**ABAP’ ta Modülarizasyon**

Program kodlarını farklı birimlere ayırarak düzenli ve anlaşır hale getirmek için farklı nesneler kullanılır. Modülarizasyon aşağıdaki nesneler kullanılarak sağlanabilir.

Include ProgramlarFonksiyon ModüllerAltyordamlarYöntemlerMakrolar

**Include**

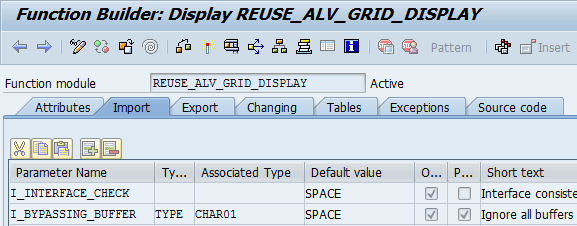
Tek başlarına çalıştırılabilir olmayan kod bloklarıdır. Programa dâhil etmek için **INCLUDE** anahtar kelimesi kullanılır. Tek başına çalıştırılabilir program değillerdir. Oluştururken program tipi “I” olarak seçilir.

INCLUDE include\_ismi

**Fonksiyon Modüller (Function Modules)**

Fonksiyon gruplarına bağlı yeniden kullanılabilir prosedürlerdir. Function Builder kullanılarak oluşturulurlar. Function Builder’ da fonksiyona içerisine ve dışına aktarılan aktarılan parametreler tanımlanır. Kaynak kodu (Source code) kısmına FUNCTION ve ENDFUNCTION ifadeleri arasına kod bloğu yazılır. Gerekirse fonksiyon grubuna bağlı inlude’ larda oluşturulabilir.

Function Builder’ a SE37 işlem kodu ile erişilebilir.



**Fonksiyon Modul Arayüzü**

Fonksiyonda kullanılan parametrelerin tanımlandığı arayüzdür.

IMPORTING: Fonksiyon içerisine aktarılan parametrelerdir.

EXPORTING: Fonksiyondan dışarı aktarılan parametrelerdir.

CHANGING: Fonksiyon içerisine gönderilen ve gönderilen parametre üzerinde değişiklik yapılıp tekrar fonksiyon dışarısına aktarılabilen parametrelerdir.

TABLES: İnternal tablo aktarmak için kullanılır. Artık kullanılmayan parametrelerdir. Sadece geriye dönük uyum için kullanımı vardır.

ABAP kodundan fonksiyon çağrılırken IMPORTING, CHANGING ve TABLES parametreleri isteğe bağlı olarak function builder’ dan seçilebilir. EXPORTING parametreleri daima isteğe bağlıdır.

… [IMPORTING parameters]

[EXPORTING parameters]

[CHANGING parameters]

[TABLES table\_parameters]

[{RAISING exc1|RESUMABLE(exc1) exc2|RESUMABLE(exc2) …}

|{EXCEPTIONS exc1 exc2 …}]

**Subroutines**

Bir program içerisinde modülarizasyonu sağlamak için kullanılırlar. PERFORM kullanılarak çağrılırlar. Subroutine’ ler artık kullanılmamalıdır. Bunların yerine yerel sınıflardaki metodlar kullanılmalıdır. Başka programlar tarafından da çağrılabilirler. FORM ve ENDFORM bildirimleri arasına yazılırlar. USING ve CHANGING kullanılarak parametre aktarımı yapılabilir.

DATA: num1 TYPE i,

      num2 TYPE i,

mean TYPE p DECIMALS 2.

START-OF-SELECTION.

  num1 = 4.

  num2 = 5.

PERFORM mean\_value USING num1 num2

                   CHANGING mean.

FORM mean\_value USING p1 TYPE i p2 TYPE i

                CHANGING result TYPE numeric.

  result = ( p1 + p2 ) / 2.

ENDFORM.

**Yöntemler (Methods)**

METHOD ve ENDMETHOD ifadeleri arasına yazılan kod bloklarıdır. CALL METHOD ile çağrılırlar.

**MAKROLAR (Macros)**

ABAP programlarında kaynak farklı birimlere ayırmak için kullanılır. DEFINE ve END-OF-DEFINATION ifadeleri arasına yazılırlar.