

Java.io.FileInputStream Class in Java

Difficulty Level : Hard Last Updated : 21 Dec, 2021

FileInputStream class is useful to read data from a file in the form of sequence of bytes.

FileInputStream is meant for reading streams of raw bytes such as image data. For reading streams of characters, consider using FileReader.

Constructors of FileInputStream Class

- FileInputStream(File file):** Creates an input file stream to read from the specified File object.
- FileInputStream(FileDescriptor fdobj) :**Creates an input file stream to read from the specified file descriptor.
- FileInputStream(String name):** Creates an input file stream to read from a file with the specified name.

Methods of FileInputStream Class

Methods	Action Performed
available()	Returns an estimate of the number of remaining bytes that can be read (or skipped over) from this input stream.
close()	Closes this file input stream and releases any system resources associated with the stream.
finalize()	Ensures that the close method of this file input stream is called when there are no more references to it.
getChannel()	Returns the unique FileChannel object associated with this file input stream.
getFD()	Returns the FileDescriptor object that represents the connection to the actual file in the file system being used by this FileInputStream.
read()	Reads a byte of data from this input stream
read(byte[] b)	Reads up to b.length bytes of data from this input stream into an array of bytes.
read(byte[] b, int off, int len)	Reads up to len bytes of data from this input stream into an array of bytes.

Methods	Action Performed
skip()	Skips over and discards n bytes of data from the input stream

Now when we do use these methods we do generically follow these steps to read data from a file using FileInputStream which is ultimatum the goal of FileInputStreamClass

Step 1: Attach a file to a FileInputStream as this will enable us to read data from the file as shown below as follows:

```
FileInputStream fileInputStream =new FileInputStream("file.txt");
```

Step 2: Now in order to read data from the file, we should read data from the FileInputStream as shown below:

```
ch=fileInputStream.read();
```

Step 3-A: When there is no more data available to read further, the read() method returns -1;

Step 3-B: Then we should attach the monitor to the output stream. For displaying the data, we can use System.out.print.

```
System.out.print(ch);
```

Implementation:

Original File content:

```
This is my first line  
This is my second line
```

Example:

```
// Java Program to Demonstrate FileInputStream Class  
  
// Importing I/O classes  
import java.io.*;
```

```
// Main class
// ReadFile
class GFG {

    // Main driver method
    public static void main(String args[])
        throws IOException
    {

        // Attaching the file to FileInputStream
        FileInputStream fin
            = new FileInputStream("file1.txt");

        // Illustrating getChannel() method
        System.out.println(fin.getChannel());

        // Illustrating getFD() method
        System.out.println(fin.getFD());

        // Illustrating available method
        System.out.println("Number of remaining bytes:"
            + fin.available());

        // illustrating skip() method
        fin.skip(4);

        // Display message for better readability
        System.out.println("FileContents :");

        // Reading characters from FileInputStream
        // and write them
        int ch;

        // Holds true till there is data inside file
        while ((ch = fin.read()) != -1)
            System.out.print((char)ch);

        // Close the file connections
        // using close() method
        fin.close();
    }
}
```

Output:

```
sun.nio.ch.FileChannelImpl@1540e19d
java.io.FileDescriptor@677327b6
Number of remaining bytes:45
FileContents :
```

```
is my first line  
This is my second line
```

BufferedInputStream can be used to read a buffer full of data at a time from a file. This improves the speed of execution.

This article is contributed by **Nishant Sharma**. If you like GeeksforGeeks and would like to contribute, you can also write an article using write.geeksforgeeks.org or mail your article to review-team@geeksforgeeks.org. See your article appearing on the GeeksforGeeks main page and help other Geeks. Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above.