



# Java da Constructor ve Method Arasındaki Farklar & Objenin Genel Tanımı

- **Object-Oriented Programming** (Nesne Yönelimli Programlama - OOP) prensiplerinin temel unsurları olan **class** (sınıf), **object** (nesne), **method** (metot) ve **constructor** (yapıcı), programlama dünyasının yapı taşlarıdır."

# Class Nedir ?



- **Class (Sinif):** Bir urun oluřturmak iin kullanılan **kaliplardir.**
- Java'da bir class , bir object nin **zelliklerini** ve **davranıřlarını** tanımlar..
- **Class,** objenin nasıl olacađını tanımlar ama henz ortada bir nesne yoktur.

# Java' da Class

## Keywords in a Class

```
public class MyFirstClass {}
```

**public** bir Access Modifier'dır.

Access Modifier bir Class'a kimin ulaşp kimin ulaşamayacağını belirler

**MyFirstClass** Class ismidir.

Class isimleri her zaman büyük harfle baslar.

{Variable, Metod} ile başlayan kısım **Class Body** olarak isimlendirilir

```
public class Araba {  
    // Özellikler (attributes)  
    String renk;  
    String model;  
    int yakitSeviyesi;  
    boolean farlarAcik;  
    boolean motorCalisiyor;  
  
    // Metotlar (methods)  
  
    // Araba hızlanıyor  
    void hizlan() {  
        hiz += 10;  
    }  
  
    // Araba fren yapıyor, hız azalıyor  
    void frenYap() {  
        if (hiz > 0) {  
            hiz -= 10;  
        }  
    }  
  
    // Motoru çalıştırma metodu  
    void motoruCalistir() {  
        if (!motorCalisiyor) {  
            motorCalisiyor = true;  
            System.out.println("Motor çalıştı.");  
        }  
    }  
  
    // Farları açma metodu  
    void farlariAc() {  
        if (!farlarAcik) {  
            farlarAcik = true;  
            System.out.println("Farlar açıldı.");  
        }  
    }  
  
    // Araba yakıt dolduruyor  
    void yakitDoldur(int miktar) {  
        yakitSeviyesi += miktar;  
    }  
}
```





# Object (Nesne) Nedir ?

- **Object (Nesne):** Kalıptan üretilmiş elle tutulur somut bir örnektir.
- **Java'da Nesnenin Tanımı:**
  - **Object**, bellekte yer kaplayan ve classın özelliklerini ve davranışlarını taşıyan bir somut bir örnektir
  - Class bir **kalıp** ya da **şablon** gibiyken, object o kalıptan oluşturulan gerçek, somut bir varlık.
  - **Java'da object**, sınıftan **new** anahtar kelimesiyle oluşturulur.



# Javada Object

```
public class Araba {  
    // Özellikler (attributes)  
    String renk;  
    int hiz;  
    String model;  
    int yakitSeviyesi;  
  
    // Araba hızlanıyor  
    void hizlan() {  
        hiz += 10;  
    }  
  
    // Araba fren yapıyor, hız azalıyor  
    void frenYap() {  
        if (hiz > 0) {  
            hiz -= 10;  
        }  
    }  
  
    // Yakıt doldurma metodu  
    void yakitDoldur(int miktar) {  
        yakitSeviyesi += miktar;  
        System.out.println("Yakıt seviyesi: " + yakitSeviyesi);  
    }  
  
    public static void main(String[] args) {  
        // Araba sınıfından bir nesne oluşturma  
        Araba araba1 = new Araba();  
  
        // Nesneye ait özelliklere değer atama  
        araba1.renk = "Kırmızı";  
        araba1.hiz = 50;  
        araba1.model = "BMW";  
        araba1.yakitSeviyesi = 20;  
  
        // Nesne üzerinde metod çağırma  
        araba1.hizlan(); // Hızlanma işlemi  
        araba1.frenYap(); // Fren yapma işlemi  
        araba1.yakitDoldur(10); // Yakıt doldurma işlemi  
    }  
}
```

➡ **Araba**   **araba1**   =   **new**   **Araba()**;

**Araba** class ismi

**araba1** uretilecek object

= Atama operatörü

**new** yeni object uretilecek

**Araba ()** .....?



# Metot Nedir ?



- **Metot:** Bir nesnenin gerçekleştirebileceği eylemleri ya da işlemleri tanımlayan bir fonksiyondur. Nesneler üzerindeki işlevsel davranışları yönetir. Metotlar, sınıfın özelliklerine ve işlevlerine göre belirleni.

Mesela:

**Karıştır:** "Karıştır" methodu kekin malzemelerini bir araya getirir.

**Pişir:** "Pişir" methodu kekin fırında pişmesini sağlar.

**Soğut:** "Soğut" methodu keki dinlenmeye bırakarak servise hazır hale getirir.

- **Java da Method,** bir sınıfa ait belirli bir görevi gerçekleştirmek amacıyla kullanılır. Bu görev, veriyi işlemek, hesaplama yapmak ya da sınıfın özelliklerini değiştirmek olabilir.

## Bir Metod nasıl oluşturulur (Create)?

```
public int myFirstMethod () {}
```

**public** bir Access Modifier'dır.

Access Modifier bir metod'a kimin ulaşip kimin ulaşamayacağını belirler

**int** return type'dır.

return type metodun ürettiği sonucun tipidir.

**myFirstMethod()** metod ismidir.

Metod isimleri küçük harfle başlar.

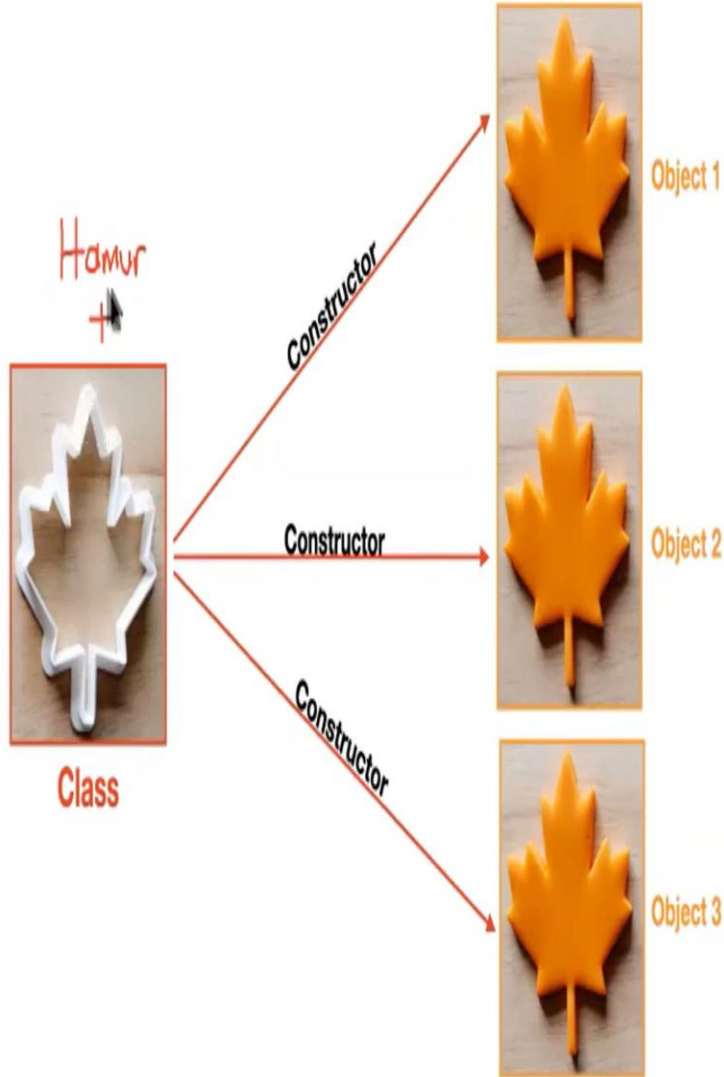
Birden fazla kelime içeriyorsa diler kelimelerin ilk harfleri büyük olur.

"{" ile başlayıp "}" ile biten kısım **Metod Body** olarak isimlendirilir.

## Javada Metod

```
public class Araba {  
    String renk;  
    int hiz;  
    boolean calisiyorMu;  
  
    // Araba çalıştırma metodu  
    public void calistir() {  
        if (!calisiyorMu) {  
            calisiyorMu = true;  
            System.out.println("Araba çalıştı.");  
        } else {  
            System.out.println("Araba zaten çalışıyor.");  
        }  
    }  
  
    // Araba durdurma metodu  
    public void durdur() {  
        if (calisiyorMu) {  
            calisiyorMu = false;  
            System.out.println("Araba durdu.");  
        } else {  
            System.out.println("Araba zaten durdu.");  
        }  
    }  
  
    // Arabanın hızını döndüren metod  
    public int getHiz() {  
        return hiz;  
    }  
  
    // Arabanın rengini döndüren metod  
    public String getRenk() {  
        return renk;  
    }  
  
    // Arabanın çalışıp çalışmadığını döndüren metod  
    public boolean isCalisiyorMu() {  
        return calisiyorMu;  
    }  
}
```

# Constructor Nedir ?



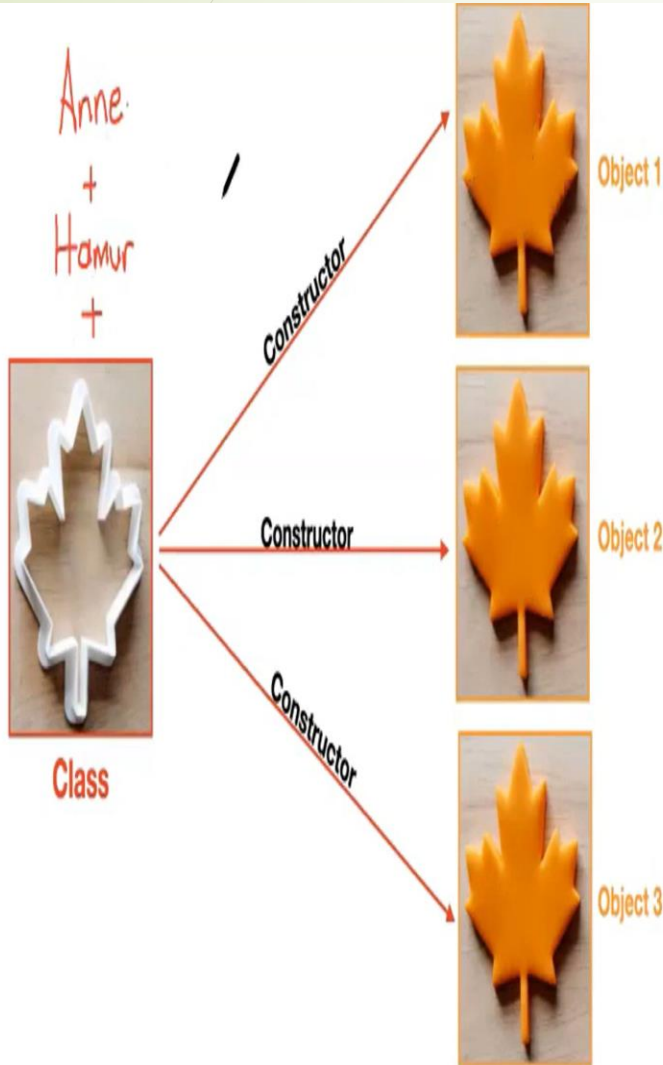
❖ . **Constructor (Yapıcı)** : Her şey hazırlandıktan sonra işi devralan ve işlemi **başlatan** biridir.

➤ Java'da **constructor** (yapıcı metot), bir sınıftan yeni bir nesne oluşturulurken **çağrılan özel bir metottur**. Constructor, nesnenin başlangıç durumunu ayarlamak için kullanılır, yani bir **nesneye ilk değerlerini atar**.



# Constructor Nedir ?

- ❖ . **Constructor (Yapıcı)** : Her şey hazırlandıktan sonra işi devralan ve işlemi **başlatan** biridir.
- Java'da **constructor** (yapıcı), bir sınıftan yeni bir nesne oluşturulurken çağrılan **sınıfa özgü bir yapıdır**. Constructor, nesnenin başlangıç durumunu ayarlamak için kullanılır, yani bir **nesneye ilk değerlerini atar**.



# Javada consrtuctori

## Constructor 'in Özellikleri

```
public Araba() {}
```

### Java'da Constructor'ın Temel Özellikleri:

- **Sınıf Adıyla Aynı İsimde Olur:** Constructor'ın adı, sınıfın adıyla aynı olmalıdır.
- **Return type Yoktur:** Constructor'ların return type olmaz (void bile yazılmaz).
- **Nesne Oluşturulurken Otomatik Olarak Çağılır:** **new** anahtar kelimesiyle bir nesne oluşturulduğunda constructor otomatik olarak çalıştırılır.

```
Araba araba1 = new Araba();
```

- **Varsayılan Constructor:** Eğer bir sınıfta constructor tanımlanmazsa, Java otomatik olarak parametresiz bir varsayılan constructor oluşturur.

```
public MyFirstConstructor ( ) {}  
Public MyFirstConstructor (String name, int age) {}
```

**public** bir **Access Modifier**'dir.

**Access Modifier**, bir constructor'a kimin ulaşım kimin ulaşamayacağını belirler.

**MyFirstConstructor** constructor ismidir.

Bu aslında Class ismidir. Constructor ismi **her zaman class ismi ile aynıdır**.

**()** parametre kısmıdır.

**Parametreler**, constructor'ın çağrıldığı anda girilecek değerleri temsil eder.

**String name, int age** parametrelerdir.

**String name:** İsmi temsil eden bir String türünde parametredir.

**int age:** Yaşı temsil eden bir int türünde parametredir.

**{ }** Constructor Body olarak isimlendirilir.

İçinde genellikle nesne değişkenlerine ilk değerler atanır.



Araba.java



Runner.java



# Constructor ve Method Arasındaki Farklar

Özellik	Constructor	Method
Amaç	Nesne oluşturmak ve başlatmak	Belirli bir işlev gerçekleştirmek
İsim	Sınıfla aynı isim	Herhangi bir isim
Return Type	Return type yoktur (void veya başka bir return type belirtilmez)	Belirtilmelidir (void, int, String, vb.)
Çağrı Zamanı	Nesne oluşturulduğunda otomatik olarak çağrılır	Kullanıcı tarafından manuel olarak çağrılır
Parametreler	Parametreler olabilir ancak zorunlu değildir. Parametreler, nesneye başlangıç değerleri atamak için kullanılır.	Parametreler olabilir, metodun işlevine bağlıdır. Parametreler, metodun aldığı girdilerle belirli işlemleri gerçekleştirmesine olanak tanır.


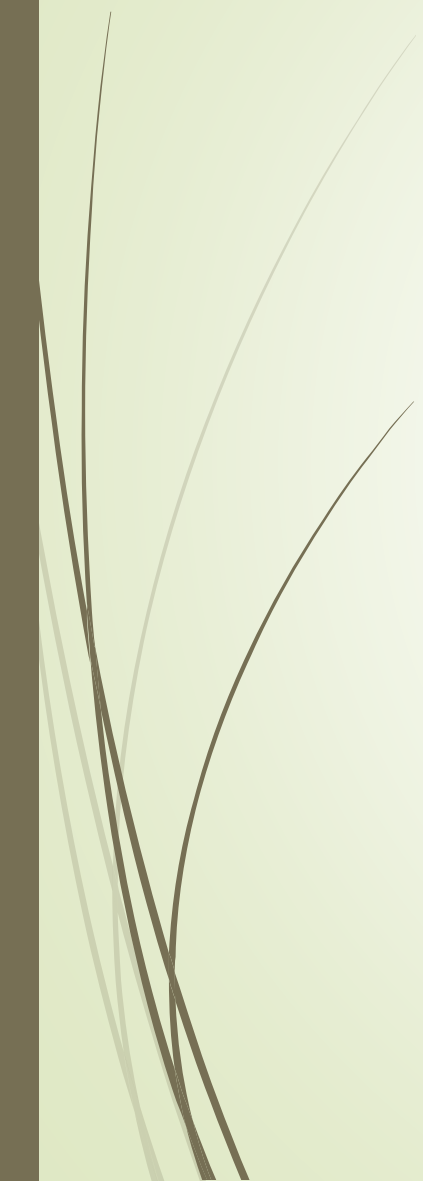
# Sonuç ve Özet

- **Objeler ve Sınıflar:** Sınıflar, objeleri oluşturan kalıplardır. Örnek: Car **sınıfı** BMW gibi **objeler** oluşturabilir
- **Constructor vs Method:** **Constructor**, objeyi başlatır; **method**, objenin işlevlerini gerçekleştirir
- Örnek: `new Car("Toyota")`  
`myCar.hızlan()`

# Java Constructor ve Method ile İlgili Mülakat Soruları

- 1. Constructor ve method arasındaki temel fark nedir?
- Cevap: Constructor, sınıfın ismiyle aynı isimde olup, dönüş tipi belirtmez. Obje oluşturulurken otomatik olarak çağrılır. Methodlar ise belirli bir işlevi yerine getirmek için kullanılır ve dönüş tipi belirtilir
- 2. Constructor'ın amacı nedir?
- Cevap: Constructor, bir sınıftan yeni bir nesne oluşturulduğunda bu nesnenin başlangıç durumunu ayarlamak için kullanılır.
- 3. Parametrelili constructor nedir?
- Cevap: Parametrelili constructor, obje başlatılırken dışarıdan değerler alarak objeye atanmasını sağlar.
- 4. Java'da default constructor nedir?
- Cevap: Default constructor, bir sınıfta hiçbir constructor tanımlanmadığında Java tarafından otomatik olarak sağlanan constructor'dır. Parametre almaz ve nesneyi varsayılan değerlerle başlatır.
- 5. Bir sınıfta birden fazla constructor tanımlanabilir mi? Eğer tanımlanabiliyorsa buna ne ad verilir?
- Cevap: Evet, bir sınıfta birden fazla constructor tanımlanabilir. Buna "constructor overloading" denir.



- 
- 
- 6.Methodun amacı nedir ve ne zaman kullanılır?
  - Cevap: Method, bir sınıfın belirli bir işlevi yerine getirmek için kullanılır. Methodlar, nesneler üzerinde işlemler yapmak ya da bilgileri geri döndürmek için kullanılır.
  - 7. Java'da bir nesne (object) nasıl oluşturulur?
  - Cevap:Nesne, new anahtar kelimesi kullanılarak oluşturulur
  - Örneğin: Araba araba= new Araba()
  - 8. Parametrelili constructor nedir?
  - Cevap: Parametrelili constructor, obje başlatılırken dışarıdan değerler alarak objeye atanmasını sağlar.
  - 9.Method'ların return type'ı ne anlama gelir ve nasıl belirlenir?
  - Cevap: Method'un return type'ı, methodun çağırıldıktan sonra döndüreceği veri türünü belirtir. Dönüş türü, methodun işlevine göre belirlenir.



➤ **SABIRLA BENİ DİNLEDİĞİNİZ İÇİN TEŞEKKÜR EDERİM!**