**java.lang.String**

/\*

**new** operatörü kullanıldığında yeni bir nesne oluşturulur.

String başlatıcısı kullanıldığında eğer **intern**(iç) nesne zaten oluşturulmuşsa yeni bir nesne oluşturulmaz

\*/

**public** **class** Hosgeldiniz {

**public** **static** **void** main(String[] args) {

String s1 = "Java’ya hoşgeldiniz";

String s2 = **new** String("Java’ya hoşgeldiniz ");

String s3 = "Java’ya hoşgeldiniz"; // daha önce oluşturulduğu için aynı referans verilir.

System.*out*.println("s1 == s2 : " + (s1 == s2)); // false

System.*out*.println("s1 == s3 : " + (s1 == s3)); // true referansları aynıdır.

}

}

------ String metotları -------------------

Eğer String nesnesi String classındaki metotlardan birini kullanacaksa s1.equals(s2) gibi nokta operatöründen sonra metot ismi ile kullanır. Parametresinin tipi ise eclipse tarafından gösterilir.

**-------length(): int =** String içinde kaç tane karakter olduğunu döndürür.

**-------charAt(index: int): char =** Belirtilen indisteki karakteri döndürür.

*Örnek :* Kullanıcının yazdığı string **palindrom** mu değil mi?

**public** **static** **void** main(String[] args) {

Scanner input = **new** Scanner(System.*in*);

**int** i,uz;

System.*out*.print("Bir karakter dizisi girin: ");

String s = input.nextLine();

uz=s.length();

**for**(i=0;i<uz;i++) {

**if** (s.charAt(i) != s.charAt(uz-1-i)){

System.*out*.println(" palindrom değil");

**break**;

}

}

**if**(i==uz)

System.*out*.println(" palindrom ");

}

--------**equals(s1: Object): boolean =** Bu string(çağıran string)in içeriği s1’in içeriğine eşitse true döndürür.

--------**equalsIgnoreCase(s1: String): boolean =** Bu string(çağıran string)in içeriği büyük-küçük harf duyarlılığı olmadan s1’in içeriğine eşitse true döndürür.

**public** **class** equals {

**public** **static** **void** main(String[] args) {

String s1 = **new** String("Hoşgeldiniz");

String s2 = "Hoşgeldiniz";

**if** (s1.equals(s2)){

System.*out*.println("s1 ve s2 aynı içeriğe sahip :"+(s1==s2));

}

**if** (s1==s2) {

System.*out*.println("s1 ve s2 aynı referansa sahip:"+(s1==s2));

}

}

}

-------**compareTo(s1: String): int =** Bu string’in s1 stringinden büyük, küçük yada eşit olması durumuna göre sırasıyla 0’dan büyük, 0’dan küçük bir integer yada 0 döndürür. Büyüklük küçüklük Unicode tablosundaki sıraya göredir. Bu da sözlükteki sıraya benzer. Mesela “Merhaba” stringi “Müzik” stringinden küçüktür. Çünkü ‘e’ harfi ‘ü’ harfinden önce gelir. Unicode tablosu için bakınız:**<http://en.wikipedia.org/wiki/List_of_Unicode_characters>**

-------**compareToIgnoreCase(s1: String): int** = compareTo ile aynıdır ancak büyük-küçük harfe duyarsızdır.

**public** **class** Compare {

**public** **static** **void** main(String[] args) {

String s1 = **new** String("Hoşgeldiniz");

String s2 = "Hoşgeldinis";

**if** (s1.compareTo(s2) > 0) {

System.*out*.println(" s1 s2’den daha büyük ");

}

**else** **if** (s1.compareTo(s2) == 0) {

System.*out*.println(" s1 ve s2 aynı içeriğe sahip ");

}

**else**

System.*out*.println("s1 s2’den daha küçük ");

}} // s1 s2’den daha büyük

**-----------regionMatches(toffset: int, s1: String, offset: int, len: int): boolean =** Eğer bu stringin belirtilen alt alanı tam olarak s1 için belirtilen alt alanına eşitse true döndürür.

*Örnek:* s.regionMatches(6, a, 5, 3)

**toffset:** bu stringin (**s** stringinin) kaçıncısından itibaren**, bu örnekte : 6**

**s1:** yeniString**, bu örnekte : a**

**offset:** yeni stringin kaçıncısından itibaren**, bu örnekte : 5**

**len:** kaç karakter karşılaştırılacağı, **bu örnekte :** 3

Sonuç olarak döndürdüğü değer boolean

**public** **static** **void** main(String[] args) {

String s=**new** String("Hoşgeldiniz");

System.*out*.println(s.regionMatches(6, "gitmediniz", 5, 3));

} // true

**-----------startsWith(prefix: String): Boolean =** Bu string belirtilen ön ekle başlıyorsa true döndürür.

**public** **static** **void** main(String[] args) {

String s=**new** String("Hoşgeldiniz");

System.*out*.println(s.startsWith("Hoşg"));// true

System.*out*.println(s.startsWith("din")); // false

}

**----------endsWith(suffix: String): Boolean =** Bu string belirtilen son ekle bitiyorsa true döndürür.

**public** **static** **void** main(String[] args) {

String s=**new** String("Hoşgeldiniz");

System.*out*.println(s.endsWith("diniz"));// true

System.*out*.println(s.endsWith("eldin"));// false

}

**-------concat(s1: String): String =** Stringi s1 stringi ile birleştirip yeni bir string döndürür.

String a=" Merhaba ";

String b="Mine";

String c = a.concat(b); //String c = a + b;

System.*out*.println(c); // Merhaba Mine

//System.*out*.println(((a.concat(b)).concat(c)));//a+b+c// Merhaba Mine Merhaba Mine

**-------------substring(baslaIndis: int): String =** baslaIndis’ten başlayarak string sonuna kadar olan alt stringi döndürür.

String s1 = "Java’ya hoşgeldiniz";

String s2 = s1.substring(4);

System.*out*.println(s2); // ’ya hoşgeldiniz

String s3 = s1.substring(4) + "HTML";

System.*out*.println(s3); // ’ya hoşgeldinizHTML

**---------substring(baslaIndis: int, bitisIndis: int): String =** Stringin baslaIndis ile bitisIndis-1 arasındaki alt stringi döndürür. bitisIndıs indisindeki karakterin dahil olmadığına dikkat edin. (bitisIndis-1)

String s1 = "Java’ya hoşgeldiniz";

String s2 = s1.substring(0, 11);

System.*out*.println(s2); //Java’ya hoş

String s3 = s1.substring(0, 11) + "gel";

System.*out*.println(s3); //Java’ya hoşgel

**-------toLowerCase(): String =** Stringin bütün karakterlerinin küçük harfe çevrilmiş halini döndürür

**-------toUpperCase(): String =** Stringin bütün karakterlerinin büyük harfe çevrilmiş halini döndürür

**-------trim(): String =** Stringin iki taraftaki boşluk karakterlerinin silinmiş halini döndürür

**-------replace(oldChar: char, newChar: char): String =** Stringin içindeki bir *karakterin* her geçtiği yerin belirtilen karakterle değiştirilmiş halini döndürür

**-------replaceFirst(oldString: String, newString: String): String =** Stringin içindeki bir alt stringin ilk geçtiği yerin belirtilen alt stringle değiştirilmiş halini döndürür.

**-------replaceAll(oldString: String, newString: String): String =** Stringin içindeki bir *alt* *stringin* her geçtiği yerin belirtilen alt stringle değiştirilmiş halini döndürür. ***replace*** *komutu ile tek farkı orada karakter değiştirilip burada string değiştirilmesidir.*

**public** **static** **void** main(String[] args) {

String s= "Hoşgeldiniz";

System.*out*.println(s.toLowerCase()); //şu string döner: hoşgeldiniz.

System.*out*.println(s.toUpperCase());// şu string döner: HOŞGELDİNİZ.

System.*out*.println(s.replace('i','A')); // şu string döner: HoşgeldAnAz.

System.*out*.println(s.replaceFirst("in", "ABC"));// şu string döner: HoşgeldABCiz.

System.*out*.println(s.replace("i", "AB"));// şu string döner: HoşgeldABnABz.

System.*out*.println(s.replace("el", "AB"));// şu string döner: HoşgABdiniz.

String s2=" Hoş geldiniz "; // baştan ve sondan boşluklar silinir

System.*out*.println(s2.trim()); // şu string yazılır: Hoş geldiniz.

}

**--------split(delimiter: String): String[] =** Stringin belirtilen ayıraçla ayrılmış alt stringlerini dizi halinde döndürür.

**String s=**"Java#HTML#Perl";

String[] parcalar=s.split("#");

**for** (**int** i = 0; i < parcalar.length; i++)

System.*out*.println(parcalar[i] );

**Output : Java**

**HTML**

**Perl**

**+indexOf(ch: char): int =** ch karakterinin string içindeki ilk geçtiği yerin indisini döndürür. Bulamazsa -1 döndürür.

**+indexOf(ch: char, fromIndex: int): int =** ch karakterinin string içinde fromIndex indisinden sonra ilk geçtiği yerin indisini döndürür. Bulamazsa -1 döndürür.

**+indexOf(s: String): int =** s stringinin string içindeki ilk geçtiği yerin indisini döndürür. Bulamazsa -1 döndürür.

**+indexOf(s: String, fromIndex: int): int =** s stringinin string içindeki fromIndex indisinden sonra ilk geçtiği yerin indisini döndürür. Bulamazsa -1 döndürür.

**+lastIndexOf(ch: int): int =** ch karakterinin string içindeki son geçtiği yerin indisini döndürür. Bulamazsa -1 döndürür.

**+lastIndexOf(ch: int, fromIndex: int): int =** ch karakterinin string içinde fromIndex indisinden önce son geçtiği yerin indisini döndürür. Bulamazsa -1 döndürür.

**+lastIndexOf(s: String): int =** s stringinin string içinde son geçtiği yerin indisini döndürür. Bulamazsa -1 döndürür.

**+lastIndexOf(s: String, fromIndex: int): int =** s stringinin string içinde fromIndex indisinden önce son geçtiği yerin indisini döndürür. Bulamazsa -1 döndürür.

String s= "Hoşgeldiniz";

System.*out*.println(s.indexOf('H'));// 0 döndürür.

System.*out*.println(s.indexOf('x'));// -1 döndürür.

String s2="Welcome to Java";

System.*out*.println(s2.indexOf('o', 5));// 9 döndürür.

System.*out*.println(s2.indexOf("come"));// 3 döndürür.

System.*out*.println(s2.indexOf("Java", 5));// 11 döndürür.

System.*out*.println(s2.indexOf("java", 5));// -1 döndürür.

System.*out*.println(s2.lastIndexOf('a'));// 14 döndürür.

**----------Karakter ve sayıları Stringlere dönüştürmek**

String sınıfı, *karakterleri, karakter dizilerini, sayısal değerleri* stringlere çevirmek için birçok **static valueOf** metodu içerir. Bu metotların hepsi **valueOf** adına sahiptir ancak farklı char, char [], long, int, double, float gibi farklı tipte parametreleri (argümanları) vardır.

**public** **static** **void** main(String[] args) {

String s="";

**double** x=5.44;

s=String.*valueOf*(x) ;

System.*out*.println(x+" sayısının string’e dönüştürülmesi "+s);

**int** n= 5500;

s=String.*valueOf*(n);

System.*out*.println(n+ " sayısının string’e dönüştürülmesi "+s);

}

---------------Palindrom 2

**import** java.util.Scanner;

**public** **class** Palindrom {

**public** **static** **void** main(String[] args) {

Scanner input = **new** Scanner(System.*in*);

System.*out*.print("Bir karakter dizisi girin: ");

String s = input.nextLine();

**if** (*isPalindrome*(s))

System.*out*.println(s + " palindromdur");

**else**

System.*out*.println(s + " palindrom değildir");

}

**public** **static** **boolean** isPalindrome(String s) {

**int** ilk=0; // string'deki ilk karakterin indisi

**int** son=s.length()-1; // string'deki son karakterin indisi

**while** (ilk<son) {

**if** (s.charAt(ilk) != s.charAt(son))

**return** **false**; // palindrom değildir

ilk++;

son--;

}

**return** **true**; // Buraya geldiyse palindromdur.

}

}

-------- Örnek : a dizisinde geçen kelimeleri b dizisine alıyor. Tekrar eden kelimeleri bir defa yazıyor.

**public** **class** KelimeSay {

**public** **static** String temizle(String s){

StringBuilder sb=**new** StringBuilder();

**for** (**int** i = 0; i < s.length(); i++) {

**if** (Character.*isLetterOrDigit*(s.charAt(i)) || s.charAt(i)==' ') {

sb.append(s.charAt(i)); //sayı, harf veya boşluk ise ekle

}

}

**return** sb.toString();

}

**public** **static** **void** main(String[] args) {

String x="Orda bir köy var uzakta. O köy bizim köyümüzdür. Gezmesek de " +

"tozmasak da. O köy bizim köyümüzdür. Orda bir ev var uzakta. O ev " +

"bizim evimizdir. Yatmasak da kalkmasak da. O ev bizim evimizdir. ";

**int** indis=0;

**boolean** var=**false**;

String[] a=x.split(" ");

String[] b=**new** String[a.length];

**for**(**int** i=0;i<a.length;i++){ // kelimelerin noktalama işaretleri temizlenir.

a[i]=*temizle*(a[i]);

}

**for**(**int** i=0;i<b.length;i++){

b[i]="";

}

**for** (**int** i = 0; i < a.length; i++) {

var=**false**;

**for**(**int** j=0; j<=indis;j++){

**if**(a[i].equalsIgnoreCase(b[j]))

var=**true**;

}

**if**(!var){

b[indis]=a[i];

indis++;

}

}

**for**(**int** i=0;i<b.length;i++){

**if**(!b[i].isEmpty())

System.*out*.println(b[i]);

}

}

}