abc.txt

100 characters

reading one character

performance is low

```
int i = fr.read();

while(i!=-1)
{
    System.out.println((char)i);

    i=fr.read();

}
```

read one character from file,
again go to a file and read,
        continue this process.....

abc.txt

100 characters

read and store the data in char array

performanc is high

```
File f = new File("abc.txt");
FileReader fr =new FileReader(f);

char[] ch =new char[ (int) f.length() ];

fr.read(ch);//read and store in character array

//from char[] read the data

 for(char data : ch)
    System.out.println(data);
```

read one character from a file,
store in array,continue the reading.....

now use the file data from char[]

BufferedObjects
===============

abc.txt

fw

bw

BufferedReader
or
BufferedWriter

FileReader
or
FileWriter

FileWriter
BufferedWriter

write(int)
write(ch[])
only one character at a time
String data writing is also possible
write(String)

boolean,long,float,double

To OverCome the limitation of writing any type of data we need to use "PrintWriter"
To OverCome the limitation of \n, we use newLine(),but remembering a method name
   is challenging.
To resolve this problem we need to use println()

FileReader
BufferedReader

readLine()

Diagram:
_____

Object

(write operation)

Writer(AC)

OutputStreamWriter BufferedWriter   PrintWriter

FileWriter

(read operation)

Reader(AC)

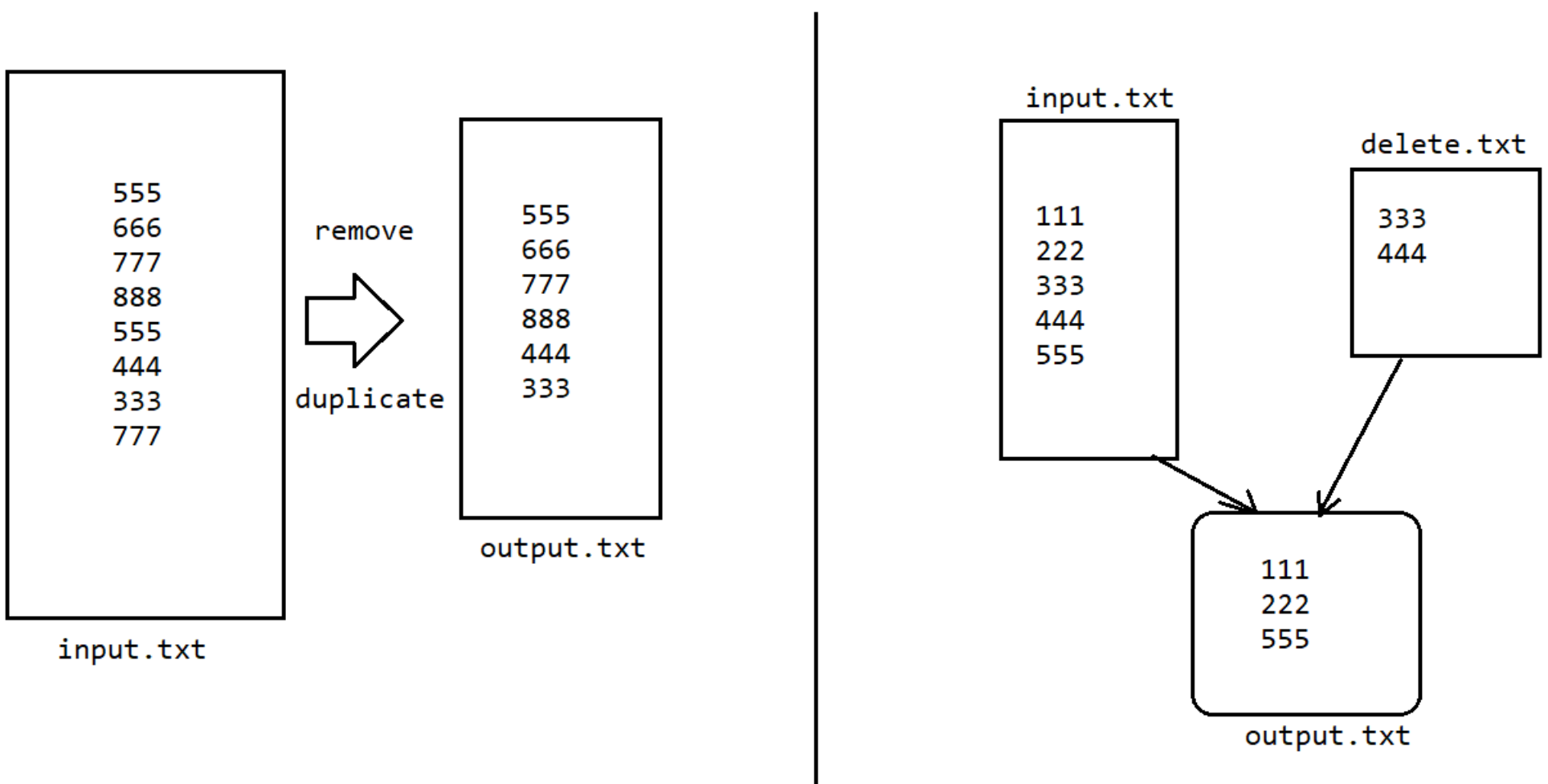InputStreamReader BufferedReader

FileReader

file1.txt

aaa
bbb
ccc

reading

file1.txt

111
222
333

aaa
bbb
ccc
111
222
333

writing

file3.txt

file3.txt create
through program

file1.txt

aaa
bbb
ccc

reading

file1.txt

111
222
333

aaa
111
bbb
222
ccc
333

writing

file3.txt

file3.txt create
through program

input.txt

555
666
777
888
555
444
333
777

remove

duplicate

555
666
777
888
444
333

output.txt

input.txt

111
222
333
444
555

delete.txt

333
444

111
222
555

output.txt