Java.io.RandomAccessFile Class Method | Set 1

Last Updated : 23 May, 2019

Java.io.RandomAccessFile Class provides a way to random access files using reading and writing operations. It works like an array of byte storted in the File.

Declaration :

```
public class RandomAccessFile
   extends Object
      implements DataOutput, DataInput, Closeable
```

Methods of RandomAccessFile Class :

1. **read() : java.io.RandomAccessFile.read()** reads byte of data from file. The byte is returned as an integer in the range 0-255

   ```
   Syntax :
   public int read()
   Parameters :
   --------
   Return  :
   reads byte of data from file, -1 if end of file is reached.
   ```

2. **read(byte[] b) java.io.RandomAccessFile.read(byte[] b)** reads bytes upto b.length from the buffer.

   ```
   Syntax :
   public int read(byte[] b)
   Parameters :
   b : buffer to be read
   Return  :
   byte of data from file upto b.length, -1 if end of file is reached.
   ```

3. **read((byte[] b, int offset, int len) : java.io.RandomAccessFile.read((byte[] b, int offset, int len)** reads bytes initialising from offset position upto b.length from the buffer.

   ```
   Syntax :
   public int read(byte[] b, int offset, int len)
   Parameters :
   b : buffer to read
   offset : starting position to read
   ```

```
len : max no of bytes to read
Return  :
reads bytes initialising from offset position upto b.length from the buffer
```

4. **readBoolean() : java.io.RandomAccessFile.readBoolean()** reads a boolean from the file.

```
Syntax :
public final boolean readBoolean()
Parameters :
------
Return  :
boolean value
```

5. **readByte() : java.io.RandomAccessFile.readByte()** reads a signed eight-bit value from file, start reading from the File Pointer.

```
Syntax :
public final byte readByte()
Parameters :
-------
Return  :
signed eight-bit value from file
```

6. **readChar() : java.io.RandomAccessFile.readChar()** reads a character from the file, start reading from the File Pointer.

```
Syntax :
public final char readChar()
Parameters :
-------
Return  :
character from the file.
```

7. **readDouble() : java.io.RandomAccessFile.readDouble()** reads a double value from the file, start reading from the File Pointer.

```
Syntax :
public final double readDouble()
Parameters :
------
Return  :
reads a double value from the file.
```

8. **readFloat() : java.io.RandomAccessFile.readFloat()** reads a float value from the file, start reading from the File Pointer.

```
Syntax :
public final double readFloat()
Parameters :
------
Return  :
reads a float value from the file.
```

9. **readFully(byte[] b) : java.io.RandomAccessFile.readFully(byte[] b)** reads bytes upto b.length from the buffer, start reading from the File Pointer.

```
Syntax :
public final void readFully(byte[] b)
Parameters :
b : buffer to be read
Return  :
reads bytes initialising from offset position upto b.length from the buffer
```

10. **readInt() : java.io.RandomAccessFile.readInt()** reads a signed 4 bytes integer from the file, start reading from the File Pointer.

```
Syntax :
reads a signed 4 bytes integer from the file
Parameters :
-----
Return  :
reads a signed 4 bytes integer from the file
```

11. **readFully(byte[] b, int offset, int len) : java.io.RandomAccessFile.readFully(byte[] b, int offset, int len)** reads bytes initialising from offset position upto b.length from the buffer, start reading from the File Pointer.

```
Syntax :
public final void readFully(byte[] b, int offset, int len)
Parameters :
b : buffer to read
offset : starting position to read
len : max no of bytes to read
Return  :
bytes initialising from offset position upto b.length from the buffer
```

12. **readLong() : java.io.RandomAccessFile.readLong()** reads a signed 64 bit integer from the file, start reading from the File Pointer.

Syntax :
public final long readLong()
Parameters :

Return :
signed 64 bit integer from the file

```
// Java Program illustrating use of io.RandomAccessFile class methods
// read(), read(byte[] b), readBoolean(), readByte(), readInt()
// readFully(byte[] b, int off, int len), readFully(), readFloat()
// readChar(), readDouble(),

import java.io.*;
public class NewClass
{
    public static void main(String[] args)
    {
        try
        {
            double d = 1.5;
            float f = 14.56f;

            // Creating a new RandomAccessFile - "GEEK"
            RandomAccessFile geek = new RandomAccessFile("GEEK.txt", "rw");

            // Writing to file
            geek.writeUTF("Hello Geeks For Geeks");

            // File Pointer at index position - 0
            geek.seek(0);

            // read() method :
            System.out.println("Use of read() method : " + geek.read());

            geek.seek(0);

            byte[] b = {1, 2, 3};

            // Use of .read(byte[] b) method :
            System.out.println("Use of .read(byte[] b) : " + geek.read(b));

            // readBoolean() method :
            System.out.println("Use of readBoolean() : " + geek.readBoolean(

            // readByte() method :
```

```
            System.out.println("Use of readByte() : " + geek.readByte());

            geek.writeChar('c');
            geek.seek(0);

            // readChar() :
            System.out.println("Use of readChar() : " + geek.readChar());

            geek.seek(0);
            geek.writeDouble(d);
            geek.seek(0);

            // read double
            System.out.println("Use of readDouble() : " + geek.readDouble())

            geek.seek(0);
            geek.writeFloat(f);
            geek.seek(0);

            // readFloat() :
            System.out.println("Use of readFloat() : " + geek.readFloat());

            geek.seek(0);
            // Create array upto geek.length
            byte[] arr = new byte[(int) geek.length()];
            // readFully() :
            geek.readFully(arr);

            String str1 = new String(arr);
            System.out.println("Use of readFully() : " + str1);

            geek.seek(0);

            // readFully(byte[] b, int off, int len) :
            geek.readFully(arr, 0, 8);

            String str2 = new String(arr);
            System.out.println("Use of readFully(byte[] b, int off, int len)
        }
        catch (IOException ex)
        {
            System.out.println("Something went Wrong");
            ex.printStackTrace();
        }
    }
}
```

Output :

```
Use of read() method : 0
Use of .read(byte[] b) : 3
Use of readBoolean() : true
Use of readByte() : 108
Use of readChar() : c
Use of readDouble() : 1.5
Use of readFloat() : 14.56
Use of readFully() : Geeks For Geeks
Use of readFully(byte[] b, int off, int len) : Geeks For Geeks
```

Next: Set 2, Set 3

This article is contributed by **MohitGupta_OMG**

. If you like GeeksforGeeks and would like to contribute, you can also write an article using contribute.geeksforgeeks.org or mail your article to contribute@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.