ER FUNDAMENTAL NETWORKING C LANGUAGE

JAVA C PROGRAMMING MANAGEMENT

You are here: Home Java Programming Stream Java.io.Reader.read() Method Example

Byte Stream Classes in java

BY DINESH THAKUR Category: Stream

Byte stream classes are used to perform reading and writing of 8-bit bytes. Streams being unidirectional in nature can transfer bytes in one direction only, that is, either reading data from the source into a program or writing data from a program to the destination. Therefore, Java further divides byte stream classes into two classes, namely, InputStream class and OutputStrearn class. The subclasses of InputStrearn class contain methods to support input and the subclasses of OutputStrearn class contain output related methods.

Input Stream Classes

Java's input stream classes are used to read 8-bit bytes from the stream. The InputStrearn class is the superclass for all byte-oriented input stream classes. All the methods of this class throw an IOException. Being an abstract class, the InputStrearn class cannot be instantiated hence, its subclasses are used. Some of these are listed in Table

Table Input Stream Classes

Class	Description

Copy Data From One File To Another File In Java Example

BitSet In Java Example

BufferedReac And Writer Example In Java

Random Access Files

How To Read A Character From The Buffered

Search Content

Custom Search

Search

Advance Courses

Website Developement

Struts 2 Tutorial

PHP (Hypertext Preprocessor)

How to

python

AngularJS Tutorial

Compiler Design

Troubleshooting

Basic Courses

BufferedInputStream	contains methods to read bytes from the buffer (memory area)
ByteArrayInputStream	contains methods to read bytes from a byte array
DataInputStream	contains methods to read Java primitive data types
FileInputStream	contains methods to read bytes from a file
FilterInputStream	contains methods to read bytes from other input streams which it uses as its basic source of data
ObjectInputStream	contains methods to read objects
PipedInputStream	contains methods to read from a piped output stream. A piped input stream must be connected to a piped output stream
SequenceInputStream	contains methods to concatenate multiple input streams and then read from the combined stream

The Input Stream class defines various methods to perform reading operations on data of an input stream. Some of these methods along with their description are listed in Table

Table InputStream Class Methods

Method	Description
int read()	returns the integral representation of the next available byte of input. It returns -1 when end of file is encountered

Reader Object.
Write Data To A File Using Scanner In Java Example
RandomAcce In Java Examples
How To Reading And Writing On Files
FileReader And FileWriter In Java Example
Write Byte Array To A File Using FileOutputSt | Java Examples
Create A File And Write Data In Java Example

Computer Fundamental
Computer Networking
Structured Query (SQL)
Java Script
HTML Language
Cascading Style Sheet
C Programming (Theory)
C Programming (Pratical)
Visual Basic
C++ Programming
Java Programming
Introduction to Java
Java Evolution
Control Structures in Java
Data Type
Classes in Java
Exception Handling
Java Language
Operators & Expression
Java Methods
Multithreading
Java Array
Inheritance
Packages
AWT and Applets
Swing
Stream
Java 2D Example
JDBC
Java Servlet
Java Server Pages

int read (byte buffer [])	attempts to read buffer. length bytes into the buffer and returns the total number of bytes successfully read. It returns -1 when end of file is encountered
int read (byte buffer [], int loc, int nBytes)	attempts to read 'nBytes' bytes into the buffer starting at buffer [loc] and returns the total number of bytes successfully read. It returns -1 when end of file is encountered
int available ()	returns the number of bytes of the input available for reading
Void mark(int nBytes)	marks the current position in the input stream until 'nBytes' bytes are read
void reset ()	Resets the input pointer to the previously set mark
long skip (long nBytes)	skips 'nBytes' bytes of the input stream and returns the number of actually skippedbyte
void close ()	closes the input source. If an attempt is made to read even after closing the stream then it generates IOException

Using ByteArrayinputStream Class

The ByteArrayinputStream class opens an input stream to read bytes from a byte array. It contains an internal buffer that holds bytes that are read from the stream. It should be noted that closing the stream does not have any consequences. That is, methods of this class can be invoked even after closing the stream without generating any IOException.

The ByteArrayinputstream object can be created using one of the following constructors.

Append Text To File Java Example
Counting Uppercase, LowerCase, Digits In A File Java Example
Java Program Which Can List All Files In A Given Directory | Java Example
Java.Io.Read Method Example
How To Writing On Console Output
Reading Data From File In Java Example
BufferedInpu
,

Computer Graphics
Electronic Commerce
Management (MIS)
Data Structures
Software Engineering
Digital Electronics
C# Programming
Database System

ByteArrayinputStream(byte[] buffer) //first

ByteArrayinputStream(byte[] buffer, int loc, int nBytes)

//second

The first constructor creates a ByteArrayinputStream which uses a byte array buffer as its input source. The second constructor creates a ByteArrayinputStream which uses a subset of byte array buffer as its input source. The reading begins from the index specified by loc and continues until nBytes are read.

```
// A Program to demonstrate the use of
ByteArrayInputStream class

import java.io.*;

class ByteArrayInputStreamExample

{

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

      {

           byte b[]="this is my first
program".getBytes();

           ByteArrayInputStream inp =new
ByteArrayInputStream(b);

           int n=inp.available();

           System.out.println("Number of
available bytes: "+n);

           long s=inp.skip(11);
//skipping 11 bytes

           System.out.println("Number of
skipped bytes: "+s);

           int i;

           System.out.print("String
after skipping s bytes: ");

           while((i=inp.read()) != -1)

                  {
```

BufferedOutp In Java Example
How To Reading String From The Buffered Reader Object.
Hashtable In Java Example
StringTokeni In Java Example
File Class Methods In Java With Examples
Java I/O FileOutputSt In Java Example
Stack In Java Example
RandomAcce In Java Example

```
System.out.print((char)i);
                   }
           inp.reset(); /*reset the
pointer to the beginning of the stream*/
           System.out.println(); //new
line
           int j;
           System.out.print("String in
uppercase: ");
           while((j=inp.read()) != -1)
                  {

System.out.print(Character.toUpperCase((c
j));
                  }
      }
}
```

The output of the program is

```
Number of available bytes: 24
Number of skipped bytes: 11
String after skipping s bytes: first program
String in uppercase: THIS IS MY FIRST PROGRAM
```

In this example, the getBytes () method is used to convert string into bytes. The use of available () and skip () methods is demonstrated here. Once the entire stream is read, reset () method is invoked to set the pointer at the start of the stream.

Output Stream classes

Java's output stream classes are used to write 8-bit bytes to a stream. The OutputStream class is the superclass for all byte-oriented output stream classes. All the methods of this class throw an IOException. Being an abstract class, the OutputStream class cannot

SequenceInp
In Java
Example
Java
StringTokeni
Example
Counting
Lines,Words,
In A File
Java
Example
PrintWrite In
Java
Example
DataInputStr
DataOutputS
In Java
Example
Split
Function In
Java With
Example
Reading File
In Java
Using
FileInputStre
Example
Character
Stream

be instantiated hence, its subclasses are used. Some of these are listed in Table

Table Output Stream Classes

Class	Description
BufferedOutputStream	Contains methods to write bytes into the buffer
ByteArrayOutputStream	Contains methods to write bytes into a byte array
DataOutputStream	Contains methods to write Java primitive data types
FileOutputStream	Contains methods to write bytes to a file
FilterOutputStream	Contains methods to write to other output streams
ObjectOutputStream	Contains methods to write objects
PipedOutputStream	Contains methods to write to a piped output stream
PrintStream	Contains methods to print Java primitive data types

The OutputStream class defines methods to perform writing operations. These methods are discussed in Table

TableOutputStream Class Methods

.Method	Description

Classes In
Java
FileReader
In Java
Example
Byte Stream
Classes In
Java
Copying
Contents
After
Removing
Vowels In
Java
Example
ObjectIOStre
In Java
Example
FileInputStre
In Java
Example
LinkedList
Java
Example
Copying
Contents Of
One File To
Another
Java
Example

void write (int i)	writes a single byte to the output stream
void write (byte buffer [])	writes an array of bytes to the output stream
Void write(bytes buffer[],int loc, int nBytes)	writes 'nBytes' bytes to the output stream from the buffer b starting at buffer [loc]
void flush ()	Flushes the output stream and writes the waiting buffered output bytes
void close ()	closes the output stream. If an attempt is made to write even after closing the stream then it generates IOException

Using ByteArrayOutputStream Class

TheByteArrayOutputStreamclass,anoutputcounterpartc writes streams of bytes to the buffer. Similar to ByteArrayinputStream, closing this stream has no effect. That is, methods of this class can be invoked even after closing the stream without generating any IOException.

The ByteArrayOutputStream object can be created using one of the following constructors.

ByteArrayOutputStream () *I* /first

ByteArrayOutputStream(int nBytes) //second

The first constructor creates a buffer of 32 bytes. The second constructor creates a buffer of size equal to nBytes. The size of the buffer increases as bytes are written to it.

```
// A Program to demonstrate the use of
ByteArrayOutputStream class

import java.io.*;

class ByteArrayOutputStreamExample

{
```

Java
RandomAcce
Read
Example
Random File
Handling In
Java
Example
File Input
Stream And
File Output
Stream
Properties In
Java
Example
Copying
Contents
After
Converting
Each
Character In
Capital In A
File Java
Example
Serialization
In Java
Filter
Streams
I/O Streams
File Streams

```
     public static void main(String
args[]) throws IOException
     {
        ByteArrayOutputStream out=new
ByteArrayOutputStream();
        byte b[]="Today is a bright
sunny day".getBytes();
        out.write(b);

System.out.println(out.toString() );
/*converting byte array to String*/
        out.close(); //closing the
stream
     }
}
```

PrintStream Class In Java

The output of the program is

Today is a bright sunny day

In this example, the getBytes () method is used to convert string into bytes. Once the entire stream is written, the to String ( ) method is invoked to convert the contents (bytes) of the buffer into a string.

Java Stream

- Copy Data From One File to Another File in Java Example
- ObjectIOStream in Java Example

Most Read Article

What is Computer?
Uses of Computer
Block Diagram of Computer and Explain

Latest News

What is Flat-File Database?
How we are Protecting the Data within the Database.

- Java Program Which Can List All Files in a Given Directory | Java Example
- Copying Contents after Converting Each Character in Capital in a File Java Example
- How to read a character from the Buffered Reader object.
- File Class Methods in Java with Examples
- FileReader in Java Example

its Various Components
Types of Operating System
Classification of Computers
Functions of Operating System
Characteristic of a Computer

How Recover from Database Failures.
Data Warehousing Architecture
What is Database Integrity
Comparison of OLTP Systems And Data Warehousing
What are the Features for Any DBMS to Qualify as OODBMS?

Advance Courses

Website Developement
Struts 2 Tutorial
PHP (Hypertext Preprocessor)
How to Python
AngularJS Tutorial
Compiler Design

Basic Courses

Computer Fundamental
Computer Networking
Structured Query (SQL)
C Programming
C++ Programming
Java Programming
Computer Graphics

Top Pages

What is Computer?
Types of Operating System
Applications of Computer Graphics
What is Constructors?
Features of java
Different Transmission Modes
What is DBMS?

Top Category

Basic Computer
Networking
C Programming
C++ Programming
Java Programming
Management (MIS)
Computer Graphics

Contact Us About Us Privacy Policy

Dinesh Thakur is a Technology Columinist and founder of Computer Notes and Technology Motivation. Mail Me At (@Computer Notes)