

# Java Dilinde Metodlar

Sınıfların asıl işlevleri olan metodlar Java dilinde yazılan programların çalışmasını sağlar.

Üç tip metod türü vardır:

1. **Instance Metodlar:** Nesne yaratılıp kullanılan metodlar
2. **Static Metodlar:** `SınıfAdi.MetodAdi` şeklinde kullanılan metodlar
3. **Constructor'lar:** Sınıfı memory'e yükleyen metodlar

## Bir metod tanımlamak

Java'da metod tanımı C dilinden farklıdır

Genel metod tanımı şu şekildedir:

`modifier returnType methodName(list of parameters)`

**Örnek:** `public static int max(int num, int num2);`

Bu örnekte;

Modifier: **public static**

returnValueType: **int**

methodName: **max**

list of parameters: **int num1** ve **int num2**

method signature: `max(int num, int num2)`

## Visible Type Nedir?

Nesneye yönelik programlama dilleri ile gelişmiş uygulamalar yapılmaktadır. Bu uygulamalar geliştirilirken kodların miktarı arttıkça yönetmesi zorlaşmaktadır. Bazı durumlarda bir metodun başka bir sınıf içerisinde çağrılmasını engellemek gibi ihtiyaçlar doğmuştur. Örneğin, bir **Date** sınıfının bazı metodları ve sınıf değişkenleri **private** tanımlanarak bunlara erişilip değiştirilmesi düzenlenmesi önlenmiştir(Başlangıç Tarihi 1 Ocak 1970 gibi..).

**public**, **protected**, **default**, **private** olmak üzere 4 farklı visible tip vardır. **public** ile tanımlanan bir metod bir başka sınıf içerisinde çağrılabilir. **private** ile tanımlanan bir metod başka bir sınıf içinde çağrılmaz. Sadece ait olduğu sınıf içerisinde çağrılabilir. **default** ve **protected** sonra anlatılacaktır.

**Örnek:**

```
public class TestMax{
    public static void main(String[] args){
        int i=5;
        int j=2;
        int k=max(i,j);
        int p=min(i,j);/*Hata verir. Çünkü min instance metod.
        Instance metod static metod içinde bu şekilde çağrılmaz. Nesne yarattıktan sonra çağrılabilir*/
        System.out.println("The maximum between "+ i + " and "+ j + " is" + k);
    }

    public static int max(int num1, int num2){
        int result=0;
        if(num1>num2){
            result=num1;
        }else{
            result=num2;
        }
        return result;
    }

    public int min(int num1,int num2){
        int result=0;
        if(num1>num2){
            result=num2;
        }else{
            result=num1;
        }
        return result;
    }
}
```

**Soru:** max metodu neden **static** tanımlandı?

**Cevap:** main metodu **static** olduğu için ve **static** bir metod ancak **static** metodu çağırabildiği için max **static** tanımlanmıştır.

## Overloading Methods

Diyelim ki programımızda iki **int** türünden sayının ve iki **double** türünden sayının maksimum değerlerini bulmak istiyoruz. Fakat basitlik olsun diye aynı metod ismini kullanacağız. Yani **int** türünden iki sayıyı parametre olarak gönderirsek bize bu iki sayının büyük olanını dönderecek, **double** türünden iki sayıyı parametre olarak gönderdiğimiz zaman ise bu iki sayıdan büyük olanını dönderecek. İşte bu tarz işlemler için metodlar overload edilir.

**Örnek:**

```
public static int max(int num1, int num2){
    int result=0;
    if(num1>num2){
        result=num1;
    }
}
```

```
    }else{
        result=num2;
    }
    return result;
}
public static double max(double num1, double num2){
    double result=0.0;
    if(num1>num2){
        result=num1;
    }else{
        result=num2;
    }
    return result;
}
```

**Sonuç:** Dikkat edersek, metod isimleri aynı parametreleri ise farklıdır. **return** tipleri aynı da olabilir farklı da olabilir. Bu örneğimizde farklıdır. Metod signature farklı olduktan sonra metod overload yapılabilir.