

JAVA.IO.LINENUMBERINPUTSTREAM. GETLINENUMBER METHOD

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

Copyright © tutorialspoint.com

Description

The **java.io.LineNumberInputStream.getLineNumber** method returns the current line number.

Declaration

Following is the declaration for **java.io.LineNumberInputStream.getLineNumber** method:

```
public int getLineNumber()
```

Parameters

- NA

Return Value

The method returns the current line number.

Exception

- NA

Example

The following example shows the usage of **java.io.LineNumberInputStream.getLineNumber** method.

```
package com.tutorialspoint;

import java.io.FileInputStream;
import java.io.IOException;
import java.io.LineNumberInputStream;

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

        LineNumberInputStream lnis = null;
        FileInputStream fis = null;
        int i,j;
        char c;

        try{
            // create new input streams
            fis = new FileInputStream("C:/test.txt");
            lnis = new LineNumberInputStream(fis);

            // reads till the end of the stream
            while((i=lnis.read())!=-1)
            {
                // converts int to char
                c=(char)i;

                // if the character is not new line
                if(i!=10)
                {
                    // prints char
                    System.out.print("Character read: "+c);

                    // get the line number
                    j=lnis.getLineNumber();
                    System.out.println(" at line: "+j);
                }
            }
        }
    }
}
```

```

    }
}
}catch(Exception e){
    // if any error occurs
    e.printStackTrace();
}finally{
    // closes the stream and releases any system resources
    if(fis!=null)
        fis.close();
    if(lnis!=null)
        lnis.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:

```

Character read: A at line: 0
Character read: B at line: 0
Character read: C at line: 0
Character read: D at line: 0
Character read: E at line: 0
Character read: F at line: 1
Character read: G at line: 1
Character read: H at line: 1
Character read: I at line: 1
Character read: J at line: 1
Character read: K at line: 2
Character read: L at line: 2
Character read: M at line: 2
Character read: N at line: 2
Character read: O at line: 2
Character read: P at line: 3
Character read: Q at line: 3
Character read: R at line: 3
Character read: S at line: 3
Character read: T at line: 3

```

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