Java.io.Writer Class in Java

Last Updated : 23 Dec, 2021

Method	Syntax	Description
append(char Sw)	public Writer append(char Sw)	appends a single character to the Writer.
append(CharSequence char_sq)	public Writer append(CharSequence char_sq)	appends specified character sequence to the Writer.
append(CharSequence char_sq, int start, int end)	public Writer append(CharSequence char_sq, int start, int end)	appends specified part of a character sequence to the Writer.
flush()	public void flush()	flushes the Writer stream. Flushing one stream invocation will flush all other buffer in chain.
write(int char)	public void write(int char)	writes a single character to character stream.
write(String str)	public void write(String str)	writes string to the character stream.
write(String str, int offset, int maxlen)	public void write(String str, int offset, int maxlen)	writes some part of the string to the character stream.
write(char[] carray)	public void write(char[] carray)	writes character array to the character stream.
write(char[] carray, int offset, int maxlen)	public abstract void write(char[] carray, int offset, int maxlen)	writes some part of the character array to the character stream.
close()	public abstract void close	closes String Buffer. Since, closes character stream, flushing it first.

java.io.Writer class is an abstract class. It is used to write to character streams.

Declaration :

```
public abstract class Writer
  extends Object
    implements Appendable, Closeable, Flushable
```

Constructors :

- **protected Writer() :** Creates a new character stream that can itself synchronize on the writer.
- **protected Writer(Object obj) :** Creates a new character stream that can itself synchronize on the given object – 'obj'.

Methods:

- **write(int char) : java.io.Writer.write(int char)** writes a single character to character stream. Characters being written is contained in 16 lower bits of the 'char' integer value, rest of the 16 higher bits are ignored by the method.
  Syntax:

```
public void write(int char)
Parameters :
char : int value of the character to be written.
Return  :
void
Exception :
-> IOException : if in case I/O error occurs.
```

- **write(String str) : java.io.Writer.write(String str)** writes string to the character stream.
  Syntax:

```
public void write(String str)
Parameters :
str : string to be written to the character stream.
Return  :
void
Exception :
-> IOException : if in case I/O error occurs.
```

- **write(String str, int offset, int maxlen) : java.io.Writer.write(String str, int offset, int maxlen)** writes some part of the string to the character stream.
  Syntax:

```
public void write(String str, int offset, int maxlen)
Parameters :
str : string to be written to the character stream.
offset : start position of the String
maxlen : maximum length upto which string has to written
Return  :
void
Exception :
-> IOException : if in case I/O error occurs.
-> IndexOutOfBoundsException : if offset is -ve or offset + maxlen = -ve || >
```

- **write(char[] carray) : java.io.Writer.write(char[] carray)** writes character array to the character stream.
  Syntax:

```
public void write(char[] carray)
Parameters :
carray : character array to be written to the character stream
Return  :
void
Exception :
-> IOException : if in case I/O error occurs.
```

- **write(char[] carray, int offset, int maxlen) : java.io.Writer.write(char[] carray, int offset, int maxlen)** writes some part of the character array to the character stream.
  Syntax:

```
public abstract void write(char[] carray, int offset, int maxlen)
Parameters :
carray : character to be written to the character stream
offset : start position of the character array
maxlen : maximum no. of the character of the carray has to written
Return  :
void
Exception :
-> IOException : if in case I/O error occurs.
```

- **close() : java.io.Writer.close()** closes character stream, flushing it first.
  Syntax:

```
public abstract void close()
Parameters :
-----------
```

Return :

void

Exception :

-> IOException : if in case I/O error occurs.

-

- **flush() : java.io.Writer.flush()** flushes the Writer stream. Flushing one stream invocation will flush all other buffer in chain.

  Syntax:

```
public void flush()
```

Parameters :

Return :

void

Exception :

-> IOException : if in case I/O error occurs.

- **Java program illustrating use of Writer class methods :**

```
// Java program illustrating the working of Writer class methods
// write(int char), write(String str), close()
// write(String str, int offset, int maxlen), flush()
// write(char[] carray, int offset, int maxlen), write(char[] carray)

import java.io.*;

public class NewClass
{
    public static void main(String[] args) throws IOException
    {
        char[] carray = {'G', 'E', 'E', 'K', 'S'};

        // Initializing Writer
        Writer geek_writer1 = new PrintWriter(System.out);
        Writer geek_writer2 = new PrintWriter(System.out);
        Writer geek_writer3 = new PrintWriter(System.out);
        Writer geek_writer4 = new PrintWriter(System.out);
```

```
        Writer geek_writer5 = new PrintWriter(System.out);

        // Use of write(int char) : to write a character
        geek_writer1.write(71);
        geek_writer1.write(70);
        geek_writer1.write(71);

        // Use of flush() method
        System.out.print("Using write(int char[]) : ");
        geek_writer1.flush();

        String str = "Hello Geeks";
        // Use of write(String str) : to write string
        geek_writer2.write(str);

        // Value written by write(String str)
        System.out.print("\nUsing write(String str) : ");
        geek_writer2.flush();

        // Use of write(String str, int offset, int maxlen)
        //: to write part of string
        geek_writer3.write(str, 2, 4);
        geek_writer3.write(str, 5, 6);

        // Value written by write(String str, int offset, int maxlen)
        System.out.print("\nUsing write(str, offset, maxlen) : ");
        geek_writer3.flush();

        geek_writer4.write(carray);
        System.out.print("\nUsing write(char[] carray) : ");
        geek_writer4.flush();

        // Use of write(char[] carray, int offset, int maxlen):
        // to write part of char array
        geek_writer5.write(carray, 1, 3);

        // Value written by write(String str, int offset, int maxlen)
        System.out.print("\nUsing write(carray, offset, maxlen) : ");
        geek_writer5.flush();

        // Use of close() method
        geek_writer1.close();
        geek_writer2.close();
        geek_writer3.close();
        geek_writer4.close();
        geek_writer5.close();

    }
}
```

- **Output :**

```
Using write(int char[]) : GFG
Using write(String str) : Hello Geeks
Using write(str, offset, maxlen) : llo  Geeks
Using write(char[] carray) : GEEKS
Using write(carray, offset, maxlen) : EEK
```

- **append(char Sw) : java.io.Writer.append(char Sw)** appends a single character to the Writer.
  Syntax:

```
public Writer append(char Sw)
Parameters :
Sw : character to be append
Return  :
Writer
Exception :
-> IOException : if in case I/O error occurs.
```

- **append(CharSequence char_sq) : java.io.Writer.append(CharSequence char_sq)** appends specified character sequence to the Writer.
  Syntax:

```
public Writer append(CharSequence char_sq)
Parameters :
char_sq : Character sequence to append.
Return  :
Writer, if char sequence is null, then NULL appends to the Writer.
Exception :
-> IOException : if in case I/O error occurs.
```

- **append(CharSequence char_sq, int start, int end) : java.io.Writer.append(CharSequence char_sq, int start, int end)** appends specified part of a character sequence to the Writer.

Syntax:

```
public Writer append(CharSequence char_sq, int start, int end)
```

Parameters :

char_sq : Character sequence to append.

start : start of character in the Char Sequence

end : end of character in the Char Sequence

Return :

void

Exception :

-> IOException : if in case I/O error occurs.

-> IndexOutOfBoundsException : if start or end are -ve or start > end or end > char_sq.length

- **Java program illustrating use of Writer class methods :**

```
// Java program illustrating the working of Writer class methods
// append(CharSequence char_sq), append(char Sw)
// append(CharSequence char_sq, int start,int end)
// flush()

import java.io.*;

public class NewClass
{
    public static void main(String[] args) throws IOException
    {
        // Initializing String Writer
        Writer geek_writer1 = new PrintWriter(System.out);
        Writer geek_writer2 = new PrintWriter(System.out);
        Writer geek_writer3 = new PrintWriter(System.out);

        // Use of write(int char) : to write a character
        geek_writer1.append('G');
        geek_writer1.append('G');
        geek_writer1.append('G');
        geek_writer1.append('G');
        geek_writer1.append('G');
```

```
        // Use of append(char Sw)
        System.out.print("append(char Sw) : ");
        geek_writer1.flush();


        // Initializing Character Sequence
        CharSequence char_sq1 = "1 Hello 1";
        CharSequence char_sq2 = " : 2 Geeks 2";

        // Use of append(CharSequence char_sq)
        geek_writer2.append(char_sq1);
        geek_writer2.append(char_sq2);

        System.out.print("\nappend(char_sq) : ");
        geek_writer2.flush();

        // Use of append(CharSequence char_sq,int start,int end)
        geek_writer3.append(char_sq1, 0, 3);
        geek_writer3.append(char_sq2, 3, 6);

        System.out.print("\nappend(char_sq,start,end) : ");
        geek_writer3.flush();

    }
}
```

- **Output :**

```
Using write(int char) : GFG
append(char Sw) : GGGGG
append(char_sq) : 1 Hello 1 : 2 Geeks 2
append(char_sq,start,end) : 1 H2 G
```

This article is contributed by **Mohit Gupta** . 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.