

java.lang

Class StringBuffer

java.lang.Object

java.lang.StringBuffer

Kılgıladığı Arayüzler:

CharSequence, Serializable

Bildirimi:

```
public final class StringBuffer
    extends Object
    implements Serializable, CharSequence
```

Buffer'da değişebilir bir karakter dizisi yaratır. Yaratılan dizi String gibidir; ama ondan farkı değişebilir olmasıdır. StringBuffer dizisine (array) bileşen ekleyen, bileşen silen, uzunluğu değiştiren ve başka işleri yapan çok sayıda metot vardır. Çoklu threadler güvenle kullanılabilir, çünkü metotları eşzamanlı (senkronize) edilmişlerdir.

Kurucuları:

[StringBuffer\(\)](#)

Başlangıç sığası 16 karakter olan boş bir StringBuffer nesnesi yaratır.

[StringBuffer\(CharSequence seq\)](#)

Parametrede verilen seq karakter dizisini içeren bir StringBuffer nesnesi yaratır.

[StringBuffer\(int capacity\)](#)

Başlangıç sığası parametrede belirtilen capacity kadar olan boş bir StringBuffer nesnesi yaratır.

[StringBuffer\(String str\)](#)

Parametrede verilen str stringini içeren bir StringBuffer nesnesi yaratır.

Metotları:

Hemen her veri tipini StringBuffer'ın sonuna ekleyen *append* metotları, StringBuffer'ın içinde istenen konuma yerleştiren *insert* metotları, istenen bileşen ya da bölgeyi silen metotlar, bileşen değerlerini bildiren, v.b. çok sayıda metodu vardır. Bu metotların tam listesi java.lang.StringBuffer API'sinden görülebilir.

Burada bazı kurucularını ve metotlarına örnekler vermekle yetineceğiz.

Program 1. *StringBuffer'a klavyeden giriş*

```

package inputstream;
public class KlavyedenOku {
    public static void main(String[] args) {
        StringBuffer strb = new StringBuffer();
        char ch;
        try {
            while ((ch = (char) System.in.read()) != '\n') {
                strb.append(ch);
            }
        } catch (Exception e) {
            System.out.println("Error: " + e.toString());
        }
        System.out.println(strb);
    }
}

```

/**

asdfghjk34567 89 hjk

asdfghjk34567 89 hjk

*/

Açıklamalar:

KlavyedenOku sınıfı strb adlı StringBuffer tipinden bir nesne (object) yaratıyor; yani ana bellekte strb 'ye bir yer ayırıyor. (ch =(char)System.in.read()) deyiminde eşitliğin sağındaki metot klavyeden girilen veriyi okuyup, onu char tipine dönüştürüyor. Böylece ch değişkenine klavyeden girilen karakter atanmış oluyor. Sonra strb.append(ch) fonksiyonu, ch değişkeninin tuttuğu karakteri strb nesnesine ekliyor. while döngüsü, klavyeden Enter tuşuna basılana kadar girilen karakterleri okur ve okunuş sırasıyla strb nesnesine ekler. try/catch hata yakalama bloku read() metodunda oluşabilecek hatayı yakalar. Hiçbir hata oluşmazsa yalnızca try altbloku çalışır, catch altblokuna girilmez; catch altblokundan sonraki ilk deyim geçilir. Bir hata oluşursa hemen catch altblokuna geçer; bu bloktaki catch(Exception e) fonksiyonu, oluşan hatayı e değişkenine atar. e.toString() fonksiyonu, bu hatayı String tipine dönüştürür. System.out.println("Error: " + e.toString()) fonksiyonu oluşan hatayı ekrana yazar ve sonraki deyim(ler)e geçmeden program akışını durdurur. Bir hata oluşmazsa, döngüyü bitirdikten sonra, catch altblokundan sonraki ilk deyim olan System.out.println(strb) deyimine atlar. Bu fonksiyon strb 'nin değerini standart çıkışa (ekrana) gönderir. Bu değer klavyeden girilen karakterler blokudur.