



CLARUSWAY

WAY TO REINVENT YOURSELF



clarusway



clarusway



clarusway



clarusway



clarusway

1

String Manipulation

Lesson:
JAVA Chapter 10

String Class

String class, **java.lang** kütüphanesinin altında bulunan ve bizlere karakter dizeleri üzerinde işlemler gerçekleştirmemizi kolaylaştıran bir imkan sağlar. String class'ı metotları, **immutable** metotlardır. Yani bu sınıfın içerisinde bulunan metotlar **değiştirilemez**.

* **TRICK :)** String sınıfı değişmezdir, bu nedenle bir kez oluşturulduktan sonra bir String objesi değiştirilemez. Karakter dizilerinde çok fazla değişiklik yapmak gerekiyorsa, **String Buffer & String Builder** Classes kullanılır.

String sınıfı **miras** alınarak, yeni bir sınıf **türetilmez**. String sınıfları temel olarak bir dizi metin karakterini depolamaya yarayan nesnelerdir. String ifadesi diğer nesnelerden ayrı olarak **new** ile tanımlama yapmamıza gerek duyulmadan çalışabilmektedir.

Bir **String değişken**, çift tırnak içine alınmış bir karakter koleksiyonu içerir:

```
String myBootcamp = "clarusway";
```



String Manipulation / Methods

1) concatenation()

Belirtilen String'i bu String'in sonuna birleştirir ve birleştirilmiş String'i döndürür. Başka bir string ekleme işlemi iki şekilde yapılır.

+ (toplama) işlemi

```
String bootcamp = "Clarusway";  
String success = "gayet başarılı";
```



Output : "Clarusway gayet başarılı"

```
System.out.println(bootcamp+" "+success);
```

concat () method

```
String bootcamp = "Clarusway";  
String success = "gayet başarılı";
```



Output : "Clarusway gayet başarılı"

```
System.out.println(bootcamp.concat(success));
```



String Manipulation / Methods

2) charAt()

Verilen String'in istenen indexteki karakteri (char) döndürür.

* **TRICK**) Index 0'dan başlar, maximum index (String'in uzunluğu -1)'dir.

Eğer method'da index olarak maximum indexden büyük bir sayı kullanılırsa JAVA hata verir.

(**StringIndexOutOfBoundsException**)

```
Exception in thread "main" java.lang.StringIndexOutOfBoundsException: String index out of range: 20
    at java.lang.String.charAt(Unknown Source)
    at _00_anlik.asd.main(asd.java:11)
```

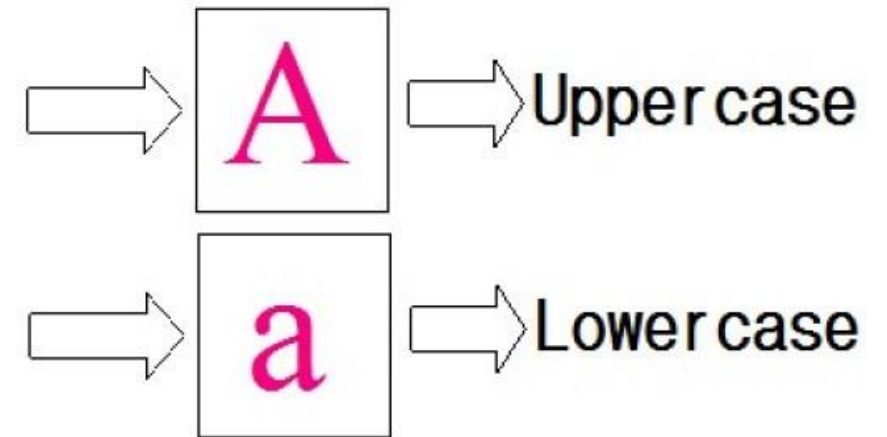


String Manipulation / Methods

3) toUpperCase()

4) toLowerCase()

Girilen String variable'daki tüm harfleri istenen forma çevirir.



* **TRICK :**) toLowerCase (Locale Local) method'u

Girilen String variable'daki tüm harfleri istenen local dilde istenen forma çevirir.



String Manipulation / Methods

5) **equals()** method'u verilen iki String'in içeriğinin birbirine eşit olup olmadığını kontrol eder.
== karşılaştırma operatörü ise verilen iki String objesinin değerinin yanında reference (adres)' lerinede bakar.

Aynı değere sahip olsa da farklı iki objeyi **==** ile karşılaştırdığımızda sonuç **FALSE** olur.

```
String bootcamp = "Clarusway";  
String success = bootcamp + "";
```

```
System.out.println(bootcamp == success); // false
```

```
System.out.println(success == bootcamp); // false
```

```
System.out.println(bootcamp.equals(success)); // true
```

```
System.out.println(success.equals(bootcamp)); // true
```



String Manipulation / Methods

String s1 = "text"; kullanımı, eğer varsa, string sabit havuzunda (string constant pool) bir örneği yeniden kullanır. Yoksa havuz içerisinde yeniden alan oluşturur. Bu durum Java'ya özgü Java Sanal Makinesi (JVM) tarafından gerçekleştirilen bir durumdur.

s1 ve s2 String constant pool içerisinde aynı referans değere (reference) sahiptir. s3 ve s4 ise farklı nesneleri ifade eder. s1 == s2 , s1 ve s2 'nin hafızadaki adres değerlerini karşılaştırır ve String constant pool üzerinde aynı adresi gösterirler. Böylece ekrana TRUE yazdırır.

s1.equals(s2) ifadesi ise, nesneler içerisindeki değerleri (clarusway) karşılaştırdığı için ekrana TRUE yazdırır. Bildiğimiz gibi == operatörü hafızadaki adresleri karşılaştırır, s3 ve s4 birbirinden farklı nesneler oldukları için farklı adreslerde tutulurlar. Bu yüzden s3 == s4 ifadesi ekrana FALSE yazdırır.

s3.equals(s4) ifadesi ise, nesneler içerisindeki değerleri (clarusway) karşılaştırdığı için ekrana TRUE yazdırır.

```
String s1 = "clarusway";  
String s2 = "clarusway";  
String s3 = new String( original: "clarusway");  
String s4 = new String( original: "clarusway");
```

```
System.out.println(s1 == s2); // True  
System.out.println(s1.equals(s2)); // True  
System.out.println(s3 == s4); // FALSE  
System.out.println(s3.equals(s4)); // True
```



String Manipulation / Methods

6) equalsIgnoreCase()

Verilen iki String değişkeni BÜYÜK HARF / küçük harf farkı gözetmeksizin karşılaştırır. Büyük / küçük harf farklılığı dışında herhangi bir karakter farklılığı olduğunda equals method'unda olduğu gibi **FALSE** döndürür.

```
String bootcamp = "Clarusway";  
String success = "clarusway";  
  
System.out.println(bootcamp.equalsIgnoreCase(success)); // true
```



String Manipulation / Methods

7) length()

Verilen String'deki karakter sayısını döndürür.

```
String bootcamp = "Clarus Way";
```



Output : 10

```
System.out.println(bootcamp.length());
```

```
String bootcamp = "";
```



Output : 0

```
System.out.println(bootcamp.length());
```

```
String bootcamp = null;
```

```
System.out.println(bootcamp.length());
```

```
Exception in thread "main" java.lang.NullPointerException  
at _00_anlik.asd.main(asd.java:11)
```



String Manipulation / Methods

8) indexOf()

Verilen String'de istenen karakterin kullanıldığı ilk index'i döndürür.

- 1) char'in index'i sorgulanabilir
- 2) Parametre String olabilir
- 3) Olmayan karakter sorgulanırsa
- 4) Parametre kelime olabilir
- 5) Belli bir index'ten sonrası sorgulanabilir

```
String bootcamp = "Clarusway ile java gayet başarılı :)";
```

```
System.out.println(bootcamp.indexOf('a')); // 2
```

```
System.out.println(bootcamp.indexOf("a")); // 2
```

```
System.out.println(bootcamp.indexOf("r")); // 3
```

```
System.out.println(bootcamp.indexOf("java")); // 14
```

```
System.out.println(bootcamp.indexOf(ch: 'a', fromIndex: 5)); // 7
```



String Manipulation / Methods

9) lastIndexOf()

Verilen String'de istenen karakterin kullanıldığı son index'i döndürür.

- 1) char'in son index'i sorgulanabilir
- 2) Parametre String olabilir
- 3) Olmayan karakter sorgulanırsa
- 4) Parametre kelime olabilir
- 5) Belli bir index'ten oncesi sorgulanabilir

```
String bootcamp = "Clarusway ile java gayet başarılı :)";  
  
System.out.println(bootcamp.lastIndexOf(ch: 'a')); // 28  
  
System.out.println(bootcamp.lastIndexOf(str: "a")); // 28  
  
System.out.println(bootcamp.lastIndexOf(str: "r")); // 29  
  
System.out.println(bootcamp.lastIndexOf(str: "java")); // 14  
  
System.out.println(bootcamp.lastIndexOf(ch: 'a', fromIndex: 5)); // 2
```



String Manipulation / Methods

10) contains()

Verilen String'in istenen karakterleri içerip içermediğini kontrol eder. İçeriyorsa **TRUE**, içermiyorsa **FALSE** döndürür.

- 1) Parametre String olmalıdır
- 2) Olmayan karakter sorgulanırsa
- 3) Parametre metin olabilir

```
String bootcamp = "Clarusway ile java gayet başarılı :)";  
  
System.out.println(bootcamp.contains('a')); // CTE  
  
System.out.println(bootcamp.contains("a")); // true  
  
System.out.println(bootcamp.contains("r")); // true  
  
System.out.println(bootcamp.contains("java")); // true
```

* **TRICK :**) contains() method'u için parametre olarak char data type kullanıldığında CTE verir, String kullanılmalıdır.



String Manipulation / Methods

11) endsWith()

Verilen String'in istenen karakterler ile bitip bitmediğini kontrol eder. İstenen karakterler ile bitiyorsa **TRUE**, yoksa **FALSE** döndürür.

- 1) Parametre String olmalıdır
- 2) Yanlış karakter sorgulanırsa
- 3) Parametre kelime olabilir

```
String bootcamp = "Clarusway ile java gayet başarılı :)";  
  
System.out.println(bootcamp.endsWith('.')'); // CTE  
  
System.out.println(bootcamp.endsWith(")")); // true  
  
System.out.println(bootcamp.contains(" ")); // false  
  
System.out.println(bootcamp.contains("java")); // false
```

* **TRICK :** endsWith() method'u için parametre olarak char data type kullanıldığında CTE verir, String kullanmalıdır.



String Manipulation / Methods

12) startsWith()

Verilen String'in istenen karakterler ile bitip başlayıp başlamadığını kontrol eder. İstenen karakterler ile başlıyorsa **TRUE**, yoksa **FALSE** döndürür.

- 1) Parametre String olmalıdır
- 2) Parametre kelime olabilir
- 3) Belirli karakterden sonrası olabilir

```
String bootcamp = "Clarusway ile java gayet başarılı :)";

System.out.println(bootcamp.startsWith('(')); // CTE

System.out.println(bootcamp.startsWith("C")); // true

System.out.println(bootcamp.startsWith("Clarus")); // true

System.out.println(bootcamp.startsWith(prefix: "a", toffset: 2)); // true

System.out.println(bootcamp.startsWith(prefix: "java", toffset: 14)); // true
```

* **TRICK :** `startsWith()` method'u için parametre olarak char data type kullanıldığında CTE verir, String kullanmalıdır.



String Manipulation / Methods

13) isEmpty()

Verilen String'in uzunluğu 0 (sıfır) ise (hiçbir karakter içermiyorsa) **TRUE**, yoksa **FALSE** döndürür.

```
String bootcamp = "Clarusway ile java gayet başarılı :)";  
System.out.println(bootcamp.isEmpty()); // false
```

```
String str = "";  
System.out.println(str.isEmpty()); // true
```

```
String str1 = null;  
System.out.println(str1.isEmpty()); // CTE
```

```
Exception in thread "main" java.lang.NullPointerException  
at _00_anlik.asd.main(asd.java:19)
```



String Manipulation / Methods

14) replace()

Verilen String'deki istenen karakterleri istenen yeni karakterler ile değiştirir.

```
String bootcamp = "Clarusway ile java gayet başarılı :);  
  
System.out.println(bootcamp.replace(target: "a", replacement: "x")); // Clxruswxy ile jxvx gxyet bxşxrılı :)  
  
System.out.println(bootcamp.replace(target: "java", replacement: ":)")); // Clarusway ile :) gayet başarılı :)  
  
System.out.println(bootcamp.replace(target: "ı", replacement: "i")); // Clarusway ile java gayet başarılı :)  
  
System.out.println(bootcamp.replace(target: "a", replacement: "")); // Clruswy ile jv gyet bşrılı :)  
  
System.out.println(bootcamp.replace('a', "x")); // CTE  
  
System.out.println(bootcamp.replace(oldChar: 'a', newChar: 'x')); // Clxruswxy ile jxvx gxyet bxşxrılı :)
```

* **TRICK :**) replace() method'u char data type için de kullanılır.



String Manipulation / Methods

15) replaceAll()

replace() method'u ile benzer olarak verilen String'deki istenen karakterleri istenen yeni karakterler ile değiştirir. Aralarındaki farklar:

- replace() method'unda char kullanılabilir, replaceAll()'da char kullanılamaz.
- replaceAll() method'unda Regular Expressions kullanılabilir.

\\s: Boşluk (space)

\\S: Boşluk dışındaki tüm karakterler

\\w: Harfler ve rakamlar (a-z, A-Z, 0-9)

\\W: Harfler ve rakamlar dışındaki tüm karakterler

\\d: Rakamlar (0-9)

\\D: Rakamlar dışındaki tüm karakterler



String Manipulation / Methods

replaceAll()

```
String bootcamp = "Clarusway ile java'dan 9876543210 kursiyer offer aldı :)";

System.out.println(bootcamp.replaceAll(regex: "a", replacement: "*")); // Cl*rusw*y ile j*v*'d*n 9876543210 kursiyer offer *ldı :)

System.out.println(bootcamp.replaceAll(regex: "\\s", replacement: "*")); // Clarusway*ile*java'dan**9876543210*kursiyer*offer*aldı*:)

System.out.println(bootcamp.replaceAll(regex: "\\S", replacement: "*")); // ***** *** ***** ***** ***** ***** **

System.out.println(bootcamp.replaceAll(regex: "\\w", replacement: "*")); // ***** *** ****'*'*** ***** ***** ***** ***1 :)

System.out.println(bootcamp.replaceAll(regex: "\\W", replacement: "*")); // Clarusway*ile*java*dan**9876543210*kursiyer*offer*ald****

System.out.println(bootcamp.replaceAll(regex: "\\d", replacement: "*")); // Clarusway ile java'dan ***** kursiyer offer aldı :)

System.out.println(bootcamp.replaceAll(regex: "\\D", replacement: "*")); // *****9876543210*****
```



String Manipulation / Methods

16) replaceFirst()

Verilen String'deki istenen karakterin ilkinin, istenen yeni karakter ile değiştirir.

```
String bootcamp = "Clarusway ile java'dan 9876543210 kursiyer offer aldı :)";
```

```
System.out.println(bootcamp.replaceFirst(regex: "a", replacement: "*")); // Cl*usway ile java'dan 9876543210 kursiyer offer aldı :)
```

```
System.out.println(bootcamp.replaceFirst(regex: "dan", replacement: "*")); // Clarusway ile java'* 9876543210 kursiyer offer aldı :)
```

```
System.out.println(bootcamp.replaceFirst(regex: "\\s", replacement: "*")); // *larusway ile java'dan 9876543210 kursiyer offer aldı :)
```

```
System.out.println(bootcamp.replaceFirst(regex: "\\d", replacement: "*")); // *larusway ile java'dan 9876543210 kursiyer offer aldı :)
```



String Manipulation / Methods

17) substring()

Index'i girilen ve verilen String'in istenen parçasını döndürür.

- Parametre olarak index 1 sayı girilirse, girilen index'den String'in sonuna kadar ki bölümü,
- Parametre olarak index 2 sayı girilirse, girilen 1. sayının index'den (inclusive) başlayıp, 2. sayıya kadar (exclusive) karakteri döndürür.

```
String bootcamp = "Clarusway ile java'dan 9876543210 kursiyer offer aldı :)";  
  
System.out.println(bootcamp.substring(beginIndex: 0)); // Clarusway ile java'dan 9876543210 kursiyer offer aldı :)  
  
System.out.println(bootcamp.substring(beginIndex: 7)); // ay ile java'dan 9876543210 kursiyer offer aldı :)  
  
System.out.println(bootcamp.substring(beginIndex: 57)); //  
  
System.out.println(bootcamp.substring(beginIndex: 60)); // *larusway ile java'dan 9876543210 kursiyer offer aldı :)
```

```
Exception in thread "main" java.lang.StringIndexOutOfBoundsException Create breakpoint : String index out of range: -3  
    at java.base/java.lang.String.substring(String.java:1841)  
    at ClarusWay.main(ClarusWay.java:16)
```



String Manipulation / Methods

substring()

```
String bootcamp = "Clarusway ile java'dan 9876543210 kursiyer offer aldı :)";
```

```
System.out.println(bootcamp.substring(3,5)); // ru
```

```
System.out.println(bootcamp.substring(7,15)); // ay ile j
```

```
System.out.println(bootcamp.substring(11,11)); //
```

```
System.out.println(bootcamp.substring(9,8)); // *larusway ile java'dan 9876543210 kursiyer offer aldı :)
```

```
Exception in thread "main" java.lang.StringIndexOutOfBoundsException Create breakpoint : begin 9, end 8, length 57
    at java.base/java.lang.String.checkBoundsBeginEnd(String.java:3319)
    at java.base/java.lang.String.substring(String.java:1874)
    at ClarusWay.main(ClarusWay.java:15)
```

- 1) **Compile Time Error (CTE):** Code create edilirken kod altının kırmızı çizilmesidir.
- 2) **Run Time Error (RTE):** Code run edildiğinde (execute) verilen hatadır.



String Manipulation / Methods

18) trim()

İstediğimiz String'in başında veya sonunda var olan boşluk (space)'leri temizler.

```
String bootcamp = "  Clarusway ile java çok başarılı :)  ";
```

```
System.out.println(bootcamp); //  Clarusway ile java çok başarılı :)
```

```
System.out.println(bootcamp.length()); // 43
```

```
System.out.println(bootcamp.trim()); // Clarusway ile java çok başarılı :)
```

```
System.out.println(bootcamp.trim().length()); // 36
```



String Manipulation / Methods

19) valueOf()

Farklı data type (int, long float, char, double, boolean, object) String data type dönüştürür.

```
int age = 33;
double db = 14.53;
boolean isLead = true;
char ch = '$';
String yas = String.valueOf(i: 33);
String dabil = String.valueOf(db);
String lead = String.valueOf(isLead);
String karakter = String.valueOf(ch);

System.out.println(yas); // 33
System.out.println(dabil); // 14.53
System.out.println(lead); // true
System.out.println(karakter); // $
```



THANKS!

Any questions?

Garry T. - Full Stack Automation Engineer

Haluk B. - JAVA Backend Developer

