

JAVA.IO.BUFFEREDINPUTSTREAM.RESET METHOD

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

Copyright © tutorialspoint.com

Description

The **java.io.BufferedInputStream.reset** method repositions the stream to the position where the **mark** method was last called on the input stream.

Declaration

Following is the declaration for **java.io.BufferedInputStream.reset** method

```
public void reset()
```

Parameters

- **NA**

Return Value

This method does not return any value.

Exception

- **IOException** -- -- if this stream is not marked, if the mark is invalid, or the stream is closed by **close** method, or an IO error occurs.

Example

The following example shows the usage of java.io.BufferedInputStream.reset method.

```
package com.tutorialspoint;

import java.io.BufferedInputStream;
import java.io.FileInputStream;
import java.io.IOException;
import java.io.InputStream;

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

        InputStream iStream = null;
        BufferedInputStream bis = null;

        try{

            // read from file c:/test.txt to input stream
            iStream = new FileInputStream("c:/test.txt");

            // input stream converted to buffered input stream
            bis = new BufferedInputStream(iStream);

            // read and print characters one by one
            System.out.println("Char : "+(char)bis.read());
            System.out.println("Char : "+(char)bis.read());
            System.out.println("Char : "+(char)bis.read());

            // mark is set on the input stream
            bis.mark(0);
            System.out.println("Char : "+(char)bis.read());
            System.out.println("reset() invoked");

            // reset is called
            bis.reset();
```

```

        // read and print characters
        System.out.println("char : "+(char)bis.read());
        System.out.println("char : "+(char)bis.read());

    }catch(Exception e){
        e.printStackTrace();
    }finally{

        // releases any system resources associated with the stream
        if(iStream!=null)
            iStream.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:

```

Char : A
Char : B
Char : C
Char : D
reset() invoked
char : D
char : E

```

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