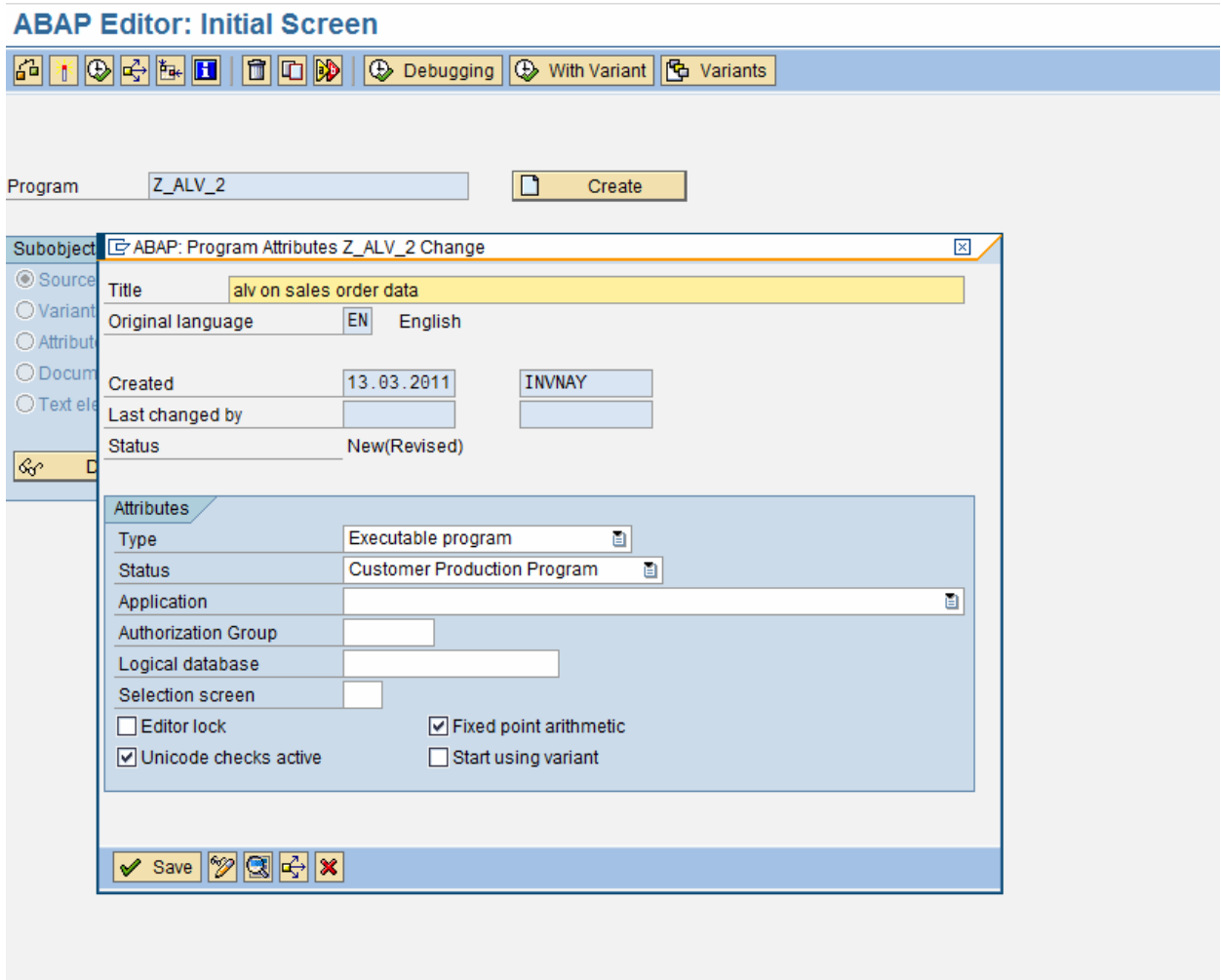


Scenario – This is an ALV report on the Sales Order Data as per the billing date given in the selection screen with the help of Oops Concept

- 1) Create the program with the following attributes



- 2) We initially have to do all the declarations required for the program.
- 3) Code for the program is in the following attachment

In get\_billing\_data using the selection screen parameters get the \*

\* billing data from billing header(VBAK) and Item (VBAP) \*

\* 2. Get the SHIP-TO partner data from VBPA using Customer number

as \*

\* one condition and store in it\_vbpa\_sh

\* 9. Popultae the ALV display table



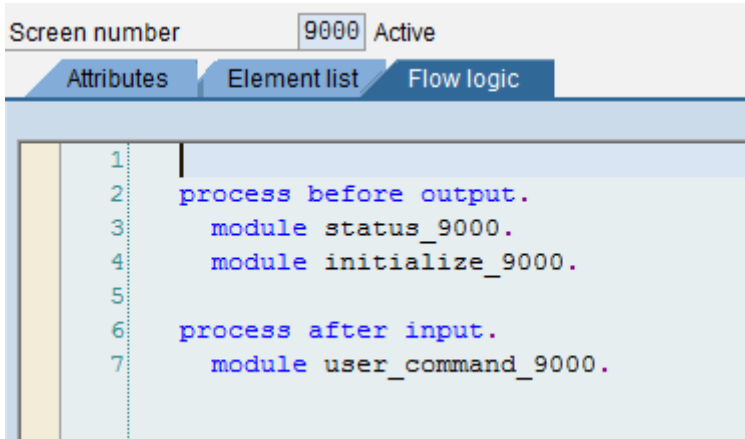
2 alv.txt

- 4) Now for displaying the data in an interactive report using oops, we need to make sure, we have the data in an internal table that has to be displayed in the output. (perform data\_retrieval)

```
126 start-of-selection.
127
128
129 * get billing data from delivery header(vbrk) and item (vbrp)
130 perform get_billing_data.
131
132 * get ship-to country
133 perform get_ship_to_data.
134
135 * get material group description 3
136 perform get_material_group_3.
137
138 * get material group description 4
139 perform get_material_group_4.
140
141 * form the alv grid title
142 perform form_title.
143
144 * process data
145 perform process_data.
146
147 * populate text descriptions
148 perform populate_description.
149
150
151 *----- event end-of-selection -----*
152 end-of-selection.
153 * set the text to when date high is there
154 if not s_fkdat-high is initial.
155 |   w_to = 'to' (c16).
156 |   endif.
157
158 * call the screen 9000 for alv display
159 call screen 9000.
```

- 5) We do the data retrieval in the first four performs.
- 6) In the perform form\_title, we fill the variable 'w\_title' with the title that has to come for the alv report output.

- 7) In the perform process\_data, we do all the calculations required in the FS and fill the data into the final internal table it\_data.
- 8) And finally in the perform populate\_description we fill the final it\_data with the description fields and also we append this table with the Grand Total value in a different color.
- 9) After all this process we call the screen 9000.

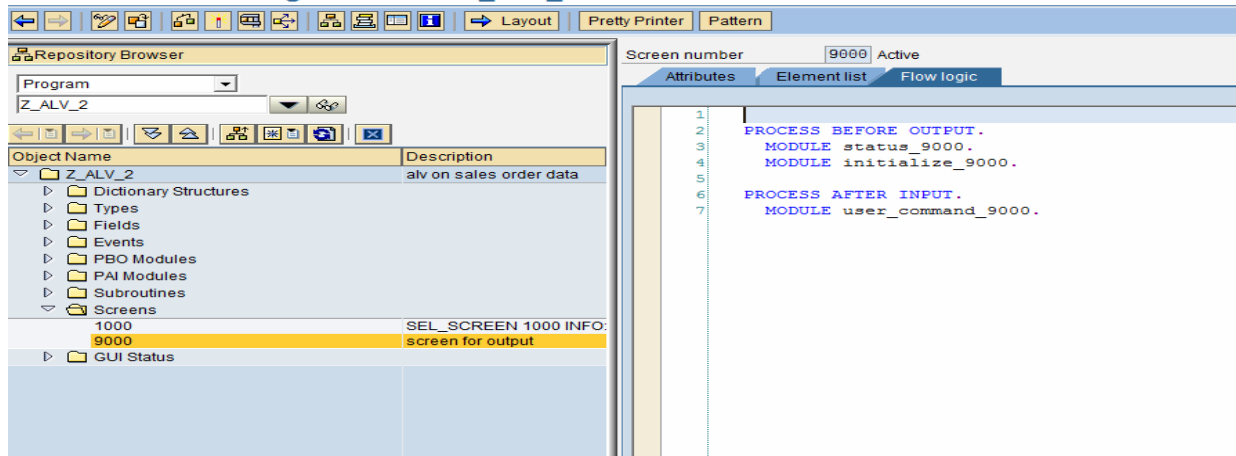


Screen number 9000 Active

Attributes Element list Flow logic

```
1  
2 process before output.  
3 module status_9000.  
4 module initialize_9000.  
5  
6 process after input.  
7 module user_command_9000.
```

Screen Painter: Change Screen for Z\_ALV\_2



Repository Browser

Program Z\_ALV\_2

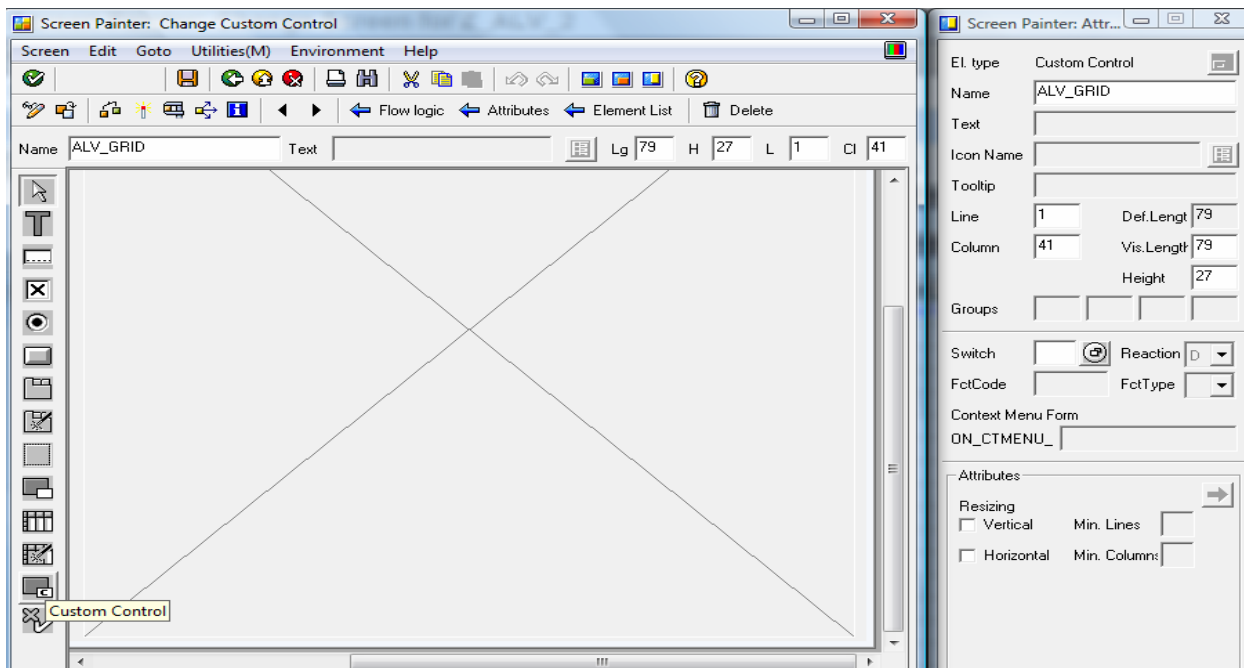
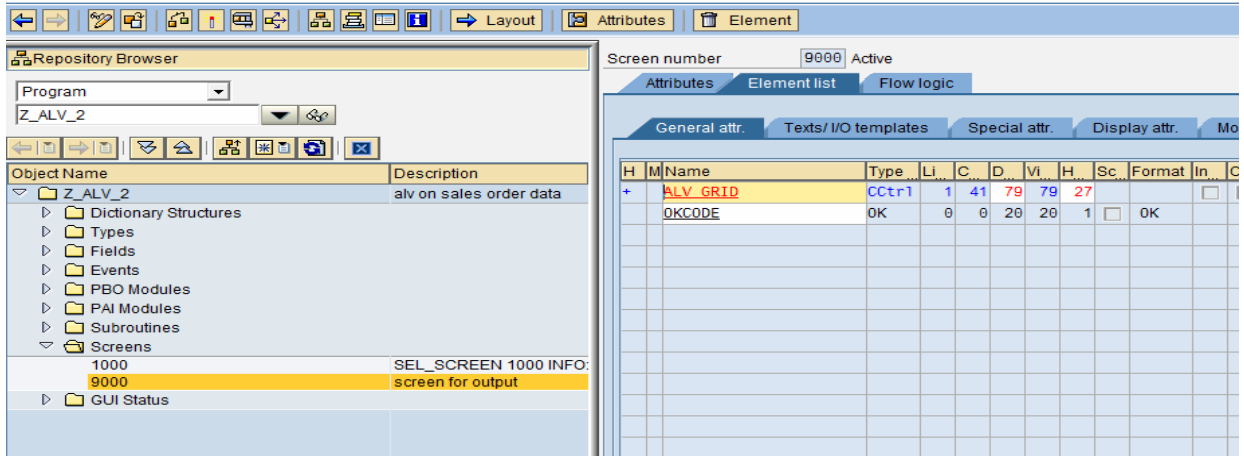
Object Name	Description
Z_ALV_2	alv on sales order data
Dictionary Structures	
Types	
Fields	
Events	
PBO Modules	
PAI Modules	
Subroutines	
Screens	
1000	SEL_SCREEN 1000 INFO:
9000	screen for output
GUI Status	

Screen number 9000 Active

Attributes Element list Flow logic

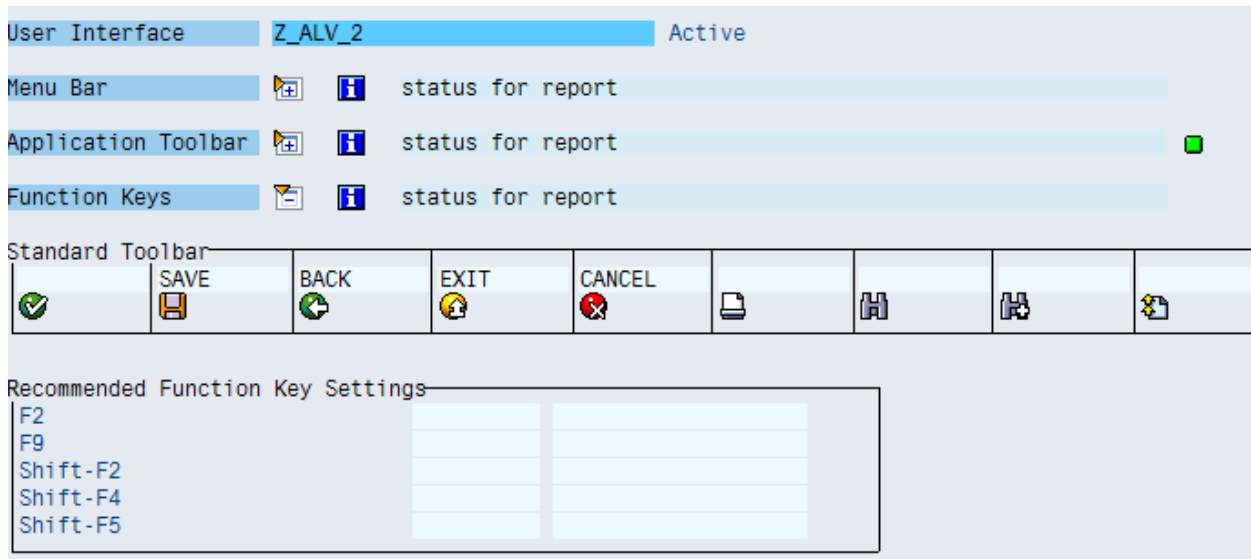
```
1  
2 PROCESS BEFORE OUTPUT.  
3 MODULE status_9000.  
4 MODULE initialize_9000.  
5  
6 PROCESS AFTER INPUT.  
7 MODULE user_command_9000.
```

### Screen Painter: Change Screen for Z\_ALV\_2



10) Now the screen should have a status bar and the code for this is

```
module status_9000 output.  
  set pf-status 'MAIN9000'.  
  set titlebar 'TITLE_9000'.  
endmodule.
```



11) Take the standard required fields and give the names of the push buttons as shown above.

12) In the module initialize\_9000, do the following code.

```
module initialize_9000 output.  
* call the alv grid to display data  
  perform display_alv_grid.  
  
endmodule.
```

13) In this perform ,we need to create an object of the instance of the class  
cl\_gui\_custom\_container and pass the container name which we created in the screen.

14) Now create an object for the instance of the class 'cl\_gui\_alv\_grid' and pass the object created  
in the step 13 as the parent in the export parameter.

15) Now create the field catalog in wt\_filecatalog.

16) Now call the method in the object created in step 14 as follows –

```
call method w_alv_grid->set_table_for_first_display
```

In this method pass the parameters layout, it\_outtab and the field catalog mandatorily.

17) In the PAI , user command input put in the following code.

### ABAP Editor: Change Report Z\_ALV\_2

The screenshot shows the ABAP Editor interface. On the left is the Repository Browser with the following structure:

Object Name	Description
Z_ALV_2	alv on sales order data
Dictionary Structures	
Types	
Fields	
Events	
END-OF-SELECTION	
START-OF-SELECTION	
PBO Modules	
INITIALIZE_9000	
STATUS_9000	
PAI Modules	
USER_COMMAND_9000	
Subroutines	
DISPLAY_ALV_GRID	
FORM_TITLE	
GET_BILLING_DATA	
GET_MATERIAL_GROUP_3	
GET_MATERIAL_GROUP_4	
GET_SHIP_TO_DATA	
POPULATE_DESCRIPTION	
PREPARE_FIELDCAT	
PROCESS_DATA	
Screens	
1000	SEL_SCREEN 1000 INFO:
9000	screen for output
GUI Status	
MAIN9000	status for report

The code editor on the right shows the following code:

```
478 * call the alv grid to display data
479 PERFORM display_alv_grid.
480
481 ENDMODULE. " initialize 9000 OUI
482
483 *-----
484 * module user_command_9000 input
485 *-----
486 * module to handle user command
487
487 MODULE user_command_9000 INPUT.
488 CASE okcode.
489 WHEN 'BACK'.
490 SET SCREEN 0.
491 CLEAR okcode.
492 LEAVE SCREEN.
493 WHEN 'CANCEL'.
494 SET SCREEN 0.
495 CLEAR okcode.
496 LEAVE SCREEN.
497 WHEN 'EXIT'.
498 CLEAR okcode.
499 LEAVE PROGRAM.
500 WHEN OTHERS.
501 ENDCASE.
502 ENDMODULE. " USER COMMAND 9000 I
503
504 *-----
505 * form display_alv_grid
506 *-----
507 * display the data in alv grid
508
508 FORM display_alv_grid .
509 IF w_alv_container IS INITIAL.
510
```

Now we are ready with the code for ALV using Oups, screen 9000 with a container which is mandatory while working on ALV with oops, and the status bar for the screen. Now execute the report and we get the following selection screen.

### alv on sales order data

The screenshot shows the selection screen for report Z\_ALV\_2. The title bar contains a green arrow icon. The status bar is empty. The main area has a 'Billing Date' field with the value '03.08.2009' and a 'to' field.

Input some test data for the billing date and you get the output in the container area that is defined in the screen 9000.



The image shows a screenshot of an SAP report titled "TTD - Statement of Export -From 03/08/2009". The report is displayed in a table format with a standard SAP toolbar at the top. The table has seven columns: Market Name, Flavour Type, Tea Type, Tea Bag(MT), PKT Tea(MT), Bulk Tea(MT), and Total(MT). The data rows show zero values for all categories across various markets, with a Grand Total row at the bottom.

Market Name	Flavour Type	Tea Type	Tea Bag(MT)	PKT Tea(MT)	Bulk Tea(MT)	Total(MT)
China			0.000	0.000	0.000	0.000
Japan			0.000	0.000	0.000	0.000
Taiwan			0.000	0.000	0.000	0.000
Germany			0.000	0.000	0.000	0.000
Colombia			0.000	0.000	0.000	0.000
Dominican Rep.			0.000	0.000	0.000	0.000
Malaysia			0.000	0.000	0.000	0.000
<b>Grand Total:</b>			<b>0.000</b>	<b>0.000</b>	<b>0.000</b>	<b>0.000</b>