

JAVA - DATAINPUTSTREAM

http://www.tutorialspoint.com/java/java_datainputstream.htm

Copyright © tutorialspoint.com

The `DataInputStream` is used in the context of `DataOutputStream` and can be used to read primitives.

Following is the constructor to create an `InputStream`:

```
InputStream in = DataInputStream(InputStream in);
```

Once you have `DataInputStream` object in hand, then there is a list of helper methods, which can be used to read the stream or to do other operations on the stream.

SN Methods with Description

1 **public final int readbyte[], intoff, intlen throws IOException**

Reads up to len bytes of data from the input stream into an array of bytes. Returns the total number of bytes read into the buffer otherwise -1 if it is end of file.

2 **Public final int readbyte[]bthrows IOException**

Reads some bytes from the inputstream and stores in to the byte array. Returns the total number of bytes read into the buffer otherwise -1 if it is end of file.

3 **a public final Boolean readBooleanthrows IOException, b public final byte readBytethrows IOException, c public final short readShortthrows IOException d public final Int readIntthrows IOException**

These methods will read the bytes from the contained `InputStream`. Returns the next two bytes of the `InputStream` as the specific primitive type.

4 **public String readLine throws IOException**

Reads the next line of text from the input stream. It reads successive bytes, converting each byte separately into a character, until it encounters a line terminator or end of file; the characters read are then returned as a `String`.

Example:

Following is the example to demonstrate `DataInputStream` and `DataOutputStream`. This example reads 5 lines given in a file `test.txt` and convert those lines into capital letters and finally copies them into another file `test1.txt`.

```
import java.io.*;

public class DataInput_Stream{

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

        //writing string to a file encoded as modified UTF-8
        DataOutputStream dataOut = new DataOutputStream(new
FileOutputStream("E:\\file.txt"));
        dataOut.writeUTF("hello");

        //Reading data from the same file
        DataInputStream dataIn = new DataInputStream(new FileInputStream("E:\\file.txt"));

        while(dataIn.available(>0)){
```

```
String k = dataIn.readUTF();
System.out.print(k+" ");
}
}
}
```

Here is the sample run of the above program:

hello

Loading [Mathjax]/jax/output/HTML-CSS/fonts/TeX/fontdata.js