MODULE POOL SUB SCREEN

(ADDITIN, SUBTRACTION, MULTIPLICATION).

Here I'm adding only the code, screens and what are all the fields I'm using….

First write a code inside the include function INCLUDE ZTESTTOP if we double click on the highlighted one then we can see the below code.

Here we are declaring the fields…**code**

PROGRAM ZSCREEN_TEST.

***INCLUDE ZTESTTOP**

DATA : num1(10) TYPE c,

num2(10) TYPE c,

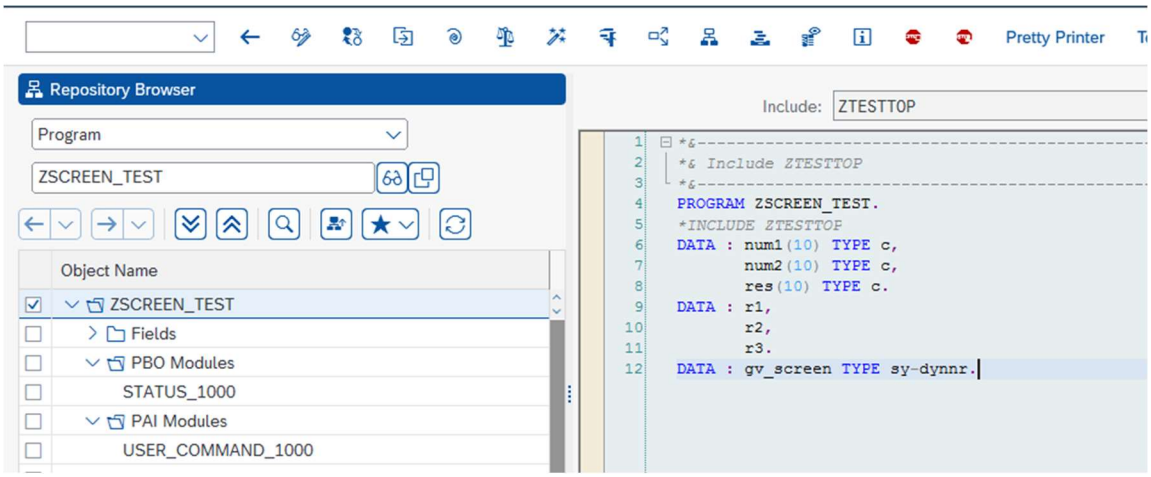
res(10) TYPE c.

DATA : r1,

r2,

r3.

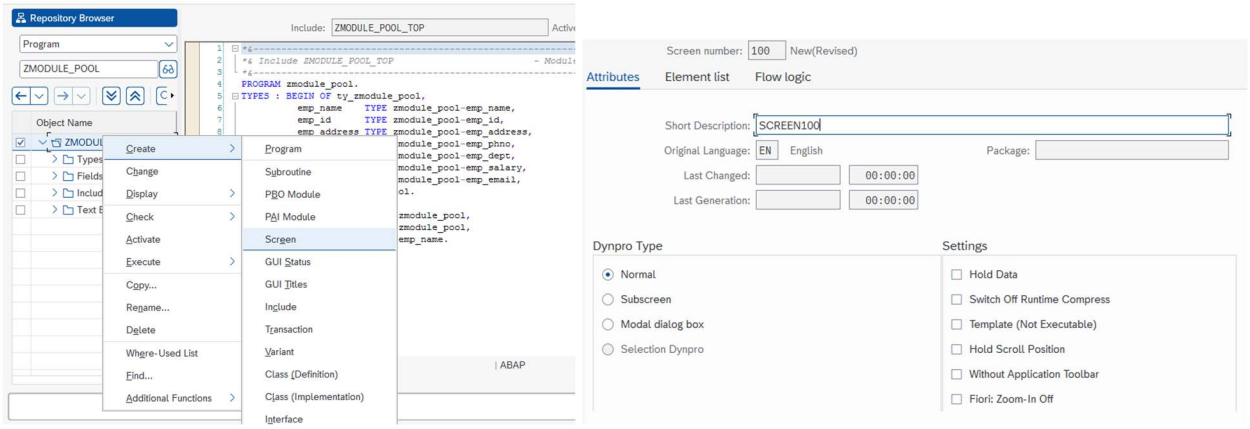
DATA : gv_screen TYPE sy-dynnr.

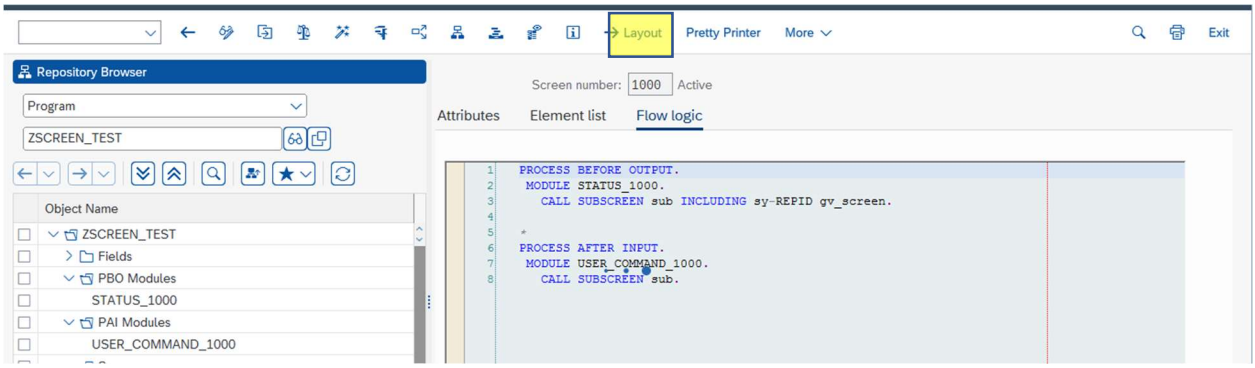


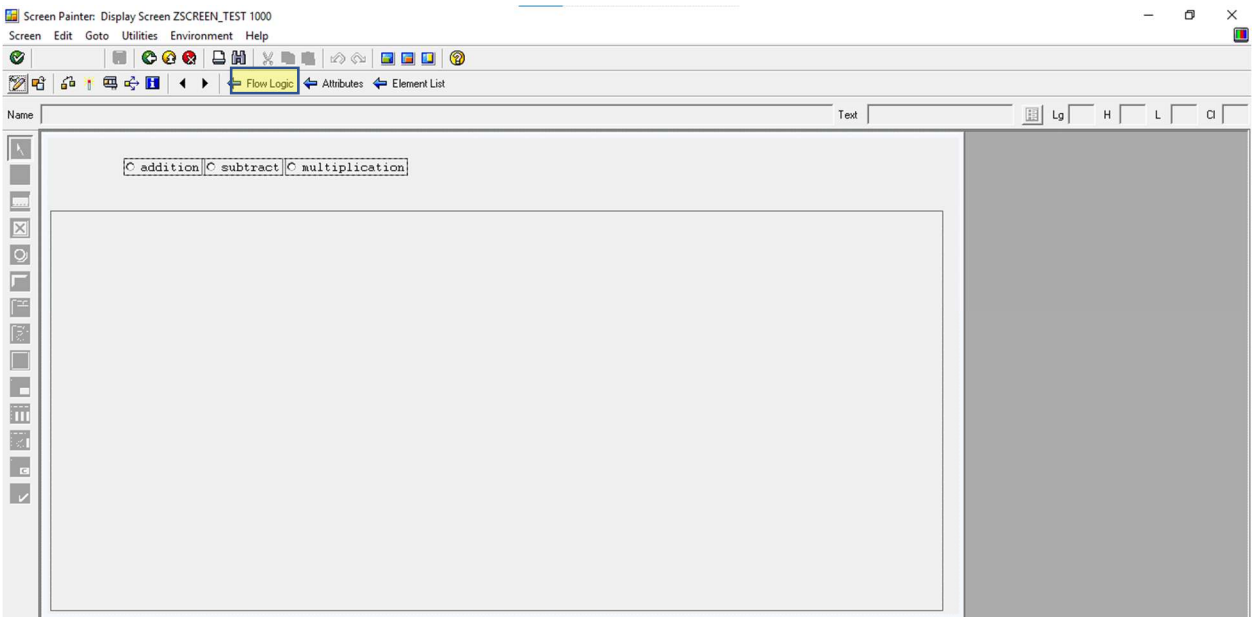
After declaring the key words, variables, fields then we can write a logic under the screen so for that..

Create a screen as main screen with short description then the PBO and PAI will open then go to layout.

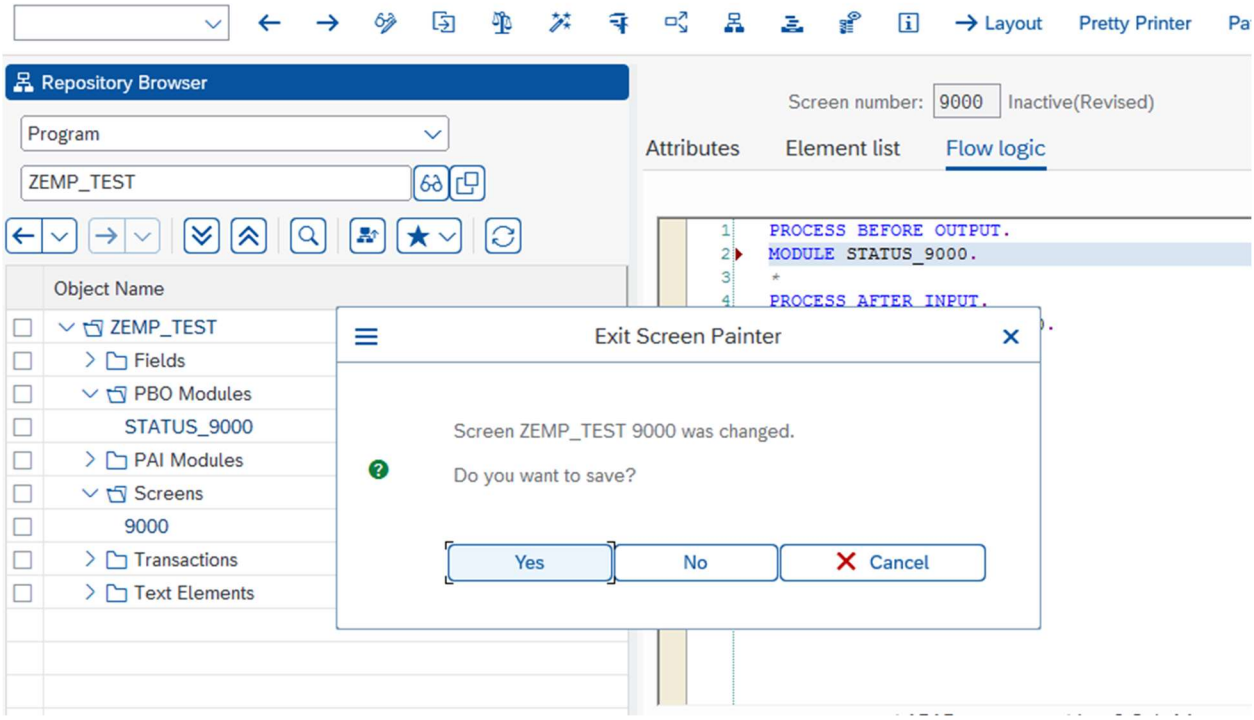
After that use the input /output field , push button, box , radio buttons, and also here mainly using the sub screen in scree using all these things then click on flow logic..





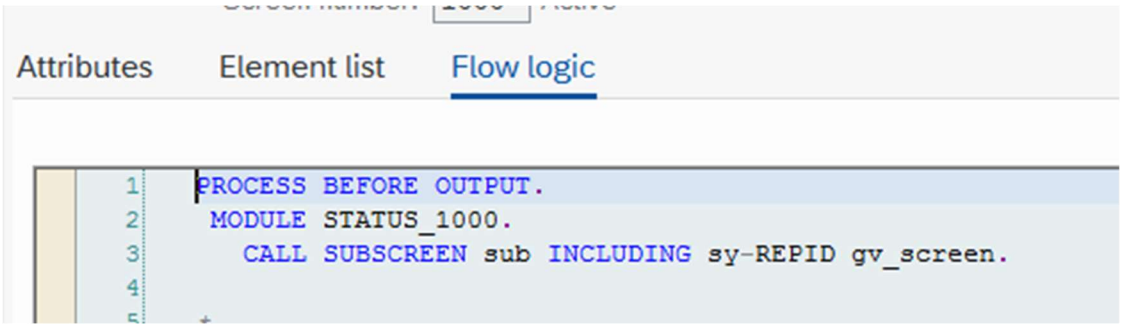


Here change the PBO and PAI logic from comment to uncomment. Then double click on the uncommented one. Under that write a logic of the program.



Under the PBO we have to call the sub screen using the sy-repid

Here the sub is the functional code and dynamic screen using for main screen using sub screen.

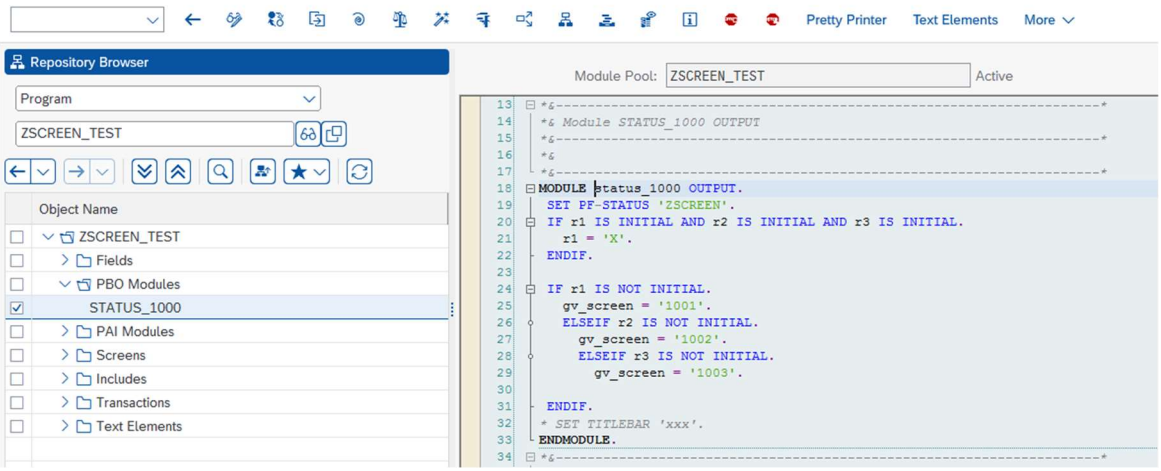


Code(for calling the sub screen) :

```
PROCESS BEFORE OUTPUT.
 MODULE STATUS_1000.
   CALL SUBSCREEN sub INCLUDING sy-REPID gv_screen.
```

PBO :

Under the PBO written the logic of initializing the radio buttons, and using the dynamics screens we can use the sub screens of the addition, subtraction, multiplication.



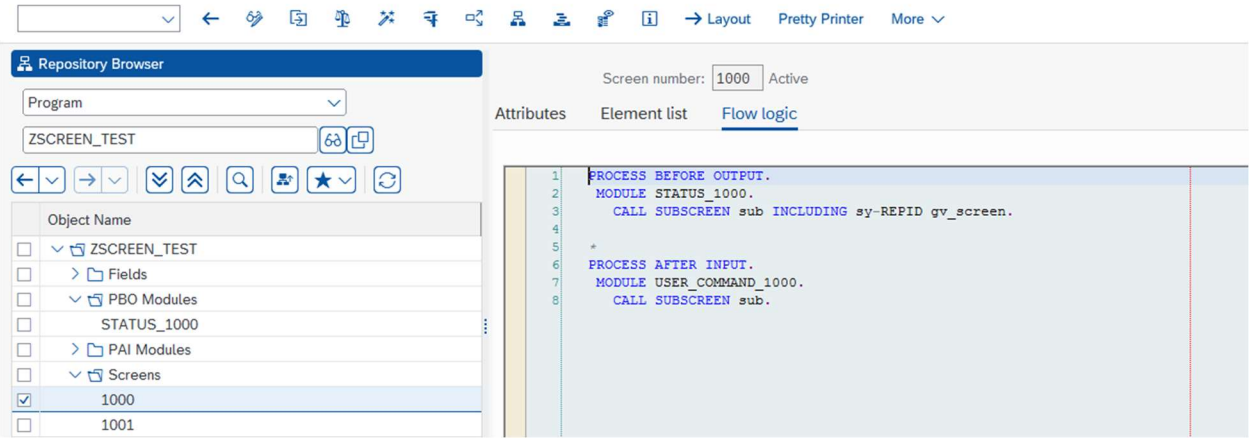
PBO CODE(For Initialization of Sub screen using the radio buttons)

```
MODULE status_1000 OUTPUT.
 SET PF-STATUS 'ZSCREEN'.
 IF r1 IS INITIAL AND r2 IS INITIAL AND r3 IS INITIAL.
  r1 = 'X'.
 ENDIF.
 IF r1 IS NOT INITIAL.
  gv_screen = '1001'.
  ELSEIF r2 IS NOT INITIAL.
    gv_screen = '1002'.
    ELSEIF r3 IS NOT INITIAL.
      gv_screen = '1003'.
```

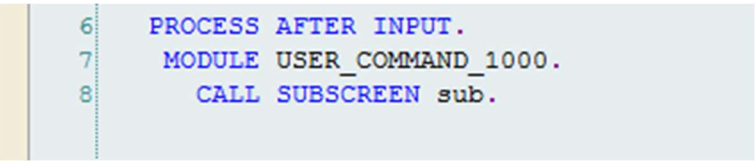
ENDIF.

*** SET TITLEBAR 'xxx'.**

ENDMODULE.



Under the PAI we have to call the sub screen using the functional code what we are declared in layout .



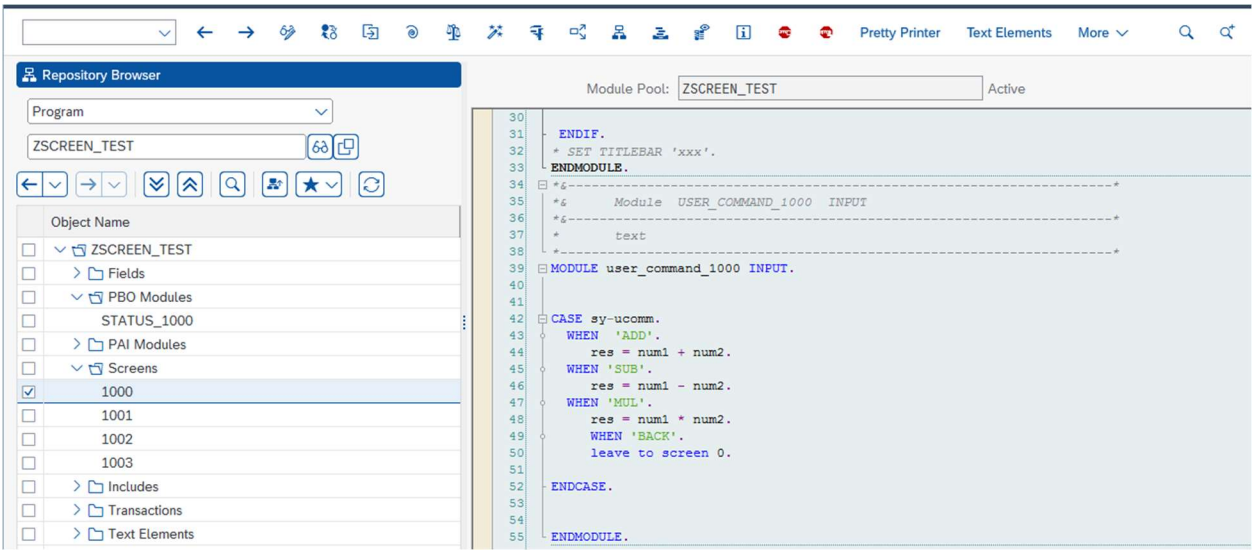
Code: (for calling the sub screen using functional code)

PROCESS AFTER INPUT.

MODULE USER_COMMAND_1000.

CALL SUBSCREEN sub.

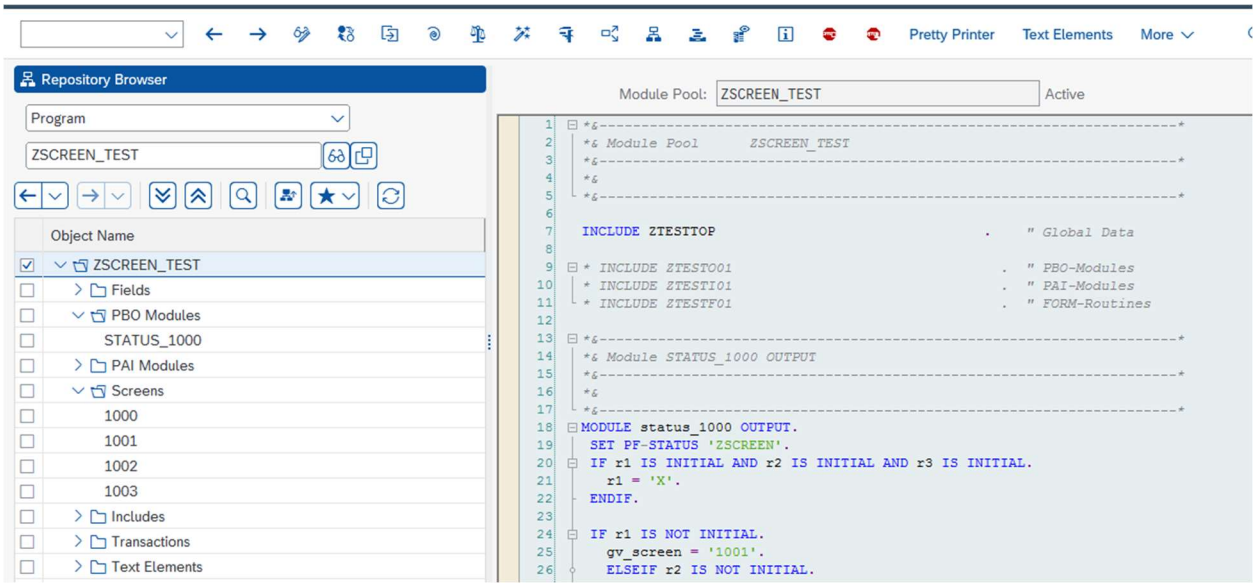
PAI: here we are using the case statement for addition, subtraction, multiplication with sy-ucomm. And after this we have to go back for that using the BACK button. Then all the process done leave the screen so for that purpose we use the leave screen 0.

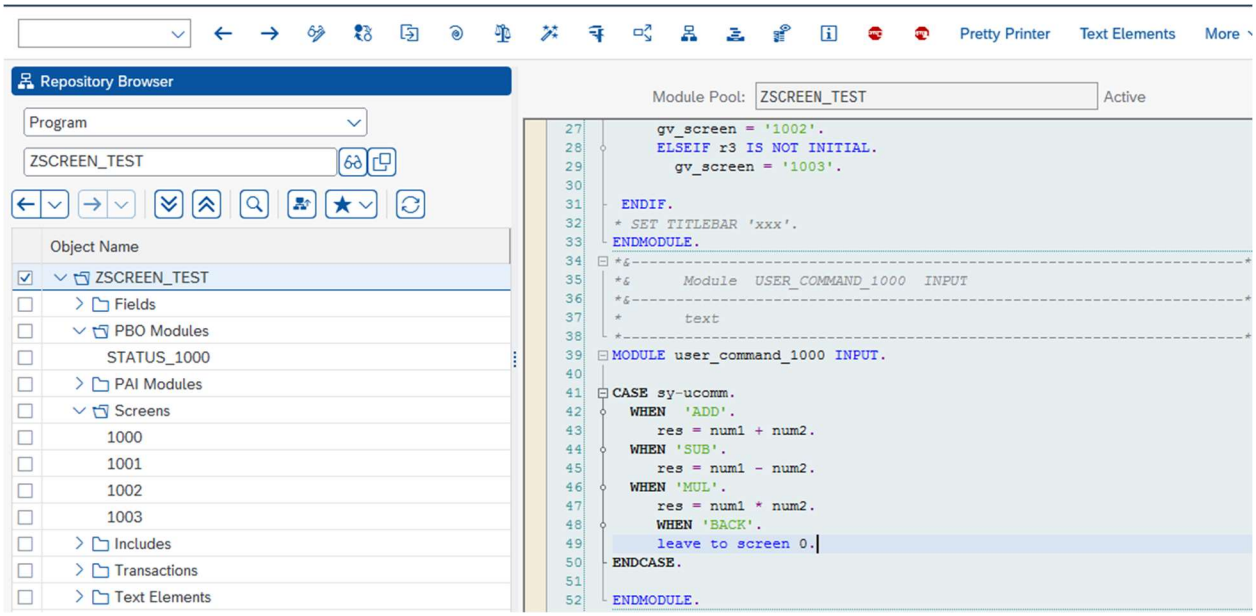


CODE:(user command)

```
MODULE user_command_1000 INPUT.

CASE sy-ucomm.

  WHEN  'ADD'.

    res = num1 + num2.

  WHEN 'SUB'.

    res = num1 - num2.

  WHEN 'MUL'.

    res = num1 * num2.

    WHEN 'BACK'.

    leave to screen 0.


ENDCASE.


ENDMODULE.
```

Done with all logic in the program active the main program and





Main program code : (includes all the logic in module pool)

CODE:

&---

*& Module Pool ZSCREEN_TEST

&---

*&

&---

```
INCLUDE ZTESTTOP                        .    " Global Data

* INCLUDE ZTESTO01                       .  " PBO-Modules
* INCLUDE ZTESTI01                       .  " PAI-Modules
* INCLUDE ZTESTF01                       .  " FORM-Routines

*&---------------------------------------------------------------------*
*& Module STATUS_1000 OUTPUT
*&---------------------------------------------------------------------*
*&
*&---------------------------------------------------------------------*
MODULE status_1000 OUTPUT.
 SET PF-STATUS 'ZSCREEN'.
 IF r1 IS INITIAL AND r2 IS INITIAL AND r3 IS INITIAL.
   r1 = 'X'.
 ENDIF.

 IF r1 IS NOT INITIAL.
   gv_screen = '1001'.
   ELSEIF r2 IS NOT INITIAL.
     gv_screen = '1002'.
     ELSEIF r3 IS NOT INITIAL.
       gv_screen = '1003'.

 ENDIF.
* SET TITLEBAR 'xxx'.
```

```
ENDMODULE.
*&---------------------------------------------------------------------*
*&      Module  USER_COMMAND_1000  INPUT
*&---------------------------------------------------------------------*
*       text
*----------------------------------------------------------------------*
MODULE user_command_1000 INPUT.


CASE sy-ucomm.
  WHEN  'ADD'.
    res = num1 + num2.
  WHEN 'SUB'.
    res = num1 - num2.
  WHEN 'MUL'.
    res = num1 * num2.
    WHEN 'BACK'.
    leave to screen 0.
ENDCASE.


ENDMODULE.
```

MODULE POOL MAIN PRG.txt

PBO & PAI.txt

INCLUDE PRG.txt

COMPLETED 😊😊