CSCI 2120:
Software Design & Development II

UNIT3: I/O management

io api

FileReader

Overview

Introduction

- **FileReader in Java** is an input stream that reads data in the form of characters from a text file.
- In other words, `FileReader` is a character-based input stream that is used to read characters from a file.

FileReader class declaration

FileReader class is a subclass of InputStreamReader *class* that extends Reader *class*. It implements Closeable, AutoCloseable, and Readable interfaces.

The general syntax to declare FileReader class in Java is given below:

```
public class FileReader
       extends InputStreamReader
       implements Closeable, AutoCloseable, Readable
```

The inheritance hierarchy for this class is as follows:

```
java.lang.Object
    java.io.Reader
        java.io.InputStreamReader
            java.io.FileReader
```

FileReader constructors

FileReader class provides the following constructors in java that are as follows:

---

1. FileReader(File file):
This constructor constructs a FileReader object that opens the specified file in reading form. If the file doesn’t exist, it throws FileNotFoundException.

---

2. FileReader(String fileName):
This constructor constructs a FileReader object with the specified name of file to read form. If the file doesn’t exist, it throws FileNotFoundException.

---

3. FileReader(FileDescriptor fd):
This constructor constructs a FileReader object with the specified file descriptor to read.

FileReader constructors

FileReader class provides the following constructors in java that are as follows:

---

4. FileReader(File file, Charset charset):
This constructor creates a FileReader object with the specified file to read using the given charset.
If the named file does not exist, this form of constructor throws an exception named IOException.

---

5. FileReader(String fileName, Charset charset):
This constructor creates a FileReader object with the specified name of the file to read using given charset.
If the named file does not exist, this form of constructor throws an exception named IOException.

FileReader constructors

Note:

The `FileReader` constructor internally creates a `FileInputStream` to read bytes from the specified file and converts them into Unicode characters with the help of its superclass `InputStreamReader.`

Because `FileReader` does not define any `read()` method or other methods of its own. It inherits all its methods from its superclass.

FileReader Methods

`FileReader` class does not define any method. It inherits methods of `Reader` class and `InputStreamReader` class.

1. **Methods declared in class `java.io.InputStreamReader:`**
    `getEncoding, read, read, ready`

2. **Methods declared in class `java.io.Reader`**:
    `close, mark, markSupported, nullReader, read, read, reset, skip, transferTo`

Example 1: Read data from File

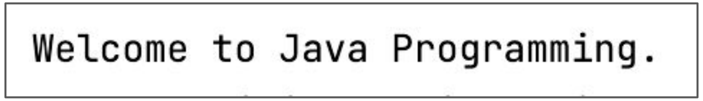
1. Let’s create a program where we will read characters from a file `myfile.txt` using `FileReader` class.

Example 1: Read data from File

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

public class FileReaderTester1 {
   public static void main(String [] args) throws IOException {
       try {
           // Create a FileReader object with file path.
           FileReader fr = new FileReader("./src/myfile.txt");

           int i;                                //return type of read() method is int.
           while((i = fr.read()) != -1) {       // Read characters at a time,
               System.out.print((char) i);     // Converting int into char.
           }
           System.out.println("");
           fr.close();                           // Closing file reader.

       } catch (FileNotFoundException e) {
           System.out.println("Error: " +e.getMessage());
       }
   }
}
```

Example 1: Read data from File

Output:



```
Welcome to Java Programming.
```

myfile.txt:

```
Welcome to Java Programming.
```

Explanation:

a. In this program, we are creating an object of `FileReader` class and passing the name of the file to be opened. If the file is not found, the `FileReader` constructor will throw a `FileNotFoundException`.

b. Once the file is open, we have called the `read()` method to fetch characters from the underlying file. The `read()` method will read character by character.

c. The `read()` method returns an `int` instead of a `char`. This is because when `read()` reaches the end of file, it returns `-1`, which is outside the range of char.
Therefore, the `read()` method returns an `int` that indicates the end of file has been reached and should stop trying to read any more characters from the underlying stream.

d. Alternatively, we can also use `readLine()` method to read a text of line.

e. The `close()` method is used to close FileReader streams.

END