

# Nesnelerin Klonlanması

# Nesnelerin Klonlanması

- Java nesnelerinin **bire bir kopyalarını** oluşturmaya yarar
- **java.lang.Object** sınıfı içerisinde **protected clone** metodu tanımlıdır
- Uygulama sınıfları bu metodu override ederek içerisinde **super.clone()**'u çağırmalıdır
- Uygulama sınıflarının ayrıca **Cloneable** arayüzünü implement etmeleri gerekir
- Aksi durumda clone metodu çağrıldığı vakit **CloneNotSupportedException** fırlatılır

# Java Object clone() Metodu

```
public class Araba implements Cloneable {  
    private int hiz;  
    private String adi;  
  
    public Araba(int hiz, String adi) {  
        super();  
        this.hiz = hiz;  
        this.adi = adi;  
    }  
  
    @Override  
    public Araba clone() throws CloneNotSupportedException {  
        return (Araba) super.clone();  
    }  
  
    public int getHiz() {  
        return hiz;  
    }  
  
    public String getAdi() {  
        return adi;  
    }  
}
```

# Clone Metodunun Problemleri

- Herhangi bir nesnenin klonlanabilir olması için öncelikle sınıfının **Cloneable** arayüzünü implement etmesi gerekir
- Ancak clone metodu bu arayüzde değildir, doğrudan **java.lang.Object** sınıfında tanımlıdır
- Fakat burada da **protected** olarak tanımlıdır
- Dolayısı ile sınıf içerisinde clone metodunu **public modifer ile override etmeden çağırmak** mümkün değildir

# Clone Metodunun Problemleri

- Normal şartlarda bir arayüz istemcilerine onu implement eden sınıfın **ne kabiliyete sahip olduğunu** içerdiği metot tanımları ile ilan eder
- **Cloneable** arayüzü ise Object sınıfındaki **clone metodunun davranışını** değiştirir
- Cloneable arayüzü implement edili değilken bir nesnenin clone metodu çağrılırsa **CloneNotSupportedException** fırlatılır
- Eğer arayüz implement edilirse metot bu durumda nesnenin **attribute'ları teker teker kopyalayarak** klon nesneyi oluşturur

# Clone Metodunun Problemleri

- Clone metodunun spesifikasyonuna göre **`x.clone().getClass() == x.getClass()`** olmalıdır
- Ayrıca nesnenin kopyası yaratılırken **herhangi bir constructor çağrılmayacağı** belirtilmiştir
- Ancak bunu clone metodunu **override eden alt sınıflarda** garanti edecek bir mekanizma mevcut değildir
- Aslında **eğer sınıf final ise alt sınıf olamayacağı için** constructor çağrısı ile nesne kopyası oluşturmak herhangi bir sorun da çıkarmayacaktır

# Clone Metodunun Problemleri

- Ancak final olmayan sınıflarda clone metodu override edildiğinde nesne kopyası mutlaka **super.clone()** çağrısı ile oluşturulmalıdır
- Klonlanan nesnenin attribute'ları kopyalanırken **yüzeysel kopyalama** (swallow copy) yapılır
- Bu nedenle asıl nesne ve klon **aynı nesnelere refer** ediyor olabilirler
- Attribute'lardaki **mutable nesneler** için bu ciddi sorun teşkil edebilir

# Clone Metodunun Problemleri

- Böyle durumlarda **derin kopyalama** (deep copy) yapmak ve attribute'lara değerlerini bu derin kopyalama sırasında **ayrıca oluşturarak atamak** gerekebilir
- Bunun için de attribute'ların **final tanımlanmamış olması** gerekir
- Constructor'larda olduğu gibi clone metodunda da **final olmayan metotların invoke edilmesi** risk teşkil edebilir
- Burada meydana gelebilecek bir hata nesnenin yaratımını başarısız kılacaktır



# Clone Metoduna Bir Alternatif: JAVA Eğitimleri

## Copy Constructor

- Clone metodunu kullanmak yerine önerilen çözümlerden birisi “**copy constructor**” yaklaşımıdır
- Bir sınıf **constructor argümanı** olarak **kendi tipinden bir nesne** kabul eder
- Constructor içerisinde **input nesnenin state'i** yeni yaratılan nesneye kopyalanır

# Copy Constructor Örneği

```
public class Araba {  
    private int hiz;  
    private String adi;  
  
    public Araba(int hiz, String adi) {  
        this.hiz = hiz;  
        this.adi = adi;  
    }  
  
    public Araba(Araba araba) {  
        this.hiz = araba.hiz;  
        this.adi = araba.adi;  
    }  
  
    public int getHiz() {  
        return hiz;  
    }  
  
    public String getAdi() {  
        return adi;  
    }  
}
```

Aynı sınıftan input argüman alan bir constructor içerisinde nesnenin bütün attribute değerleri input nesne den elde edilerek initialize edilir

# İletişim



[www.harezmi.com.tr](http://www.harezmi.com.tr)

[www.java-egitimleri.com](http://www.java-egitimleri.com)



[info@harezmi.com.tr](mailto:info@harezmi.com.tr)

[info@java-egitimleri.com](mailto:info@java-egitimleri.com)



[@HarezmiBilisim](https://twitter.com/HarezmiBilisim)

[@JavaEgitimleri](https://twitter.com/JavaEgitimleri)