

Java File Handling

Topics : [JAVA](#)

Written on [April 11, 2023](#)

Java File Handling is the mechanism of creating, reading, writing, and manipulating files on a file system using Java programming language. Java provides a set of classes and methods for file handling, which make it easy to work with files and directories.

Some of the commonly used classes for file handling in Java include:

1. File class - This class provides methods to create, delete, and manipulate files and directories. It also provides methods to check if a file or directory exists, to get the size of a file, and to get the last modified time of a file.
2. FileReader and FileWriter classes - These classes provide methods to read from and write to a file in a character stream.
3. FileInputStream and FileOutputStream classes - These classes provide methods to read from and write to a file in a byte stream.
4. BufferedReader and BufferedWriter classes - These classes provide methods to read and write data from and to a file in a buffered manner, which improves the performance of file handling operations.

Here is an example of reading data from a file using FileReader and BufferedReader classes:

```
import java.io.BufferedReader;
import java.io.FileReader;
import java.io.IOException;

public class ReadFileExample {
    public static void main(String[] args) {
        try {
            FileReader fileReader = new FileReader("input.txt");
            BufferedReader bufferedReader = new BufferedReader(fileReader);
            String line;
            while ((line = bufferedReader.readLine()) != null) {
                System.out.println(line);
            }
        }
    }
}
```

```
        }  
        bufferedReader.close();  
    } catch (IOException e) {  
        e.printStackTrace();  
    }  
}  
}
```

This code reads data from a file named "input.txt" using FileReader and BufferedReader classes, and prints it to the console. The code is enclosed in a try-catch block to handle any IOException that might occur while reading the file.

Similarly, you can use FileWriter and BufferedWriter classes to write data to a file, and FileInputStream and FileOutputStream classes to read and write binary data to a file.

© Copyright **Aryatechno**. All Rights Reserved. Written tutorials and materials by [Aryatechno](#)