



clarusway



clarusway



clarusway



clarusway



clarusway

# Scanner Class Type Casting

Lesson:  
JAVA Chapter 03

# Kullanıcıdan Değer Alma

## 1) Scanner scan = new Scanner (System.in);

**scan:** Scanner class'ından create edilene obje'dir. Naming convention(isimlendirme kuralları) olarak scan, sc input isimleri verilir. Verilen isim variable type veya metodun aksiyonuna uygun olmalı.

```
Scanner scan=new Scanner(System.in);
```

## 2) System.out.println("Lütfen password'unuzu giriniz : ");

Kullanıcıya girilmesi istenen değerler için bilgilendirme komutu girilmeli.

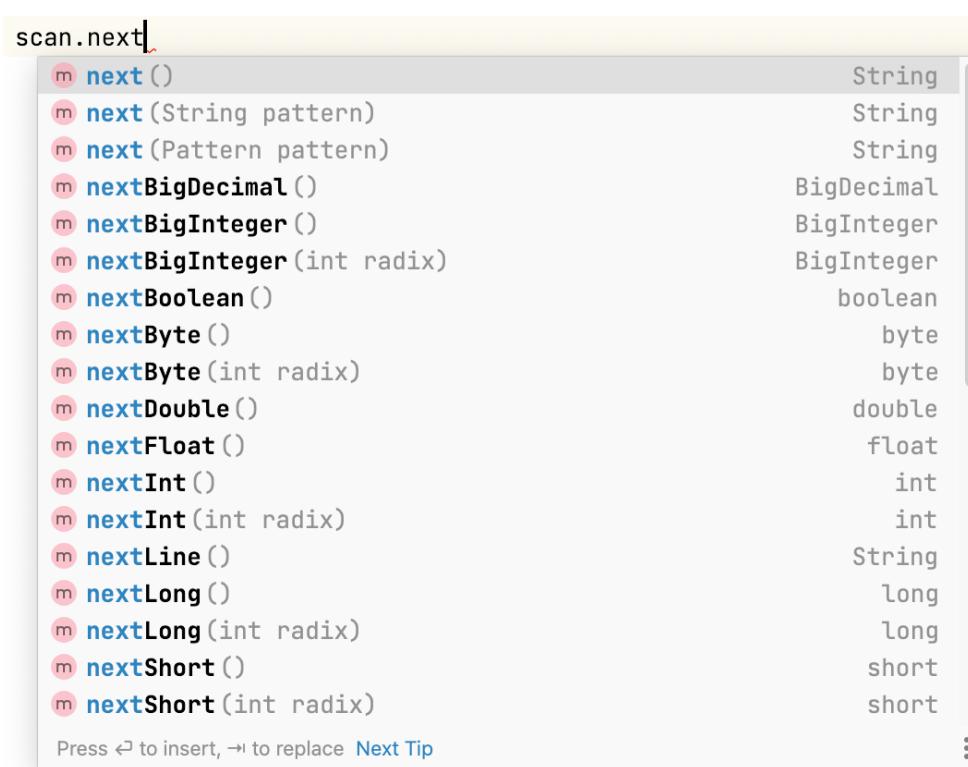
```
System.out.println("Lütfen password'unuzu giriniz : ");
```



# Kullanıcıdan Değer Alma

**3) Scan obje'sine kullanıcının girdiği data type'a göre ilgili next() method call ederek girilen değerleri scan objesine atar.**

```
int age = scan.nextInt()  
String name = scan.nextLine()
```



# Type Casting (Veri Türü Dönüşümü)

- JAVA'da kod yazarken bir veri tipinden diğer bir veri tipine aktarım yapmamız gerekebilir.
- **Type Casting:** Bir tipin doğal olarak uyumlu olduğu diğer bir tip olarak kullanımı. Örneğin, float bir sayıdır, int de sayıdır. Double da sayıdır. Elimizdeki bir int değişkeni/değeri float ya da double olarak kullanabilmemize casting diyoruz. Peki float değeri int olarak alabilir miyiz? Cevap evet. Yalnızca bir küçük "fakat" koşulu var. float kesirli bir sayı olduğundan 17,1 değeri varsa int kullanımında 0,1 değerini kaybediyoruz.
- **Type Casting** yaparken data tiplerinin sınırlarına dikkat edilmelidir. Data tipinin sınırlarını aşan Type Casting işlemlerinde hata almamamız için bazı komutlar kullanılır.

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



# Type Casting (Veri Türü Dönüşümü)

## 1) Auto Widening (Otomatik Genişletme)

Dar veri tipinden daha geniş bir veri tipine atama yapıldığında JAVA Type Casting otomatik olarak yapar.

byte → short → int → long → float → double

widening

ÖRN:

```
byte age = 33;
short yas = age; → 33
int kuru = yas; → 33
float omur = kuru; → 33.0
double hayat = omur; → 33.0
```

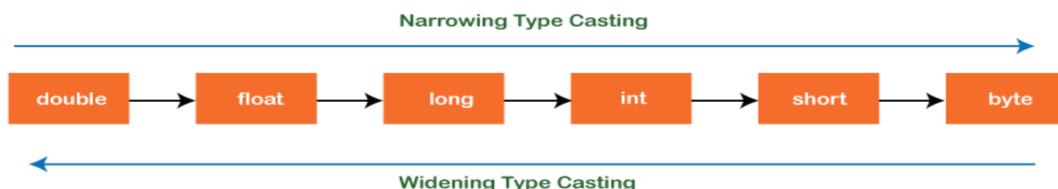


# Type Casting (Veri Türü Dönüşümü)

## 2) Explicit Narrowing (Manuel Daraltma)

```
public class ClarusWay {
    public static void main(String[] args) {
        double mySalary=12000.453;
        int massim= (int) mySalary;
        System.out.println(mySalary); //12000.453
        System.out.println(massim); //12000
    }
}
```

- Geniş veri tipinden daha dar bir veri tipine atama yapıldığında JAVA dönüşümü otomatik olarak YAPMAYACAKTIR. JAVA Casting'in bir problem oluşturabileceğini varsayıarak MANUEL ONAY isteyecektir.
- Narrowing Casting bazen veri kaybına sebep olur, bazen de veriyi kendi sınırları içinde kalan başka bir veriye dönüştürebilir.



# THANKS!

Any questions?

HalUk Bilgin | JAVA Backend Developer

BAŞARI GAYRETE AŞIKTIR ☺

