

```

1  *&-----*
2  *& Report ZTC_CALCULADORA_OO
3  *&-----*
4  *&
5  *&-----*
6  REPORT ztc_calculadora_oo.
7
8  CLASS lcl_main DEFINITION.
9
10 PUBLIC SECTION.
11 CLASS-METHODS main.
12
13 *Variaveis Globais
14 CLASS-DATA: go_calculadora TYPE REF TO ztccl_calculadora,
15             gv_result      TYPE ztce_valorescalc,
16             gv_msg         TYPE string,
17             gv_operation   TYPE domvalue_l.
18
19 PRIVATE SECTION.
20
21 * Métodos e atributos para o processamento do programa
22 CLASS-METHODS: instance_calculadora, process_calculadora, display_result,
23 check_operation RETURNING VALUE(rv_exist) TYPE bool.
24
25
26 ENDCLASS.
27
28 SELECTION-SCREEN BEGIN OF BLOCK b1 WITH FRAME TITLE TEXT-001.
29
30 PARAMETERS p_oper      TYPE char1 OBLIGATORY.
31 PARAMETERS p_valor1    TYPE ztce_valorescalc OBLIGATORY.
32 PARAMETERS p_valor2    TYPE ztce_valorescalc OBLIGATORY.
33
34 SELECTION-SCREEN END OF BLOCK b1.
35
36 CLASS lcl_main IMPLEMENTATION.
37
38 METHOD main.
39
40     gv_operation = p_oper.
41
42     instance_calculadora( ).
43
44     IF go_calculadora IS BOUND. "O objeto estar atribuido.
45
46         IF check_operation( ) EQ abap_true.
47
48             process_calculadora( ).
49             display_result( ).
50
51         ELSE.
52
53             WRITE TEXT-003.
54
55         ENDIF.
56
57     ELSE.
58
59         WRITE TEXT-002.
60
61     ENDIF.
62
63 ENDMETHOD.
64
65 METHOD instance_calculadora.
66     CREATE OBJECT go_calculadora
67     EXPORTING
68         iv_valor1 = p_valor1
69         iv_valor2 = p_valor2.
70 ENDMETHOD.
71
72 METHOD process_calculadora.
73
74     IF go_calculadora IS BOUND. "se foi instanciado

```

```

74     CASE p_oper.
75         WHEN '+' .
76             gv_result = go_calculadora->somar( ).
77         WHEN '-' .
78             gv_result = go_calculadora->subtrair( ).
79         WHEN '/' .
80             go_calculadora->dividir(
81                 IMPORTING
82                     ev_result = gv_result           " Valores calculadora
83                     ev_msg    = gv_msg
84             ).
85         WHEN '*' .
86             gv_result = go_calculadora->multiplicar( ).
87         WHEN OTHERS.
88             WRITE TEXT-003.
89     ENDCASE.
90
91     ENDIF.
92     ENDMETHOD.
93
94     METHOD display_result.
95
96         IF gv_msg IS NOT INITIAL.
97             WRITE gv_msg.
98         ELSEIF gv_result IS NOT INITIAL.
99             WRITE gv_result.
100        ENDIF.
101
102    ENDMETHOD.
103
104    METHOD check_operation.
105
106        IF p_oper IS NOT INITIAL.
107
108            SELECT SINGLE domvalue_l, ddlanguage, ddtext FROM dd07v
109                INTO @DATA(ls_dd07v)
110                WHERE domname EQ 'ZTCD_OPERACAO'
111                AND domvalue_l EQ @gv_operation
112                AND ddlanguage EQ @sy-langu.
113
114            IF sy-subrc EQ 0.
115
116                rv_exist = abap_true.
117
118            ENDIF.
119
120        ENDIF.
121
122    ENDMETHOD.
123
124    ENDCLASS.
125
126    START-OF-SELECTION.
127        lcl_main=>main( ).

```