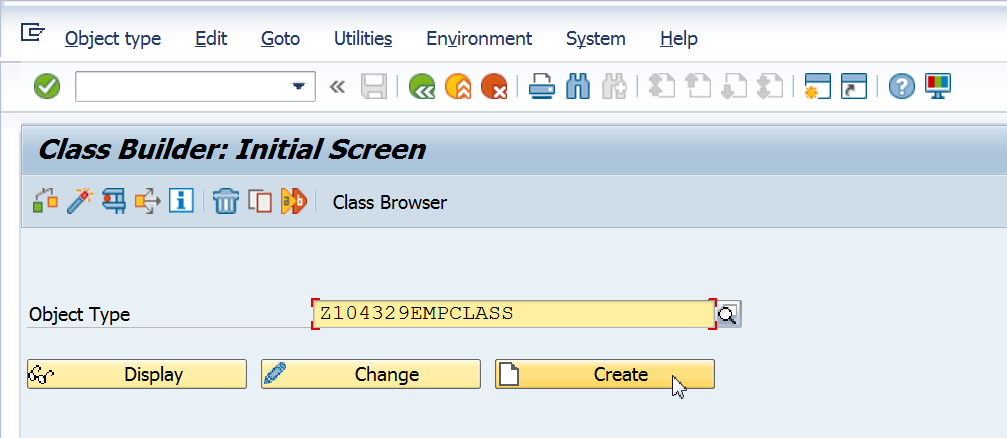
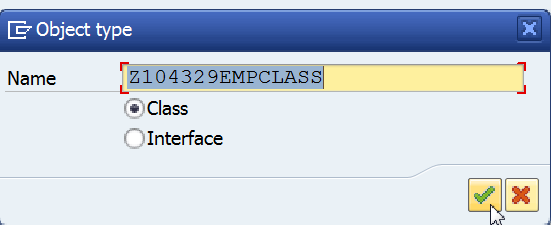
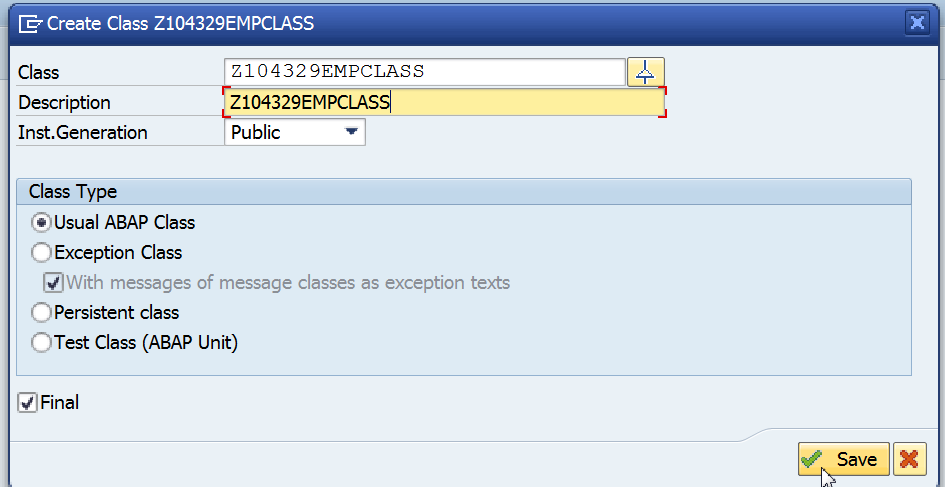
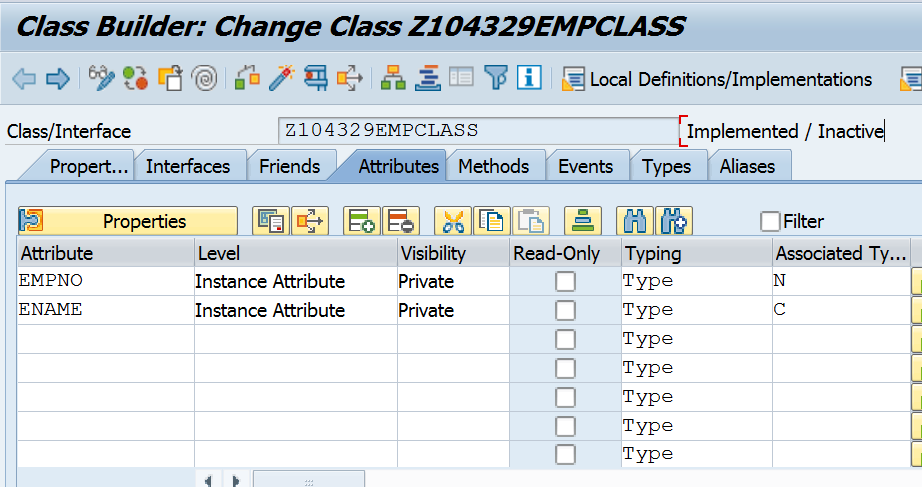
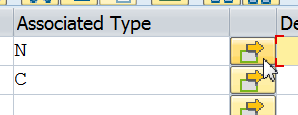
TCODE SE24

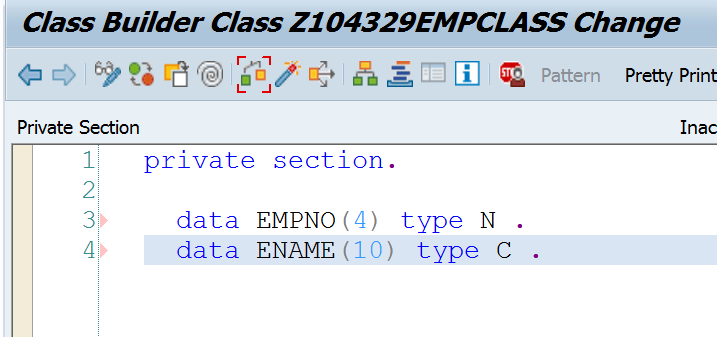


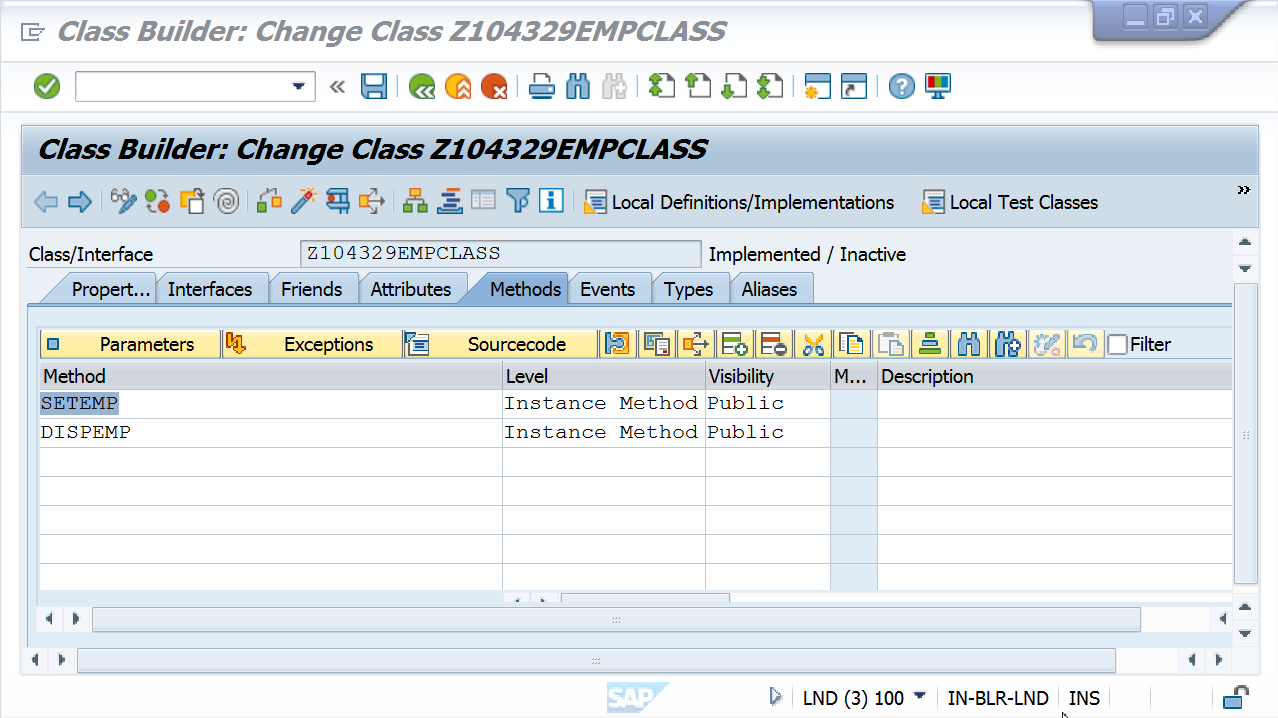


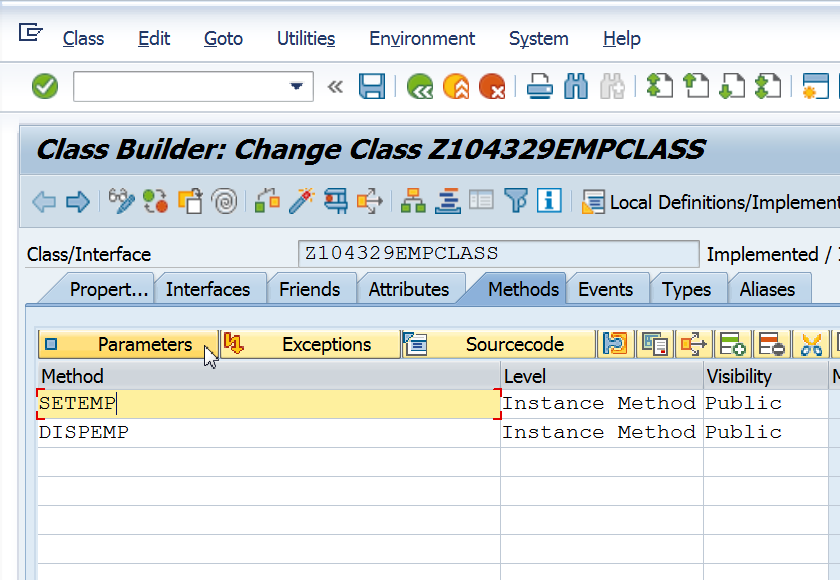


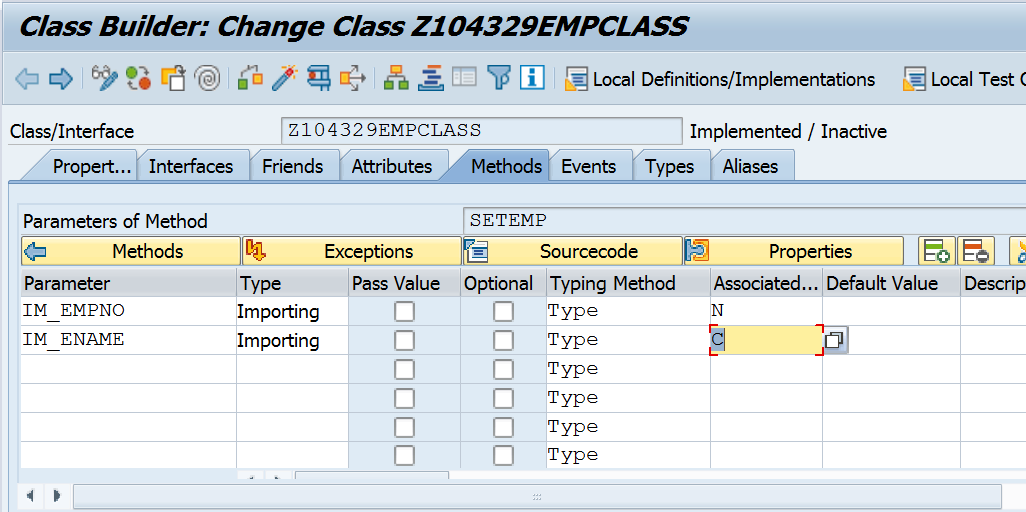
Click on



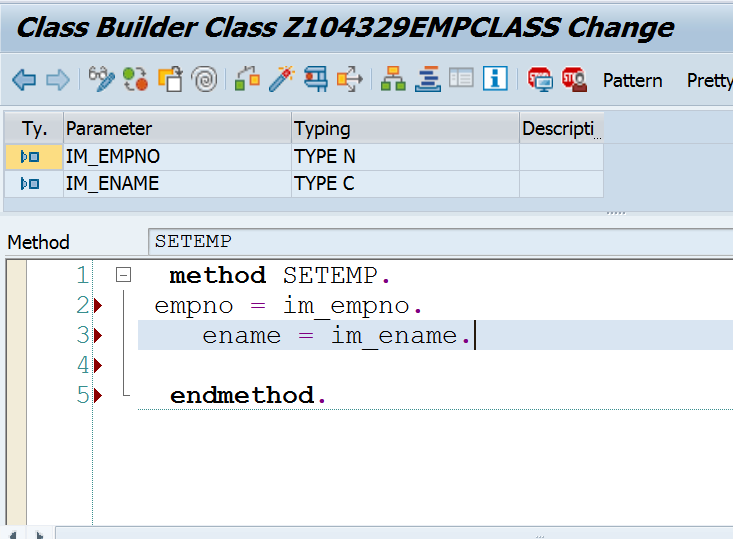








SAVE AND CLICK ON CODE BUTTON  AND CAN WRITE YOUR CODE



WAYS TO CALL FUNCTIONS

These are the possibilities of an inline method call.

If you are calling so called functional method which has only IMPORTING parameters and optionally one RETURN parameter you can call it like this.

CLASS lcl\_test DEFINITION.

PUBLIC SECTION.

CLASS-METHODS:

func\_meth

IMPORTING

i\_param TYPE i

RETURNING

VALUE(r\_res) TYPE char1.

ENDCLASS.

l\_res = lcl\_test=>func\_meth( 1 ).

\* you could also call it like this

l\_res = lcl\_test=>func\_meth( i\_param = 1 ).

\* also this variant is possible

l\_res = lcl\_test=>func\_meth( EXPORTING i\_param = 1 ).

\* the traditional CALL METHOD syntax would be like this

CALL METHOD lcl\_test=>func\_meth

EXPORTING

i\_param = 1

RECEIVING

r\_res = l\_res.

If there is more than one IMPORTING parameter you have to specify names of the parameters.

CLASS lcl\_test DEFINITION.

PUBLIC SECTION.

CLASS-METHODS:

func\_meth

IMPORTING

i\_param1 TYPE i

i\_param2 TYPE i

RETURNING

VALUE(r\_res) TYPE char1.

ENDCLASS.

l\_res = lcl\_test=>func\_meth(

i\_param1 = 1

i\_param2 = 2

).

If there are EXPORTING or CHANGING parameters in the method then an inline call is still possible but the parameter categories have to be explicitly specified.

CLASS lcl\_test DEFINITION.

PUBLIC SECTION.

CLASS-METHODS:

func\_meth

IMPORTING

i\_param TYPE i

EXPORTING

e\_param TYPE c

CHANGING

c\_param TYPE n.

ENDCLASS.

lcl\_test=>func\_meth(

EXPORTING

i\_param = 1

IMPORTING

e\_param = l\_param

CHANGING

c\_param = l\_paramc

).