JAVA 입출력 (I/O) 이란

자바 IO(입출력)는 자바 프로그래밍 언어에서 데이터를 읽고 쓰는 작업을 수행하는 데 사용되는 기능 및 라이브러리의 집합을 의미한다. 자바 IO는 파일, 네트워크 연결, 시스템 콘솔 등과 상호 작용하는 데 사용된다.

데이터 입력의 대상
- 키보드, 파일, 네트워크를 통해 들어오는 데이터

데이터 출력의 대상
- 모니터, 파일, 외부 네트워크

스트림(Stream) 이란

자바에서 데이터는 스트림(Stream)을 통해 입출력된다.

스트림(Stream) → 데이터를 전달하는데 사용되는 장치

스트림은 단일 방향으로 연속적으로 흘러가는 것을 말하는데, 높은 곳에서 낮은 곳으로 흐르듯이 데이터는 출발지에서 도착지로 흘러간다.

프로그램이 출발지냐 또는 도착지냐에 따라서 사용하는 스트림의 종류가 결정

프로그램이 도착지면 흘러온 데이터를 입력받아야 하므로 입력스트림을 사용

반대로 프로그램이 출발지면 데이터를 출력해야 하므로 출력스트림을 사용

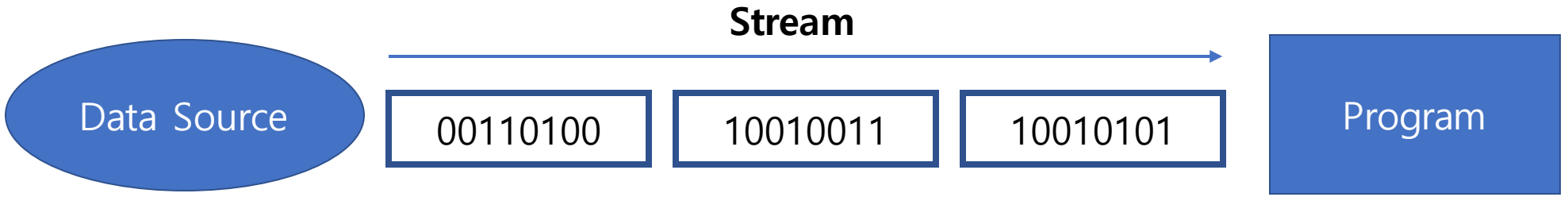
입출력 스트림의 종류

다음은 바이트 기반 / 문자기반 스트림의 최상위클래스와 하위 클래스의 이름이다.

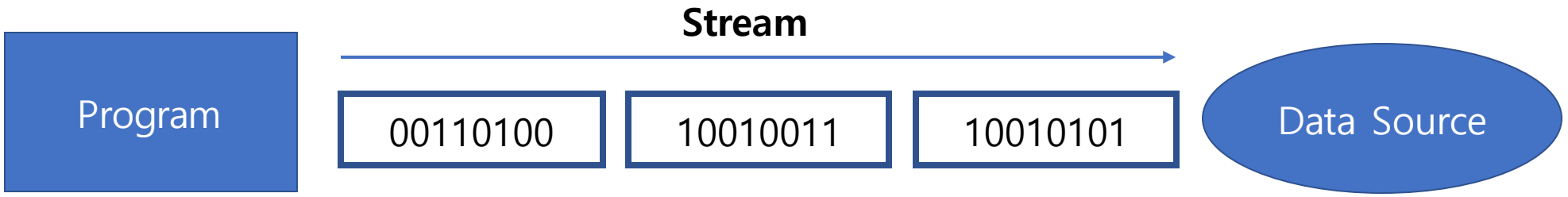
구분	바이트 기반 스트림		문자 기반 스트림	
	입력 스트림	출력 스트림	입력 스트림	출력 스트림
최상위 클래스	InputStream	OutputStream	Reader	Writer
하위 클래스 (예)	XXXInputStream (FileInputStream)	XXXOutputStream (FileOutputStream)	XXXReader (FileReader)	XXXWriter (FileWriter)

(I/O) 프로그래밍 핵심 키워드

모든 작업 시에 in 이나 read 라는 단어가 있다면 '읽어' 내는 기능을 의미



모든 작업 시에 out 이나 write 라는 단어가 있다면 '쓰는' 기능을 의미



InputStream의 read()

InputStream은 바이트 기반의 데이터를 읽어들이는 스트림이다.

주요 메서드로 read(), skip(), available() 등이 있다.

read() → int : 한 byte에 쓰여진 데이터

read(byte[]) → int : 읽어낸 데이터는 byte[]안으로 들어가고, int는 몇 개나 새로운 데이터를 읽어냈는지를 말한다.

[read() 메소드는 더 이상 읽어낼 데이터가 없는경우 -1을 반환한다.]

OutputStream의 write()

OutputStream은 바이트 기반의 데이터를 출력하는 스트림이다.
주요 메서드로 write(), flush(), close() 등이 있다.

write(int) → 한 바이트의 데이터를 전송

write(byte[]) → byte[]안에 있는 데이터들을 전송

write(byte[], int, int) → byte[]안에 있는 데이터들을 시작 위치에서부터
몇개의 개수만큼 데이터들을 전송

InputStreamReader / OutputStreamWriter

InputStreamReader과 OutputStreamWriter는 자바의 입출력 스트림(InputStream과 OutputStream)과 문자 인코딩 사이의 브리지 역할을 하는 클래스이다. 이 클래스들은 바이트 기반 스트림과 문자 기반 스트림 간의 상호 변환을 수행하여, 문자 데이터를 효율적으로 처리할 수 있도록 도와준다.

InputStreamReader은 InputStream을 받아들여 바이트를 문자로 변환하는 데 사용된다. InputStreamReader의 생성자에는 InputStream과 문자 인코딩을 지정하는 Charset 객체나 문자 인코딩 이름을 전달할 수 있다.

InputStreamReader / OutputStreamWriter

OutputStreamWriter는 OutputStream을 받아들여 문자를 바이트로 변환하는 데 사용된다. 마찬가지로, OutputStreamWriter의 생성자에는 OutputStream과 문자 인코딩을 지정하는 Charset 객체나 문자 인코딩 이름을 전달할 수 있다.

InputStreamReader와 OutputStreamWriter는 주로 텍스트 파일을 읽거나 쓸 때, 특히 다양한 문자 인코딩을 처리해야 할 때 유용하다. 이를 통해 애플리케이션은 표준화되지 않은 인코딩을 사용하는 파일을 읽고 쓸 수 있으며, 특정 인코딩을 사용하는 파일을 다른 인코딩으로 변환할 수도 있다.

PrintStream / PrintWriter

PrintStream과 PrintWriter 클래스는 자바의 표준 출력 스트림(System.out)이나 파일 등에 텍스트를 출력하는 데 사용되는 클래스이다.

PrintStream 클래스는 바이트 기반 출력 스트림을 래핑하여 텍스트를 출력한다. 표준 출력 스트림(System.out)에 연결되어 주로 콘솔에 출력하는 데 사용된다.

PrintStream ps = new PrintStream(바이트기반 출력스트림);
PrintWriter pw = new PrintWriter(문자기반 출력스트림);

PrintStream / PrintWriter

PrintWriter 클래스도 문자 기반 출력 스트림을 래핑하여 텍스트를 출력한다.

PrintWriter는 주로 파일에 출력하는 데 사용된다. PrintWriter는 문자를 자동으로 인코딩하지 않으며, 문자 인코딩을 지정할 수 있는 생성자를 제공한다.

메소드	설명
print()	주어진 값을 출력한다.
println()	주어진 값을 출력하고, 줄바꿈 문자를 추가한다.
printf()	형식화된 문자열을 출력한다.
format()	형식화된 문자열을 출력한다.
flush()	출력 버퍼를 비운다.
close()	스틀림을 닫는다.

File class

File 클래스는 파일과 디렉터리를 다루는 데 사용되는 클래스이다.

(하드 디스크의 위치를 잡아줄때 사용되는 클래스이다.)

File 클래스는 파일의 경로, 이름, 디렉터리 구조 등을 조작할 수 있는 다양한 메서드를 제공한다.

파일 및 디렉토리의 메타데이터와 경로를 다루는 기능을 제공하지만, 실제 파일 내용의 읽기와 쓰기에는 사용되지 않는다.

파일 byte (입력 / 출력)

스트림은 바이트 단위로 데이터를 전송

※ byte 입력
- FileInputStream 클래스를 이용하여 입력
- read()메소드를 이용하여 파일의 자료를 가져온다.
- 데이터의 끝은 -1 이다.

※ byte 출력
- FileOutputStream 클래스를 이용하여 출력
- 주로 write() 메소드를 통해 내보내기를 한다.

File class 생성자

생성자	설명
File(String pathname)	실제 경로명을 문자열로 처리
File(String parent, String child)	폴더명과 파일명으로 처리
File(File parent, String child)	폴더객체와 파일명으로 처리
File(URI uri)	웹상의 주소로 처리

File class 메소드

메소드	설명
createNewFile()	파일 생성 후 (true / false) 반환
createTempFile(String prefix, String suffix, File dir)	임시파일 생성
deleteOnExit()	프로그램끝날때 삭제
delete()	즉시 삭제
length()	파일의 크기(반환형은 long형)

성능향상 보조스트림

기본적으로 출력스트림은 내부에 작은 버퍼를 가지고 있다.
하지만 이것만으로는 불충분하다. 보조스트림 중에서는 메모리버퍼를 추가로 제공하여 프로그램의 실행 성능을 향상시키는 보조스트림 클래스가 있다.

바이트 기반 스트림 : BufferedInputStream, BufferedOutputStream
문자기반 스트림 : BufferedReader, BufferedWriter

성능향상 보조스트림

BufferedOutputStream과 BufferedWriter는 프로그램에서 전송한 데이터를 내부 버퍼에 쌓아두었다가 버퍼가 꽉 차면, 버퍼의 모든 데이터를 한꺼번에 보낸다.

메모리버퍼로 데이터를 고속 전송하기 때문에 출력성능이 향상된다.

BufferedInputStream과 BufferedReader는 입력 소스로부터 데이터를 미리 읽고 버퍼에 저장해둔다.

외부의 입력 소스로부터 읽는 대신 버퍼로부터 읽음으로써 읽기 성능이 향상된다.

DataInputStream / DataOutputStream

InputStream / OutputStream을 좀 더 편리하게 사용하기 위해서
제공되는 확장된 클래스이다.

ex) readInt()의 경우는 한 번에 4byte의 데이터를 읽어서
int 타입으로 변환하는 기능

기본자료형에 맞게 데이터들을 처리하거나 UTF-8 방식의 문자 데이터를 처리하는 기능이 있다.

DataInputStream / DataOutputStream

DataInputStream과 DataOutputStream 보조스트림을 연결하면 기본타입인 boolean, char, short, int, long, float, double을 입출력할 수 있다.

DataInputStream과 DataoutputStream 은 다른 보조스트림과 마찬가지로 연결할 바이트 입출력스트림을 생성자의 매개값으로 주면된다.

DataInputStream dis = new DataInputStream(바이트기반 입력스트림);

DataOutputStream dos = new DataOutputStream(바이트기반 출력스트림);

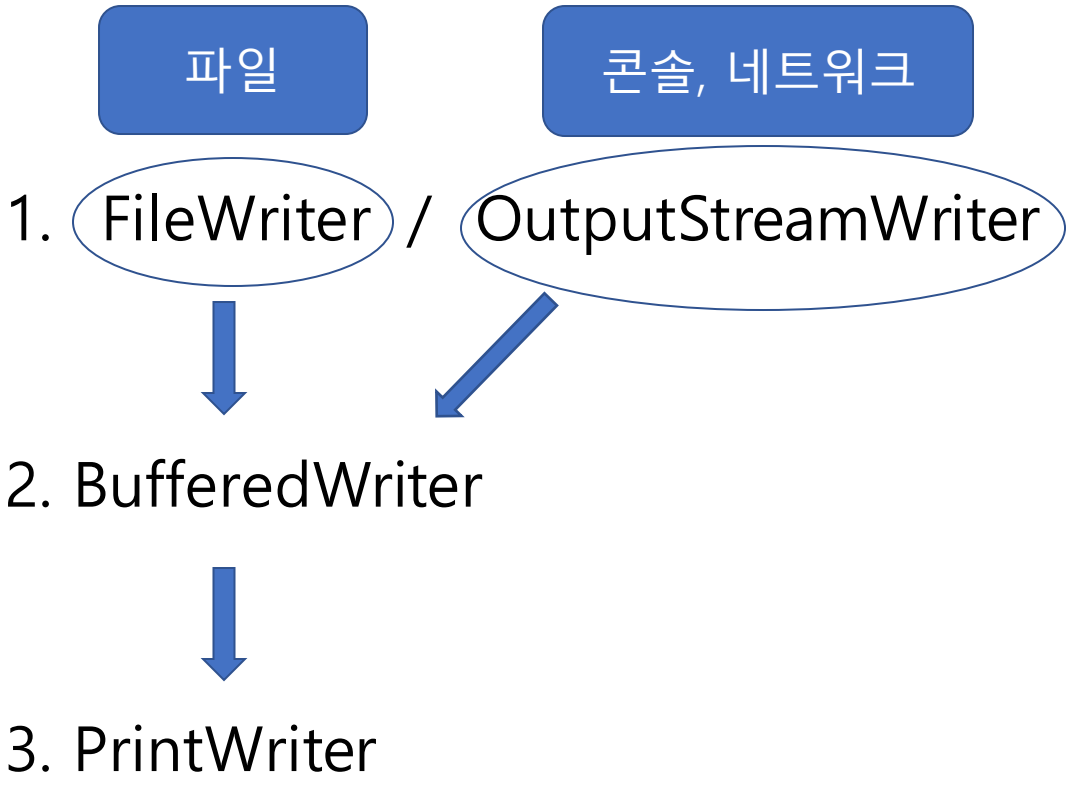
Reader / Writer

※ InputStream / OutputStream의 char처리 대용

※ char의 2bytes 기반의 처리 기능 지원

※ char기반의 처리라는 점을 제외하면 모든 처리는
InputStream / OutputStream과 동일

Text 출력


파일
콘솔, 네트워크
1. FileWriter / OutputStreamWriter
2. BufferedWriter
3. PrintWriter

객체입출력 보조스트림

ObjectOutputStream과 ObjectInputStream 보조스트림을 연결하면 메모리에 생성된 객체를 파일 또는 네트워크로 출력 할 수 있다. ObjectOutputStream은 객체를 직렬화하는 역할을 하고, ObjectInputStream은 객체로 역직렬화하는 역할을 한다. 직렬화란 객체를 바이트 배열로 만드는 것을 말하고, 역직렬화란 바이트 배열을 다시 객체로 복원하는것을 말한다.

ObjectOutputStream과 ObjectInputStream은 다른 보조스트림과 마찬가지로 연결할 바이트 기반 입출력스트림을 생성자의 매개값으로 받는다.

ObjectInputStream ois = new ObjectInputStream(바이트기반 입력스트림);
ObjectOutputStream ois = new ObjectOutputStream(바이트기반 출력스트림);

객체입출력 보조스트림

ObjectOutputStream의 writeObject()메소드는 객체를 직렬화해서 출력스트림으로 보낸다.

oos.writeObject(객체);

반대로 ObjectInputStream의 readObject()메소드는 입력스트림에서 읽은 바이트를 역직렬화해서 객체로 다시 복원해서 리턴한다. 리턴타입은 Object 타입이기 때문에 원래타입으로 다음과 같이 강제 변환해야 한다.

객체타입 변수 = (객체타입) ois.readObject();