ARYATECHNO

Java Read Files

Topics : JAVA
Written on April 11, 2023

In Java, you can read files using the `FileReader` and `BufferedReader` classes.

Here's an example:

```
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();
        }
    }
}
```

In this example, we create a `FileReader` object for a file named "input.txt". We also create a `BufferedReader` object, which allows us to read the file in a buffered manner, improving performance.

We then use a `while` loop to read each line from the file using the `readLine()` method, which returns `null` when the end of the file is reached. We print each line to the console using the `println()` method of the `System.out` object.

Finally, we close the `BufferedReader` object to release any resources held by the object.

If an `IOException` occurs while reading the file, we catch it and print the stack trace.

Note that the file must exist in the specified path for this code to work. If the file does not exist, a `FileNotFoundException` will be thrown.

ARYATECHNO

You can also read binary data from files using the `FileInputStream` and `BufferedInputStream` classes, as shown in the following example:

```
import java.io.BufferedInputStream;
import java.io.FileInputStream;
import java.io.IOException;

public class ReadBinaryFileExample {
    public static void main(String[] args) {
        try {
            FileInputStream fileInputStream = new
FileInputStream("image.jpg");
            BufferedInputStream bufferedInputStream = new
BufferedInputStream(fileInputStream);
            byte[] data = new byte[1024];
            int bytesRead;
            while ((bytesRead = bufferedInputStream.read(data, 0, 1024)) !=
-1) {
                // process data here
            }
            bufferedInputStream.close();
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
```

In this example, we create a `FileInputStream` object for a file named "image.jpg". We also create a `BufferedInputStream` object, which allows us to read the file in a buffered manner, improving performance.

We then use a `while` loop to read binary data from the file using the `read()` method of the `BufferedInputStream` object. The `read()` method returns the number of bytes read, or `-1` if the end of the file is reached. We process the data in the loop as necessary.

Finally, we close the `BufferedInputStream` object to release any resources held by the object.

Note that the file must exist in the specified path for this code to work. If the file does not exist, a `FileNotFoundException` will be thrown.

ARYATECHNO

© Copyright **Aryatechno**. All Rights Reserved. Written tutorials and materials by Aryatechno