

ByteArrayOutputStream

java.io

java.lang.Object

java.io.OutputStream

java.io.ByteArrayOutputStream

Kılgıladığı Arayüzler:

Closeable, Flushable, AutoCloseable

```
public class ByteArrayOutputStream  
    extends OutputStream
```

Java 1.1 sürümüyle gelen `ByteArrayOutputStream` sınıfı bir çıkış akımı yaratır ve çıktıyı byte array'i biçiminde yazar. Veri yazıldıkça buffer büyür. Veriye erişmek için `toByteArray()` ve `toString()` metotları kullanılır.

`ByteArrayOutputStream` sınıfına ait bir akımın `close()` metodu ile kapanmasının etkisi yoktur. Akım kapandıktan sonra da metotları çağrılabilir ve bu çağrı bir hata üretmez.

Değişkenleri:

protected byte[] buf

protected byte[] count

Kurucuları:

[ByteArrayOutputStream](#) ()

Creates a new byte array output stream.

[ByteArrayOutputStream](#)(int size)

Creates a new byte array output stream, with a buffer capacity of the specified size, in bytes.

Metotlari:

void [close](#) ()

Closing a `ByteArrayOutputStream` has no effect.

void [reset](#) ()

Resets the `count` field of this byte array output stream to zero, so that all currently accumulated output in the output stream is discarded.

int [size](#) ()

Returns the current size of the buffer.

byte[] [toByteArray](#) ()

Creates a newly allocated byte array.

[String](#) [toString](#) ()

Converts the buffer's contents into a string decoding bytes using the platform's default character set.

[String](#) [toString](#) (int hiByte)

Deprecated.

This method does not properly convert bytes into characters. As of JDK 1.1, the preferred way to do this is via the `toString(String enc)` method, which takes an encoding-name argument, or the `toString()` method, which uses the platform's default character encoding.

[String](#) [toString](#) ([String](#) charsetName)

Converts the buffer's contents into a string by decoding the bytes using the specified [charsetName](#).

void [write](#) (byte[] b, int off, int len)

Writes `len` bytes from the specified byte array starting at offset `off` to this byte array output stream.

void [write](#) (int b)

Writes the specified byte to this byte array output stream.

void [writeTo](#) ([OutputStream](#) out)

Writes the complete contents of this byte array output stream to the specified output stream argument, as if by calling the output stream's write method using `out.write(buf, 0, count)`.

java.io.OutputStream Sınıfından Kalıtıl Gelen Değişkenler:

[flush](#), [write](#)

java.lang.Object Sınıfından Kalıtıl Gelen Metotlar:

[clone](#), [equals](#), [finalize](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [wait](#), [wait](#), [wait](#)

```
// ByteArrayOutputStream örneği
import java.io.*;

class Demo {
    public static void main(String args[]) throws IOException {
        ByteArrayOutputStream baos = new ByteArrayOutputStream();
        String str = "Çiçeğin dikenini var diye üzüleceğimize dikenin çiçeği var diye sevinelim.";
        byte buf[] = str.getBytes();
        baos.write(buf);

        System.out.println(baos.toString());

        byte b[] = baos.toByteArray();
        for (int i = 0; i < b.length; i++) {
            System.out.print((char) b[i]);
        }

        OutputStream fos = new FileOutputStream("test.txt");
        baos.writeTo(fos);
        fos.close();

        baos.reset();
        for (int i = 0; i < 5; i++)
            baos.write('!');
        System.out.println(baos.toString());
    }
}

/**
Çiçeğin dikenini var diye üzüleceğimize dikenin çiçeği var diye sevinelim.
?i?e?in dikenini var diye ?z?lece?imize dikenin ?i?e?i var diye
sevinelim!!!!!!
*/
```

Açıklamalar:

10.satırdaki **byte** `buf[] = str.getBytes()` deyimi, `str` stringini `byte` olarak `buf` arrayi üzerine yazıyor.

11.satırdaki `baos.write(buf)` deyimi `buf` `byte` arrayini `baos` üzerine yazıyor.

13.satırdaki `baos.toString()` metodu, `baos`'taki `byte` arrayini `string` tipine dönüştürerek, standart çıkış akımına (ekran) yolluyor.

byte `b[] = baos.toByteArray();` deyimi, `baos`'taki `byte` array'inden yeni bir `b` `byte` arrayi yaratıyor.

16. ile 18.satırlar arasındaki `for` döngüsü `b` `byte` arrayinin bileşenlerini tek tek `byte` tipinden `char` tipine dönüştürerek standart çıkış akımına (ekran) yolluyor.

20.satır `test.txt` adlı bir hedef dosyası (çıkış akımı) yaratıyor ve onu `fos` referansı (pointer) ile gösteriyor.

21.satırdaki `baos.writeTo(fos);` deyimi, `baos`'taki bütün veriyi `test.txt` dosyasına yazıyor.

25.satırdaki `for` döngüsü, `test.txt` dosyasının sonuna beş tane '!' karakteri ekliyor.

```
// ByteArrayOutputStream örneği
import java.io.*;

class Demo {

    public static void main(String args[]) throws IOException {

        ByteArrayOutputStream bOutput = new ByteArrayOutputStream(12);

        System.out.println("Lütfen bir şeyler yazınız:");
        while (bOutput.size() != 10) {
            // Klavyeden girdi alıyor
            bOutput.write(System.in.read());
        }

        byte b[] = bOutput.toByteArray();

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

            System.out.print((char) b[x] + " ");
        }
        System.out.println(" ");
    }
}
```

```
int c;

ByteArrayInputStream blInput = new ByteArrayInputStream(b);

System.out.println("Karakterleri büyük harfe dönüştürüyor ");
for (int y = 0; y < 1; y++) {
    while ((c = blInput.read()) != -1) {
        System.out.print(Character.toUpperCase((char) c));
    }
    blInput.reset();
}
}
```

```
/**
Lütfen bir şeyler yazınız:
abc 123 def 456 ghijkl
a b c 1 2 3 d e
Karakterleri büyük harfe dönüştürüyor
ABC 123 DE
*/
```