

Javada Final Anahtar Kelimesi Ve Kullanımı

Bu yazımızda java programlama dilinde önemli bir yere sahip olan final anahtar kelimesi ve kullanımı ile ilgili örnekler vereceğiz. Final anahtar kelimesini kullanım bakımından 3 kısımda inceleyebiliriz.

1. Değişken seviyesinde final kullanımı
2. Metod seviyesinde final kullanımı
3. Sınıf seviyesinde final kullanımı

1. Değişken Seviyesinde Final Kullanımı

Sınıflar içerisinde bulunan değişkenler, oluşturulan her nesne için farklı alanda tekrar tekrar oluşturulmaktadır. final anahtar kelimesini kullanılan değişkenler ilk değerleri girildikten sonra değiştirilemezler.

Final değişkenlerine değerler ya tanımlamanın gerçekleştirildiği esnada yada sınıfa ait yapıcı metodu çağrıldığında verilmelidir. Bu belirtilen yerlerde değeri atanmazsa uygulamada hatalar alırız.

```
public class Urun{  
  
    #urunadi final değişken olarak setlendi.  
    private final String urunadi = "FLOWMETRE";  
    public void urunAdiYazdir() {  
        System.out.println("urunadi : "+urunadi);  
    }  
}
```

Bu verdiğimiz örnekte final olarak tanımlanan değişkenin değeri doğru yerde verildiği için herhangi bir hata ile karşılaşmayacağız. Eğer final kelimesi ile belirtilen değişkenin değerini vermeseydik bize derleme sırasında "The

blank final field urunadi may not have been initialized” şeklinde bir hata verecektir.

2. Metod seviyesinde final kullanımı

Java’da sınıflar içerisinde bulunan methodlarda final anahtar kelimesi kullanılarak oluşturulabilmektedir. Ancak final tanımlı olan method hiçbir şekilde başka sınıf tarafından miras alınsa bile override edilemez.

```
public class Urun{
    #urunsayisi Sınıf değişkeni. Program çalışır çalışmaz
    bellekte yer oluşturulur.
    public static int urunsayisi = 0;

    public final int urunsayisiGetir() {
        return urunsayisi;
    }
}

public class Arac extends Urun {
    public int urunsayisiGetir(){
        System.out.println("urunsayisi getir");
    }
}
```

Bu örneğimizde verilen kodlara bakarsak eğer Arac sınıfı Urun sınıfını miras almaktadır. Urun sınıf içerisinde bulunan urunsayisiGetir() metodu final olarak ayarlanmıştır. Aynı isimde bir metod Arac sınıfı içerisinde de tanımlanmış ve override edilmiştir. Bu da “Cannot override the final method from urunsayisiGetir” şeklinde bir hataya sebep olacaktır. Bu tür hatalardan uzaklaşmak için alt sınıflarda miras alınan sınıflardaki final metodları override edilmemelidir. Bu duruma bir örnek daha verelim.

```

public class FinalSuperClass{

    final void yaz() {
        System.out.println("Ana sınıf final metodu");
    }
}
public class FinalSubClass extends FinalSuperClass{
    void yaz() {

        System.out.println("Alt sınıftaki final metodu");

    }
}

```

3.Sınıf seviyesinde final kullanımı

Bu kullanımı incelersek eğer final olarak hazırlanan sınıflar başka sınıflar tarafından kesinlikle miras alınamazlar. Bir sınıfın zaten hiçbir şekilde başka sınıf tarafından miras alınmasını istemiyorsak bu yapıyı kullanırız. ufak bir kod örneği verelim.

```

public final class Dikdortgen {

}
class Kare extends Dikdortgen{

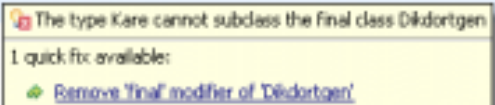
}

```

```

public final class Dikdortgen {
}
class Kare extends Dikdortgen{
}

```



Bu yazığımız satırlar sonucunda yukarıdaki şekilde gördüğümüz hata ile karşılaşacağız. Çünkü biz sınıfımızı final değişkeni ile tanımladık bu durumda Kare sınıfı Dikdörtgen sınıfının kalıtımını alamaz.

KAYNAKÇA

- <http://yazilimdarsi.info/makaleler/detay/86/javada-final-anahtar-kelimesinin-kullanimi>
- <https://umiitkose.com/2016/08/java-final-deyimi-nedir-ne-ise-yarar/>
- <http://www.ugurkizmaz.com/YazilimMakale-1635-Java-Final-Kullanimi.aspx>
- <http://www.kurumsaljava.com/2010/02/18/javada-final-anahtar-kelimesi/>
- https://www.dijitalders.com/icerik/44/1078/final_ozelliginin_kullanimi.html