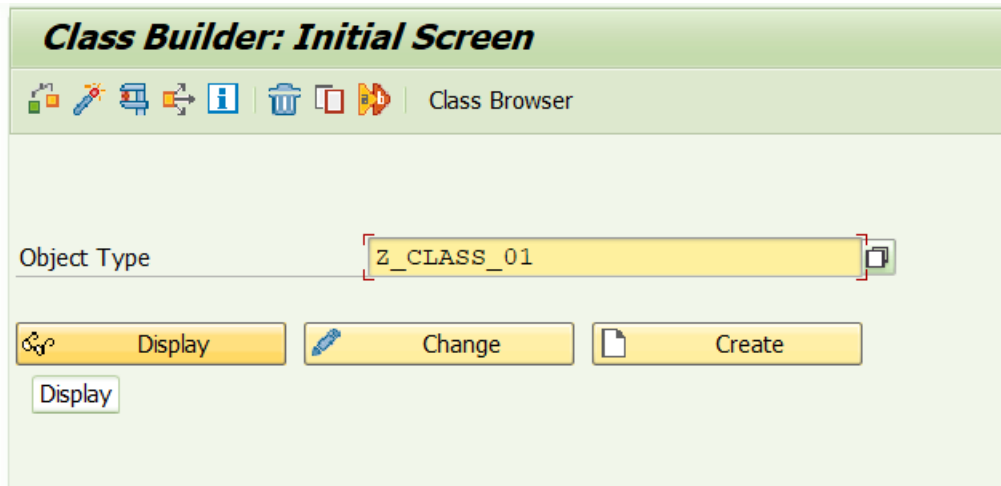


Requirement: To display the sales order details when a sales document number is inserted in the given field using a class.

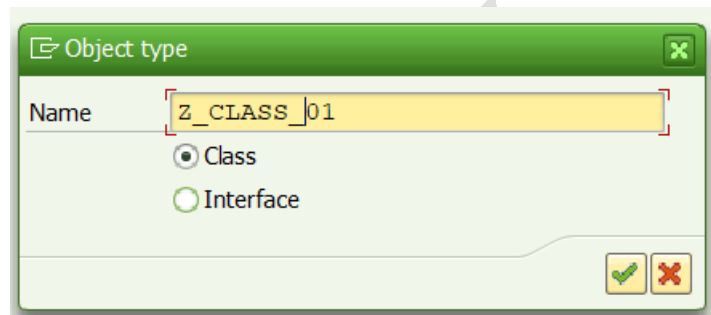
Solution:

Step 1.1: Create a class through class builder t-code (SE24).



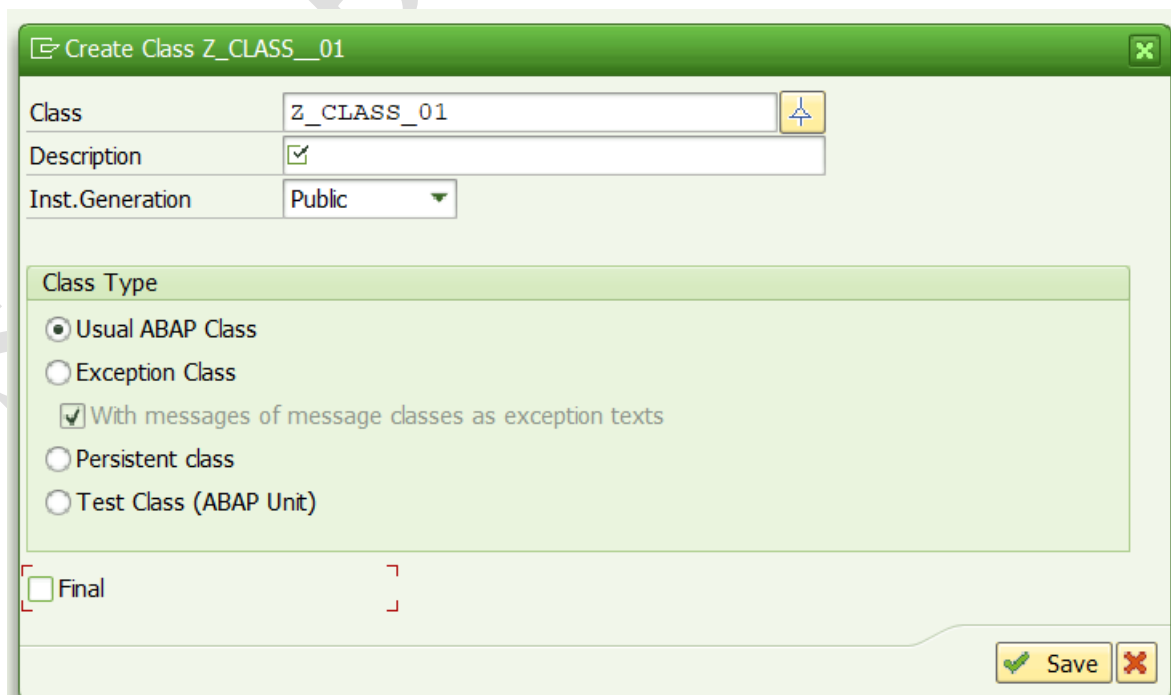
The screenshot shows the 'Class Builder: Initial Screen' in SAP. At the top, there is a title bar and a toolbar with icons for creating, editing, and deleting classes. Below the toolbar, the 'Object Type' field is set to 'Z_CLASS_01'. There are three main buttons: 'Display', 'Change', and 'Create'. A smaller 'Display' button is also visible below the 'Display' button.

Step 1.2: Choose class from the following.



The screenshot shows the 'Object type' dialog box. The 'Name' field is set to 'Z_CLASS_01'. There are two radio buttons: 'Class' (selected) and 'Interface'. There are 'OK' and 'Cancel' buttons at the bottom right.

Step 1.3: Choose class types as Usual ABAP Class and uncheck final .



The screenshot shows the 'Create Class Z_CLASS_01' dialog box. The 'Class' field is set to 'Z_CLASS_01'. The 'Description' field has a checkmark. The 'Inst. Generation' dropdown is set to 'Public'. Under the 'Class Type' section, 'Usual ABAP Class' is selected. There are checkboxes for 'With messages of message classes as exception texts' (checked), 'Persistent class', and 'Test Class (ABAP Unit)'. At the bottom, there is a 'Final' checkbox which is unchecked. There are 'Save' and 'Cancel' buttons at the bottom right.

Step 1.4: Save it as local.

Step 1.5: Create a method, choose level as static/instance, select visibility type and write a meaningful description.

Method	Level	Visibility	M...	Description
DISPLAY	Static Method	Public		To display sales order data (parameter)

Step 1.6: Select importing and exporting parameters.

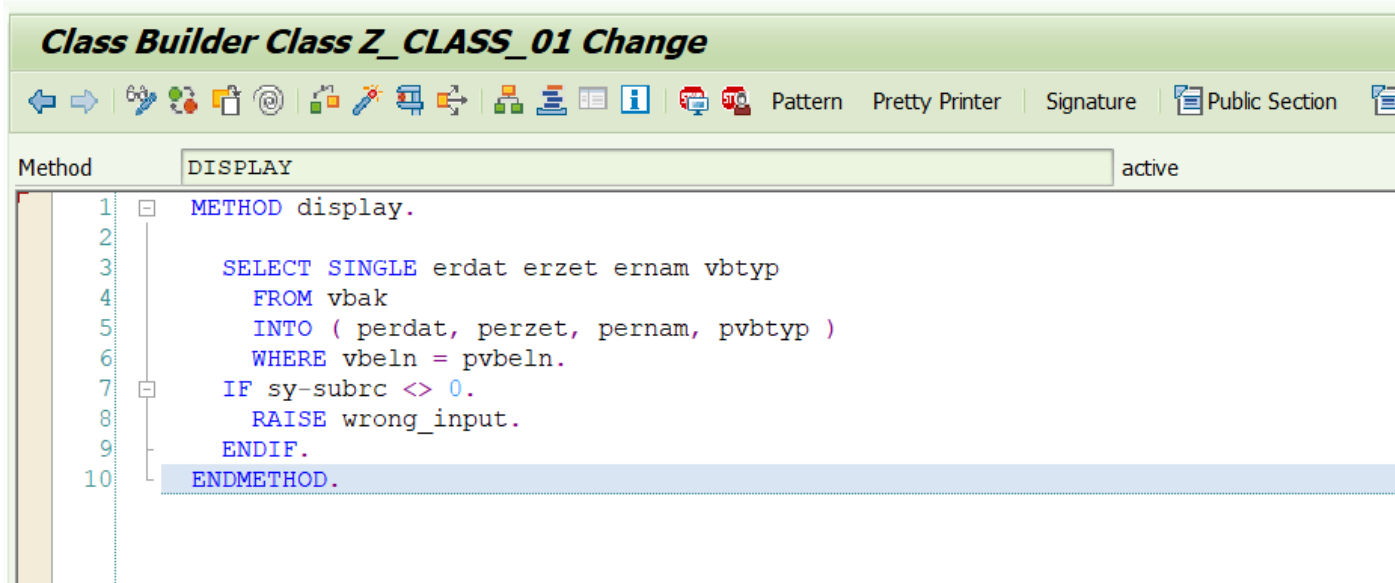
Parameter	Type	P...	O...	Typing Method	Associated Type	Default Value	Description
PVBELN	Importing	<input type="checkbox"/>	<input type="checkbox"/>	Type	VBELN_VA		Sales Document
PERDAT	Exporting	<input type="checkbox"/>	<input type="checkbox"/>	Type	ERDAT		Record Created On
PERZET	Exporting	<input type="checkbox"/>	<input type="checkbox"/>	Type	ERZET		Entry time
PERNAM	Exporting	<input type="checkbox"/>	<input type="checkbox"/>	Type	ERNAM		Name of Person Responsible for Creating the Object
PVBTYPL	Exporting	<input type="checkbox"/>	<input type="checkbox"/>	Type	VBTYPL		SD Document Category

Step 1.7: Select exceptions button and write required number of exceptions.

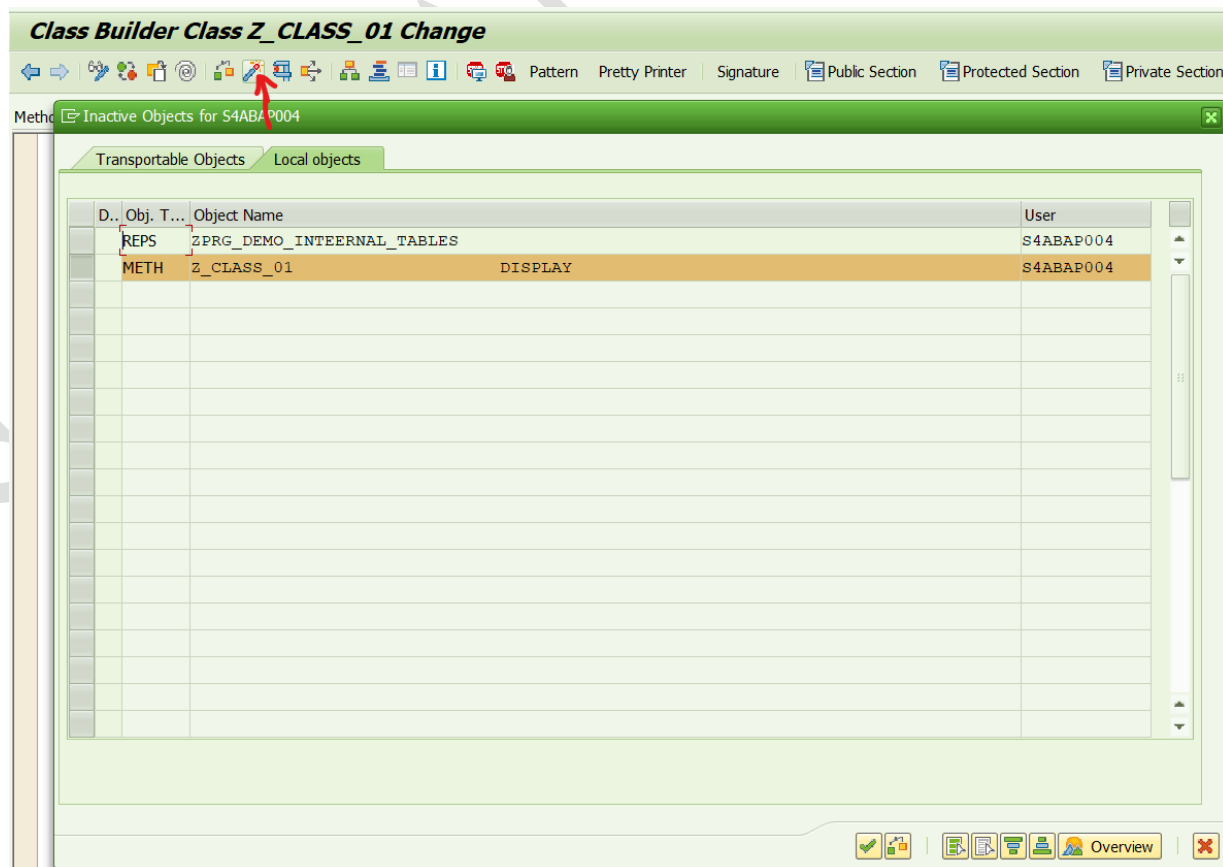
Exception	Description
WRONG_INPUT	Wrong Input

Step 1.8: Write a source code to the method for class.

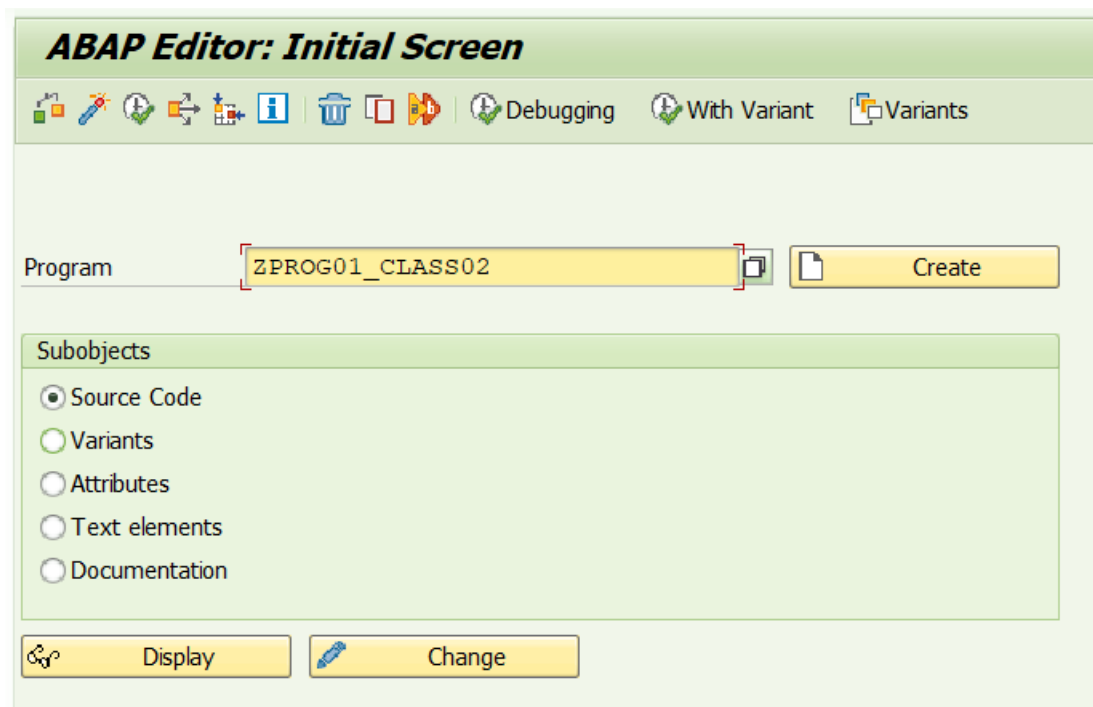
```
SELECT SINGLE erdat erzet ernam vbtyp
FROM vbak
INTO ( perdat, perzet, pernam, pvbtyp )
WHERE vbeln = pvbeln.
IF sy-subrc <> 0.
  RAISE wrong_input.
ENDIF.
```



Step 1.9: Check if there is any error, then save and activate.



Step 2.1: Create a Report through ABAP Editor T-code (SE38).



The screenshot shows the 'ABAP Editor: Initial Screen'. At the top, there is a title bar and a toolbar with icons for creating, saving, and other functions. Below the toolbar, the 'Program' field is set to 'ZPROG01_CLASS02'. To the right of this field is a 'Create' button. Below the 'Program' field is a 'Subobjects' section with radio buttons for 'Source Code' (selected), 'Variants', 'Attributes', 'Text elements', and 'Documentation'. At the bottom, there are 'Display' and 'Change' buttons.

ABAP Editor: Initial Screen

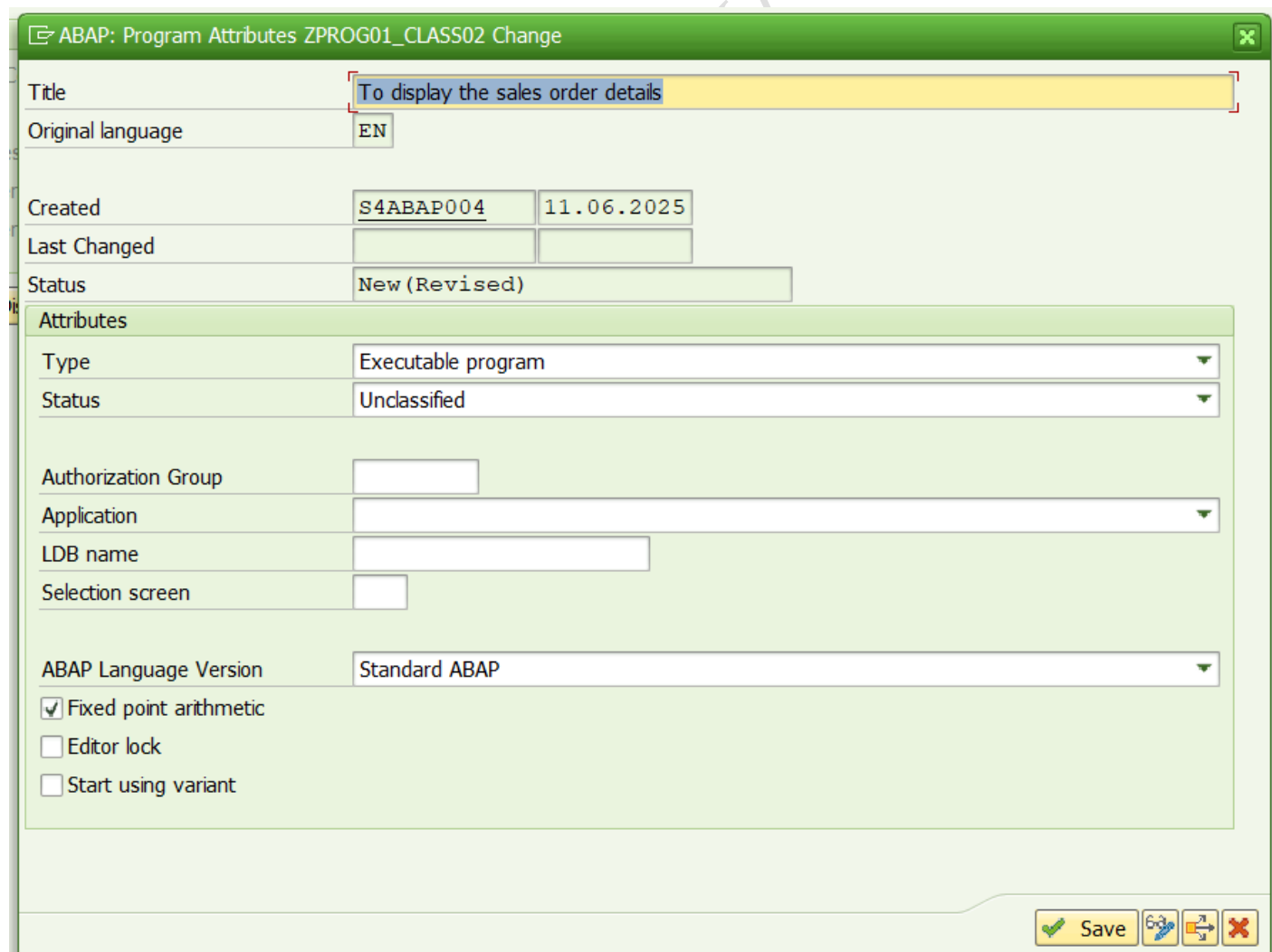
Program: ZPROG01_CLASS02 [Create]

Subobjects:

- ☒ Source Code
- ☐ Variants
- ☐ Attributes
- ☐ Text elements
- ☐ Documentation

[Display] [Change]

Step 2.2: Give a meaningful name and choose type as executable program. Select save button.



The screenshot shows the 'ABAP: Program Attributes ZPROG01_CLASS02 Change' dialog box. It contains various fields for program attributes. The 'Title' field is highlighted with a red box and contains the text 'To display the sales order details'. The 'Original language' is set to 'EN'. The 'Created' field shows 'S4ABAP004' and '11.06.2025'. The 'Last Changed' field is empty. The 'Status' is set to 'New (Revised)'. The 'Attributes' section has a 'Type' dropdown set to 'Executable program' and a 'Status' dropdown set to 'Unclassified'. Other fields include 'Authorization Group', 'Application', 'LDB name', 'Selection screen', and 'ABAP Language Version' (set to 'Standard ABAP'). There are checkboxes for 'Fixed point arithmetic' (checked), 'Editor lock', and 'Start using variant'. At the bottom right, there is a 'Save' button and other icons.

ABAP: Program Attributes ZPROG01_CLASS02 Change

Title: To display the sales order details

Original language: EN

Created: S4ABAP004 11.06.2025

Last Changed:

Status: New (Revised)

Attributes:

Type: Executable program

Status: Unclassified

Authorization Group:

Application:

LDB name:

Selection screen:

ABAP Language Version: Standard ABAP

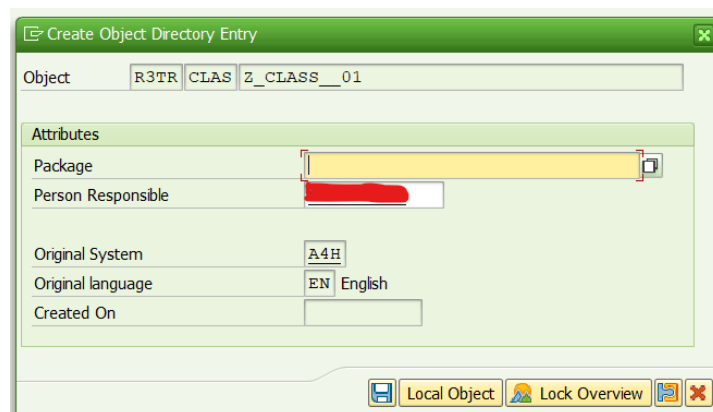
☒ Fixed point arithmetic

☐ Editor lock

☐ Start using variant

[Save] [Go] [Print] [Close]

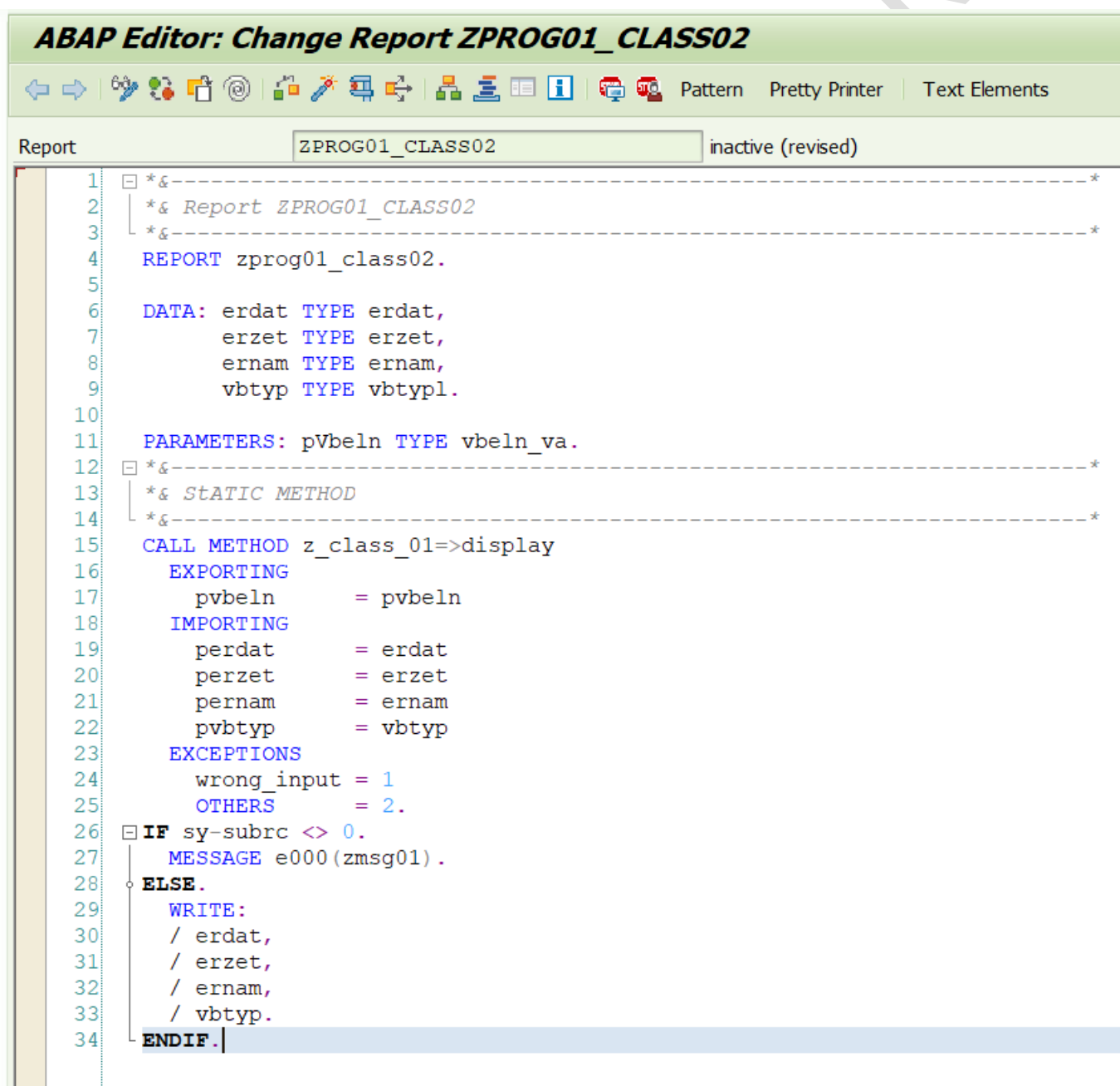
Step 2.3: Save it as local.



The 'Create Object Directory Entry' dialog box is shown. The 'Object' field contains 'R3TR CLAS Z_CLASS__01'. The 'Attributes' section includes fields for 'Package' (highlighted in yellow), 'Person Responsible' (redacted), 'Original System' (A4H), 'Original language' (EN English), and 'Created On'. At the bottom, there are buttons for 'Local Object', 'Lock Overview', and a close button.

Step 2.4: Write a code to call the function to display the sales document number.

STATIC METHOD



The screenshot shows the ABAP Editor for report 'ZPROG01_CLASS02', which is marked as 'inactive (revised)'. The editor contains the following code:

```
1  *-----*
2  *& Report ZPROG01_CLASS02
3  *-----*
4  REPORT zprog01_class02.
5
6  DATA: erdat TYPE erdat,
7         erzet TYPE erzet,
8         ernam TYPE ernam,
9         vbtyp TYPE vbtyp1.
10
11  PARAMETERS: pvbeln TYPE vbeln_va.
12  *-----*
13  *& STATIC METHOD
14  *-----*
15  CALL METHOD z_class_01=>display
16    EXPORTING
17      pvbeln      = pvbeln
18    IMPORTING
19      perdat      = erdat
20      perzet      = erzet
21      pernam      = ernam
22      pvbtyp      = vbtyp
23    EXCEPTIONS
24      wrong_input = 1
25      OTHERS      = 2.
26  IF sy-subrc <> 0.
27    MESSAGE e000(zmsg01).
28  ELSE.
29    WRITE:
30      / erdat,
31      / erzet,
32      / ernam,
33      / vbtyp.
34  ENDIF.
```

REPORT zprog01_class02.

DATA: erdat TYPE erdat,
 erzet TYPE erzet,
 ernam TYPE ernam,
 vbtyp TYPE vbtypl.

PARAMETERS: pVbeln TYPE vbeln_va.

&-----

*& *Static METHOD*

&-----

CALL METHOD z_class_01=>display

EXPORTING

 pvbeln = pvbeln

IMPORTING

 perdat = erdat

 perzet = erzet

 pernam = ernam

 pvbtyp = vbtyp

EXCEPTIONS

 wrong_input = 1

 OTHERS = 2.

IF sy-subrc <> 0.

 MESSAGE e000(zmsg01).

ELSE.

 WRITE:

 / erdat,

 / erzet,

 / ernam,

 / vbtyp.

ENDIF.

INSTANCE METHOD

```
REPORT zprog01_class02.
```

```
DATA: erdat TYPE erdat,  
      erzet TYPE erzet,  
      ernam TYPE ernam,  
      vbtyp TYPE vbtyp.
```

```
PARAMETERS: pVbeln TYPE vbeln_va.
```

```
*-----  
* Instance method  
*-----
```

```
*DATA: lo_object TYPE REF TO z_class_01.
```

```
*CREATE OBJECT lo_object.
```

```
*CALL METHOD lo_object->display1
```

```
* EXPORTING
```

```
*   pvbeln   = pvbeln
```

```
* IMPORTING
```

```
*   perdat   = erdat
```

```
*   perzet   = erzet
```

```
*   pernam   = ernam
```

```
*   pvbtyp   = vbtyp
```

```
* EXCEPTIONS
```

```
*   wrong_input = 1
```

```
*   OTHERS      = 2.
```

```
*IF sy-subrc <> 0.
```

```
* MESSAGE e000(zmsg01).
```

```
* ELSE.
```

```
* WRITE:
```

```
*   / erdat,
```

```
*   / erzet,
```

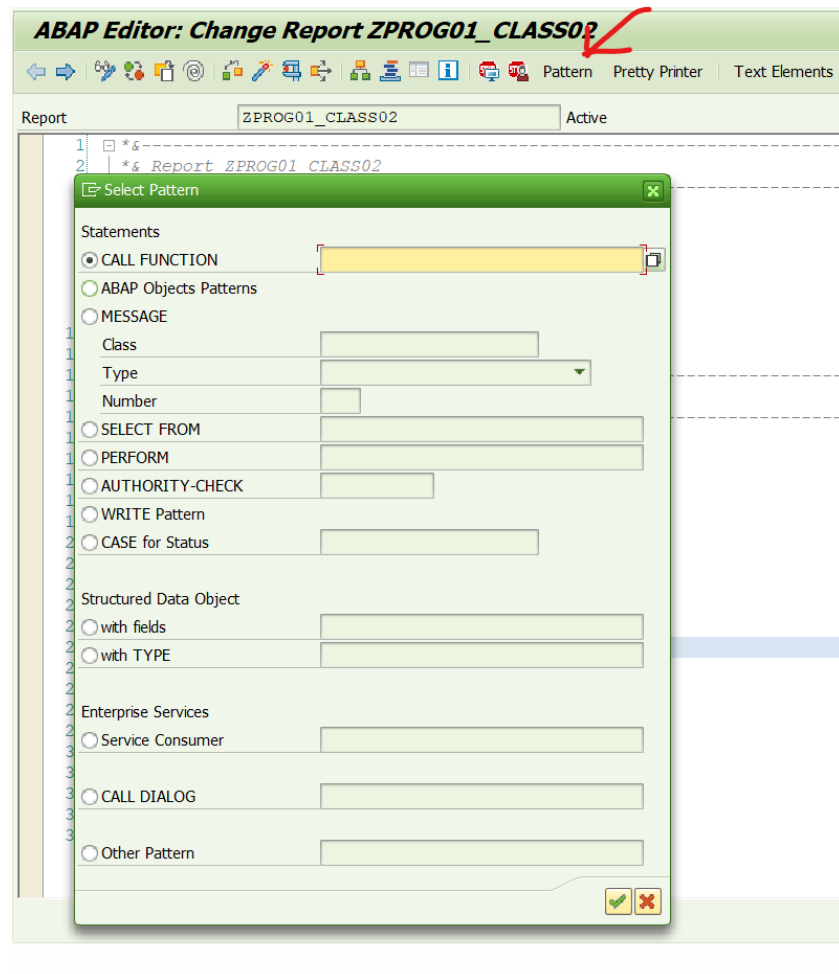
```
*   / ernam,
```

```
*   / vbtyp.
```

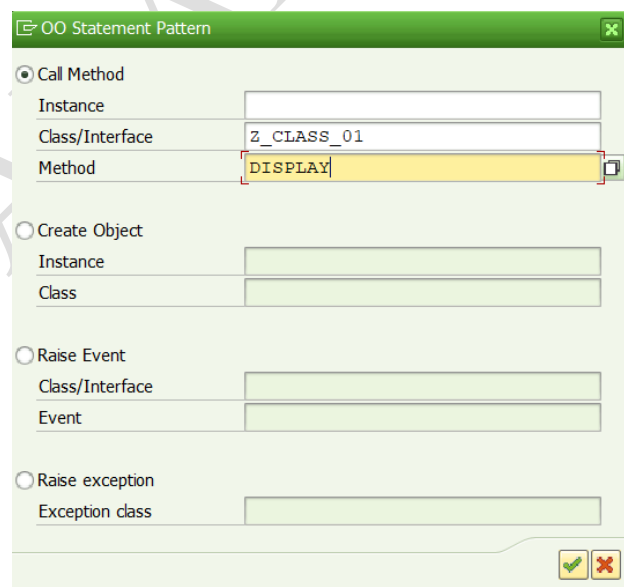
```
*ENDIF.
```

Step 2.5: Check if there is any error, then save and activate.

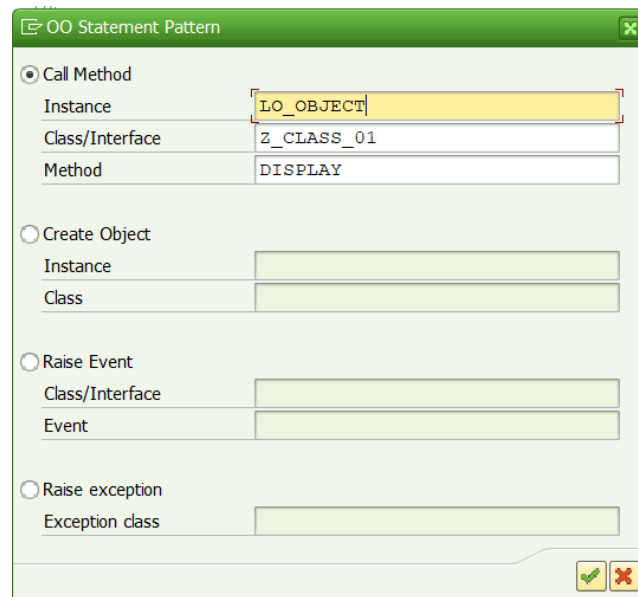
Note:- Call these class by pressing PATTERN button, click on the radio button ABAP OBJECT PATTERNS



FOR STATIC METHOD



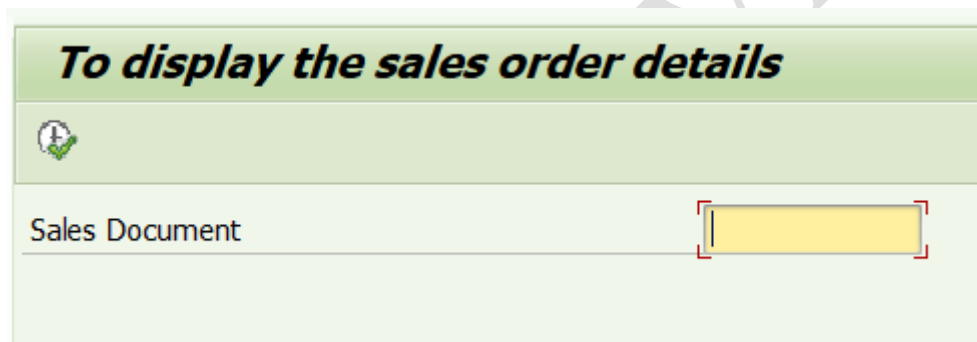
FOR INSTANCE METHOD




The dialog box titled "OO Statement Pattern" has a green header bar with a close button. It contains four radio button options: "Call Method" (selected), "Create Object", "Raise Event", and "Raise exception". Under "Call Method", there are three text fields: "Instance" with the value "LO_OBJECT", "Class/Interface" with the value "Z_CLASS_01", and "Method" with the value "DISPLAY". The other options have empty text fields. At the bottom right, there are green and red checkmark buttons.

<input checked="" type="radio"/> Call Method	Instance	LO_OBJECT
	Class/Interface	Z_CLASS_01
	Method	DISPLAY
<input type="radio"/> Create Object	Instance	
	Class	
<input type="radio"/> Raise Event	Class/Interface	
	Event	
<input type="radio"/> Raise exception	Exception class	

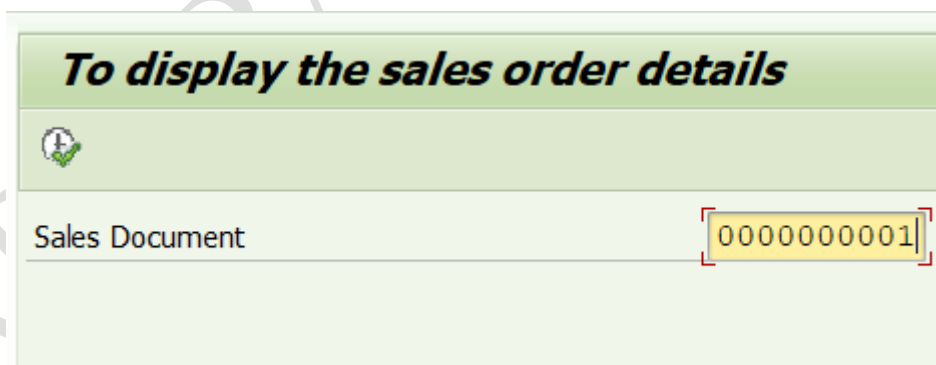
Step 2.6: Execute the code to check output.




The dialog box titled "To display the sales order details" has a green header bar. Below the header is a green bar with a green checkmark icon. The main area has a text field labeled "Sales Document" with a yellow highlight and a red border.

To display the sales order details	
	
Sales Document	

Inset Document Number



The dialog box titled "To display the sales order details" has a green header bar. Below the header is a green bar with a green checkmark icon. The main area has a text field labeled "Sales Document" with the value "0000000001" and a yellow highlight and red border.

To display the sales order details	
	
Sales Document	0000000001

OUTPUT

To display the sales order details

To display the sales order details

20.11.2024

04:22:49

ABAP10

C