```
*&---------------------------------------------------------------------*
*& Report ZTC_CALCULADORA_OO
*&---------------------------------------------------------------------*
*&
*&---------------------------------------------------------------------*
REPORT ztc_calculadora_oo.

CLASS lcl_main DEFINITION.

  PUBLIC SECTION.
    CLASS-METHODS main.

*Variaveis Globais
    CLASS-DATA: go_calculadora TYPE REF TO ztccl_calculadora,
                gv_result      TYPE ztce_valorescalc,
                gv_msg         TYPE string,
                gv_operation   TYPE domvalue_l.

  PRIVATE SECTION.

*   Métodos e atributos para o processamento do programa
    CLASS-METHODS: instance_calculadora, process_calculadora, display_result,
check_operation RETURNING VALUE(rv_exist) TYPE bool.

ENDCLASS.

SELECTION-SCREEN BEGIN OF BLOCK b1 WITH FRAME TITLE TEXT-001.

  PARAMETERS p_oper     TYPE char1 OBLIGATORY.
  PARAMETERS p_valor1   TYPE ztce_valorescalc OBLIGATORY.
  PARAMETERS p_valor2   TYPE ztce_valorescalc OBLIGATORY.

SELECTION-SCREEN END OF BLOCK b1.

CLASS lcl_main IMPLEMENTATION.

  METHOD main.

    gv_operation = p_oper.

    instance_calculadora( ).

    IF go_calculadora IS BOUND. "O objeto estar atribuido.

      IF check_operation( ) EQ abap_true.

        process_calculadora( ).
        display_result( ).

      ELSE.

        WRITE TEXT-003.

      ENDIF.

    ELSE.

      WRITE TEXT-002.

    ENDIF.

  ENDMETHOD.

  METHOD instance_calculadora.
    CREATE OBJECT go_calculadora
          EXPORTING
            iv_valor1 = p_valor1
            iv_valor2 = p_valor2.
  ENDMETHOD.

  METHOD process_calculadora.

    IF go_calculadora IS BOUND. "se foi instanciado

```

```
     CASE p_oper.
       WHEN '+'.
         gv_result = go_calculadora->somar( ).
       WHEN '-'.
         gv_result = go_calculadora->subtrair( ).
       WHEN '/'.
         go_calculadora->dividir(
           IMPORTING
             ev_result = gv_result           " Valores calculadora
             ev_msg    = gv_msg
         ).
       WHEN '*'.
         gv_result = go_calculadora->multiplicar( ).
      WHEN OTHERS.
        WRITE TEXT-003.
     ENDCASE.

    ENDIF.
  ENDMETHOD.

  METHOD display_result.

    IF gv_msg IS NOT INITIAL.
      WRITE gv_msg.
    ELSEIF gv_result IS NOT INITIAL.
      WRITE gv_result.
    ENDIF.

  ENDMETHOD.

  METHOD check_operation.

    IF p_oper IS NOT INITIAL.

      SELECT SINGLE domvalue_l, ddlanguage, ddtext FROM dd07v
        INTO @DATA(ls_dd07v)
        WHERE domname EQ 'ZTCD_OPERACAO'
        AND domvalue_l EQ @gv_operation
        AND ddlanguage EQ @sy-langu.

        IF sy-subrc EQ 0.

          rv_exist = abap_true.

        ENDIF.

    ENDIF.

  ENDMETHOD.

ENDCLASS.

START-OF-SELECTION.
  lcl_main=>main( ).
```