

{ JAVA }

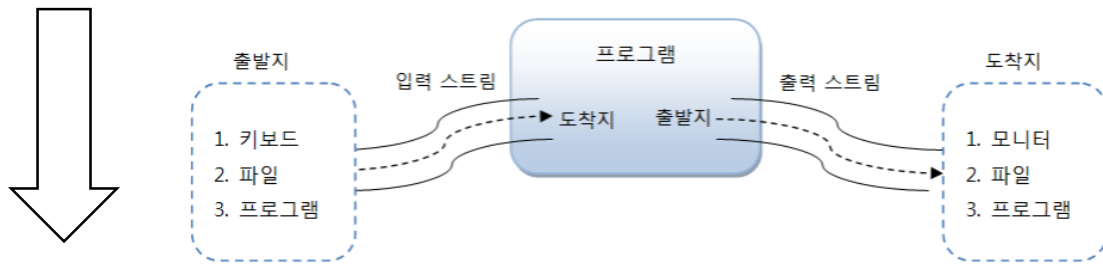
선린인터넷

웹 운영 과

I/O 입출력

java.io 패키지

데이터를 입/출력하고 싶다 : 키보드, 모니터, 파일, 네트워크 등



스트림 생성 = 해당 클래스 타입의 객체를 생성 (단방향: FIFO)

스트림으로부터 데이터 입출력 = 객체가 제공하는 메소드 이용

I/O 입출력

(input/output : **Stream**)

스트림은 데이터들이 들어오고 나가는 통로이며, 단방향이다.

Stream : 자바 프로그램에서 데이터를 입출력하기 위해서는 스트림을 이용한다.

1. 내, 시내, 개울

유의어 brook, rill

어법 1. stream 강(river) 다음으로 작은 중간 정도의 강. 2.... 더보기

참고 (※보통 river보다 작음)

a large[a little] stream

큰[작은] 내

2. (보통 단수형) (강·액체·공기 등의) 흐름; 수류, 해류, 조류; 기류; 광선

the Gulf Stream

멕시코 만류

3. [사람·일·사건 등의] 끊임없는 움직임, 연속, 계속[of ...]

a stream of curses

쉴 새 없이 퍼붓는 악담

4. (보통 the **stream**) (시간·역사·여론 등의) 흐름; 동향, 경향, 추세, 풍조

On January 12, lots of kids enjoyed ice sledding at Yangjae **Stream** in Seoul. Kidstimes

1월 12일, 많은 어린이들이 서울 양재천에서 얼음 썰매를 탔습니다.

But a few days ago, I visited the Cheonggye **Stream** with my cousin. Juniortimes

그런데 며칠 전에, 나는 사촌과 함께 청계천에 갔어.

However, there are some concerns about **streaming** services as well. Teentimes

하지만, 스트리밍 서비스에 대한 우려들 또한 존재한다.

❖ 자바 I/O 준비학습

- Java 입출력 방식은 크게 ① **바이트(Byte)** 단위 ② **문자(Char)** 단위로 구분된다.

※ 단방향

- Java I/O는 **데코레이터 패턴**으로 구성된다. (주인공과 장식의 구분이 있다.)

- **표준 입/출력** : 미리 생성된 스트림 = 사용자가 직접 생성할 필요 없음

- 교과서125 쪽 참고 -

- 표준입력 : **System.in** (InputStream 타입) - 키보드로 데이터를 입력하는 것
- 표준출력 : **System.out** (PrintStream 타입) - 화면(모니터)으로 메시지를 출력하는 것
- 표준에러 : **System.err** (PrintStream 타입) - 화면(모니터)으로 오류를 출력하는 것

❖ 1. 자바의 입출력 방식은 바이트 스트림 & 문자 스트림 으로 분류할 수 있다.

■ 바이트(Byte) 스트림 : jdk1.0

- 8bit의 1바이트 단위로 읽고 쓰기 위한 스트림 : **read(), write() 메소드**
- 그림, 멀티미디어, 문자 등 모든 종류의 데이터를 받고 보내는 것 가능

■ 문자(Char) 스트림 : jdk1.1

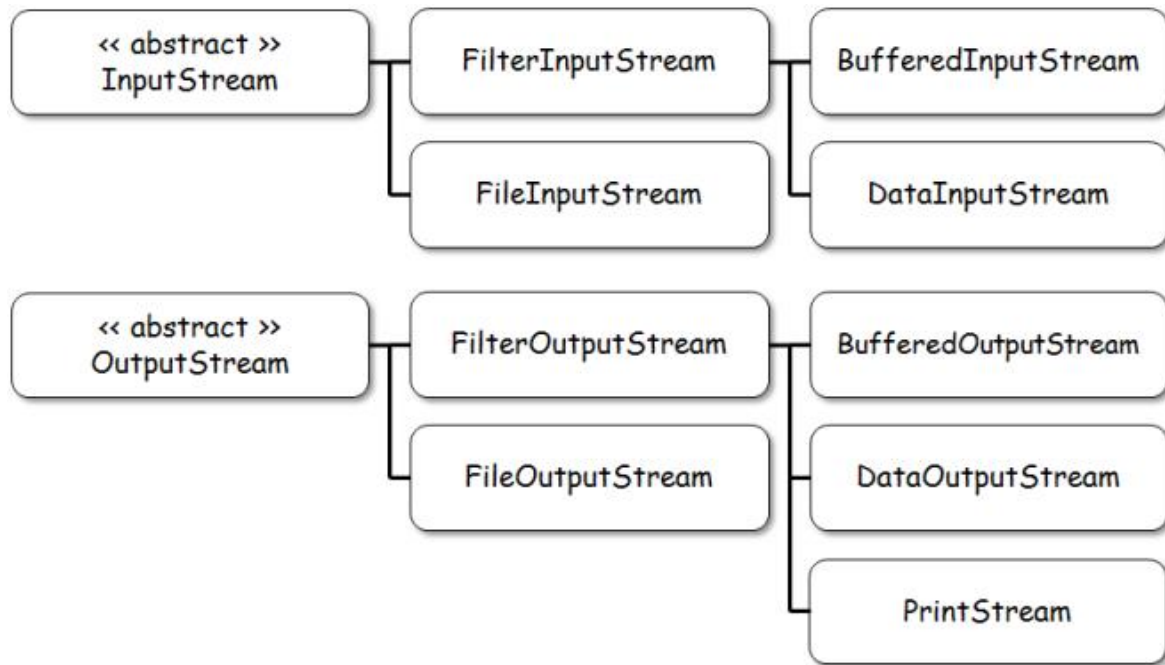
- 2바이트 단위로 문자나 문자열을 읽고 쓰기 위한 스트림 : **read(), write() 메소드**
- 문자만 받고 보낼 수 있도록 특화

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

❖ 1. 자바의 입출력 방식은 바이트 스트림 & 문자 스트림 으로 분류할 수 있다.

■ 바이트(Byte) 스트림 : jdk1.0

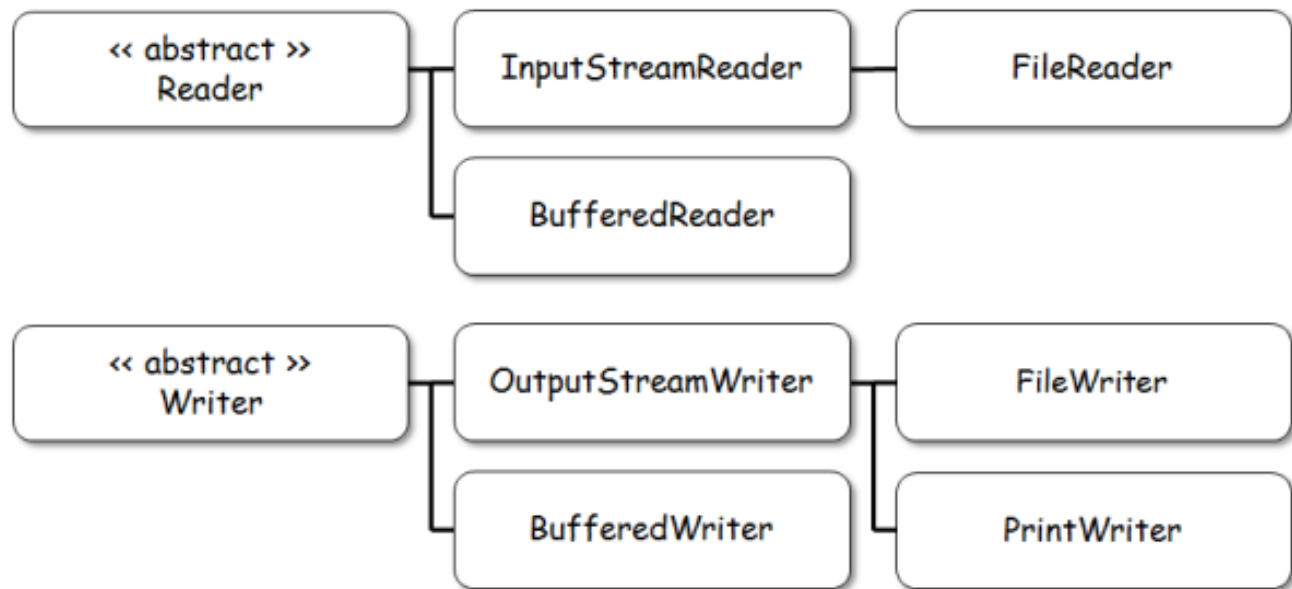
- 8bit의 1바이트 단위로 읽고 쓰기 위한 스트림 : **read(), write()** 메소드



❖ 1. 자바의 입출력 방식은 바이트 스트림 & 문자 스트림 으로 분류할 수 있다.

■ 문자(Char) 스트림 : jdk1.1

- 2바이트 단위로 문자나 문자열을 읽고 쓰기 위한 스트림 : **read(), write()** 메소드



❖ 2. Java I/O는 데코레이터(Decorator) 패턴 으로 구성된다. = 주인공 + 장식

decoration은 '장식(포장)'이란 뜻이다. 빵집에서 케이크를 만들 때 먼저 둥근 모양의 빵을 만든다. 이 위에 초콜릿을 바르면 초콜릿 케이크가 되고, 치즈를 바르면 치즈 케이크가 된다. 또 생크림을 바르고 과일을 많이 올려놓으면 과일 생크림 케이크가 된다.

이처럼 기존에 구현되어 있는 **클래스**(둥근 모양의 빵)에 그때그때 필요한 기능(초콜릿, 치즈, 생크림)을 추가(장식, 포장)해나가는 설계 패턴을 decorator 패턴이라고 한다. 이것은 기능 확장이 필요할 때 **상속**의 대안으로 사용한다.



<http://terms.naver.com/entry.nhn?docId=3532967&cid=58528&categoryId=58528>

❖ 3. 자바에서 표준 입/출력은 미리 생성된 스트림이다.

- Java에서는 키보드와 모니터를 이용한 표준 입출력 방식을 제공한다.
- **java.lang.System** 클래스에서 제공하는 **in, out, err** 멤버 변수에 의해 제공한다.

Fields

<http://docs.oracle.com/javase/8/docs/api/java/lang/System.html>

Modifier and Type	Field and Description
static PrintStream	err The "standard" error output stream.
static InputStream	in The "standard" input stream.
static PrintStream	out The "standard" output stream.

예제

```
1 import java.io.*;
2 public class Exam1 {
3     public static void main(String[] args) {
4         int bt = 0;
5         try {
6             bt = System.in.read();
7             System.out.print((char)bt);
8         } catch (IOException e) {
9             System.err.println(e); // 생략가능
10        }
11    }
12 }
```

Console

<terminated> Exam1 [J

T

T

in

```
public static final InputStream in
```

The "standard" input stream. This stream is already open and ready to supply input data. Typically this stream corresponds to keyboard input or another input source specified by the host environment or user.

read

```
public abstract int read()
    throws IOException
```

Reads the next byte of data from the input stream. The value byte is returned as an int in the range 0 to 255. If no byte is available because the end of the stream has been reached, the value -1 is returned. This method blocks until input data is available, the end of the stream is detected, or an exception is thrown.

실습문제1 : Ctrl+Z로 종료될 때까지 여러 문자를 입력받아 출력하도록 수정해보자.

```
1 import java.io.*;
2 public class Exam2 {
3     public static void main(String[] args) {
4         int bt = 0;
5         try {
6             // Ctrl+Z입력하면 종료됨
7             // 관련 코드 작성
8         } catch (IOException e) {
9             System.err.println(e); // 생략가능
10        }
11    }
12 }
```

Console

<terminated> Exa

Test

Test

ctrl+z 입력

// while문 이용하자

read

```
public abstract int read()
    throws IOException
```

Reads the next byte of data from the input stream. The value byte is returned as an int in the range 0 to 255. If no byte is available because the end of the stream has been reached, the value -1 is returned. This method blocks until input data is available, the end of the stream is detected, or an exception is thrown.

실습문제2 : 이미지 파일 복사하기

이미지 파일은 이진파일입니다.

a.jpg파일을 b.jpg파일로 복사하는 프로그램을 작성해봅시다.

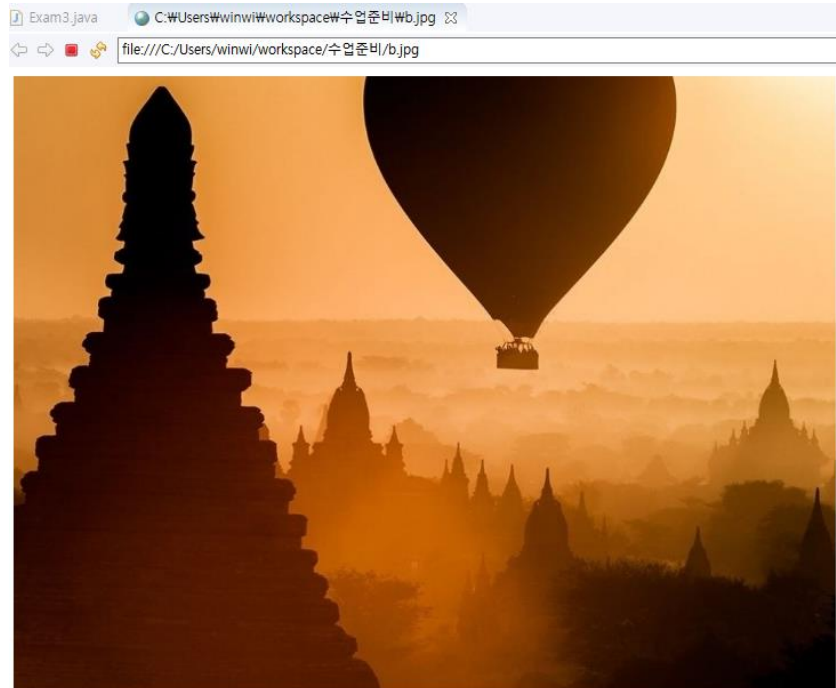
= Hint =

```
InputStream is = null;  
OutputStream os = null;  
int bin = 0;  
try{  
    is = new FileInputStream("a.jpg");  
    os = new FileOutputStream("b.jpg");
```

```
// ... 이하 코드 생략
```

실습문제2 : 참고하세요.

```
1 import java.io.*;
2 public class Exam3 {
3     public static void main(String[] args) {
4         InputStream is = null;
5         OutputStream os = null;
6         int bin = 0;
7         try {
8             is = new FileInputStream("a.jpg");
9             os = new FileOutputStream("b.jpg");
10            while ((bin = is.read()) != -1) {
11                os.write(bin);
12            }
13        } catch (IOException e) {
14            System.err.println(e);
15        } finally{
16            try{ os.close();
17            }catch(IOException e){ }
18            try{ is.close();
19            }catch(IOException e){ }
20        }
21    }
22 }
```



I/O Stream

정리하기

❖ I/O (Input/Output: 입출력)

- 프로그램 외부 소스 혹은 목적지와의 데이터 정보의 교환
- **입력:** 키보드, 파일 등으로 부터 프로그램으로 들어오는 데이터
- **출력:** 프로그램으로 부터 화면, 파일 등으로 나가는 데이터

❖ 스트림 (Stream)

- 프로그램과 I/O 객체를 연결하여 데이터의 흐름을 연결해주는 통로
- 스트림은 한방향으로 연결

출발지

- 1.키보드
- 2.파일
- 3.프로그램

입력스트림

데이터를
읽어 들이는
객체

program

출력스트림

데이터를
써서 보내는
객체

도착지

- 1.모니터
- 2.파일
- 3.프로그램

Byte 스트림

