ASSIGNMENT

ON

Java I/O Classes and Methods

COURSE NAME: **ADVANCE ENTERPRISE JAVA (THEORY)**

COURSE CODE: **SE 409**

SUBMITTED TO:

MD. SAFAET HOSSAIN

ASSOCIATE PROFESSOR AND HEAD

DEPARTMENT OF CSE

SUBMITTED BY:

ZULKAR NINE

163432601

43RD BATCH

DEPARTMENT OF CSE

SUBMITTED DATE: 30-09-2020

Java i/o classes and Methods

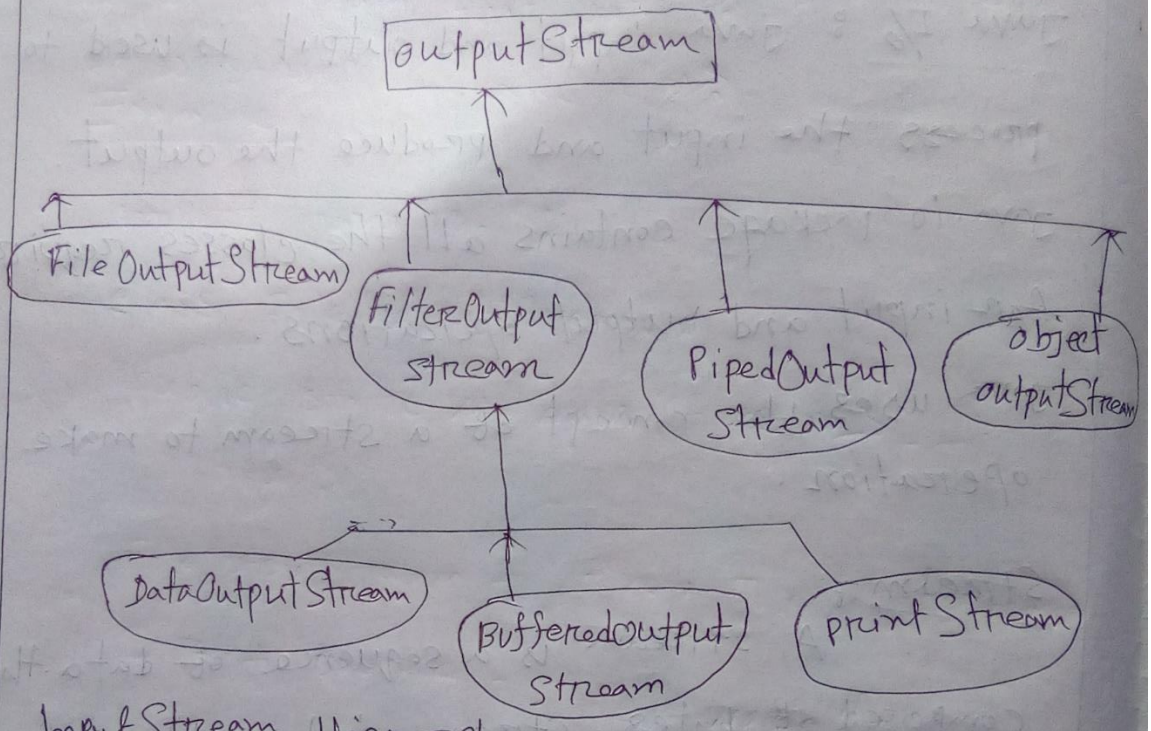
Java I/O : Java input and output is used to process the input and produce the output.
java.io package contains all the classes required for input and output operations.
Java uses the concept of a stream to make I/O operation.

Stream :
A stream is a sequence of data that composed of bytes. Stream is Linked to a physical device by the java I/O System.

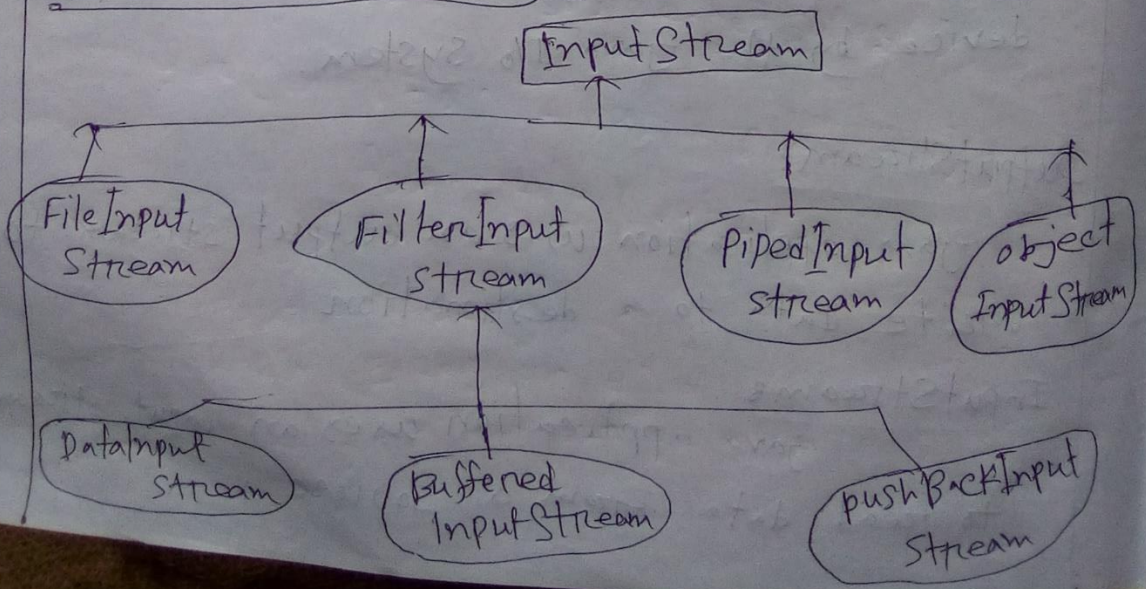
OutputStream:
Java application uses an output stream to write data to a destination.

InputStream:
Java application uses an input stream to read data from a source.

OutputStream Hierarchy



InputStream Hierarchy



Character Stream Classes:-

Stream class	Description
FileReader	Input stream that reads from a file.
FileWriter	Output stream that writes to a file.
BufferedReader	Buffered input character stream.
BufferedWriter	Buffered output character stream
CharArrayReader	Input stream Reads character array
CharArrayWriter	Output stream writes chacter array
FilterReader	Filtered reader
FilterWriter	Filtered writer
PipedReader	Input pipe
PipedWriter	Output pipe
PrintWriter	Output stream that contains print() and println()
PushbackReader	Input stream that allows characters to be returned to the input stream.
StringReader	Input stream that ~~writes~~ reads from a string.
StringWriter	Output stream that writes to a string.

Byte Stream classes:

Stream class	Description
FileInputStream	Input stream that reads from a file.
FileOutputStream	output stream that write to a file
BufferedInputStream	Buffered input stream.
BufferedOutputStream	Buffered output stream
ByteArrayInputStream	Input stream that reads byte array
ByteArrayOutputStream	output stream that write byte array
FilterInputStream	Implements InputStream
FilterOutputStream	Implements outputStream
PipedInputStream	Input pepe
PipedOutputStream	Output pipe
PrintStream	output stream that contains print();
PushbackInputStream	Input stream that allows bytes to be returned to input stream
objectInputStream	Input stream for objects
objectOutputStream	Output stream for objects.

Most commonly used Stream classes

	characters	Bytes
Files	FileReader FileWriter	FileInputStream FileOutputStream
Buffering	BufferedReader BufferedWriter	BufferedInputStream BufferedOutputStream
printing	printWriter	PrintStream

Java File :

The File class is an abstract representation of file and directory pathname.

The File class methods creating new directories and files, deleting and renaming directories, listing the contents of a directory.

Example

```
import java.io.*;
class File {
public static void main (String [] args){
    try {
    File file = new File ("File.txt");
if (file.createNewFile()){
  System.out.println("New File is created");
else {
   system.out.println("File exists");}
catch (IOException e){
      system.out.println(e);}
   }
}
```

FileInputStream:

Input stream that reads from a file bytes by bytes.

Example

```
import java.io.*;
class FileInputStream {
Public static void main (String args[]) {
try{
FileInputStream fin = new FileInputStream
                        ("file.txt");
 int b;
while (b = fin.read() != -1) {
          System.out.print (b);
        }
    fin.close(); }
catch (Exception e) {
        System.out.println (e); }
    }
}
```

FileOutputStream:

output stream that writes to a file Bytes by
It is string oriented class. Bytes.

Example

```
import java.io.*;
class FileOutputStream {
public static void main (String args[]){
    try {
FileOutputStream fout = new FileOutputStream
String source = "Hello world";          ("file.txt");
byte buf[] = source.getBytes();
for(int i=0; i< buf.length; i++) {
        fout.write(buf[i]);
    }
fout.close();
catch (Exception e) {
        system.out.println(e);
} } }
```

Java Buffered Output Stream :

This class used for buffering an output stream. It add more efficiency than to write data directly into a stream.

Example

```
import java.io.*;
class BufferedOutput {
public static void main(String args[]){
  FileOutputStream fout = new FileOutputStream
                                ("File.txt");
  BufferedOutputStream bout = new BufferedOutputStream
                                (fout);
  String s = "welcome to java";
  byte[b] = s.getBytes();
   bout.write(b);
   bout.close();
   fout.close();
   System.out.println("Success");
  }
}
```

java BufferedInputStream :

This class used for buffering input stream. It read data many bytes at a time.

Example

```
import java.io.*;
class BufferedInput {
public static void main (String args[]) {
FileInputStream fin = new FileInputStream("File.txt");
BufferedInputStream bin = new BufferedInputStream (fin);
int i;
while (i = bin.read() != -1) {
system.out.print (i);
}
bin.close();
fin.close();
}
}
```

Methods of BufferedWriter

method	description
newLine()	It is added a new Line by writing a line separator.
write()	It is used to write
close()	It is used to close the input stream.
flush()	It is used flushes input stream.

Methods of BufferedReader:

method	Description
read()	it used for reading a single character.
ready()	It is used input stream is ready to be read.
readLine()	It is used for reading a line of text.
reset()	It repositions input stream
skip()	skipping the characters.
mark()	It marking the present position
close()	closses the input stream.

FileWriter

javaFilewriter :-

It is an abstract class for writing to Character streams.

Example

```
import java.io.*;
class writer{
public static void main(String[] args){
  try{
      FileWriter fw=new FileWriter("File.txt");
      fw.write("welcome to java");
      fw.close();
    }
  catch(Exception e){
      System.out.println(e)
    }
  }
}
```

Java FileReader

Java FileReader class is used to read from the file.

Example

```
import java.io.*;
class FileReader {
public static void main (String args[]) {
FileReader fr = new FileReader ("File.txt");
String i;
while (i = fr.read ()! = -1)
    System.out.print (i);
fr.close ();
}
}
```

Methods of FileWriter

Method	Description
write ()	It is used to write into FileWriter
flush ()	It is flushes the data of Filewriter
close ()	It is closed the FileWriter

Methods of FileReader

Method	Description
read ()	It is used to Read a file
close ()	It is used to close the FileReader class.

java printwriter :-

printwriter class implementation writer class. It is used print formatted representation of object to the output stream.

Example:

```
import java.io.*;
class printwriter{
public static void main (String[] args) {
    FileWriter FW = new FileWriter("File.txt");
  printWriter pw = printWriter (FW);
   pw.write ("welcome to java");
   pw.println();
   pw.close();
  System.out.println ("completed");
    }
}
```

Methods of printStream :-

Method	Description
print()	It prints the specified value
println()	It prints the value and terminate the line.
printf()	It writes the formatted to the current stream.
format()	It writes values specified format

Methods of printWriter :-

Method	Description
println()	It is used for print characters
append()	append to specified characters to write
checkError()	It check error and flushes stream.
clearError()	It is used clean error of stream
format()	It use formated string and format string
flush()	It is used to flushes the stream
close()	It is used to the close stream