

Programlama Dillerinin Prensipleri

ALTPROGRAMLAR

Hedef

- Bu bölümde, programlarda sıkça kullanılan işlemlerin bir araya gruplanması ile oluşturulan altprogramlar incelenecektir. Altprogramlar ile, bir programda birden çok kez gereksinim duyulan kod bölümleri programda bir kez yer alır ve gerekince yeniden kullanılır.
- Bölüm tamamlandığında,
- Altprogramların Genel Özellikleri,
- Parametre Aktarım Yöntemleri,
- Etkinlik Kayıtları ve
- Çeşitli Programlama Dillerinde Uygulanan Yaklaşımlar
- öğrenilmiş olacaktır.

ALTPROGRAMLAR

- Programlama dillerinde her bir modülün (alt parça) bağımsız olarak ele alınmasını sağlayan ve alt program adı verilen birimler tanımlanır.
- Altprogramlar büyük ve karmaşık problemin çözümünde yapısal programlamanın etkin kullanımı için gerekli temel taşlarındandır. Altprogramlar bir kere yazılır ve program içerisinde bir yada daha çok yerde çağrılarak kullanılır.
- Altprogramlar ile işlev soyutlaması amaçlanır.

Neden Altprogram

- Program kodlarının gereksiz yere uzaması önlenir.
- Programın tasarlanmasını kolaylaştırır ve okunulabilirliğini artırır.
- Büyük bir programın, daha küçük ve yazılması daha kolay fonksiyonel parçalara bölünür.
- Ekip çalışması
- Altprogramlar birden çok uygulama arasında paylaşılabilir.

Altprogramların Genel Yapısı

Program kodu Altprogram çağrım deyimi Program kodu	
Altprogram Başlığı	Altprogram ismini, parametreler listesini (formal parametreler) ve varsa altprogramın döndürdüğü değer tipini bildirir.
Lokal değişkenler Deyimler	Altprogramda geçerli değişkenler Altprogramda tanımlı deyimler
Altprogram sonu Program kodu	Altprogramın çalışması bittikten sonra kontrol çağrıldığı noktaya geçer.
call prog1 (a,b,c)	call:çağrım deyimi prog1:altprogram ismi a,b,c: gerçek parametre

altprogram ismi
resmi parametreler
procedure **ornek** (a: integer, b: real)

- Bir altprogram bir fonksiyon ise, parametre profiline ek olarak fonksiyonun sonuç değerinin dönüş tipini de içerir ve bu bilgiye **altprogram protokolü** adı verilir.
- C programlama dilinde, altprogramların tanımlandığı deyimler yer alır. Bu deyimlerde altprogramların parametre tipleri belirtilir ve bu deyimlere **prototip** (*prototype*) adı verilir.
- Bir altprogram, bir **yordam** veya bir **fonksiyon** olabilir
- İki altprogram türü arasındaki fark, fonksiyonların çağıran program birimine bir sonuç değeri döndürmelerinin şart olmasıdır.

procedure ortalama (parametreler);

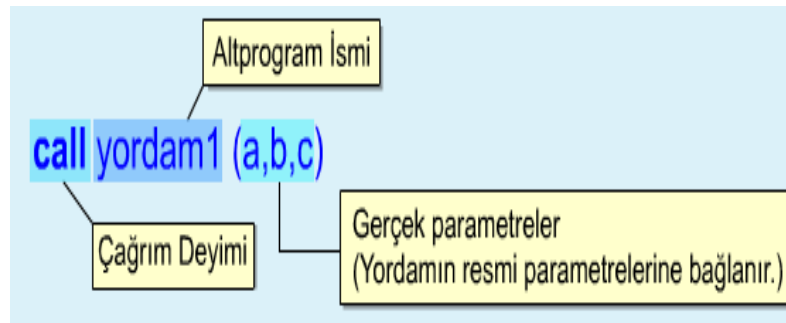
ortalama (parametreler);

Çeşitli dillerde Altprogramlar

- Yordamlar, Pascal ve Ada'da procedure anahtar kelimesi ile belirtilirler.
- C'de yordamların belirtilmesi için özel bir anahtar kelime kullanılmaz.
- Çoğu popüler *imperative* dilde hem fonksiyonlar hem de yordamlar bulunur.
- C ve C++'da sadece fonksiyonlar yer alırsa da, bu fonksiyonlar yordamların işlevlerini gerçekleştirebilirler.
- FORTRAN'da genel özellikleri diğer dillerdeki yordamlara benzer olmakla birlikte, yordamlara SUBROUTINE ismi verilir.

Parametreler

- **gerçek parametrelerin** ve **resmi parametrelerin** ilişkilendirilmesi için kullanılan iki yöntem:
- **konumsal** ve **anahtar kelime parametre** yöntemleridir.
- çağırım deyiminde, altprogramın ismine ek olarak, yordamın resmi parametreleriyle eşleştirilecek **gerçek (*actual*) parametreler** de yer alır.



- Bir yordam çağrımı gerçekleşince, etkin olan program birimindeki komutların çalışması durdurulur ve yordam etkin duruma geçer. Yordam gövdesindeki komutlar çalıştırıldıktan sonra etkinlik, yeniden çağrımın yapıldığı program birimine geçirilir ve yordam çağrım deyimini izleyen ilk deyim etkin olur

Konumsal Parametreler

- Eğer bir yordam çağrımında gerçek parametreler ile resmi parametreler arasındaki bağlama, parametrelerin çağrım deyimindeki ve yordam başlığındaki konumuna göre yapılıyorsa, bu parametrelere **konumsal** (*positional*) **parametre** adı verilir. Birçok programlama dilinde uygulanan bu yöntem, parametre sayısı az olduğu zaman kullanışlıdır.

- Aşağıda görüldüğü gibi bir çağrım deyimi ile etkin duruma geçen *ortalama* yordamında, *c* resmi parametresi, *a* gerçek parametresi ile; *d* resmi parametresi, *b* gerçek parametresi ile bağlanıyorsa, konumsal parametreler yaklaşımı uygulanmıştır.

```
call ortalama(a,b);
```

```
Procedure ortalama(c: integer; d:integer);
```

```
begin
```

```
    ...
```

```
end;
```

Resmi Parametre	Gerçek Parametre
c	a
d	b

Anahtar Kelime Parametre Yöntemi

- Gerçek ve resmi parametreler arasındaki bağlamayı konuma göre belirlemek yerine, her iki parametrenin de ismini belirterek gösterme, **anahtar kelime parametre** yöntemi olarak adlandırılır. Bu durumda çağırım deyiminde, hem resmi, hem de gerçek parametrelerin isimleri belirtilir.

Call (toplam→alttoplam, liste→dizi, gelir→kazanc);

Anahtar kelime parametre yöntemi, parametre sayısının çok olduğu durumda yararlı bir yöntemdir.

Ada ve FORTRAN 90'da hem anahtar kelime hem de konumsal parametreler yöntemi kullanılabilmektedir.

Fonksiyonlar

C

C'de geri dönüş değerinin tipi, fonksiyon adından önce yazılmalıdır.

Örnek

```
int topla(int say1, int say2);
```

Fonksiyon Adı

Geri Dönüş Tipi

Pascal

Pascal'da geri dönüş değerinin tipi tanımın en sonunda yazılmalıdır.

Örnek

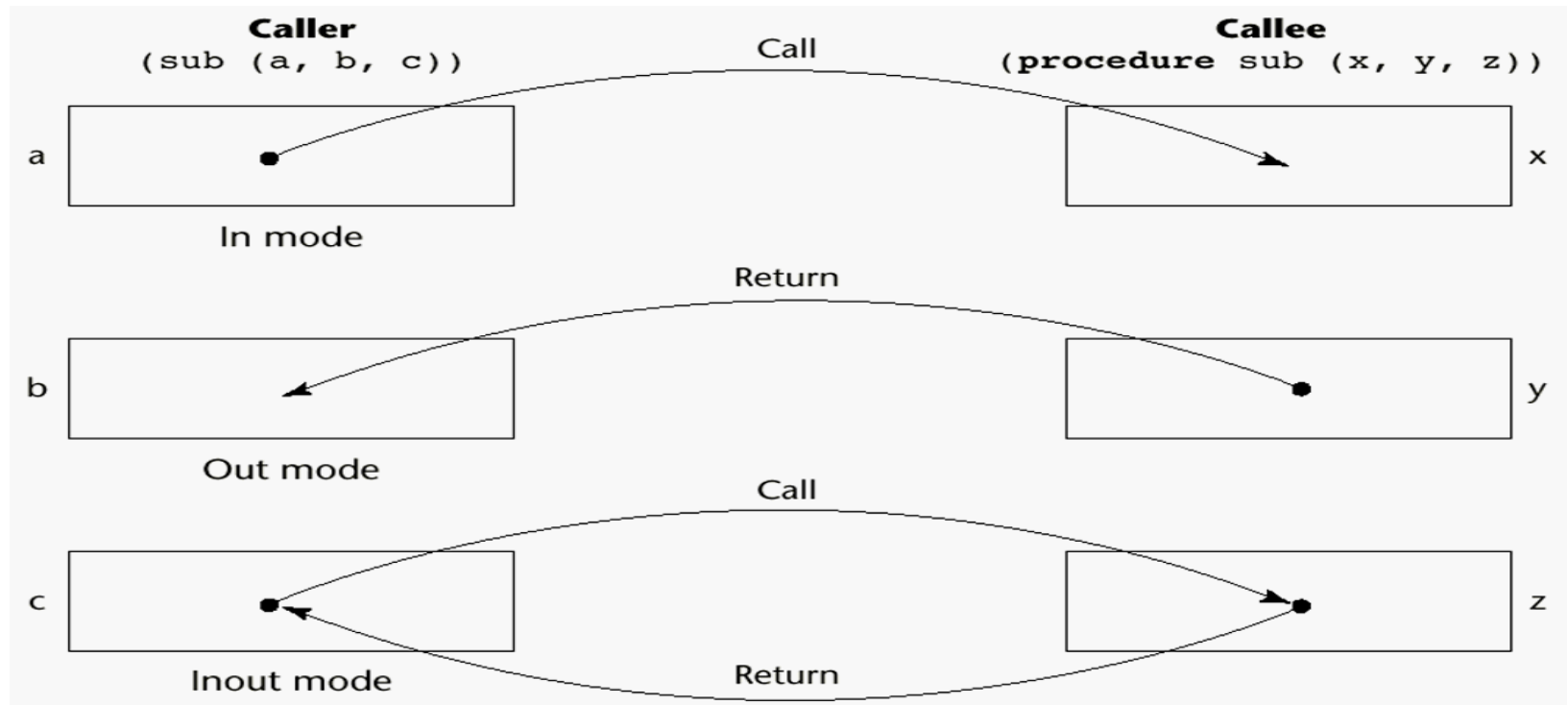
```
function kare( say3: integer): integer ;
```

Geri Dönüş Tipi

Parametre Aktarım Yöntemleri

- Bir yordam etkin duruma getirilirken, çağrım deyimindeki gerçek parametreler ile yordamın resmi parametrelerine farklı değerler aktarılabilir.

Gerçek parametreler yordamlara aktarılacak değerleri gösterdikleri için, değişken, sabit veya ifade olabilir. Resmi parametreler ise bu değerleri tutacak bellek yerlerini gösterdikleri için değişken olmak durumundadır.



Parametre Aktarım Yöntemleri

ÇAĞIRMA YÖNTEMLERİ

Değer ile Çağırma

Sonuç ile Çağırma

Değer ve Sonuç ile Çağırma

Gerçek değer fiziksel olarak kopyalanarak aktarılır.

Başvuru ile Çağırma

İsim ile Çağırma

Verinin erişim yolu aktarılır.

Parametre Aktarım Yöntemleri (Devam)

- Değer ile çağırma, sonuç ile çağırma, değer ve sonuç ile çağırma yöntemlerinde, gerçek ve formal parametreler arasındaki veri fiziksel olarak kopyalanarak aktarılmakta, başvuru ile çağırma yönteminde ise veri yerine verinin adresi aktarılmaktadır.

Değer ile Çağırma (*Call by Value*)

- Bu yöntemde formal parametre, gerçek parametrenin değeriyle ikklendikten sonra, altprograma yerel bir değişken olarak değerlendirilir.

```
void f(int n)
{   n++;}
int main() {
    int x = 2;
    f(x);
    cout << x; }
```

Sadece gerçek parametreden formal parametreye değer geçişi olduğu için en güvenilir parametre aktarım yöntemidir

Programın çıktısı= 2 olacaktır.

Sonuç ile Çağırma (*Call by Result*)

- Bu yöntemde çağırım deyimi ile altprograma bir değer aktarılmazken, gerçek bir parametreye karşı gelen formal parametrenin değeri, altprogram sonunda, denetim yeniden çağıran programa geçmeden önce, gerçek parametreyi gösteren değişkene aktarılır. (gerçek parametrenin değişken olmalı). Gerçek parametreye karşı gelen resmi parametre, altprogramın çalışması süresince yerel değişkendir.

procedure sub1(y: out Integer; z: out Integer) is

• ...;
 sub1(x, x);

Değer ve Sonuç ile Çağırma (*Call by Value Result*)

- Değer ile çağırma ve sonuç ile çağırma yöntemlerinin birleşimidir.
- Gerçek parametrenin değeri ile karşı gelen formal parametrenin değeri ilklendir ve sonra resmi parametre, altprogramın çalışması süresince yerel değişken gibi davranır ve altprogram sona erdiğinde formal parametrenin değeri gerçek parametreye aktarılır.

```
int x=0;
int main()
{
    f(x);
    .....
}
void f(int a) {
    x=3;
    a++;}
```

x'in son değeri değer-sonuç aktarımına göre 1 olacaktır.

Parametreler için birden çok bellek yeri gerekmesi ve değer kopyalama işlemlerinin zaman almaktadır.

Başvuru ile Çağırma (*Call by Reference*)

- Başvuru ile çağırma yöntemi de gerçek ve formal parametreler arasında iki yönlü veri aktarımı vardır.
- Altprograma verinin adresi aktarılır. Bu adres aracılığıyla altprogram, çağıran program ile aynı bellek yerine erişebilir ve gerçek parametre, çağıran program ve altprogram arasında ortak olarak kullanılır.

```
int x=0;  
int main()  
{  
    f(x);  
    .....  
}  
void f(int a) {  
    x=3;  
    a++;}
```

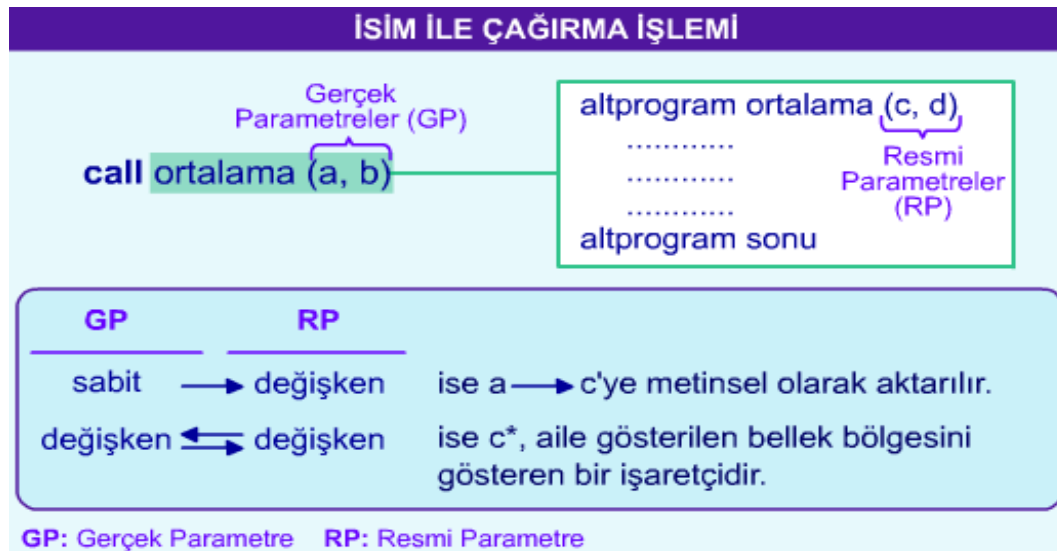
x'in son değeri değer-sonuç aktarımına göre 4 olacaktır.

Hem yer hem de zaman açısından etkindir. Fiziksel kopyalama yoktur.

Formal parametrelere erişim için bir **dolaylı erişim** gerekiyor. Sadece çağıran programdan altprograma değer aktarılması isteniyorsa, bu yöntemde gerçek parametrenin bellekteki yerine altprogram tarafından ulaşılabilirdi için, değerinde **istenmeyen değişiklikler** oluşabilir.

İsim ile Çağırma (*Call by Name*)

- Altprogramda gerçek parametreye karşı gelen formal parametrenin bulunduğu her yere metinsel olarak gerçek parametre yerleştirilir.
- Eğer gerçek parametre bir sabit değerse, isim ile çağırma yöntemi, değer ile çağırma yöntemi ile aynı şekilde gerçekleşir.
- Eğer gerçek parametre bir değişkense, isim ile çağırma yöntemi başvuru ile çağırma yöntemi ile aynı şekilde gerçekleşir.
- İsim ile çağırma yönteminin gerçekleştirilmesi güçtür ve kullanıldığı programların hem yazılmasını hem de okunmasını karmaşıktırabilir. Bu nedenle ALGOL 60 ile tanıtılan isim ile çağırma yöntemi, günümüzde popüler olan programlama dillerinde uygulanmamaktadır.



Programlama Dili	Değer	Sonuç	Değer- Sonuç	Referans	İsim
FORTRAN IV				X	
Fortran 77			X		
ALGOL 60	seçimlik				X
ALGOL W			X		
C++	X			X	
PASCAL, Modula2	X			seçimlik	
ADA	X	X	X	X	
Java	X			X	

Özet

- Altprogramlar ve altprogramların genel özellikleri
- Parametreler,
- Parametre Aktarım Yöntemleri,
- Programlama Dillerinde Parametre Aktarımı,