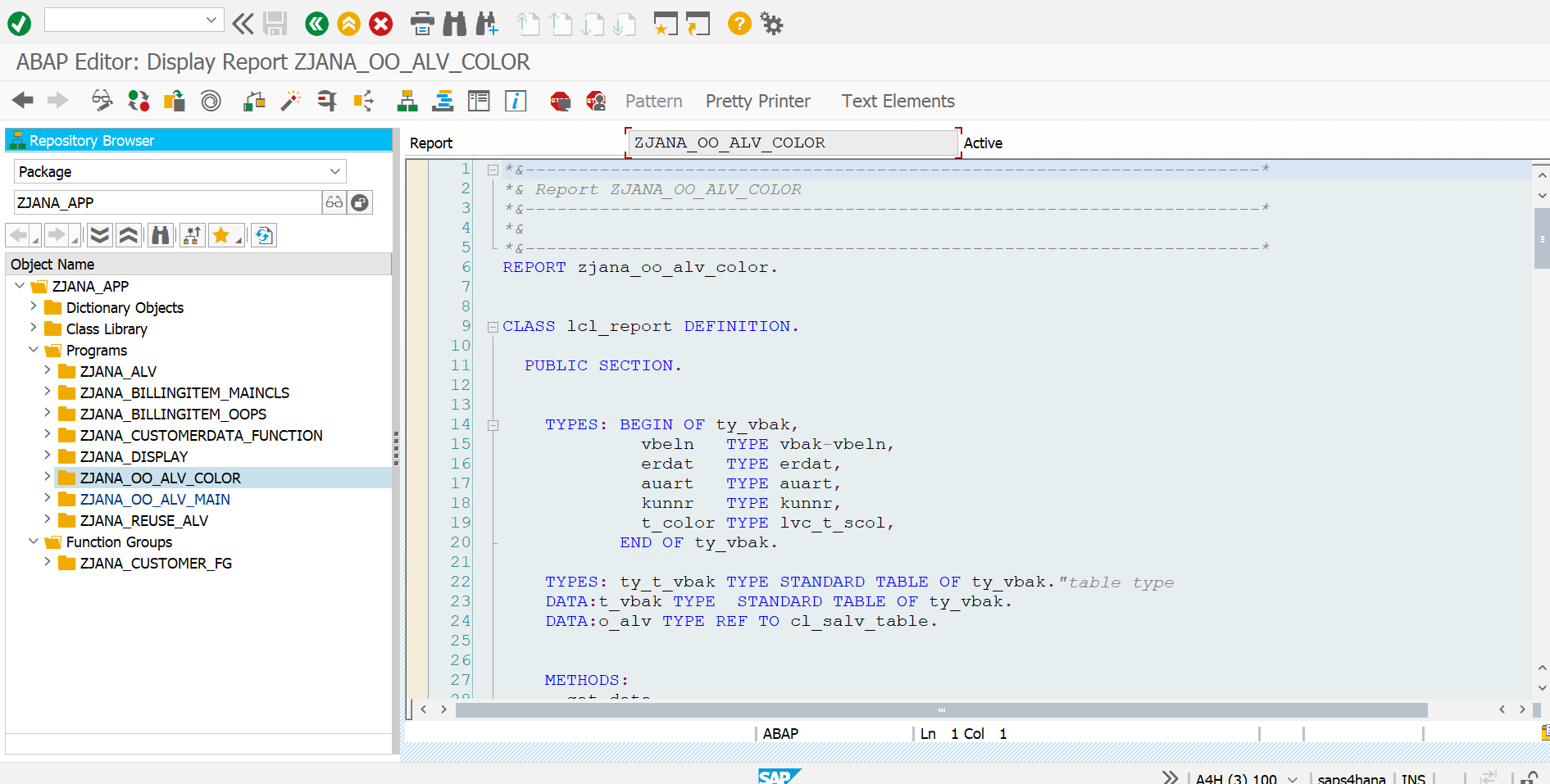
REPORT: ZJANA\_OO\_ALV\_COLOR



*\*&---------------------------------------------------------------------\**  
*\*& Report ZJANA\_OO\_ALV\_COLOR*  
*\*&---------------------------------------------------------------------\**  
*\*&*  
*\*&---------------------------------------------------------------------\**  
REPORT zjana\_oo\_alv\_color.  
  
  
CLASS lcl\_report DEFINITION.  
  
 PUBLIC SECTION.  
  
  
 TYPES: BEGIN OF ty\_vbak,  
 vbeln TYPE vbak-vbeln,  
 erdat TYPE erdat,  
 auart TYPE auart,  
 kunnr TYPE kunnr,  
 t\_color TYPE lvc\_t\_scol,  
 END OF ty\_vbak.  
  
 TYPES: ty\_t\_vbak TYPE STANDARD TABLE OF ty\_vbak.*"table type*  
 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\_colors  
 CHANGING  
  
 co\_alv TYPE REF TO cl\_salv\_table  
 ct\_vbak TYPE ty\_t\_vbak.  
  
  
  
ENDCLASS.  
  
  
  
  
  
  
  
CLASS lcl\_report IMPLEMENTATION.  
  
 METHOD get\_data.  
 SELECT vbeln erdat auart kunnr INTO CORRESPONDING FIELDS OF 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(  
*\* EXPORTING*  
*\* list\_display = if\_salv\_c\_bool\_sap=>false " ALV Displayed in List Mode*  
*\* r\_container = " Abstract Container for GUI Controls*  
*\* container\_name =*  
 IMPORTING  
 r\_salv\_table = o\_alv *" Basis Class Simple ALV Tables*  
 CHANGING  
 t\_table = t\_vbak  
 ).  
 CATCH cx\_salv\_msg. *" ALV: General Error Class with Message*  
  
  
  
  
  
  
 ENDTRY.  
  
 CALL METHOD set\_pf\_status  
 CHANGING  
 co\_alv = o\_alv.  
  
  
  
  
 CALL METHOD set\_colors  
 CHANGING  
 co\_alv = o\_alv  
 ct\_vbak = t\_vbak.  
 o\_alv->display( ).  
  
  
  
 ENDMETHOD.  
  
 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 ).  
  
 ENDMETHOD.  
 METHOD set\_colors.  
 DATA: lo\_cols\_tab TYPE REF TO cl\_salv\_columns\_table,  
 lo\_col\_tab TYPE REF TO cl\_salv\_column\_table,  
 ls\_color TYPE lvc\_s\_colo.  
  
  
 lo\_cols\_tab = co\_alv->get\_columns( ).  
  
 INCLUDE <color>.  
 TRY.  
 lo\_col\_tab ?= lo\_cols\_tab->get\_column( 'ERDAT' ).  
 ls\_color-col = col\_total.  
 lo\_col\_tab->set\_color( ls\_color ).  
 CATCH cx\_salv\_not\_found.  
 ENDTRY.  
 DATA: lt\_s\_color TYPE lvc\_t\_scol,  
 ls\_s\_color TYPE lvc\_s\_scol,  
 la\_vbak LIKE LINE OF ct\_vbak,  
 l\_count TYPE i.  
  
  
 LOOP AT ct\_vbak INTO la\_vbak.  
 l\_count = l\_count + 1.  
  
 CASE l\_count.  
 WHEN 3.  
 ls\_s\_color-fname ='AUART'.  
 ls\_s\_color-color-col = col\_negative.  
 ls\_s\_color-color-int = 0.  
 ls\_s\_color-color-inv = 0.  
  
 APPEND ls\_s\_color TO lt\_s\_color.  
  
 CLEAR ls\_s\_color.  
  
 WHEN 5.  
 ls\_s\_color-color-col = col\_positive.  
 ls\_s\_color-color-int = 0.  
 ls\_s\_color-color-inv = 0.  
  
 APPEND ls\_s\_color TO lt\_s\_color.  
 CLEAR ls\_s\_color.  
 ENDCASE.  
 la\_vbak-t\_color = lt\_s\_color.  
 MODIFY ct\_vbak FROM la\_vbak.  
 CLEAR la\_vbak.  
 CLEAR lt\_s\_color.  
 ENDLOOP.  
  
  
 TRY.  
 lo\_cols\_tab->set\_color\_column( 'T\_color' ).  
  
  
 CATCH CX\_SALV\_data\_error.  
  
 ENDTRY.  
  
  
  
 ENDMETHOD.  
  
ENDCLASS.  
  
  
  
START-OF-SELECTION.  
 DATA:lo\_report TYPE REF TO lcl\_report.  
  
  
  
 CREATE OBJECT lo\_report.  
 lo\_report->get\_data( ).  
 lo\_report->generate\_output( ).

A screenshot of a computer

Description automatically generated