

29-05-2020
Vol-3 Java IO

Now, when we have to merge two file's data into a third file, where merging should be done line by line alternatively.

FileMerger2.java

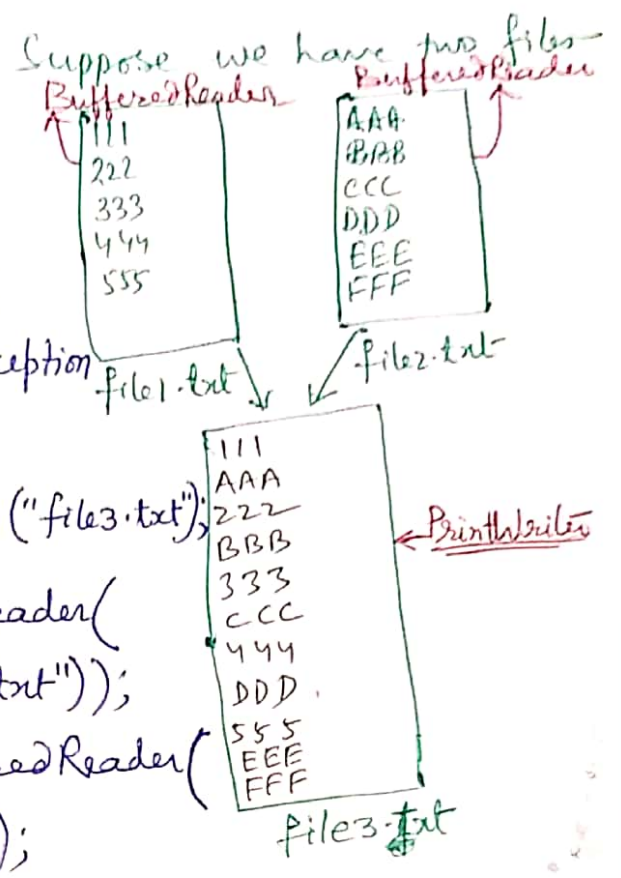
```

import java.io.*;

class FileMerger2 {
    public static void main (String
        args[]) throws IOException
    {
        PrintWriter pw = new PrintWriter ("file3.txt");
        BufferedReader br1 = new BufferedReader (
            new FileReader ("file1.txt"));
        BufferedReader br2 = new BufferedReader (
            new FileReader ("file2.txt"));

        String line1 = br1.readLine();
        String line2 = br2.readLine();
        while ( (line1 != null) || (line2 != null) )
        {
            if (line1 != null)
            {
                pw.println(line1);
                line1 = br1.readLine();
            }
            if (line2 != null)
            {
                pw.println(line2);
                line2 = br2.readLine();
            }
        }
    }
}

```



```

pw.flush();
br1.close();
br2.close();
pw.close();

```

```

BufferedReader br3 = new BufferedReader(
    new FileReader("file3.txt"));
String line3 = br3.readLine();
while (line3 != null)
{
    System.out.println(line3);
    line3 = br3.readLine();
}
br3.close();
} //close of main

```

reading data from file3.txt and printing on output (console).

```

} //close of class FileMerger2

```

Note:- Please make two text files - file1.txt and file2.txt before running the above program so that you can see the desired output - with required data.

Now, we want to write a program to merge data from all files present in ~~D:~~ D:\Java\xyz folder into one output file.

First we need:-

```

PrintWriter pw = new PrintWriter("output.txt");
File f = new File("D:\\java\\xyz");
String [] s = f.list();
for (String s1 : s)
{

```

```

File f1 = new File(f, s1);
BufferedReader br = new BufferedReader(new
FileReader(f1));

String line = br.readLine();
while (line != null)
{
    pw.println(line);
    line = br.readLine();
}
pw.flush();

```

Explanation →

- pw object opens a file "output.txt" ready for writing.
- f object contains list of all files from D:\Java\xyz folder.
- s[] → stores all list of files present in the above folder.
- for loop works for each and every file present in "D:\Java\xyz" folder, through String s1 variable
- f1 object works for each and every file name got through variable s1 from directory "D:\Java\xyz" represented by File object f.
File f1 = new File(f, s1);
- br is ready to read from f1 object through FileReader.

→ br.readLine() reads first line from the first-
available file .

→ It reads until the file has text and
writes the same onto the "output.txt" through
pw object of PrintWriter class.

The process continues for entire files using
for (String s1 : s), that means for each
String s1 containing (got) from array s[],
which contains all list of files from D:\Java\xyz
directory.

Note:- Before doing this program completely, make sure
that you should have D:\Java\xyz folder and
there you have created some text files..

Now let us write the entire program:-

```
import java.io.*;
class TotalFileMerger {
    public static void main(String args[]) throws
        IOException
    {
        PrintWriter pw = new PrintWriter("output.txt");
        File f = new File("D:\\Java\\xyz");
        String []s = f.listFiles();
        for (String s1 : s)
        {
            BufferedReader br = new BufferedReader(new FileReader(
                new File(f, s1)));
```

```
String line = br.readLine();
```

```
while ((line != null)) {
```

```
    pw.println(line);
```

```
    line = br.readLine();
```

```
}
```

```
}
```

```
pw.flush();
```

```
System.out.println("Now Data Read from output.txt file:");
```

```
br = new BufferedReader(new FileReader("output.txt"));
```

```
String line3 = br.readLine();
```

```
while (line3 != null)
```

```
{
    System.out.println(line3);
```

```
    line3 = br.readLine();
```

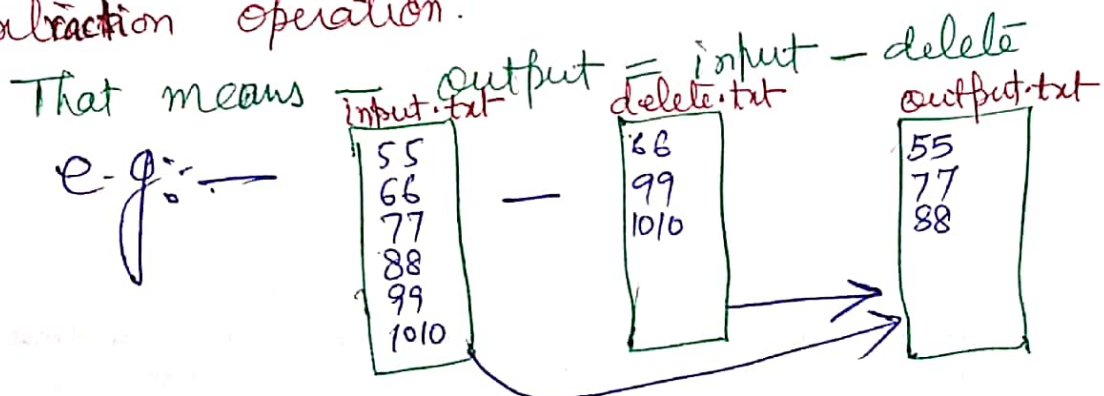
```
}
```

```
br.close();
```

```
} // close of main
```

```
} // close of TotalFileMerger class.
```

Now try to write a Program to perform file extraction operation.



FileExtraction.java

```
import java.io.*;
```

```
class FileExtraction{
```

```
public static void main(String [] args) throws  
IOException
```

```
{
```

```
PrintWriter pw = new PrintWriter("output.txt");
```

```
BufferedReader br1 = new BufferedReader(new  
FileReader("input.txt"));
```

```
String line = br1.readLine();
```

```
while ( line != null) {
```

```
boolean available = false;
```

```
BufferedReader br2 = new BufferedReader(new  
FileReader("delete.txt"));
```

```
String target = br2.readLine();
```

```
while (target != null)
```

```
{
```

```
if (line.equals(target)) {
```

```
available = true;
```

```
break;
```

```
}
```

```
target = br2.readLine();
```

```
}
```

```
if (available == false)
```

```
{
```

```
pw.println(line);
```

```
}
```

```

        line = br1.readLine();
    }
    pw.flush();
    BufferedReader br3 = new BufferedReader(new
        FileReader("output.txt"));
    String txt = br3.readLine();
    while (txt != null)
    {
        System.out.println(txt);
        txt = br3.readLine();
    }
}
}

```

Explanation of the above program:-

We have to read data from "input.txt" by BufferedReader object, read the first line, store it line, compare it with contents of "delete.txt" file line by line till end of file. If content of line is matched with content of "delete.txt" file, which is in "target" string, then make a boolean flag available to true and it is not to be written on "output.txt". If (available == false) means content of "input.txt" file does not matches with "delete.txt" then data is written on "output.txt".

Now, we can try to make a program

to remove Duplicates present in the given file.

```

import java.io.*;
class RemoveDuplicate {
public static void main (String [] args)
throws IOException {
BufferedReader br1 = new BufferedR-
eader (new FileReader ("input.txt"));
PrintWriter pw = new PrintWriter ("output.txt");
String line = br1.readLine ();
while (line != null) {
boolean flag = false;
BufferedReader br2 = new BufferedReader (new FileReader(
"output.txt"));
String target = br2.readLine ();
while (target != null) {
if (line.equals(target)) {
flag = true;
break;
}
target = br2.readLine ();
}
}
}

```

input.txt

```

222
333
222
333
566
333
262
666
226
566

```

input.txt



output.txt

```

22
333
566
262
666
226

```

output.txt


```
if (flag == false) {  
    pw.println(line);  
    pw.flush();  
}
```

```
    line = br1.readLine();  
} // close of while (line != null)  
} // close of main()  
} // close of class.
```

