DataOutputStream and DataInputStream Class

12 September 2025 09:04

Java DataOutputStream class allows an application to write primitive Java data types to the output stream in a machine-independent way.
Java application generally uses the data output stream to write data that can later be read by a DataInputStream.

Constructors

Constructor	Description
`DataOutputStream(OutputStream out)`	Creates a new data output stream to write data to the specified underlying output stream.

Method Summary

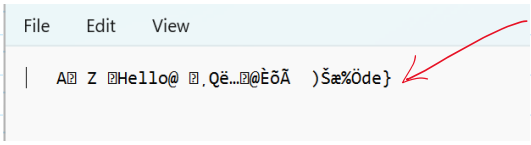
All Methods | **Instance Methods** | **Concrete Methods**

Modifier and Type	Method	Description
`void`	`flush()`	Flushes this data output stream.
`final int`	`size()`	Returns the current value of the counter `written`, the number of bytes written to this data output stream so far.
`void`	`write(byte[] b, int off, int len)`	Writes `len` bytes from the specified byte array starting at offset `off` to the underlying output stream.
`void`	`write(int b)`	Writes the specified byte (the low eight bits of the argument `b`) to the underlying output stream.
`final void`	`writeBoolean(boolean v)`	Writes a `boolean` to the underlying output stream as a 1-byte value.
`final void`	`writeByte(int v)`	Writes out a `byte` to the underlying output stream as a 1-byte value.
`final void`	`writeBytes(String s)`	Writes out the string to the underlying output stream as a sequence of bytes.
`final void`	`writeChar(int v)`	Writes a `char` to the underlying output stream as a 2-byte value, high byte first.
`final void`	`writeChars(String s)`	Writes a string to the underlying output stream as a sequence of characters.
`final void`	`writeDouble(double v)`	Converts the double argument to a `long` using the `doubleToLongBits` method in class `Double`, and then writes that `long` value to the underlying output stream as an 8-byte quantity, high byte first.
`final void`	`writeFloat(float v)`	Converts the float argument to an `int` using the `floatToIntBits` method in class `Float`, and then writes that `int` value to the underlying output stream as a 4-byte quantity, high byte first.
`final void`	`writeInt(int v)`	Writes an `int` to the underlying output stream as four bytes, high byte first.
`final void`	`writeLong(long v)`	Writes a `long` to the underlying output stream as eight bytes, high byte first.
`final void`	`writeShort(int v)`	Writes a `short` to the underlying output stream as two bytes, high byte first.
`final void`	`writeUTF(String str)`	Writes a string to the underlying output stream using modified UTF-8 encoding in a machine-independent manner.

```
package FileHandling;

import javax.swing.*;
import java.io.*;

/**
```

```
 * Write a description of class DataOutputStreamClassExample here.
 *
 * @author (your name)
 * @version (a version number or a date)
 */
public class DataOutputStreamClassExample
{
   public static void main(String[] args){
      String path = "../JavaSem2/DataFiles/";
      String fileName;
      fileName = JOptionPane.showInputDialog(null, "Enter file name: ",
      "Input file name", JOptionPane.QUESTION_MESSAGE);

      try(FileOutputStream fos = new FileOutputStream(path+fileName);
         DataOutputStream dos = new DataOutputStream(fos)){

            int i = 65;
            boolean b = true;
            char x = 'Z';
            String str = "Hello";
            double d = 3.14;
            float f = 6.28f;
            long l = 45676543454678l;
            byte bt = 100;
            short s = 25981;

            dos.writeInt(i);
            dos.writeBoolean(b);
            dos.writeChar(x);
            dos.writeUTF(str);
            dos.writeDouble(d);
            dos.writeFloat(f);
            dos.writeLong(l);
            dos.writeByte(bt);
            dos.writeShort(s);

            dos.flush();

            JOptionPane.showMessageDialog(null, "File is created!",
         "Success", JOptionPane.INFORMATION_MESSAGE);
      }
      catch(FileNotFoundException fnfe){
         JOptionPane.showMessageDialog(null, "File not found on the specified path",
         "Error Message", JOptionPane.ERROR_MESSAGE);
      }
      catch(IOException ioe){
         JOptionPane.showMessageDialog(null, "File is used by other application",
         "IOException Error Message", JOptionPane.ERROR_MESSAGE);
      }
   }
}
```

Output:



data is in encoded form.

Java DataInputStream Class

Java **DataInputStream** class allows an application to read primitive data from the input stream in a machine independent way.
Java application generally uses the data output stream to write data that can later. Be read by a data input stream.

Constructors

Constructor	Description
`DataInputStream(InputStream in)`	Creates a DataInputStream that uses the specified underlying InputStream.

Method Summary

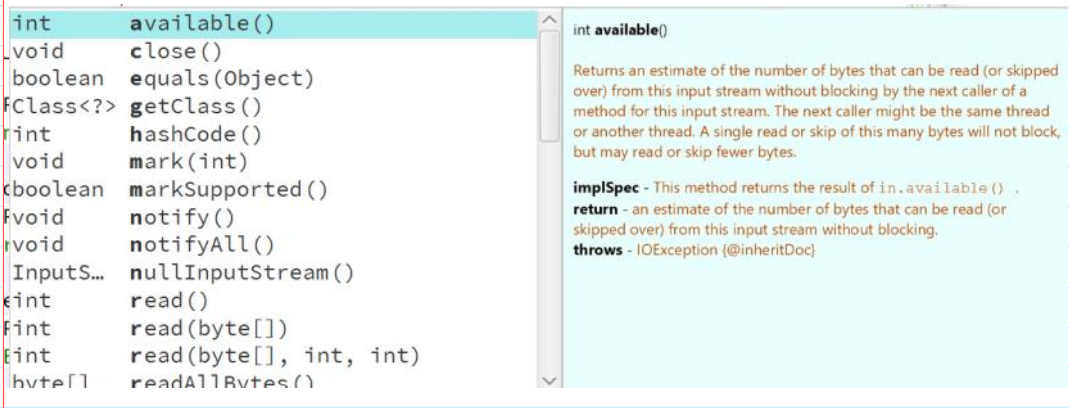
All Methods | **Static Methods** | **Instance Methods** | **Concrete Methods** | **Deprecated Methods**

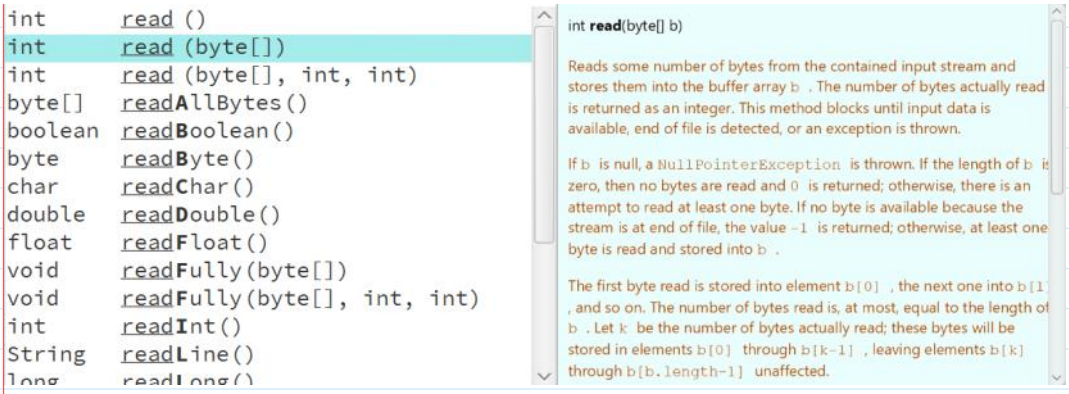
Modifier and Type	Method	Description
`final int`	`read(byte[] b)`	Reads some number of bytes from the contained input stream and stores them into the buffer array b.
`final int`	`read(byte[] b, int off, int len)`	Reads up to len bytes of data from the contained input stream into an array of bytes.
`final boolean`	`readBoolean()`	See the general contract of the `readBoolean` method of `DataInput`.
`final byte`	`readByte()`	See the general contract of the `readByte` method of `DataInput`.
`final char`	`readChar()`	See the general contract of the `readChar` method of `DataInput`.
`final double`	`readDouble()`	See the general contract of the `readDouble` method of `DataInput`.
`final float`	`readFloat()`	See the general contract of the `readFloat` method of `DataInput`.
`final void`	`readFully(byte[] b)`	See the general contract of the `readFully` method of `DataInput`.
`final void`	`readFully(byte[] b, int off, int len)`	See the general contract of the `readFully` method of `DataInput`.
`final int`	`readInt()`	See the general contract of the `readInt` method of `DataInput`.
`final String`	`readLine()`	**Deprecated.** This method does not properly convert bytes to characters.
`final long`	`readLong()`	See the general contract of the `readLong` method of `DataInput`.
`final short`	`readShort()`	See the general contract of the `readShort` method of `DataInput`.
`final int`	`readUnsignedByte()`	See the general contract of the `readUnsignedByte` method of `DataInput`.

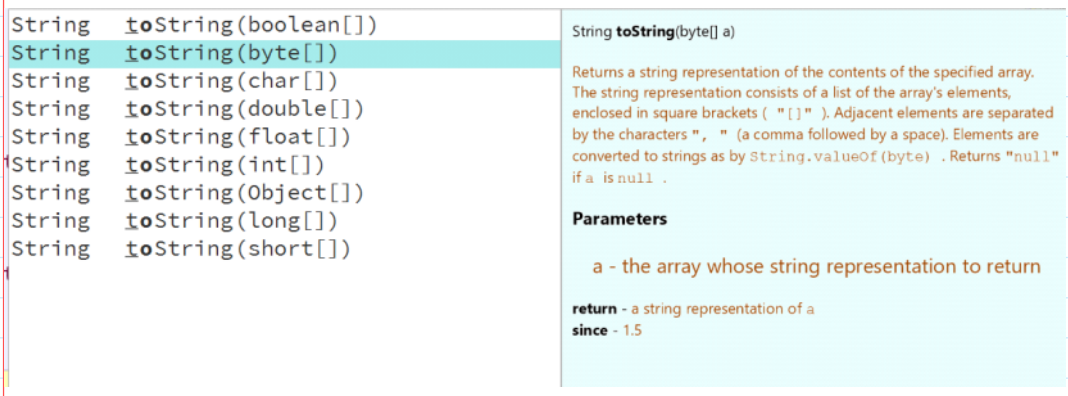
Method Summary

All Methods | Static Methods | Instance Methods | Concrete Methods | Deprecated Methods

Modifier and Type	Method	Description
final int	read(byte[] b)	Reads some number of bytes from the contained input stream and stores them into the buffer array b.
final int	read(byte[] b, int off, int len)	Reads up to len bytes of data from the contained input stream into an array of bytes.
final boolean	readBoolean()	See the general contract of the readBoolean method of DataInput.
final byte	readByte()	See the general contract of the readByte method of DataInput.
final char	readChar()	See the general contract of the readChar method of DataInput.
final double	readDouble()	See the general contract of the readDouble method of DataInput.
final float	readFloat()	See the general contract of the readFloat method of DataInput.
final void	readFully(byte[] b)	See the general contract of the readFully method of DataInput.
final void	readFully(byte[] b, int off, int len)	See the general contract of the readFully method of DataInput.
final int	readInt()	See the general contract of the readInt method of DataInput.
final String	readLine()	**Deprecated.** This method does not properly convert bytes to characters.
final long	readLong()	See the general contract of the readLong method of DataInput.
final short	readShort()	See the general contract of the readShort method of DataInput.
final int	readUnsignedByte()	See the general contract of the readUnsignedByte method of DataInput.
final int	readUnsignedShort()	See the general contract of the readUnsignedShort method of DataInput.
final String	readUTF()	See the general contract of the readUTF method of DataInput.
static final String	readUTF(DataInput in)	Reads from the stream in a representation of a Unicode character string encoded in modified UTF-8 format; this string of characters is then returned as a String.
final int	skipBytes(int n)	See the general contract of the skipBytes method of DataInput.



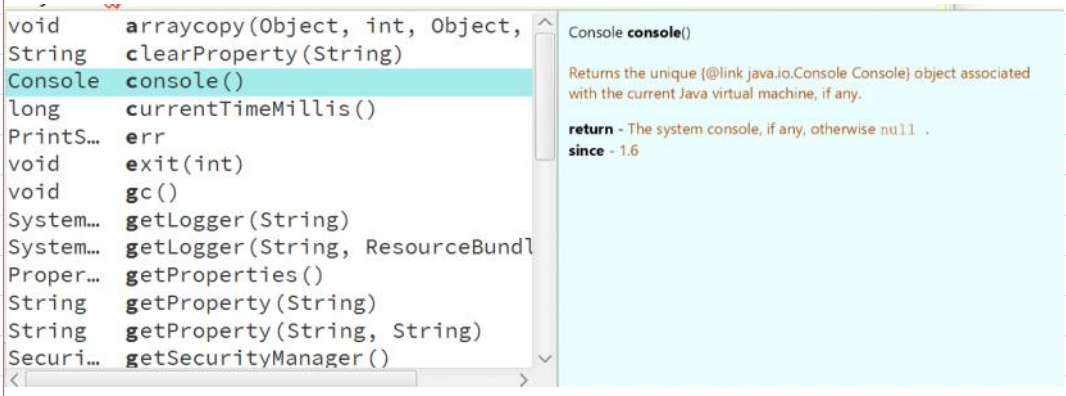





```
package FileHandling;
import java.io.*;
import javax.swing.*;


/**
 * Write a description of class ReadingTextFileUsingDataInputStream here.
 *
 * @author (your name)
 * @version (a version number or a date)
*/
public class ReadingTextFileUsingDataInputStream
{
   public static void main(String[] args){
      String path = "../JavaSem2/DataFiles/";
      String fileName;
      fileName = JOptionPane.showInputDialog(null, "Enter file name: ",
         "Input file name", JOptionPane.QUESTION_MESSAGE);

      try(FileInputStream fos = new FileInputStream(path+fileName);
      DataInputStream dis = new DataInputStream(fos)){
         int count = dis.available();
         byte[] any = new byte[count];
         dis.read(any);
         String _message_ = "";
         for(byte c: any){
            _message_ += (char)c;
         }
         JOptionPane.showMessageDialog(null, _message_,
```

```
            "Content", JOptionPane.INFORMATION_MESSAGE);
        }
        catch(FileNotFoundException fnfe){
            JOptionPane.showMessageDialog(null, "File not found on the specified path",
                "Error Message", JOptionPane.ERROR_MESSAGE);
        }
        catch(IOException ioe){
            JOptionPane.showMessageDialog(null, "File is used by other application",
                "IOException Error Message", JOptionPane.ERROR_MESSAGE);
        }
    }
}
```

Console class:



Method Summary

Methods

Modifier and Type	Method and Description
void	flush() Flushes the console and forces any buffered output to be written immediately .
Console	format(String fmt, Object... args) Writes a formatted string to this console's output stream using the specified format string and arguments.
Console	printf(String format, Object... args) A convenience method to write a formatted string to this console's output stream using the specified format string and arguments.
Reader	reader() Retrieves the unique Reader object associated with this console.
String	readLine() Reads a single line of text from the console.
String	readLine(String fmt, Object... args) Provides a formatted prompt, then reads a single line of text from the console.
char[]	readPassword() Reads a password or passphrase from the console with echoing disabled
char[]	readPassword(String fmt, Object... args) Provides a formatted prompt, then reads a password or passphrase from the console with echoing disabled.
PrintWriter	writer() Retrieves the unique PrintWriter object associated with this console.

```
import java.io.Console;
public class ConsoleClass {
   public static void main(String args[]){
        Console c = System.console();
        if(c == null){
            System.out.print("No console available");
        }
        String name = c.readLine("Enter your name: ");
```

```
        c.printf("%s", name);
        c.printf("%s","\nEnter your password: ");
        char pass[] = c.readPassword();
        String password = String.valueOf(pass);
        c.printf("%s%s","Password is: ",password);
    }
}
```

Output:

```
Enter your name: Satyender
Satyender
Enter your password:

Password is: pasword
```