**Java BufferedInputStream Class**

Java BufferedInputStream class is used to read information from stream. It internally uses buffer mechanism to make the performance fast.

**The important points about BufferedInputStream are:**

* When the bytes from the stream are skipped or read, the internal buffer automatically refilled from the contained input stream, many bytes at a time.
* When a BufferedInputStream is created, an internal buffer array is created.

**Java BufferedInputStream class declaration**

Let's see the declaration for Java.io.BufferedInputStream class:

1. **public** **class** BufferedInputStream **extends** FilterInputStream

**Java BufferedInputStream class constructors**

|  |  |
| --- | --- |
| **Constructor** | **Description** |
| BufferedInputStream(InputStream IS) | It creates the BufferedInputStream and saves it argument, the input stream IS, for later use. |
| BufferedInputStream(InputStream IS, int size) | It creates the BufferedInputStream with a specified buffer size and saves it argument, the input stream IS, for later use. |

**Java BufferedInputStream class methods**

|  |  |
| --- | --- |
| **Method** | **Description** |
| int available() | It returns an estimate number of bytes that can be read from the input stream without blocking by the next invocation method for the input stream. |
| int read() | It read the next byte of data from the input stream. |
| int read(byte[] b, int off, int ln) | It read the bytes from the specified byte-input stream into a specified byte array, starting with the given offset. |
| void close() | It closes the input stream and releases any of the system resources associated with the stream. |
| void reset() | It repositions the stream at a position the mark method was last called on this input stream. |
| void mark(int readlimit) | It sees the general contract of the mark method for the input stream. |
| long skip(long x) | It skips over and discards x bytes of data from the input stream. |
| boolean markSupported() | It tests for the input stream to support the mark and reset methods. |

### **Example of Java BufferedInputStream**

|  |  |
| --- | --- |
| **package** com.company;  **import** java.io.\*;  **public class** Main {   **public static void** main(String[] args) {  **try** {  FileInputStream fileInputStream = **new** FileInputStream(**"D:\\Cat.txt"**);  BufferedInputStream bufferedInputStream = **new** BufferedInputStream(fileInputStream);  **int** i = 0;  **while** ((i = bufferedInputStream.read()) != -1){  System.***out***.print((**char**)i);  }  } **catch** (Exception ex){  System.***out***.println(**"Message : "** + ex);  }  } } | java is my language |

**int available() & int read(byte[] b, int off, int ln)**

|  |  |
| --- | --- |
| **package** com.company; **import** java.io.\*; **public class** Main {   **public static void** main(String[] args) {  **try** {  FileInputStream fileInputStream = **new** FileInputStream(**"D:\\Cat.txt"**);  BufferedInputStream bufferedInputStream = **new** BufferedInputStream(fileInputStream);  **int** i = 0;  **int** z = bufferedInputStream.available();  System.***out***.println(**"the available no will be : "** + z);  **byte**[] b = **new byte**[z];  bufferedInputStream.read(b, 5,10);  **for** (**byte** n : b){  **char** c = (**char**)n;  System.***out***.print(c);  }  } **catch** (Exception ex){  System.***out***.println(**"Message : "** + ex);  }  } } | the available no will be : 19  java is my |

**void reset() & void mark(int readlimit)**

|  |  |
| --- | --- |
| **package** com.company;  **import** java.io.\*;  **public class** Main {   **public static void** main(String[] args) {  **try** {  FileInputStream fileInputStream = **new** FileInputStream(**"D:\\Cat.txt"**);  BufferedInputStream bufferedInputStream = **new** BufferedInputStream(fileInputStream);  System.***out***.println(**"reading with buffer"**);  System.***out***.println((**char**) bufferedInputStream.read());  System.***out***.println((**char**)bufferedInputStream.read());   bufferedInputStream.mark(0);  System.***out***.println(**".mark() method invoked"**);  System.***out***.println((**char**)bufferedInputStream.read());  System.***out***.println((**char**)bufferedInputStream.read());   bufferedInputStream.reset();  System.***out***.println(**".reset() method invoked"**);  System.***out***.println((**char**)bufferedInputStream.read());  System.***out***.println((**char**)bufferedInputStream.read());   bufferedInputStream.close();  fileInputStream.close();   } **catch** (Exception ex){  System.***out***.println(**"Message : "** + ex);  }  } } | **reading with buffer**  **j**  **a**  **.mark() method invoked**  **v**  **a**  **.reset() method invoked**  **v**  **a** |