CSCI 2120:

Software Design & Development II

UNIT3: I/O management

io api

PrintWriter

Overview

Introduction

- **PrintWriter in Java** is a character output stream that creates a file and writes output data to a text file in a human-readable format.

- It provides several methods to print strings and numbers in text format.

- The methods of `PrintWriter` never throw `IOException` when errors occur, but they set an internal flag we can check by calling `checkError()` method.

Note:

1. For reading strings and numeric values from a text file, use Scanner class.
2. For writing strings and numeric values to a text file, use PrintWriter class.

PrintWriter class declaration

PrintWriter class is a subclass of Writer class that extends Object class. It implements Closeable, Flushable, Appendable, and AutoCloseable interfaces.

The general syntax to declare PrintWriter class in Java is as follows:

```
public class PrintWriter
       extends Writer
       implements Closeable, Flushable, Appendable, AutoCloseable
```

Appendable is present in java.lang package. It defines an object to which characters can be added by using the append( ) method.

PrintWriter class declaration

The inheritance hierarchy for `PrintWriter` class is as follows:

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

`PrintWriter` class was added in Java 1.0 version. It is defined in the `java.io.PrintWriter` package that is imported into the program before using it.

PrintWriter Constructors

`PrintWriter` class defines the following constructors in Java that are as follows:

PrintWriter Constructors

1. PrintWriter(File file):

This constructor creates a `PrintWriter` object with the specified `file`, without automatic line flushing.

The general syntax to create an object of `PrintWriter` with the specified file is as follows:

```
FileWriter fw = new FileWriter("myfile.txt");
PrintWriter pw = new PrintWriter(fw);
```

PrintWriter Constructors

2. PrintWriter(File file, String charSet):
This constructor creates a PrintWriter object with the specified file and charset, without automatic line flushing.

~~3. PrintWriter(File file, Charset charset):~~
This form of constructor creates a PrintWriter object with the specified file and charset, without automatic line flushing.

~~4. PrintWriter(OutputStream os):~~
This constructor creates a PrintWriter object with specified OutputStream os without automatic line flushing.

~~5. PrintWriter(OutputStream os, boolean autoFlush):~~
This constructor creates a PrintWriter object with specified OutputStream os. If autoFlush is true, it flushes the Writer after every call to println.

PrintWriter Constructors

6. PrintWriter(OutputStream os, boolean autoFlush, Charset charset):
This overloaded constructor creates a PrintWriter object with specified OutputStream os and charset.

---

7. PrintWriter(Writer wr):
This form of constructor creates a PrintWriter object with specified Writer wr, without automatic line flushing.

---

8. PrintWriter(Writer wr, boolean autoFlush):
This constructor creates a PrintWriter object with the specified Writer wr. If the autoFlush is true, it flushes the Writer after every call to println.

---

9. PrintWriter(String filename):
This constructor creates a PrintWriter object with the specified file name, without automatic line flushing.

The general syntax to create an object of PrintWriter with the specified filename is as follows:

```
PrintWriter pw = new PrintWriter(filename);
```

PrintWriter Constructors

10. PrintWriter(String fileName, String charSet):
This constructor constructs a PrintWriter object with the specified file name and charset, without automatic line flushing.

---

11. PrintWriter(String fileName, Charset charset):
This form of constructor creates a PrintWriter object with the specified file name and charset, without automatic line flushing.

Note:
If `autoFlush` parameter defined in the above constructors is `true`, the output buffer is automatically flushed every time a `println(), printf(),` or `format()` is called.

PrintWriter Methods

In addition to methods inherited from `Writer` class, `PrintWriter` class also define some useful methods that are as follows:

PrintWriter Methods

Method	Description
PrintWriter append(char c)	This method appends the specified character to this print writer.
PrintWriter append(CharSequence csq)	This method appends the specified character sequence to this printwriter
PrintWriter append(CharSequence csq, int start, int end)	This method appends a subsequence of the given character sequence to this writer.
boolean checkError()	The checkError() method flushes the stream if it is not closed and checks its error state. If any exception has occurred while writing values to that stream, the checkError() method returns true.
protected void clearError()	The clearError() method is used to clear the error state of this stream.
void close()	This method closes the stream and releases any system resources associated with it. **Note:** You must close the print stream when you are completed writing output. If you exit from the program without closing PrintWriter, the file may not have all of the output

PrintWriter Methods

Method	Description
void flush()	This method flushes the stream.
void print(boolean b)	It is used to write a boolean value to a file. In other words, it prints a boolean value.
void print(char c)	It is used to write a character to the file. It prints a character.
void print(char[ ] cArray)	It is used to write an array of characters to the file. It prints an array of characters.
void print(double d)	It is used to write a double value to the file. That is, it prints a double-precision floating-point number.
void print(float f)	It is used to write a float value to the file. That is, it prints a floating-point number.
void print(int i)	It is used to write an int value to the file. That is, it prints an integer value.
void print(long l)	It is used to write a long value to the file. That is, it prints a long value.
void print(Object obj)	It is used to write an object to the file. That is, it prints an object.
void print(String s)	It is used to write a string value to the file. That is, it prints a string value.

PrintWriter Methods

Method	Description
void println()	This method is used to terminate the current line by writing the line separator string.
void println(boolean x)	This method is used to print a boolean value and then terminates the line.
void println(char x)	This method is used to print a character and then terminates the line.
void println(char[ ] x)	This method is used to print an array of characters and then terminates the line.
void println(double x)	This method is used to print a double-precision floating-point number and then terminates the line.
void println(float x)	This method is used to print a floating-point number and then terminates the line.
void println(int x)	It is used to print an integer value and then terminates the line.
void println(long x)	It is used to print a long integer value and then terminates the line.
void println(Object x)	It is used to print an Object and then terminates the line.
void println(String x)	It is used to print a string and then terminates the line.

PrintWriter Methods

Method	Description
protected void setError()	The setError() method is used to indicate that an error has occurred.
void write(char[ ] buf)	This method is used to write an array of characters.
void write(char[ ] buf, int offset, int length)	This method is used to write a portion of an array of characters.
void write(int c)	It is used to write a single character.
void write(String s)	It is used to write a string.
void write(String s, int offset, int length)	This method is used to write a portion of a string.
PrintWriter printf(String format, Object… args)	This method is used to write a formatted string to the writer using the specified format string and arguments.
PrintWriter format(String format, Object… args)	This method is used to write a formatted string to the writer using the specified format string and arguments.

Example 1: Write data to console and to file

1. Let’s take a simple example program where we will write data to the console and a text file. Look at the source code.

Example 1: Write data to console and to file

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

public class PrintWriterTester1 {
   public static void main(String[] args) throws IOException {
       // Data to write on Console using PrintWriter class.
       // Create an object of PrintWriter class using System.out.
       PrintWriter pw = new PrintWriter(System.out);

       pw.write("Hello from the console.");
       pw.flush();
       pw.close();

// Data to write in File using PrintWriter class.
       PrintWriter pw2 = new PrintWriter(new FileWriter("./src/myfile.txt"));
       pw2.write("Hello from a file.");
       pw2.flush();
       pw2.close();
   }
}
```

Example 1: Write data to console and to file

Output:

```
Hello from the console.
```

myfile.txt:

```
Hello from a file.
```

Explanation:

In this program, we have created a `PrintWriter` object wrapped around a `FileWriter` object, that specified the name of the file in the form of either `String` or `File` reference. The `write()` method inherited from `Writer` class has been used to write lines of text.

Note:

If you are writing multiple lines of text, must use newline or println method. Without the newline, the next string would start on the same line, immediately after the previous one.

END