functions->set\_default( abap\_true ).  
 o\_alv->display( ).  
  
  
  
  
 ENDMETHOD.  
 METHOD set\_pf\_status.  
 co\_alv->set\_screen\_status(  
 report = 'ZJANA\_DISPLAY'  
 pfstatus = 'STANDARD'  
  
 ).  
 ENDMETHOD.  
 METHOD set\_layout.  
  
 DATA: lo\_layout TYPE REF TO cl\_salv\_layout,  
 lf\_variant TYPE slis\_vari,  
 ls\_key TYPE salv\_s\_layout\_key.  
  
 lo\_layout = co\_alv->get\_layout( ).  
  
 ls\_key-report = sy-repid.  
 lo\_layout->set\_key( ls\_key ).  
  
 lo\_layout->set\_save\_restriction( if\_salv\_c\_layout=>restrict\_none ).  
  
 lf\_variant = 'DEFAULT'.  
 lo\_layout->set\_initial\_layout( lf\_variant ).  
  
  
 ENDMETHOD.  
 METHOD set\_top\_of\_page.  
*\**  
 DATA: lo\_header TYPE REF TO cl\_salv\_form\_layout\_grid,  
 lo\_h\_label TYPE REF TO cl\_salv\_form\_label,  
 lo\_h\_flow TYPE REF TO cl\_salv\_form\_layout\_flow.  
*\**  
*\* header object*  
 CREATE OBJECT lo\_header.  
*\**  
*\* To create a Lable or Flow we have to specify the target*  
*\* row and column number where we need to set up the output*  
*\* text.*  
*\**  
*\* information in Bold*  
 lo\_h\_label = lo\_header->create\_label( row = 1 column = 1 ).  
 lo\_h\_label->set\_text( 'Header in Bold' ).  
*\**  
*\* information in tabular format*  
 lo\_h\_flow = lo\_header->create\_flow( row = 2 column = 1 ).  
 lo\_h\_flow->create\_text( text = 'This is text of flow' ).  
*\**  
 lo\_h\_flow = lo\_header->create\_flow( row = 3 column = 1 ).  
 lo\_h\_flow->create\_text( text = 'Number of Records in the output' ).  
*\**  
 lo\_h\_flow = lo\_header->create\_flow( row = 3 column = 2 ).  
 lo\_h\_flow->create\_text( text = 20 ).  
*\**  
*\* set the top of list using the header for Online.*  
 co\_alv->set\_top\_of\_list( lo\_header ).  
*\**  
*\* set the top of list using the header for Print.*  
 co\_alv->set\_top\_of\_list\_print( lo\_header ).  
*\**  
 ENDMETHOD. *"set\_top\_of\_page*  
*\**  
 METHOD set\_end\_of\_page.  
*\**  
 DATA: lo\_footer TYPE REF TO cl\_salv\_form\_layout\_grid,  
 lo\_f\_label TYPE REF TO cl\_salv\_form\_label,  
 lo\_f\_flow TYPE REF TO cl\_salv\_form\_layout\_flow.  
*\**  
*\* footer object*  
 CREATE OBJECT lo\_footer.  
*\**  
*\* information in bold*  
 lo\_f\_label = lo\_footer->create\_label( row = 1 column = 1 ).  
 lo\_f\_label->set\_text( 'Footer .. here it goes' ).  
*\**  
*\* tabular information*  
 lo\_f\_flow = lo\_footer->create\_flow( row = 2 column = 1 ).  
 lo\_f\_flow->create\_text( text = 'This is text of flow in footer' ).  
*\**  
 lo\_f\_flow = lo\_footer->create\_flow( row = 3 column = 1 ).  
 lo\_f\_flow->create\_text( text = 'Footer number' ).  
*\**  
 lo\_f\_flow = lo\_footer->create\_flow( row = 3 column = 2 ).  
 lo\_f\_flow->create\_text( text = 1 ).  
*\**  
*\* Online footer*  
 co\_alv->set\_end\_of\_list( lo\_footer ).  
*\**  
*\* Footer in print*  
 co\_alv->set\_end\_of\_list\_print( lo\_footer ).  
*\**  
  
 ENDMETHOD.  
 METHOD set\_display\_setting.  
*\**  
 DATA: lo\_display TYPE REF TO cl\_salv\_display\_settings.  
*\**  
*\* get display object*  
 lo\_display = co\_alv->get\_display\_settings( ).  
*\**  
*\* set ZEBRA pattern*  
 lo\_display->set\_striped\_pattern( 'X' ).  
*\**  
*\* Title to ALV*  
 lo\_display->set\_list\_header( 'ALV Test for Display Settings' ).  
*\**  
  
  
 o\_alv->display( ).  
  
 ENDMETHOD.  
  
  
  
ENDCLASS.

A screenshot of a computer

Description automatically generated