

InputStreamReader

```
java.lang.Object
    java.io.Reader
        java.io.InputStreamReader
```

Kılgıladıđı Arayüzler:

`Closeable`, `AutoCloseable`, `Readable`

AltSınıfları:

`FileReader`

Bildirimi:

```
public class InputStreamReader
    extends Reader
```

Java 1.1 sürümüyle gelen `InputStreamReader` sınıfı `byte` akımından karakter akımına geçiş köprüsü gibidir. Giriş akımından `byte` okur, onları istenen `charset`'e dönüştürür. Söz konusu `charset`, kullanılan platformun belirlediđi istemsiz (implicit) `charset` olabileceđi gibi, programcının belirleyeceđi istemli (explicit) `charset` de olabilir.

`InputStreamReader` sınıfının üç `read()` metodu `byte` giriş akımından, parametresinde belirtilecek isteđe göre bir `byte` ya da bir `byte` dizisi okur. `Byte`'ları karektere dönüştürme işlemlerini hızlandırmak için, her okunan `byte`'ı anında karektere dönüştürmek yerine bir blok `byte` okutulur, sonra onlar toptan karektere dönüştürülebilir.

`InputStreamReader` nesnesini daha da etkin kılmak için, onu `BufferedReader` sınıfına gömmek iyi bir yoldur. Bunu yapmak için aşağıdakine benzer bir deyim kullanılır:

```
BufferedReader in
    = new BufferedReader(new InputStreamReader(System.in));
```

Deđişkenleri (veri alanları):

Yalnızca bir deđişkeni vardır, ki o da `java.io.Reader` sınıfından kalıtsal gelmektedir.

```
lock
```

Kurucuları:

```
InputStreamReader (InputStream in)
```

Creates an `InputStreamReader` that uses the default `charset`.

```
InputStreamReader (InputStream in, Charset cs)
```

Creates an InputStreamReader that uses the given charset.

```
InputStreamReader (InputStream in, CharsetDecoder dec)
```

Creates an InputStreamReader that uses the given charset decoder.

```
InputStreamReader (InputStream in, String charsetName)
```

Creates an InputStreamReader that uses the named charset.

Metotlari:

```
void close ()
```

Closes the stream and releases any system resources associated with it.

```
String getEncoding ()
```

Returns the name of the character encoding being used by this stream.

```
int read ()
```

Reads a single character.

```
int read (char[] cbuf, int offset, int length)
```

Reads characters into a portion of an array.

```
boolean ready ()
```

Tells whether this stream is ready to be read.

java.io.Reader Sınıfından Kalıtıl Gelen Metotlar:

```
mark, markSupported, read, read, reset, skip
```

java.lang.Object Sınıfından Kalıtıl Gelen Metotlar:

```
clone, equals, finalize, getClass, hashCode, notify, notifyAll,  
toString, wait, wait, wait
```

Örnek:

Aşağıdaki program `InputStreamReader` sınıfının

```
read(char[] cbuf, int offset, int length)
```

metodunu kullanarak;

1. Klavyeden girilen metni okuyup `cbuf1`'e kaydediyor.
2. `cbuf1`'den `str1` stringine aktarıyor.
3. `str1` stringini `cbuf2`'ye aktarıyor.
4. `cbuf2`'yi `str2` stringine aktarıyor
5. `str2`'yi ekrana yazıyor.

Programı inceleyerek, her aşama için gerekli veri akımının yaratıldığını görünüz.

```
package inputStreamReader;

import java.io.*;

class Demo {
    public static void main(String args[]) {
        char cbuf1[] = new char[64];
        System.out.println("Lütfen bir metin giriniz :");
        try {
            Reader isr = new InputStreamReader(System.in);
            isr.read(cbuf1, 0, 64);
        } catch (Exception e) {
            System.out.println("Error: " + e.toString());
        }

        String s1 = new String(cbuf1);

        Reader sr = new StringReader(s1);
        char cbuf2[] = new char[64];

        try {
            sr.read(cbuf2, 0, 64);
        } catch (Exception e) {
            System.out.println("Error: " + e.toString());
        }
        String s2 = new String(cbuf2);
        System.out.println(s2);
    }
}

/**
Lütfen bir metin giriniz :
abcdef ghij klmn oöprs ştuüvyz xqw 1 2 3 4 5 6 7 8 9 0
abcdef ghij klmn oöprs ştuüvyz xqw 1 2 3 4 5 6 7 8 9 0
*/
```

Örnek:

```
package inputStreamReader;

import java.io.*;

class Demo {
    public static void main(String args[]) {
        char[] cArr = new char[80];

        System.out.println("Lütfen bir metin giriniz");
        try {
            Reader isr = new InputStreamReader(System.in);
            isr.read(cArr, 0, 80);
        } catch (Exception e) {
            System.out.println("Error: " + e.toString());
        }
        String s = new String(cArr);
        System.out.println(s);
    }
}

/**
 */
```

