



Merhabalar ,

Bu anlatımda java da tip dönüşümlerinden bahsedeceğim.Nedir bu tip dönüşümü (type casting) ? Kod yazarken bir veri tipinden diğer bir veri tipine aktarım yapmamız gerekebilir.Örneğin bir değişkeni String olarak aldık varsayalım aldığımız değer 5 ancak biz bu değeri int tipinde aktarmak istediğimizi farz edelim veya tam tersi int tipinde aldığımız bir değeri String tipine aktarmamız gerektiğini varsayalım işte bu noktada tip dönüşümleri (Type Casting) yardımımıza koşmaktadır.Bu tarz tip dönüşümde aklımızdan çıkarmamız gereken bir nokta var.Her tipin bellekte belirli bir alan kapladığını tiplerin sınırlarını daha önceli yazımda([TIKLAYINIZ](#)) anlatmıştım.Bu yazımı okuduğunuzu varsayarak küçük bir veri tipini büyük bir veri tipine değiştirdiğimizde bir sorun olmayacaktır , ancak büyük bir veri tipini küçük bir veri tipine aktardığımızda veri kaybı yaşayabiliriz ...Tabiki tip dönüşümlerinin bazı kuralları vardır.Şimdi bunlardan bahsedelim ...

İlkel(Primitive) tiplerden başlayalım ve bunlar küçükten büyüğe soldan sağa yazalım ...

byte < short < int < long < float(ondalıklı) < double(ondalıklı)

Burada küçük bir tipi büyük bir veri tipine dönüştürme işlemine Otomatik tip dönüşümü denir

Java bize herhangi bir sorun çıkarmaz hata vermez.

Hedef tip kaynak tipden daha büyükse

Java bu noktada bize hata vermeyecektir...

Örnek Kod

```
package serkankaya.net.tipdonusumleri;
```

```

/**
 *
 * @author Serkan Kaya
 */
public class SkayaTipDonusumleri {

    public static void main(String[] args) {

        byte byteTipi = 15;
        short shortTipi = byteTipi;
        System.out.println("byte dan short tipine otomatik tip dönüşümü sonucu : "
+ shortTipi);
        int intTipi = shortTipi;
        System.out.println("short dan int tipine otomatik tip dönüşümü sonucu : "
+ intTipi);
        long longTipi = intTipi;
        System.out.println("int den long tipine otomatik tip dönüşümü sonucu : " +
longTipi);
        float floatTipi = longTipi;
        System.out.println("long dan float tipine otomatik tip dönüşümü sonucu : "
+ floatTipi);
        double doubleTipi = floatTipi;
        System.out.println("float dan double tipine otomatik tip dönüşümü sonucu :
" + doubleTipi);

        // ----- Console Sonuç Ekranı -----
        // byte dan short tipine otomatik tip dönüşümü sonucu : 15
        // short dan int tipine otomatik tip dönüşümü sonucu : 15
        // int den long tipine otomatik tip dönüşümü sonucu : 15
        // long dan float tipine otomatik tip dönüşümü sonucu : 15.0
        // float dan double tipine otomatik tip dönüşümü sonucu : 15.0

    }
}

```

Peki büyük veri türünün küçük veri türüne değişimi nasıl gerçekleşir?(Casting)

Kuralları ;

Tipler birbirine uygun olmalıdır,

Kaynak tip hedef tipin tutacağı değerden büyük olmamalıdır,
Küsratlı sayılar tam sayılara çevirildiğinde küsratlarını kaybederler
Örnek Kod :

```
package serkankaya.net.tipdonusumleri;

/**
 *
 * @author Serkan Kaya
 */
public class SkayaTipDonusumleri {

    public static void main(String[] args) {
        double doubleVeri = 12723.12;
        float doubleToFloat = (float) doubleVeri;
        System.out.println("double veri türünden float Veri Türüne Casting
:"+doubleToFloat);
        long longVeri = 32654;
        int longToInt = (int) longVeri;
        System.out.println("long veri türünden int Veri Türüne Casting
:"+longToInt);
        int intVeri = 4587;
        short intToShort = (short) intVeri;
        System.out.println("int veri türünden short Veri Türüne Casting
:"+intToShort);
        short shortVeri = 127;
        byte shortToByte = (byte) shortVeri;
        System.out.println("short veri türünden byte Veri Türüne Casting
:"+shortToByte);
        //      ----- Console Sonuç Ekranı -----
        //      double veri türünden float Veri Türüne Casting :12723.12
        //      long veri türünden int Veri Türüne Casting :32654
        //      int veri türünden short Veri Türüne Casting :4587
        //      int veri türünden float Veri Türüne Casting :127
    }
}
```

Bahsettiğim kurallara uyarak örnek yaptığım için Java da herhangi bir hata vermedi ve sonuçlarıda gördüğünüz gibi ... hata alan versiyonlarını denemekte size kalsın ☐

Umarım faydalı olmuştur ?

Serkan Kaya

Full Stack Java Developer