

I/O in Java



Prepared by Harish Patnaik

School of Computer Engineering, KIIT Deemed to be University

Content

1. Intro to Stream
2. Stream classes
3. Input
4. Output
5. File handling

Introduction

- Java programs perform I/O operation through streams.
- Stream is an abstraction that either produces or consumes information.
- Java stream represents a pipeline through which data will flow and it is linked to physical devices by Java IO system.
- package - `java.io`
- Two types of stream -
 - Byte stream - handling input and output of bytes
 - Character stream - handling input and output of characters

Stream classes

Byte Stream classes -

InputStream	OutputStream
BufferedInputStream	BufferedOutputStream
FileInputStream	FileOutputStream
	PrintStream

Character Stream classes -

Reader	Writer
BufferedReader	BufferedWriter
FileReader	FileWriter
InputStreamReader (byte to char)	OutputStreamWriter (char to byte)
	PrintWriter

Predefined Streams

System class-

```
static InputStream in
static PrintStream out
static PrintStream err
```

Input -

System.in → InputStreamReader → BufferedReader

BufferedReader methods -

int	read()
String	readLn()
boolean	ready()
void	close()

Example - bufreader.java

Predefined Streams

Output -

System.out → PrintWriter

PrintWriter methods -

void
void

print()
println()

Example - pwriter.java

File

- All files are byte oriented.
- Commonly used byte stream classes are -
 `FileInputStream(String)`
 `FileOutputStream(String)`
 `FileOutputStream(String, boolean)`

Commonly used methods are -
 `int read()` throws `IOException`
 `void write(int)` throws `IOException`

Example - `filebyte.java`

File

- Commonly used character stream classes are -
 `FileReader(String)`
 `FileWriter(String)`
 `FileWriter(String, boolean)`

Commonly used methods are -

- `int read()` throws `IOException`
- `String readLine()` throws `IOException`
- `void write(int)` throws `IOException`
- `void write(String)` throws `IOException`

Example - `filedemo.java`

RandomAccessFile

➤ It supports both reading and writing to a file. It behaves like a large array of bytes and is associated with a cursor called FilePointer. Read and write operations are performed at the current position of FilePointer.

➤ Commonly used constructor is -

`RandomAccessFile(String,mode)`

Commonly used methods are -

`int read()` throws `IOException`

`string readLine()` throws `IOException`

`void write(int)` throws `IOException`

`void writeChars(String)` throws `IOException`

`void seek(long)` - sets the FilePointer

Example - `rafile.java`



Thank you