Reading From Text Files with Java

In this tutorial, we'll look at different ways of reading from text files in Java. To learn how to write data into files, check our article Writing in Text Files with Java.

To learn how to read and write in binary files, we have the article Binary Files in Java.

The Reader Class

Java provides the Reader class, which is used for reading data from text files. It is an abstract class for reading character streams.

The following list shows the implementations of the Reader class:

- BufferedReader,
- CharArrayReader,
- FilterReader,
- InputStreamReader,

- PipedReader,
- StringReader

However, this article will not cover all the implementations. Instead, we will focus on the most used ones. Each implementation provides the read() method, which can be used to read data from the file.

Using FileReader Class

The FileReader class reads streams of characters from a file. For reading byte streams, we should consider using a FileOutputStream class instead.

Let's see how to read characters from the file using the FileReader class.

Firstly, we need to create an instance of a FileReader class by providing a path to the file we want to read from:

```
String filePath = "files/order.txt";
FileReader fileReader = new FileReader(filePath);
```

Secondly, we need to create a loop that will execute until there are lines in the file:

```
int i;
while ((i = fileReader.read()) != -1) {
    System.out.println((char) i);
}
```

The read() method returns a single character or the -1 value if the end of the stream has been reached. Inside the while statement, we can perform actions with the read data.

Using BufferedReader Class

The BufferedReader class reads the text from a character-input stream. In addition, it buffers characters in order to provide an efficient reading of characters, arrays, and lines. We can think of BufferedReader as a class opposite from the BufferedWriter class.

Now, let's create an instance of the BufferedReader class:

```
String filePath = "files/order.txt";
BufferedReader br = new BufferedReader(new FileReader(filePath));
```

Using the while statement, we can get each line:

```
String row;
while ((row = br.readLine()) != null) {
    System.out.println(row);
}
```

Conclusion

In this quick article, we learned how to read data from textual files.

As always, the entire code of this example can be found over on GitHub.

Read More