

JAVA.IO.BUFFEREDINPUTSTREAM.READ METHOD

http://www.tutorialspoint.com/java/io/bufferedinputstream_read_byte.htm

Copyright © tutorialspoint.com

Description

The **java.io.BufferedReader.readbyte[]b, intoff, intlen** method reads **len** bytes from byte-input stream into a byte array, starting at a given offset. This method repeatedly invokes the **read** method of the underlying stream.

The iterated read continues until one of the following conditions becomes true:

- **len** bytes read.
- Returns **-1**, indicating end-of-file.
- If the **available** method of **BufferedReader** returns 0

Declaration

Following is the declaration for **java.io.BufferedReader.readbyte[]b, intoff, intlen** method

```
public int read(byte[] b, int off, int len)
```

Parameters

- **b** - byte array to be populated.
- **off** - starts storing from the offset.
- **len** - number of bytes to read.

Return Value

This method does not return any value.

Exception

- **IOException** -- if an I/O error occurs.

Example

The following example shows the usage of **java.io.BufferedReader.readbyte[]b, intoff, intlen** method.

```
package com.tutorialspoint;

import java.io.BufferedReader;
import java.io.FileInputStream;
import java.io.InputStream;

public class BufferedReaderDemo {
    public static void main(String[] args) throws Exception {

        InputStream inStream = null;
        BufferedReader bis = null;

        try{
            // open input stream test.txt for reading purpose.
            inStream = new FileInputStream("c:/test.txt");

            // input stream is converted to buffered input stream
            bis = new BufferedReader(inStream);

            // read number of bytes available
```

```

    int numByte = bis.available();

    // byte array declared
    byte[] buf = new byte[numByte];

    // read byte into buf , starts at offset 2, 3 bytes to read
    bis.read(buf, 2, 3);

    // for each byte in buf
    for (byte b : buf) {
        System.out.println((char)b+": " + b);
    }
} catch (Exception e) {
    e.printStackTrace();
} finally {
    // releases any system resources associated with the stream
    if (inStream != null)
        inStream.close();
    if (bis != null)
        bis.close();
}
}
}

```

Assuming we have a text file **c:/test.txt**, which has the following content. This file will be used as an input for our example program:

```
ABCDE
```

Let us compile and run the above program, this will produce the following result:

```

: 0
: 0
A: 65
B: 66
C: 67

```

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