**java.lang.StringBuilder**

**StringBuilder** class’ı bir string oluşturur. Java API’sindeki bu class String class’ının bir alternatifidir. Stringin kullanıldığı her yerde kullanılabilir. String işlemlerini daha esnek hale getirir.

* **String** class’ını kulanarak oluşturduğunuz bir string nesnesinin içeriğini değiştiremezsiniz. Değiştirmek istediğinizde yeni bir String nesnesi oluşturur. Eskisi ise bellekte ulaşılamaz halde kalır. Bu da bellek kullanımı açısından iyi değildir.
* **StringBuilder** class’ı ile oluşturduğunuz bir string nesnesinin içeriğini değiştirebilir veya ekleme yapabilirsiniz.

Fazla **değiştirme gerekmeyecekse** String, eğer çok **değiştirme gerekecekse** StringBuilder kullanmak daha iyidir. Aşağıda StringBuilder class’ının bazı metotlarını göreceğiz.

**StringBuilder()** = 16 byte kapasiteli yeni bir string builder oluşturur.

**StringBuilder(kapasite: int)** = Verilen kapasitede yeni bir string builder oluşturur.

**StringBuilder(s: String)** = Verilen string + 16 byte kapasite ile yeni bir string builder oluşturur.

**append(data: char[])**: StringBuilder = Bu string builder’a bir karakter dizisi ekler.

**append(data: char[], offset: int, len: int):** StringBuilder = Data içindeki bir alt diziyi(başlangıcı ve uzunluğu verilen) bu StringBuilder’a ekler. Yeni bir StringBuilder döndürür.

**append(v: *aPrimitiveType*):** StringBuilder = Bir primitive veri türünü bu StringBuilder’a string olarak ekler.

**append(s: String):** StringBuilder = Bu StringBuilder’a bir string ekler.

**delete(baslaInd: int, bitisInd: int):** StringBuilder = baslaInd-bitisInd arasındaki karakterleri siler.

**deleteCharAt(index: int):** StringBuilder = Verilen indisteki karakteri siler.

**insert(index: int, data: char[], offset: int, len: int):** StringBuilder = Data’nın bir alt dizisini bu builder’da verilen indis’den sonraya ekler.

**insert(offset: int, data: char[]):** StringBuilder = Builder’ın offset pozisyonuna data’yı ekler.

**insert(offset: int, b: *aPrimitiveType*):** StringBuilder = Bu builder’a string’e çevrilmiş bir değer ekler

**insert(offset: int, s: String**): StringBuilder = Builder’ın offset pozisyonuna string s’yi ekler.

**replace(baslaInd: int, bitisInd: int, s: String**): StringBuilder = baslaInd-bitisInd arasını verilen string ile değiştirir.

**reverse():** StringBuilder = Builder’daki karakterleri tersine çevirir.

**setCharAt(index: int, ch: char):** void = Verilen indise verilen karakteri koyar. (eskisini değiştirir)



**public** **class** Ornek1 {

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

StringBuilder sb = **new** StringBuilder("Welcome to Java");

System.*out*.println(sb.toString());

System.*out*.println("uzunluk:"+sb.capacity());

sb.append("Java");

System.*out*.println(sb);

sb.insert(11, "HTML ve");

System.*out*.println(sb);

sb.delete(8, 22);

System.*out*.println(sb);

sb.deleteCharAt(8) ;

System.*out*.println(sb);

sb.insert(8, "to J");

sb.reverse();

System.*out*.println(sb);

sb.reverse();

sb.replace(11, 15, "HTML");

System.*out*.println(sb);

sb.setCharAt(0, 'w');

System.*out*.println(sb);

System.*out*.println("kapasite:"+sb.capacity());

System.*out*.println("uzunluk:"+sb.length());

sb.trimToSize();

System.*out*.println("trimToSize kullandıktan sonra :\n");

System.*out*.println("kapasite:"+sb.capacity());

System.*out*.println("uzunluk:"+sb.length());

}

}

Output:

Welcome to Java

uzunluk:31

Welcome to JavaJava

Welcome to HTML veJavaJava

Welcome Java

Welcome ava

avaJ ot emocleW

Welcome to HTML

welcome to HTML

kapasite:31

uzunluk:15

trimToSize kullandıktan sonra :

kapasite:15

uzunluk:15

------------------- Örnek 2: palindrom

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

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

System.*out*.print("Bir string yazın: ");

String st = t.nextLine();

StringBuilder s=**new** StringBuilder(st); // Girilen string StringBuilder nesnesi yapılıyor

StringBuilder x=**new** StringBuilder(s.toString()); // İkinci bir StringBuilder nesnesi oluşturuluyor

s.reverse(); // s nesnesi kendi içinde ters çeviriliyor.

**if**(s.toString().equals(x.toString())) // s ve x nesnesinin içerikleri aynı ise palindromdur

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

**else**

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

}

----------------- Örnek 3

**import** java.util.Scanner;

**public** **class** PalindromeIgnoreNonAlphaNumeric {

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

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

System.*out*.print("Bir string yazın: ");

String s = input.nextLine();

System.*out*.println("harf ve sayı olanları dikkate alır:" + s + " palindrom mudur? " + *isPalindrom*(s));

}

//String palindrom ise true döndürür

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

String s1 = *temizle*(s); // Rakam, harf boşlukların dışındakileri(,;:.) siler ve yeni string oluşturur

String s2 = *tersCevir*(s1);// s1'in tersini alır

**return** s2.equals(s1); // string ve tersi aynı ise palindromdur.

}

//temizleyip yeni bir string olarak döndürür.

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

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

// harf ve rakam olanları boş StringBuilder’a ekleyerek yeni bir StringBuilder oluşturur

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

**if** (Character.*isLetterOrDigit*(s.charAt(i))) {

sb.append(s.charAt(i));

}

}

**return** sb.toString(); // temizlenmiş builder'ı stringe çevirip döndürür

}

// Stringi ters çevirip döndüren metot

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

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

sb.reverse(); // StringBuilder'ın reverse metodu kullanılıyor

**return** sb.toString(); // stringe çevrilip döndürülüyor.

}

}

Output:

Bir string yazın: a+fd d/f/a

harf ve sayı olanları dikkate alır:a+fd d/f/a palindromdur? True

--------------Bir metinde en fazla geçen kelimenin bulunması.

**public** **class** MaxKelimeBul {

**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** String maxBul(String s){

**int** max=0, say;

String sakla="";

String x=*temizle*(s);

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

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

say=0;

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

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

say++;

}

**if**(say>=max){

max=say;

sakla=a[i];

}

}

**return** sakla;

}

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

String s="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. ";

System.*out*.println("\nen çok kullanılan kelime="+*maxBul*(s));

}

}