

Function Module and Function Group

Include'un yarım kalan konusu;

```
INCLUDE zbk_test_00_top.  
INCLUDE zbk_test_00_frm.  
  
START-OF-SELECTION.  
PERFORM form1.  
PERFORM form2.
```

ABAP Editor: Change Include ZBK_TEST_00_FRM

← → | 60 | 🔄 | 📄 | 🕒 | 🔒 | 🛠️ | 🚚 | 📦 | 📊 | 📋 | ⓘ | 📄 | 🛑 | 🛑 Pattern Pretty Print

Include

ZBK_TEST_00_FRM

Active

```
1  | *&-----  
2  | *& Include      ZBK_TEST_00_FRM  
3  | *&-----  
4  
5  | FORM form1.  
6  |   DATA: lv_num2 TYPE int4.  
7  |   gv_num1 = gv_num1 + 1.  
8  | ENDFORM.  
9  
10  
11 | FORM form2.  
12 |   gv_num1 = gv_num1 + 1.  
13 |   WRITE: 'Form2 calisti.', gv_num1.  
14 |   "lv_num2 = lv_num2 + 1.  
15 | ENDFORM.
```

ABAP Editor: Change Include ZBK_TEST_00_TOP

Include Active

```
1  *&-----
2  *& Include ZBK_TEST_00_TOP
3  *&-----
4
5
6  DATA: gv_num1 TYPE int4.
```

Dark Mode Test

Dark Mode Test

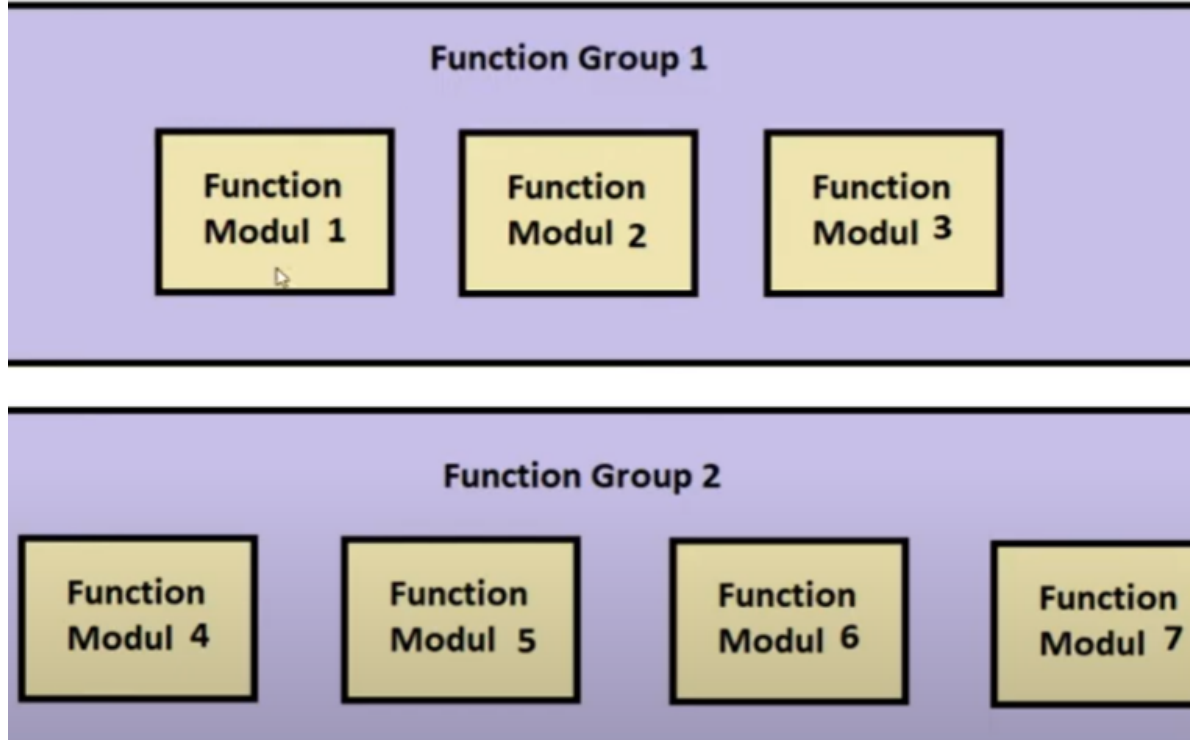
Form2 calisti. 2

Function Module and Function Group

- ☐ **FM ve FG İlişkisi**
- ☐ **Fonksiyon Grubu**
- ☐ **Fonksiyon Modülü**
- ☐ **Programda FM kullanımı**

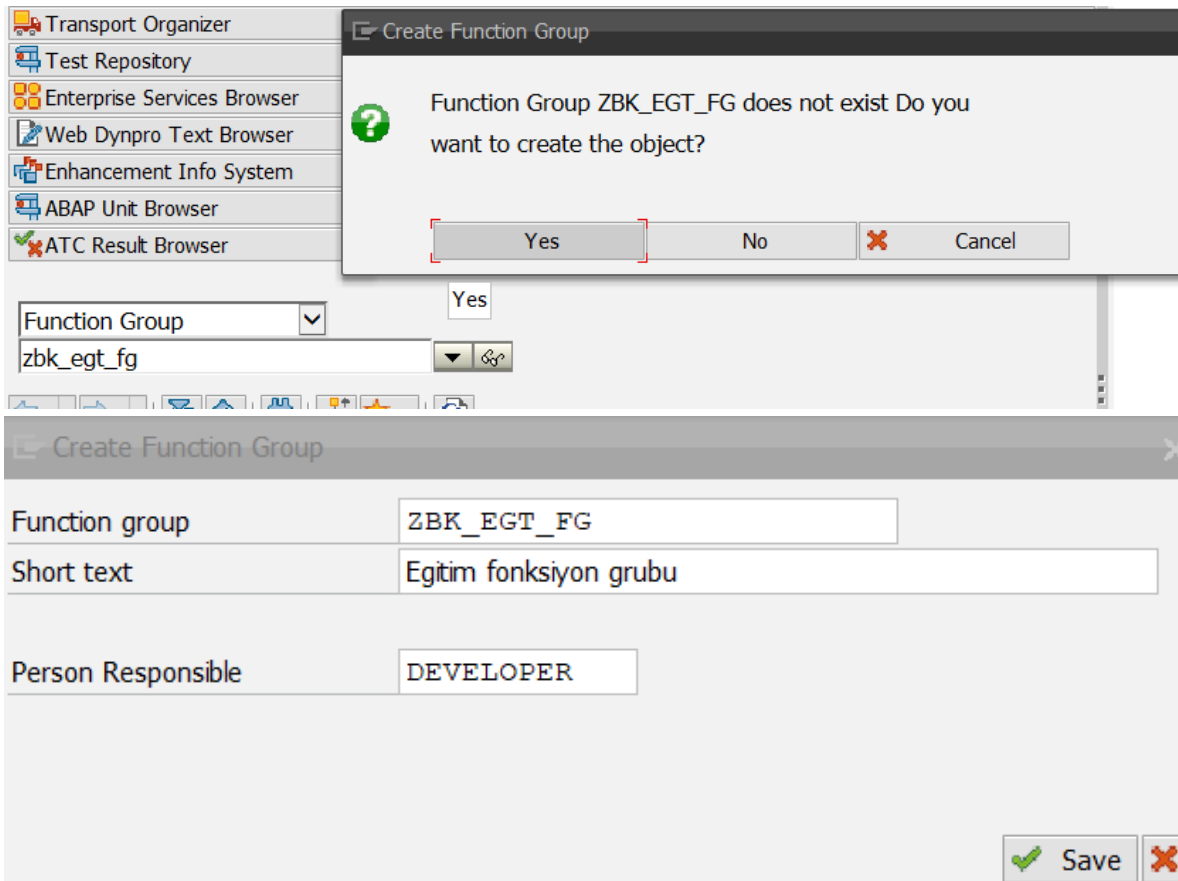
Fonksiyon modülü belirli bir amaca hizmet eden bir yapıyı bir fonksiyon altında topladığımız ve bunu her programda kullanabildiğimiz bir yapıdır, (!) bir kere yaz her yerde kullan.

Fonksiyon grubu ise belirli amaca hizmet eden fonksiyonların bir başlığın altında toplanmasıdır. Aslında bir amaca hizmet eden sürecin tüm fonksiyonları bir fonksiyon grubunun altında toplanması gibi düşünebiliriz.

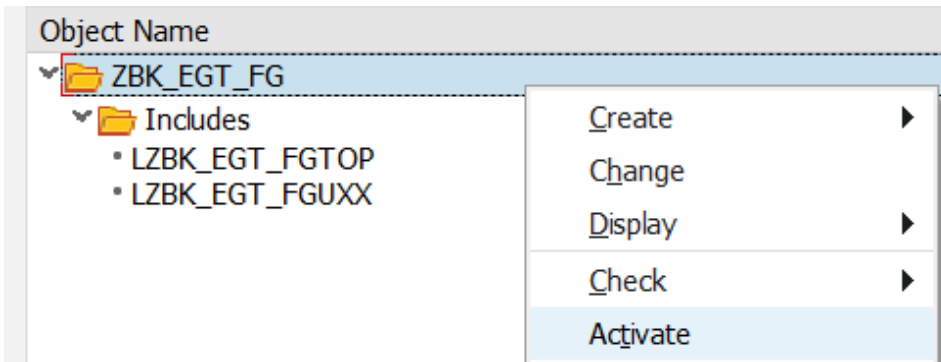


! Her fonksiyon bir fonksiyon grubunun altında olmak zorunda !

Programa girdiğimiz zaman yeni bir **TCode** olan **se80**'i kullanmakla işe başlıyoruz.






Grubun bir yapısı genellikle ilk oluşturulduğu zaman pasif durumda olur, tüm içeriği aktif etmek için şu şekilde ilerliyoruz;




(!) FONKSİYON MODULU OLUŞTURMANIN DA İŞLEM KODU SE37'DİR.

Function Module

 Display  Change  Create

Function Builder: Change ZBK_EGITIM_FONKS

 Pattern Pretty Printer Function Module Documentation Text elements

Function module Inactive

Attributes Import Export Changing Tables Exceptions Source code

Parameter Name	Typing	Associated Type	Default value	Optional	Pass by Value	Short text	Lon...
iv_num1	Type	int4	10	<input type="checkbox"/>	<input type="checkbox"/>		

İki adet değişken parametresi oluşturmakla işe başlıyoruz, atamaları yapıyoruz. Burada Optional ve Pass by Value kısımları var. Optional'a tıklandığında opsiyonel demek oluyor, bu parametreyi göndermek zorunda değiliz anlamına geliyor. Diğeri ise Pass by Value olan, içeride değiştirilebilsin veya değiştirilmesin anlamına geliyor. Pass by Value en basit örneği;

```
" iv_num1 = 10. "Pass by Value sayesinde degistirebiliyoruz bu sayiyi.  
ev_result = iv_num1 / iv_num2.
```

Source Code içerisinde F8 bastığımız zaman bu ekran karşımıza gelir;

Test Function Module: Initial Screen

 Debugging  Test data directory

Test for function group ZBK_EGT_FG
Function module ZBK_EGITIM_FONKS
Uppercase/Lowercase ☐

Import parameters	Value
IV_NUM1	10
IV_NUM2	5

Changing parameters	Value
CV_MES	[Selamlar, mesaj]

Test Function Module: Result Screen



Test for function group ZBK_EGT_FG
Function module ZBK_EGITIM_FONKS
Uppercase/Lowercase ☐

Runtime: 18 Microseconds

Import parameters	Value
IV_NUM1	10
IV_NUM2	5

Changing parameters	Value
CV_MES Result:	SELAMLAR, MESAJ Degistirilmis Mesaj

Export parameters	Value
EV_RESULT	2

Yine execute ettiğimiz zaman ise 10'u 5'e bolup değerini gösterdi, ardından elle default girilen Selamlar, Mesaj içeriği ise kod içerisinde değiştirmiş olduğumuz 'Degistirilmis Mesaj' statusune gelerek son halini aldı.

Sırada sıfıra bolme hatası konusu bulunmakta, bunu yapabilmek içinde Exceptions kısmını kullanmamız gerekiyor.

Function Builder: Change ZBK_EGITIM_FONKS

Function module ZBK_EGITIM_FONKS Inactive

Attributes Import Export Changing Tables Exceptions Source code

Classes

Exception	Short text
DIVIDED_BY_ZERO	Cannot divide by zero

Exceptions içerisinde tanımlamayı yaptıktan sonra Source Code'a gelerek ilerliyoruz ve görülen kod blogunu yazıp ilerliyoruz. Burada önemli olan kısım INITIAL ve RAISE konusudur. INITIAL hiçbir parametre girilmeyen kontrollerde işimize yarar, 0 dahil görüldüğü gibi. RAISE ise hata mesajı almak yerine gösterilecek exception'u kod içerisinde tanımladığımız kısımdır.

Function module ZBK_EGITIM_FONKS Active

Attributes Import Export Changing Tables Exceptions Source code

```
1 FUNCTION zbk_egitim_fonks.
2
3 """Local Interface:
4 """
5     IMPORTING
6         VALUE(iv_num1) TYPE INT4 DEFAULT 10
7         VALUE(iv_num2) TYPE INT4 DEFAULT 2
8     EXPORTING
9         REFERENCE(ev_result) TYPE INT4
10    CHANGING
11        REFERENCE(cv_mes) TYPE CHAR20
12    EXCEPTIONS
13        DIVIDED_BY_ZERO
14
15
16    "IF iv_num2 EQ 0.
17    IF iv_num2 IS INITIAL.
18        RAISE divided_by_zero.
19    ENDIF.
20    " iv_num1 = 10. "Pass by Value sayesinde degistirebiliyoruz bu sayiyi.
21    ev_result = iv_num1 / iv_num2.
22
23    cv_mes = 'Degistirilmis Mesaj'.
24
25 ENDFUNCTION.
```

Test Function Module: Result Screen



Test for function group ZBK_EGT_FG
Function module ZBK_EGITIM_FONKS
Uppercase/Lowercase ☐

Runtime: 4 Microseconds

Exception DIVIDED_BY_ZERO

Import parameters	Value
IV_NUM1	10
IV_NUM2	0

Changing parameters	Value
CV_MES	GONDERILEN MESAJ
Result:	GONDERILEN MESAJ

Export parameters	Value
EV_RESULT	0

Ayrı bir konu karşımıza çıkmış bulunuyor, eğer Optional'i tikli yapmazsak bizden her zaman bir deger isteyecek, ornegin şu anda çıkan messages bloğu gibi;

Test Function Module: Initial Screen

Debugging Test data directory

Test for function group ZBK_EGT_FG
Function module ZBK_EGITIM_FONKS
Uppercase/Lowercase ☐

Import parameters	Value
IV_NUM1	10
IV_NUM2	2

Changing parameters	Value
CV_MES	

Enter Parameter Value

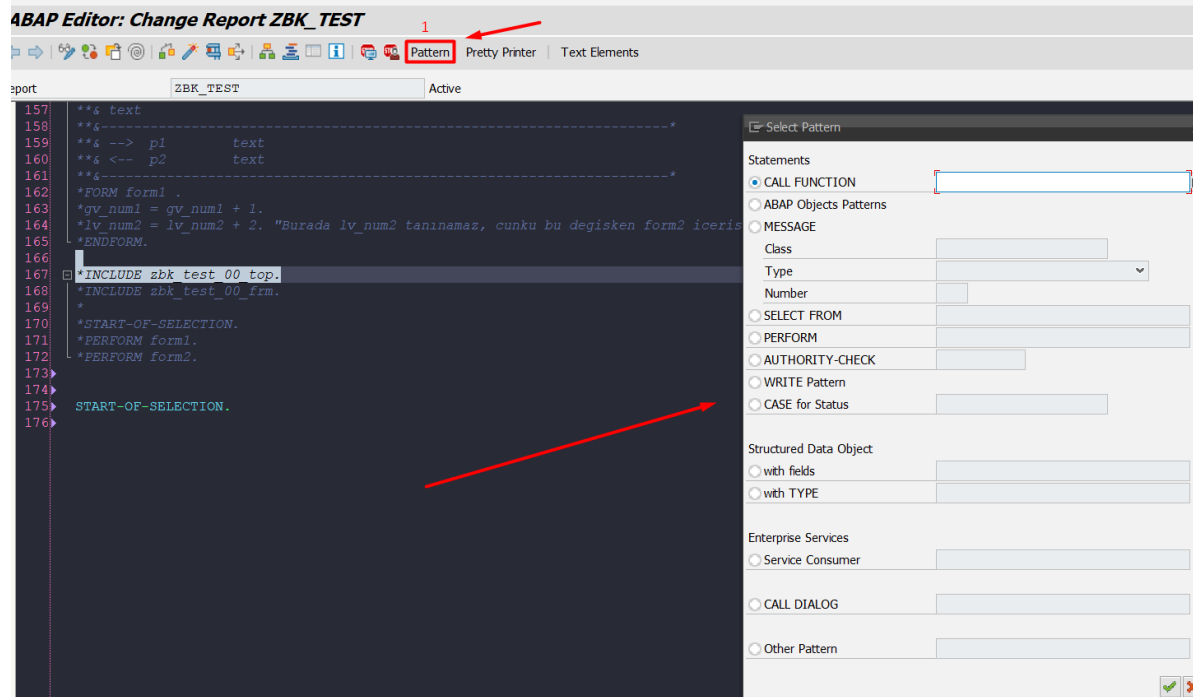
The input parameter CV_MES contains no value. Specify a value:

New Value:

< >

Copy

Ve şimdi bu yaptığımız fonksiyonu ana kod içerisinde çağırmanız gerekiyor.



```
DATA: gv_num1    TYPE int4,
      gv_num2    TYPE int4,
      gv_result  TYPE int4,
      gv_mes     TYPE char20.

CALL FUNCTION 'ZBK_EGITIM_FONKS'
  EXPORTING
    iv_num1      = gv_num1
    iv_num2      = gv_num2
  IMPORTING
    ev_result    = gv_result
  CHANGING
    cv_mes       = gv_mes
  EXCEPTIONS
    divided_by_zero = 1
    OTHERS         = 2.

IF sy-subrc <> 0.
  * Implement suitable error handling here
ENDIF.
```

Eğer ki hata yoksa bize 0 dönüyor, burada ki sy-subrc hata konularıyla ilgilidir.

```
DATA: gv_num1    TYPE int4,  
      gv_num2    TYPE int4,  
      gv_result  TYPE int4,  
      gv_mes     TYPE char20.
```

```
START-OF-SELECTION.
```

```
gv_num1 = 20.  
gv_num2 = 0.  
gv_mes = 'Ilk mesaj.'.
```

```
CALL FUNCTION 'ZBK_EGITIM_FONKS'  
  EXPORTING  
    iv_num1      = gv_num1  
    iv_num2      = gv_num2  
  IMPORTING  
    ev_result    = gv_result  
  CHANGING  
    cv_mes       = gv_mes  
  EXCEPTIONS  
    divided_by_zero = 1  
    OTHERS         = 2.
```

```
IF sy-subrc EQ 0.  
  * Implement suitable error handling here  
  WRITE: / 'Sonuc: ' , gv_result.  
  WRITE: / 'Mesaj ' , gv_mes.  
ELSEIF sy-subrc EQ 1.  
  WRITE: / ' 0 a bolunemez. ' .  
ENDIF.
```

Dark Mode Test

Dark Mode Test

Sonuc: 4
Mesaj Degistirilmis Mesaj

Dark Mode Test

Dark Mode Test

0 a bolunemez.