Program: ZJANA\_OO\_ALV\_MAIN

A screenshot of a computer

Description automatically generated

*\*&---------------------------------------------------------------------\**  
*\*& Report ZJANA\_OO\_ALV\_MAIN*  
*\*&---------------------------------------------------------------------\**  
*\*&*  
*\*&---------------------------------------------------------------------\**  
REPORT zjana\_oo\_alv\_main.  
  
CLASS lcl\_report DEFINITION.  
  
 PUBLIC SECTION.  
  
 TYPES:BEGIN OF ty\_vbak,  
 vbeln TYPE vbak-vbeln,  
 erdat TYPE vbak-erdat,  
 auart TYPE vbak-auart,  
 kunnr TYPE vbak-kunnr,  
  
 END OF ty\_vbak.  
  
 DATA: t\_vbak TYPE STANDARD TABLE OF ty\_vbak.  
  
  
  
  
 *" alv reference*  
 DATA o\_alv TYPE REF TO cl\_salv\_table.  
  
  
  
 METHODS:  
 get\_data,  
 generate\_output.  
  
 PROTECTED SECTION.  
  
 PRIVATE SECTION.  
 METHODS:  
  
 set\_pf\_status  
  
 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\_layout  
 CHANGING  
 co\_alv TYPE REF TO cl\_salv\_table.  
 METHODS:  
 set\_display\_setting  
 CHANGING co\_alv TYPE REF TO cl\_salv\_table.  
  
 METHODS:  
 set\_columns  
 CHANGING  
 co\_alv TYPE REF TO cl\_salv\_table.  
  
 METHODS:  
 set\_hotspot\_vbeln  
 CHANGING  
 co\_alv TYPE REF TO cl\_salv\_table.  
*\* co\_report TYPE REF TO lcl\_report.*  
  
 METHODS:  
 on\_link\_click  
 FOR EVENT link\_click OF cl\_salv\_events\_table  
 IMPORTING  
 row  
 column.  
  
  
  
ENDCLASS.  
  
CLASS lcl\_report IMPLEMENTATION.  
  
 METHOD set\_pf\_status.  
  
 DATA:lo\_functions TYPE REF TO cl\_salv\_functions\_list.  
 lo\_functions = co\_alv->get\_functions( ).  
 lo\_functions->set\_default( abap\_true ).  
  
 co\_alv->set\_screen\_status(  
 report = 'SALV\_DEMO\_TABLE\_SELECTIONS'  
 pfstatus = 'SALV\_STANDARD'  
set\_functions = co\_alv->c\_functions\_all ).  
 ENDMETHOD.  
  
 METHOD set\_columns.  
  
 DATA:lo\_cols TYPE REF TO cl\_salv\_columns.  
 lo\_cols = o\_alv->get\_columns( ).  
  
  
 lo\_cols->set\_optimize( 'X' ).  
  
 DATA: lo\_column TYPE REF TO cl\_salv\_column.  
  
 TRY.  
  
 lo\_column = lo\_cols->get\_column( 'KUNNR').  
 lo\_column->set\_long\_text( 'SOLD-to party').  
 lo\_column->set\_medium\_text( 'SOLD-to Party').  
 lo\_column->set\_short\_text( 'SOLD-TO').  
 lo\_column->set\_output\_length( 10 ).  
 CATCH cx\_salv\_not\_found.  
  
 ENDTRY.  
 ENDMETHOD.  
  
 METHOD SET\_DISPLAY\_setting.  
 DATA: lo\_display TYPE REF TO cl\_salv\_display\_settings.  
  
 lo\_display = co\_alv->get\_display\_settings( ).  
 lo\_display->set\_striped\_pattern( 'X' ).  
 lo\_display->set\_list\_header(' ALV Test for display settings' ).  
  
 ENDMETHOD.  
  
 METHOD set\_hotspot\_vbeln.  
  
 DATA: lo\_cols\_tab TYPE REF TO cl\_salv\_columns\_table,  
 lo\_col\_tab TYPE REF TO cl\_salv\_column\_table.  
  
 lo\_cols\_tab = co\_alv->get\_columns( ).  
  
 TRY.  
  
 lo\_col\_tab ?= lo\_cols\_tab->get\_column( 'VBELN').  
 CATCH cx\_salv\_not\_found.  
 ENDTRY.  
  
  
  
 TRY.  
 CALL METHOD lo\_col\_tab->set\_cell\_type  
 EXPORTING  
 value = if\_salv\_c\_cell\_type=>hotspot.  
 CATCH cx\_salv\_data\_error.  
 ENDTRY.  
*\* DATA: lo\_events TYPE REF TO cl\_salv\_events\_table.*  
*\* lo\_events = o\_alv->get\_event( ).*  
*\* SET HANDLER co\_report->on\_link\_click for lo\_events.*  
 ENDMETHOD.  
  
 METHOD on\_link\_click.  
DATA: lo\_report TYPE REF TO lcl\_report.  
 DATA: la\_vbak TYPE ty\_vbak.  
  
  
 READ TABLE lo\_report->t\_vbak INTO la\_vbak INDEX row.  
 IF la\_vbak-vbeln IS NOT INITIAL.  
 MESSAGE i398(00) WITH 'you have selected' la\_vbak-vbeln.  
 ENDIF.  
 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.  
  
  
 lo\_h\_label = lo\_header->create\_label( row = 1 column = 1 ).  
 lo\_h\_label->set\_text('Header in Bold').  
  
 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 ).  
  
 co\_alv->set\_top\_of\_list( lo\_header ).  
 co\_alv->set\_top\_of\_list\_print( lo\_header ).  
  
  
ENDMETHOD.  
  
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.  
  
 CREATE OBJECT lo\_footer.  
  
  
  
 lo\_f\_label = lo\_footer->create\_label( row = 1 column = 1 ).  
 lo\_f\_label->set\_text( 'Footer ..here it goes' ).  
  
  
 lo\_f\_flow = lo\_footer->create\_flow( row = 2 column = 1 ).  
 lo\_f\_flow->create\_text( text = 'this is the text of the 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 ).  
  
 co\_alv->set\_end\_of\_list( lo\_footer ).  
 co\_alv->set\_end\_of\_list\_print( lo\_footer ).  
  
  
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.  
*\**  
*\* get layout object*  
 lo\_layout = co\_alv->get\_layout( ).  
*\**  
*\* set Layout save restriction*  
*\* 1. Set Layout Key .. Unique key identifies the Differenet ALVs*  
 ls\_key-report = sy-repid.  
 lo\_layout->set\_key( ls\_key ).  
*\* 2. Remove Save layout the restriction.*  
 lo\_layout->set\_save\_restriction( if\_salv\_c\_layout=>restrict\_none ).  
*\**  
*\* set initial Layout*  
 lf\_variant = 'DEFAULT'.  
 lo\_layout->set\_initial\_layout( lf\_variant ).  
*\**  
ENDMETHOD.  
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.  
 CALL METHOD set\_pf\_status  
 CHANGING  
 co\_alv = o\_alv.  
 CALL METHOD set\_display\_setting  
 CHANGING  
 co\_alv = o\_alv.  
 CALL METHOD set\_hotspot\_vbeln  
 CHANGING  
 co\_alv = o\_alv.  
*\* co\_report = lo\_report.*  
  
 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\_layout  
 CHANGING  
 co\_alv = o\_alv.  
 CALL METHOD me->set\_columns  
 CHANGING  
 co\_alv = o\_alv.  
  
  
 o\_alv->display( ).  
  
  
  
  
  
 ENDMETHOD.  
  
  
ENDCLASS.  
  
  
  
 DATA: lo\_report TYPE REF TO lcl\_report.  
START-OF-SELECTION.  
 CREATE OBJECT lo\_report.  
 lo\_report->get\_data( ).  
 lo\_report->generate\_output( ).

A screenshot of a computer

Description automatically generated