

Java OOP Eğitimi

String Yapısı

Kasım ADALAN

Elektronik ve Haberleşme Mühendisi
Freelance Software Developer

Eğitim İçeriği

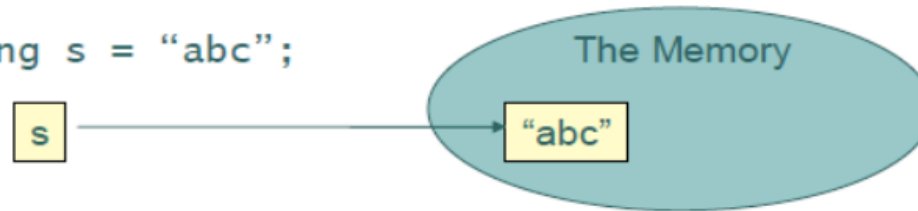
1. String değişmezliği
2. String'in önemli metodları
3. String birleştirme
4. String kıyaslama

String

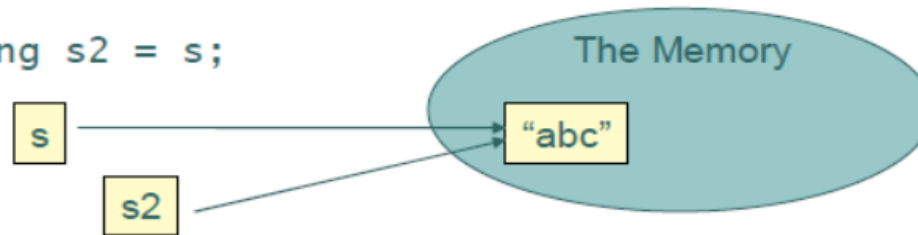
- Değer atama yapıldıktan sonra değiştirilemez
- Her birleştirme yeni bir String objesi yaratır
- String metodlarından dönen değerler herhangi bir değişkende tutulmazsa kaybolur.
- Final değişken olarak davranır bu yüzden büyük uygulamalarda sistemi yavaşlatır

String Değişmezliği

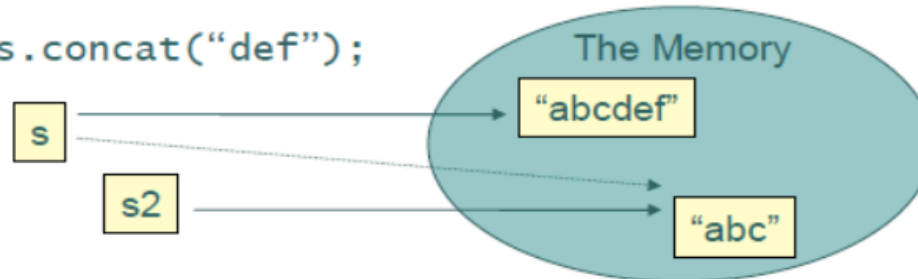
Step-1: `String s = "abc";`



Step-2: `String s2 = s;`



Step-3: `s = s.concat("def");`



String'in Önemli Metodları

- **charAt()** Belirtilen index nolu karakteri getirir.
- **concat()** Stringleri birleştirir.
- **Equals()** Stringlerin eşitliğine bakar.
- **equalsIgnoreCase()** Büyük küçük harf bakmaksızın karşılaştırır.
- **Length()** String uzunluğunu verir.
- **replace()** Karakterleri istediğimiz karakterlerle değiştir.
- **substring()** String içerisinden istediğimiz kadarını alır.
- **toLowerCase()** Küçük harfe dönüşür.
- **toUpperCase()** Büyük harfe dönüşür.
- **split()** Belli bir karaktere göre böler.
- **trim()** Başındaki ve sonundaki boşlukları alır.

Stringleri Birleştirme

- İki yöntem vardır ;

1. **+** işareti ile iki String nesnesi birleştirilir.

2. **StringBuilder** sınıfı nesne birleştirme yapar.

- Not : İkinci yöntem performans olarak daha iyidir.

String Kıyaslaması

- Stringler `equals()` metodu ile kıyaslanır.

```
if ( s.equals("mehmet" ) {  
  
}
```

Örnek

Konsoldan girilen kelimeyi tersten yazdıran programı yazınız..

Teşekkürler...



kasım-adalan



kasimadalan@gmail.com



kasimadalan