

CSCI 2120:

Software Design & Development II

UNIT3: I/O management

io api
Writer

Overview

1. Introduction
2. Writer class declaration
3. Writer constructors
4. Writer methods
5. Writer examples

Introduction

- **Writer in Java** is an abstract class that is used to **write characters** into a file or output devices.
- In other words, it defines **streaming character output**. Since **Writer** is an abstract class, we cannot instantiate it.
- Java **Writer** class is a superclass for all **other writer stream classes** that are designed for performing all output operations on files.
- Writer stream classes are similar to output stream classes with only one difference. Output stream classes are designed to write bytes whereas, **writer stream classes** are designed to **write characters**.

Writer class declaration

Writer stream class extends **Object** class and implements **Closeable**, **Flushable**, **Appendable**, and **AutoCloseable** interfaces.

The general syntax to declare writer stream class in java is as follows:

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

The inheritance diagram for the **Writer** class is as follows:

```
java.lang.Object
    java.io.Writer
```

Writer subclasses

The **most important** concrete subclass of the `Writer` class is `OutputStreamWriter`.

In addition to `OutputStreamWriter` class, `java.io` package also contains several writer classes that write characters into files.

They are as follows:

- `FileWriter`
- `BufferedWriter`
- `CharArrayWriter`
- `FilterWriter`
- `PrintWriter`
- `StringWriter`
- `PipeWriter`

Writer Fields

The **Writer** stream class in Java defines the following field that is protected access modifier.

Field	Description
protected Object lock	It is used to synchronize operations on this stream.

Writer Constructors

The **Writer** stream class in Java defines two protected constructors. It does not define any public constructor.

1. **protected Writer():**

This constructor constructs a new character-stream writer whose critical sections will synchronize on the writer itself.

2. **protected Writer(Object lock):**

This constructor constructs a new character-stream writer whose critical sections will synchronize on the given object.

Writer Methods

`Writer` class defines the following methods in java, all of which are public. All these methods are identical to methods available in the `OutputStream` class. The most important writer class methods are as follows:

Note:

All the methods defined in Java `Writer` class will throw an `IOException` on error conditions.

Writer Methods

Method	Description
Writer append(char c)	This method appends the specified character c to the end of the invoking output stream. It takes only a single character.
Writer append(CharSequence charSeq)	This method appends the specified character sequence to the end of the invoking output stream.
Writer append(CharSequence charSeq, int start, int end)	This method appends a subsequence of the specified character sequence to the end of the output stream.
abstract void close()	The close() method closes the output stream, flushing it first, and then releases the system resources.
abstract void flush()	This method flushes the stream.
static Writer nullWriter()	This method returns a new Writer which discards all characters.

Writer Methods

Method	Description
<code>void write(char[] buffer)</code>	This method writes an array of characters to the output stream.
<code>abstract void write(char[] buffer, int off, int len)</code>	This method writes a portion of an array of characters, starting from the specified offset to the output stream.
<code>void write(int c)</code>	This method writes a single character to the invoking output stream.
<code>void write(String str)</code>	This method writes specified string str to the output stream.
<code>void write(String str, int off, int len)</code>	This method writes a portion of a string, beginning at the specified offset.

Example 1: Write app data to file

1. Let's take a very simple example program where we will write data into a file named `outputfile.txt` and display a message on the console.

Example 1: Write app data to file

```
import java.io.FileWriter;
import java.io.IOException;
import java.io.Writer;

public class WriterTester1 {
    public static void main(String [] args) {
        try {
            Writer w = new FileWriter("./src/outputfile.txt");
            String data = "Hello World.";
            w.write(data);
            w.close();

            System.out.println("Successfully written...");
        }
        catch (IOException e) {
            e.printStackTrace();
        }
    }
}
```

Example 1: Write app data to file

Output:

```
Successfully written...
```

outputfile.txt:

```
Hello World.
```

Difference between Writer and OutputStream in Java

There are the following differences between `Writer` and `java.io.OutputStream` that are as follows:

1. `Writer` stream class is designed to write characters whereas, `OutputStream` class is designed to write bytes.
2. `Writer` provides several `append()` methods for appending characters to the output stream. These methods exist because of implementing `java.lang.Appendable` interface by `Writer` class. In the case of `OutputStream`, these methods do not exist.
3. `Writer` defines additional `write()` methods, including `write(String str)` method for writing a `String` object's character to the output stream. In the case of `OutputStream` class, these methods do not exist.

END