

Algoritmalar ve Programlama I – BLM 103

Hafta 6: Fonksiyonlar



Fenerbahçe Üniversitesi

6. Hafta İçeriği

- Fonksiyon Tanımlamaları
- Fonksiyon Argümanları
- Çalışma Zamanı Yığınları (Stack)
- Fonksiyonlar ile Algoritma Geliştirme

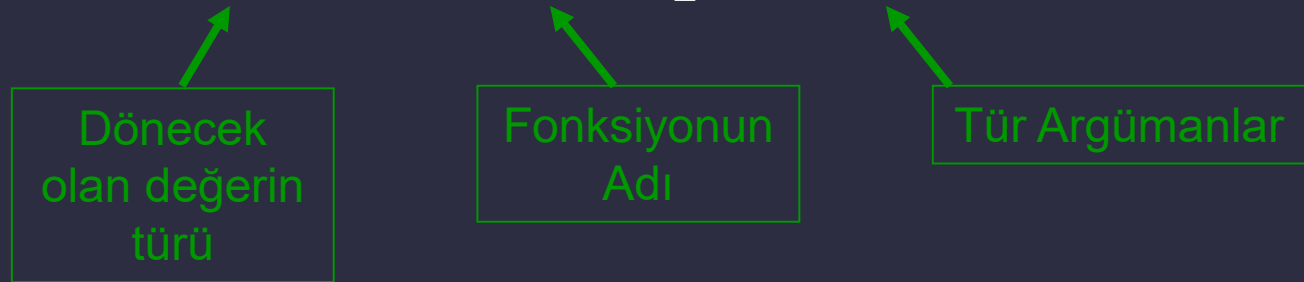
Fonksiyonlar

- Bir yazılımın, küçük parçacılarıdır.
- Yazılımı küçük parçalara ayırıp, soyutlaştırmayı sağlar
 - Düşük seviye detayların gizlenmesini sağlar
 - Programın akışının okunabilirliğini artırır
 - Birden çok yazılımcı ile çalışılan projelerde, bağımsız çalışmayı sağlar
- C Fonksiyonları
 - Argüman alabilir.
 - Tek bir değişken döndürebilir.
 - Döndüreceği değişkenin tipi sabittir.

C dilinde Fonksiyonlar

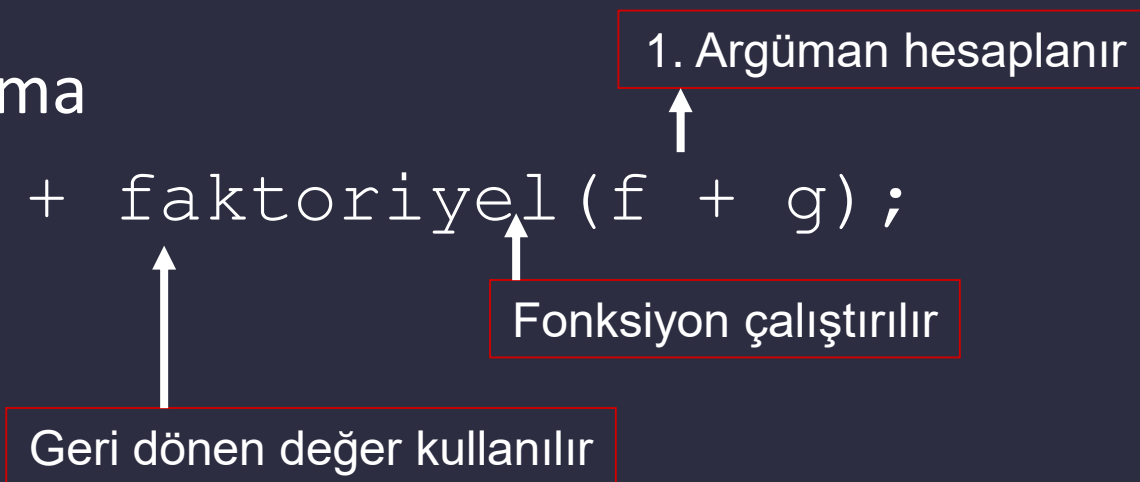
- **Tanımlama** (Prototip de denir)

```
int faktoriyel(int n);
```



- **Fonksiyon çağırma**

```
a = x + faktoriyel(f + g);
```



Yüksek Seviyeli Yaklaşım Örneği

```
int ebobBul(int s1, int s2) {  
...  
return hesap;  
}  
  
int main()  
{  
    int sayi1 = 5;  
    int sayi2 = 15;  
    int sonuc = ebobBul(sayi1, sayi2);  
  
    printf(sayi1, sayi2);  
    return 1;  
}
```

Örnek Fonksiyon Çağırılması

```
• int deneme(int q, int r)
{
    int k;
    int m;
    ...
    return k;
}
```

```
int abc(int a)
{
    int w;
    ...
    w = deneme(w, 10);
    ...
    return w;
}
```

Fonksiyon Tanımlama

- Dönecek değer türü ve argümanlar
 - Fonksiyonun prototip tanımlaması varsa, dönecek değer ve argüman türleri eşleşmelidir.
 - Argüman isimlerinin eşleşmesine gerek yoktur.

```
int faktoriyel(int n)
{
    int i;
    int sonuc = 1;
    for (i = 1; i <= n; i++)
        sonuc *= i;
    return sonuc;
}
```

Return result satırı, fonksiyonun çağrıldığı yere result değerini döndürür

Neden Tanımlama Yapılıyor?

- Fonksiyon yazılırken zaten tüm girişler ve çıkışlar yazılıyor. Ancak bir fonksiyonun kullanılacağı satırdan önce, kullanılacak olan fonksiyon henüz tanımlanmamış ise derleyici hata verecektir.
- Derleyiciyi kullanılacak olan fonksiyon hakkında ön bilgilendirme yapmak için prototip yazılmaktadır. Derleyici fonksiyonun kullanıldığı yerde, fonksiyonun giriş ve çıkışlarını bilmesi yeterli olmaktadır.
- Ayrıca fonksiyonun tanımlamaları ayrı bir dosyada olabilir. Bu dosyaya "header" denmektedir. Ayrı dosyada olması, birden çok programcı ile bir uygulama geliştirirken, fonksiyonun içine bakmadan sadece prototipe bakarak fonksiyonu kullanabilmesini sağlamaktadır.
- Diğer bir özellik ise, fonksiyonun içindeki kodun paylaşılmasının istenmediği durumlarda, sadece prototip paylaşarak güvenlik sağlanmaktadır.

Void Fonksiyonu

- Bir fonksiyon geriye bir değer döndürmeyecekse void olarak tanımlanır.

- Örn.

```
void test (int a, int b);
```

Fonksiyonlar Bellekte Nasıl Tutulur?

- Fonksiyonlar çağrıldıklarında bellekte Yığın (Stack) denen mekanizmalar ile tutulmaktadır.

Çağırılan Fonksiyon (Örn. Main)

Yığın'a fonksiyonun
ihtiyaç duyacağı
argümanları kopyala
Fonksiyonu çağır

Yığından sonuçları al

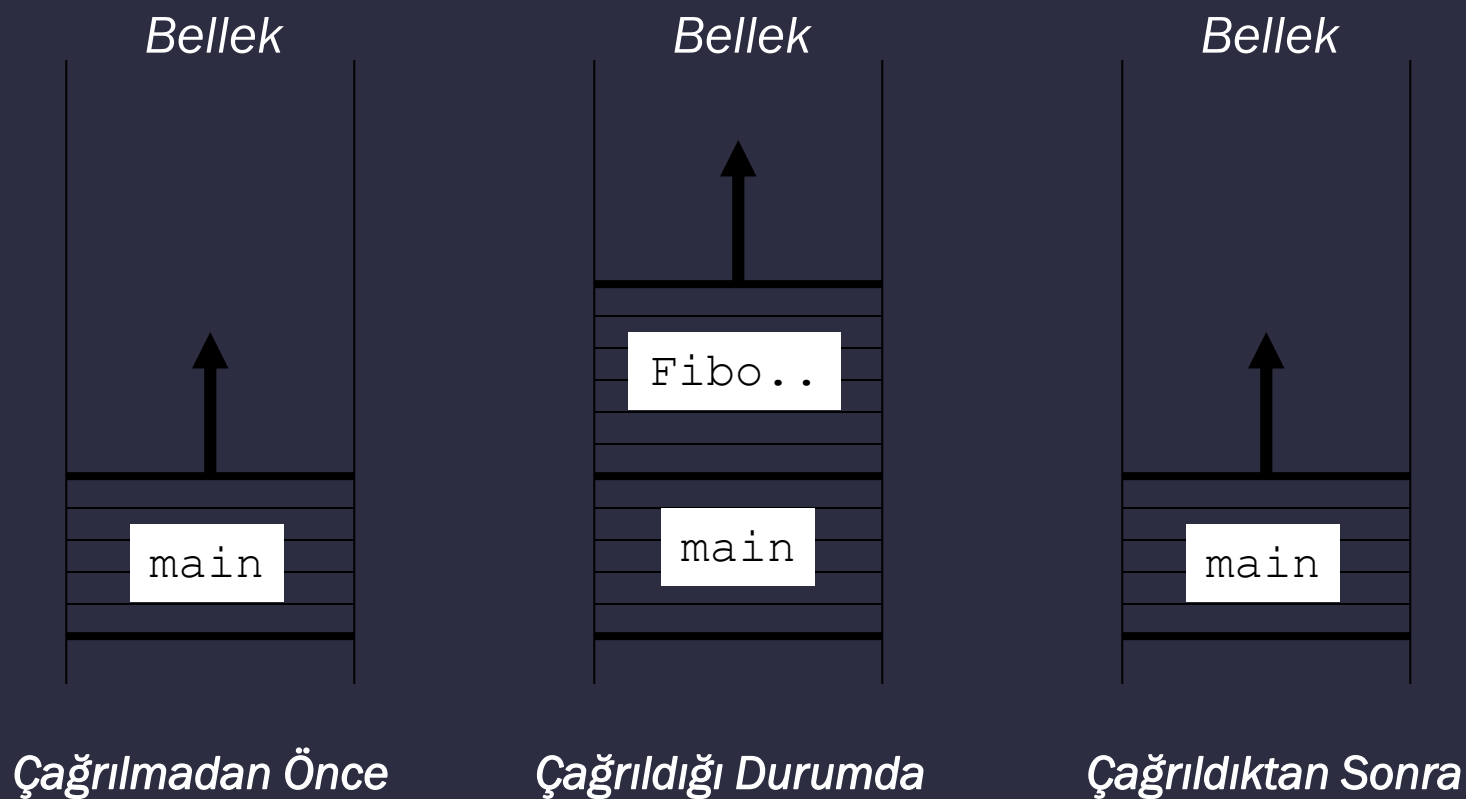
Çağırılan Fonksiyon (örn. Fibonacci)

Kodu çalıştır
sonuçları yığın'a yaz
Fonksiyon kaydını
yığından sil
çağırılan yere geri dön

Fonksiyonlar Bellekte Nasıl Tutulur?

- Yeni bir fonksiyon çağrıldığında, kaydı yığına atılır.
- Fonksiyon işini tamamlayıp, geri döndüğünde, yığından kaydı silinir.

Fonksiyonlar Bellekte Nasıl Tutulur?



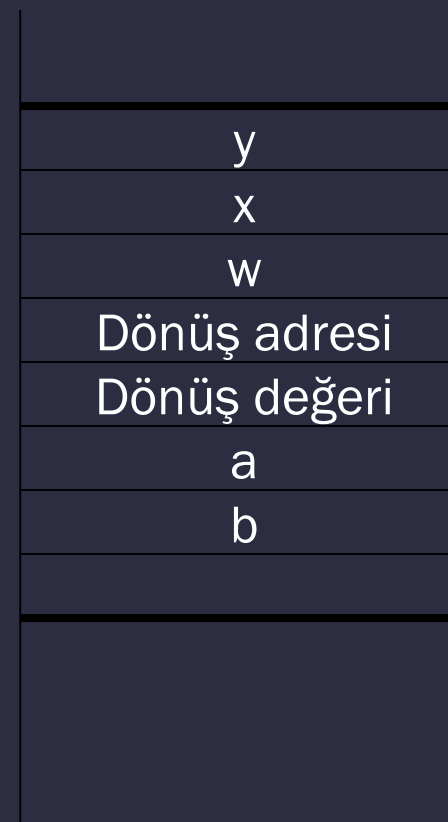
Fonksiyonlar Bellekte Nasıl Tutulur?

```

• int fibonacci(int a, int b)
{
    int w, x, y;
    .
    .
    .
    return y;
}

```

İsim	Tür	Offset	Geçerlilik
a	int	4	fibonacci
b	int	5	fibonacci
w	int	0	fibonacci
x	int	-1	fibonacci
y	int	-2	fibonacci



Yerel değişkenler

argümanlar

Dönüş Değeri ve Adresi

- Dönüş Değeri
 - Fonksiyonun hesap yaptıktan sonra döndüreceği değer
- Dönüş Adresi
 - Fonksiyon işini tamamladıktan sonra, çağrıldığı yere döneceği adres

Temel C Fonksiyonları (“math.h”)

Fonksiyon	Açıklama
abs ()	Sayının mutlak değerini bulur
floor ()	En yakın sayıdan küçük yada eşit tam sayıya yuvarlar.
round ()	Giriş verilen sayının küsüratı, .5’den büyük ise yukarıdaki tamsayıya, değilse aşağıdaki tam sayıya yuvarlar
ceil ()	En yakın sayıdan büyük yada eşit tam sayıya yuvarlar
sin ()	Sinüs hesabı yapar
cos ()	Kosinüs hesabı yapar
cosh ()	Hiperbolik kosinüs hesaplar
exp ()	e^x sayısını hesaplar
tan ()	Tanjant hesabı yapar
tanh ()	Hiperbolik tanjant hesabı yapar
sinh ()	Hiperbolik sinüs hesabı yapar
log ()	Log2 tabanında işlem yapar
log10 ()	Log 10 tabanında işlem yapar
sqrt ()	Karekök hesaplar
pow ()	Sayının kuvvetinin hesaplaması

Çeşitli C Fonksiyonları

Fonksiyon	Açıklama
clock()	Sistemin kaç clock cycle zaman geçtiğini gösterir
time()	Sistemin zamanını alır
difftime()	Ölçülen iki zaman arasındaki süreyi hesaplar
rand()	0 -32767 arasında rasgele sayı üretir
delay()	Programı verilen ms cinsi kadar bekletir