

{ JAVA }

선린인터넷
웹 운영 과

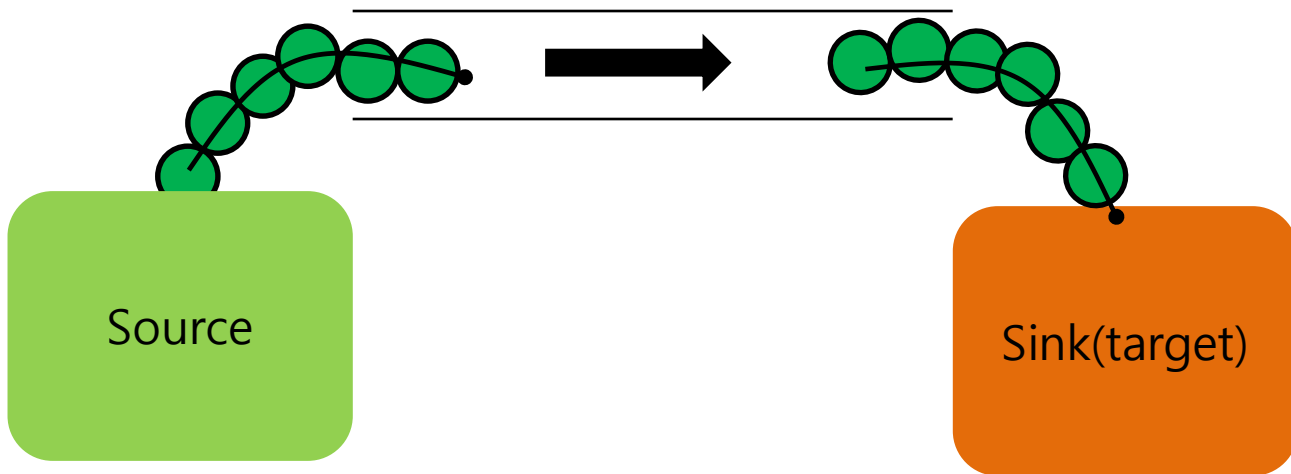
I/O 입출력

java.io 패키지 3

Byte Stream

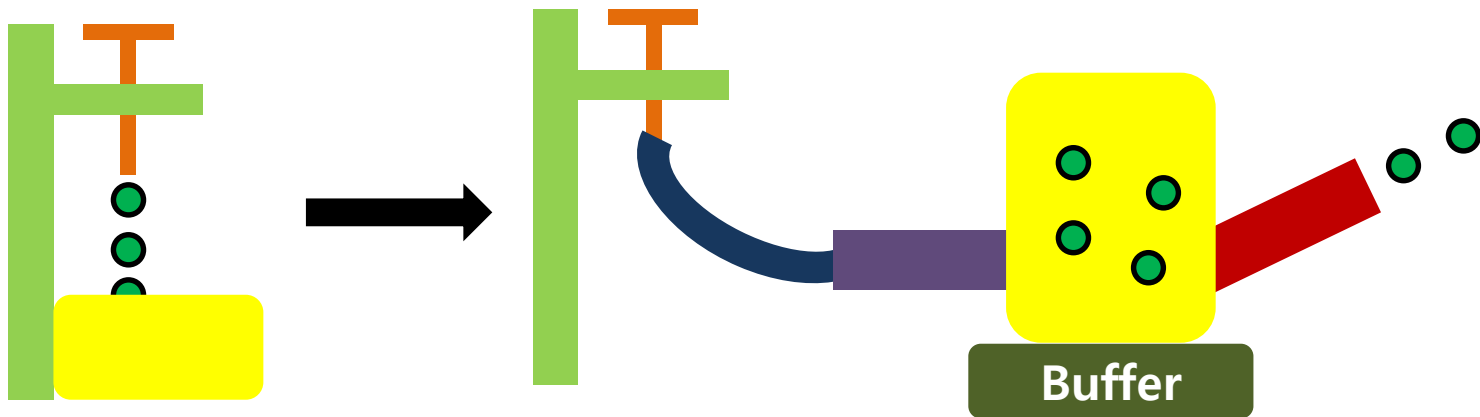
❖ 자바의 **스트림(Stream)** : 연속적인 데이터의 흐름

- Source : 스트림이 시작되는 부분
- Sink(Target) : 스트림의 목표지점



❖ 자바의 스트림(Stream)

- 노드(Node) : 입출력을 위한 장치 (예: System.in)
- 필터(Filter) : 노드 이외의 것

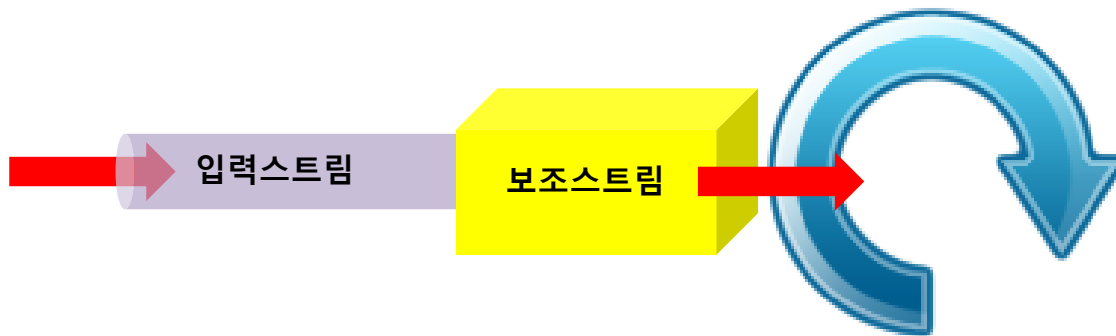


보조 스트림(Stream) = 필터 스트림(Filter Stream)



```
InputStream is = System.in;
```

```
InputStreamReader reader = new InputStreamReader( is );
```

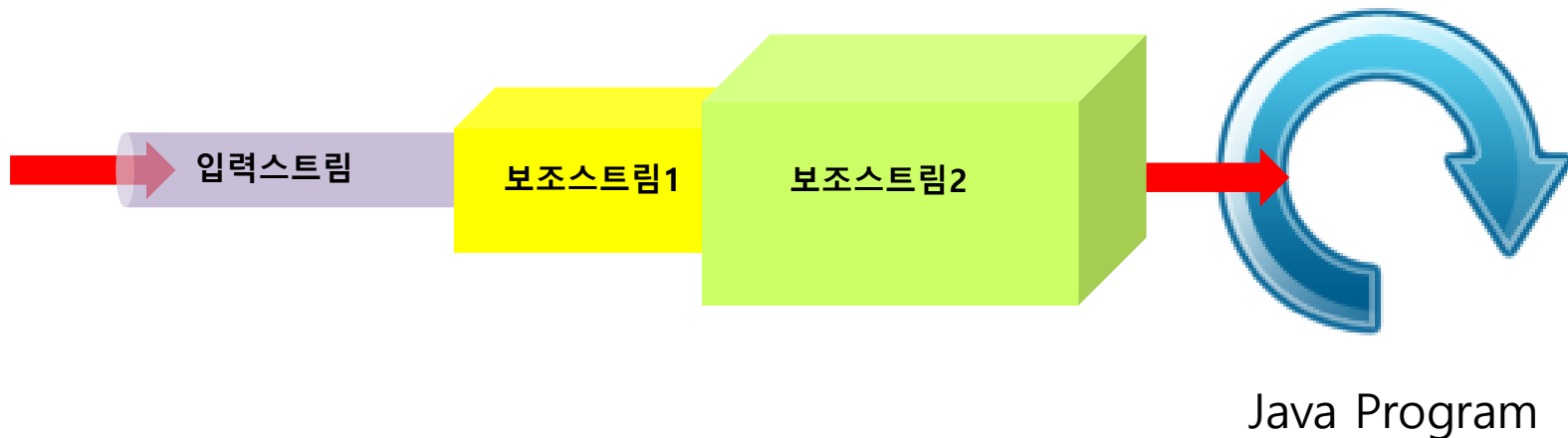


Java Program

```
InputStream is = System.in;
```

```
InputStreamReader reader = new InputStreamReader(is);
```

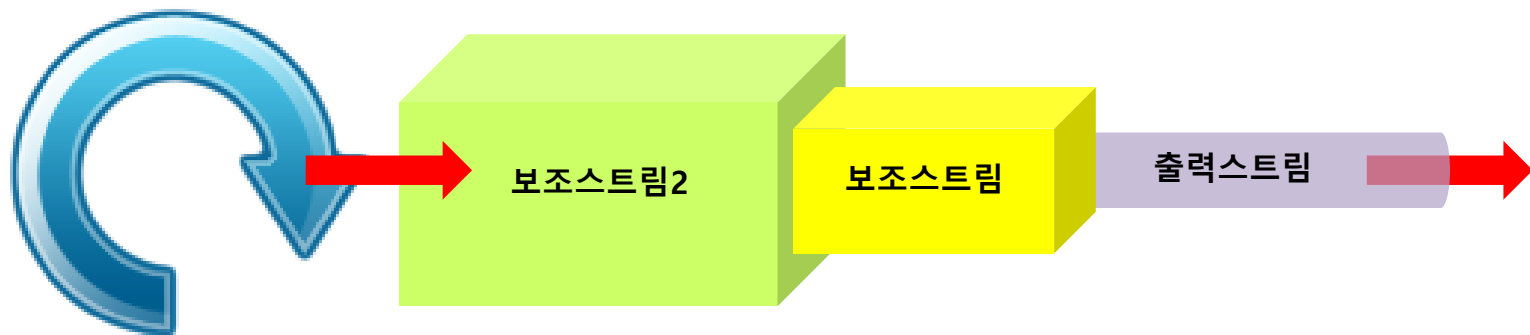
```
BufferedReader br = new BufferedReader(reader);
```



```
FileOutputStream fos = new FileOutputStream( "temp.txt" );
```

```
Writer writer= new OutputStreamWriter(fos);
```

```
BufferedWriter bw= new BufferedWriter(writer);
```



Java Program

데코레이터 패턴

❖ IO의 3대 관점

1. 노드와 필터의 구분

Node

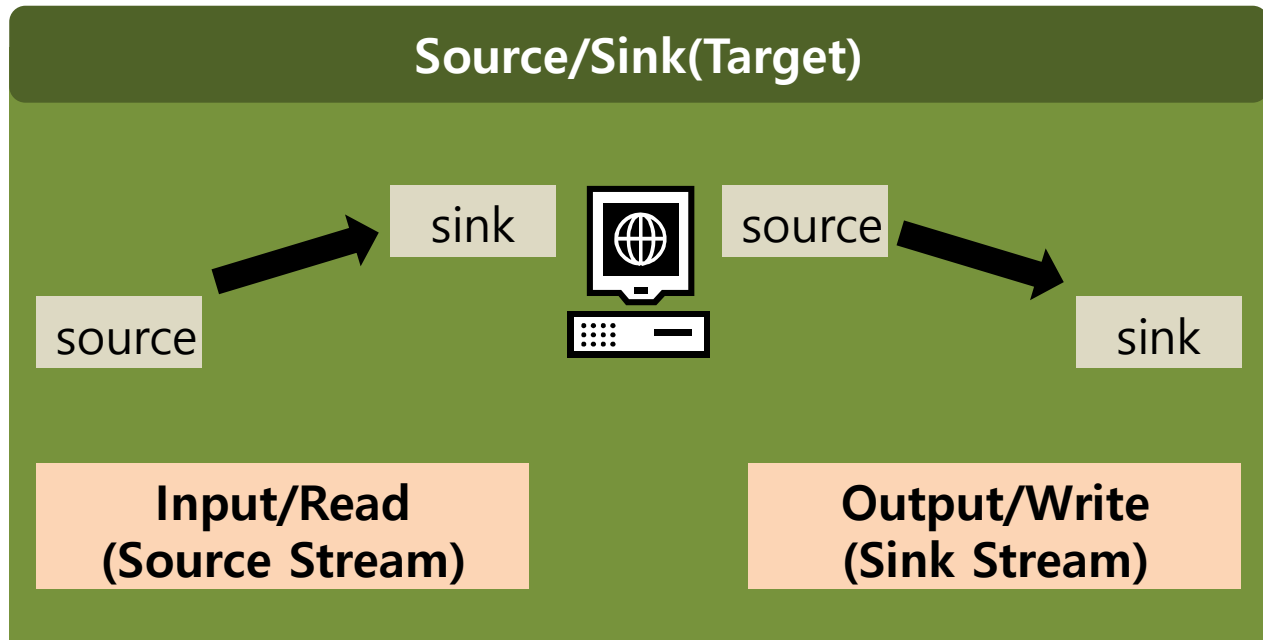
System.in, File,
ByteArray, CharArray,
String, Piped ...

Filter

Node 외
(Buffered...)

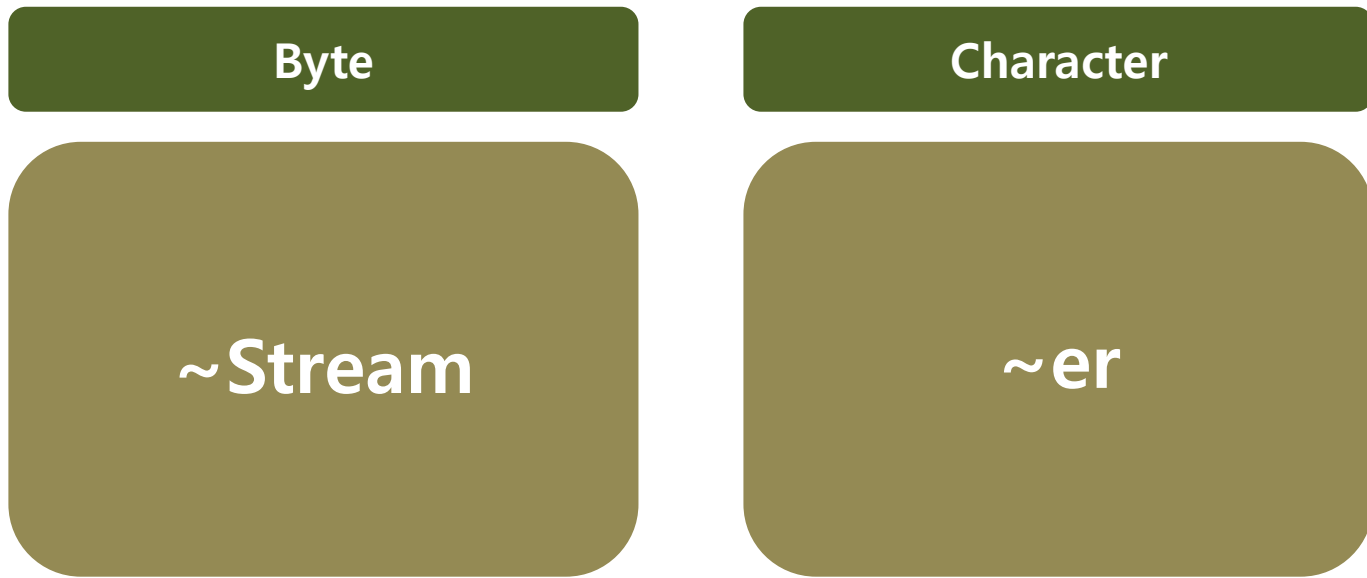
❖ IO의 3대 관점

2. 소스 스트림(Source Stream)과 싱크 스트림(Sink Stream)의 구분



❖ IO의 3대 관점

3. Byte Stream(8bit)과 Character Stream(16bit)의 구분



❖ IO의 3대 관점 - 정리

Node / Filter

Source /
Sink(Target)

Byte / Character

FileReader

File

Read

er

BufferedOutputStream

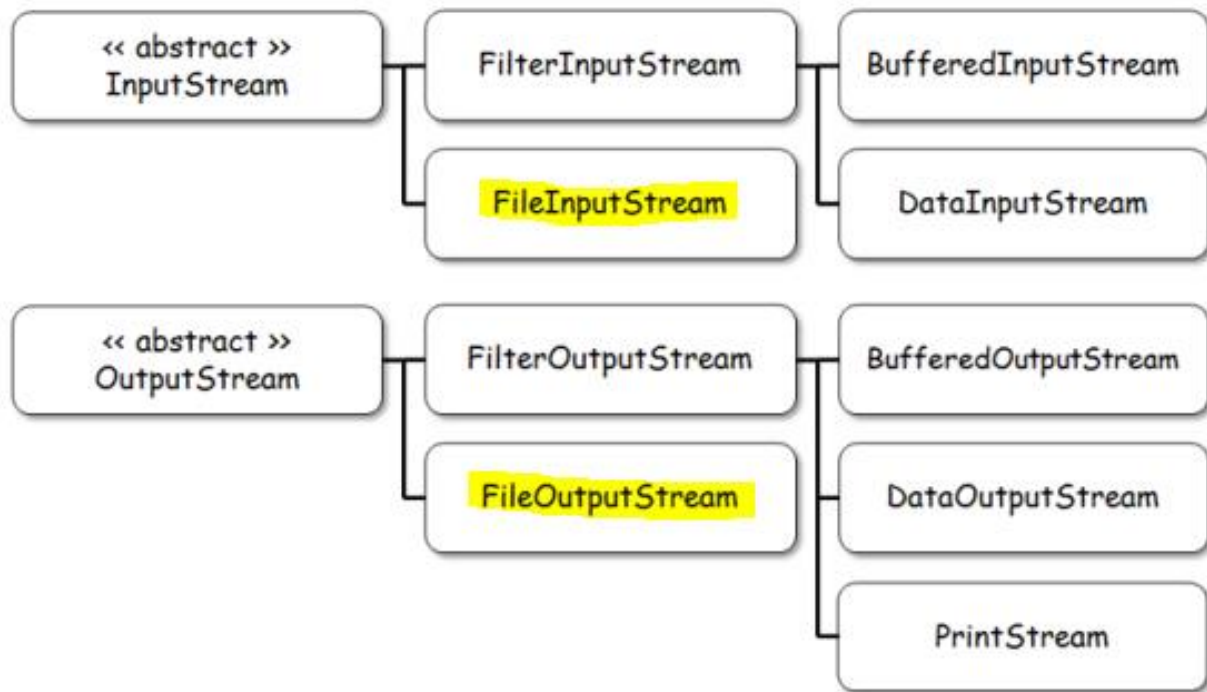
Buffered

Output

Stream

❖ 바이트(Byte) 스트림 : jdk1.0

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



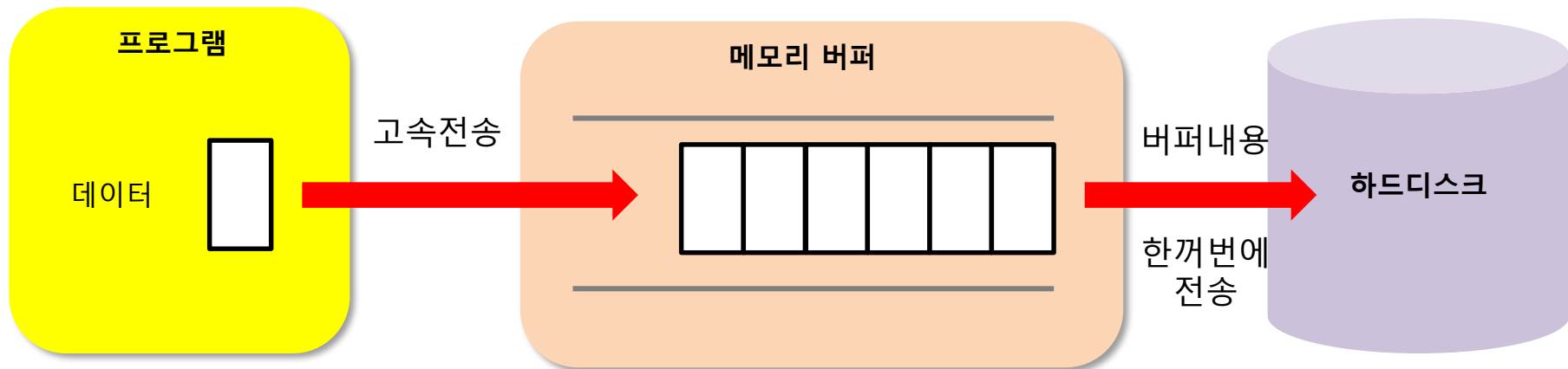
교과서 131쪽 : 입출력 스트림에 버퍼를 추가한 필터 스트림 클래스 (필터)

▷ 데이터 처리 속도 향상

BufferedInputStream 클래스

BufferedOutputStream 클래스

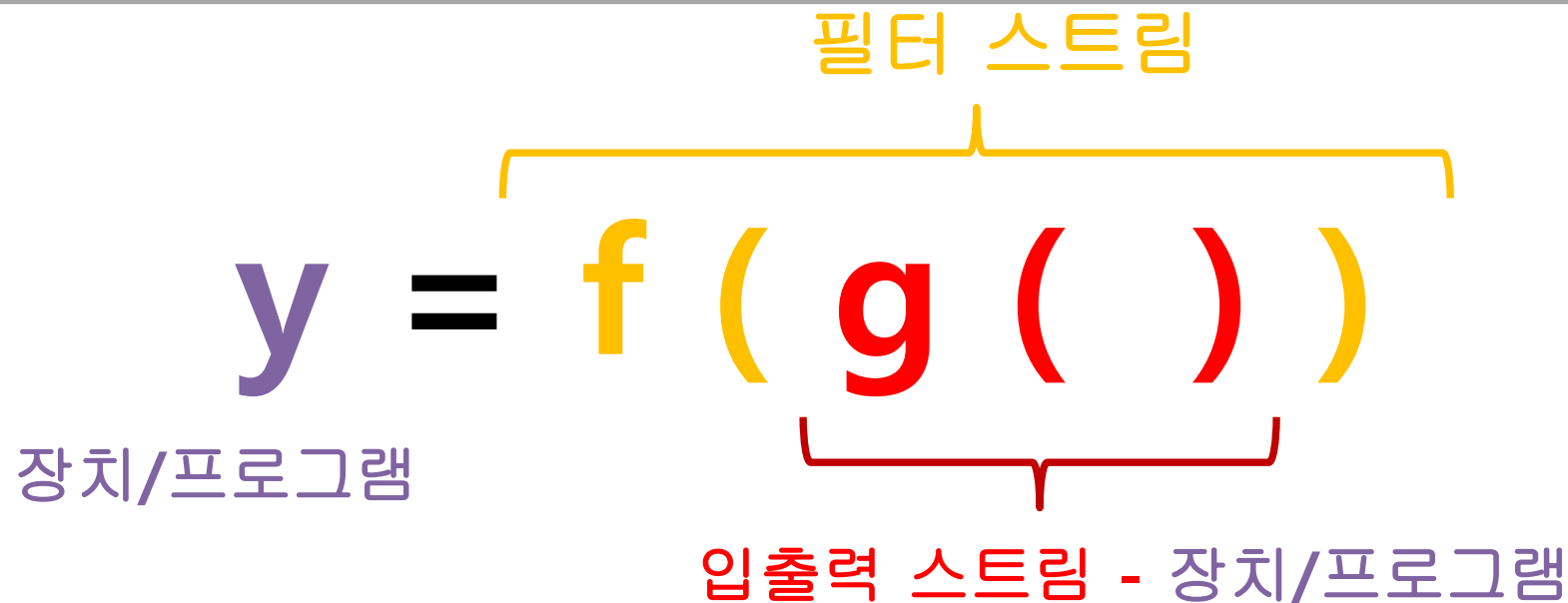
❖ BufferedInputStream와 BufferedOutputStream



❖ BufferedInputStream 및 BufferedOutputStream 클래스

- 바이트 블록 단위로 입/출력 → **데이터 처리 속도 향상**
- **flush()메소드** : 버퍼의 모든 내용을 출력 스트림에 출력하여 버퍼를 비운다.

	BufferedInputStream	BufferedOutputStream
의미	바이트 스트림으로부터 데이터를 읽어 들여 버퍼링 하는 기능을 제공	버퍼링 된 바이트 스트림을 출력하는 기능 제공 (기본적으로 8192byte 크기 버퍼 가짐)
생성자	BufferedInputStream(InputStream in) BufferedInputStream(InputStream in, int buffer_size)	BufferedOutputStream(OutputStream out) BufferedOutputStream(OutputStream out, int buffer_size)



예제4 : 키보드로부터 문자를 입력 받아 출력하는 프로그램 (입력끝: **Ctrl+z**)

```
1 import java.io.*;
2 public class ByteExam4 {
3     public static void main(String[] args) throws IOException {
4         BufferedInputStream bi = new BufferedInputStream(System.in);
5         BufferedOutputStream bo = new BufferedOutputStream(System.out);
6         int a;
7
8         while((a=bi.read())!=-1){ // 더이상 입력받을 데이터가 없을 때 = ctrl+z
9             bo.write(a);
10            if(a=='\n') // 엔터키를 입력하게 되면
11                bo.flush(); // 버퍼를 비움 : 출력스트림으로 모두 출력
12        }
13        bo.close();
14        bi.close();
15    }
16 }
```

Console

<terminated> B)

홍길동

홍길동

Test

Test

교과서 132쪽 : 파일로부터 데이터를 바이트 단위로 입출력 하기 위한 스트림 (노드)

FileInputStream 클래스

FileOutputStream 클래스

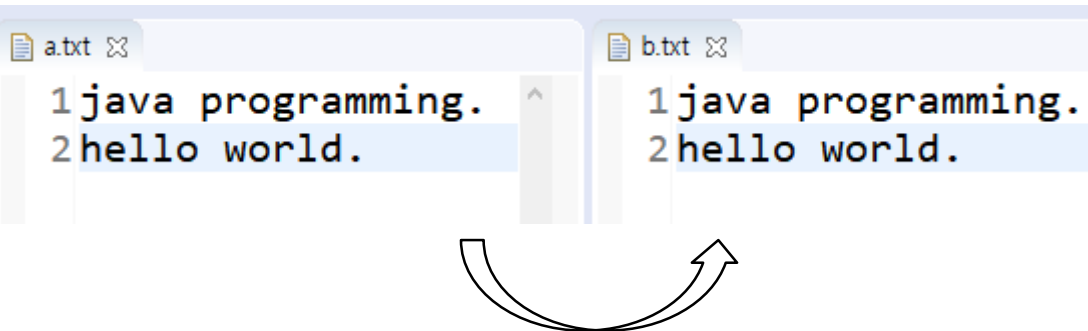
❖ FileInputStream 및 FileOutputStream 클래스

- 파일로부터 데이터를 바이트 단위로 입출력하기 위한 입출력스트림 생성 클래스

	FileInputStream	FileOutputStream
의미	파일 이름을 인자로 받아 해당 파일에서 읽는 스트림을 생성	파일 이름을 인자로 받아 해당 파일에서 쓰는 스트림을 생성
생성자	FileInputStream(String name) FileInputStream(File file)	FileOutputStream(String name) FileOutputStream(String name, boolean append) FileOutputStream(File file) FileOutputStream(File file, boolean append)
메소드	read() read(byte[] b) read(byte[] b, int off, int len)	write(int b) write(byte[] b) write(byte[] b, int off, int len)

예제5 : 파일 a.txt 를 읽어서... 파일 b.txt 로 저장(출력)하는 프로그램

```
1 import java.io.*;
2 public class ByteExam5 {
3     public static void main(String[] args) throws Exception {
4         FileInputStream fi = new FileInputStream("a.txt");
5         FileOutputStream fo = new FileOutputStream("b.txt");
6         int i;
7
8         while(true){
9             i=fi.read();
10            if(i==-1) break;
11            fo.write(i);
12        }
13
14        fo.close();
15        fi.close();
16    }
17 }
```



실습문제 : 키보드로 부터 데이터를 읽어서... 파일 c.txt 로 저장 (입력끝 : **Ctrl+z**)

소스프로그램 파일명: **Exam.java**

Console

<terminated> Exam (28) [Java Application] C:\WProgr

i love javaprogramming.

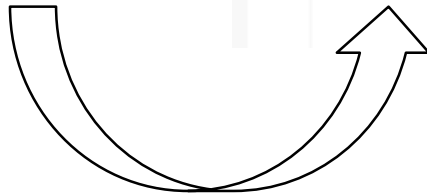
즐거운 자바 시간입니다.

c.txt

1 i love javaprogramming.

2 즐거운 자바 시간입니다.

3

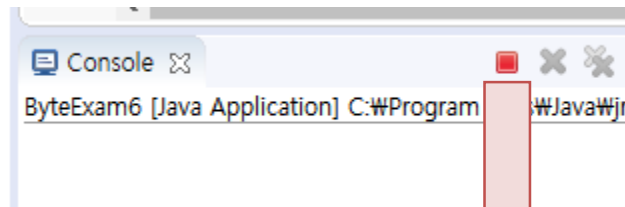


실습문제 예시

```
1 import java.io.*;
2 public class Exam {
3     public static void main(String[] args) throws IOException {
4         InputStream is = System.in;
5         BufferedInputStream bis = new BufferedInputStream(is);
6         OutputStream bos = new BufferedOutputStream(new FileOutputStream("c.txt"));
7         int i;
8
9         while(true) {
10             i=bis.read();
11             if(i==-1) break;
12             bos.write(i);
13         }
14         bos.close();
15         bis.close();
16     }
17 }
```

예제6 : 파일 heavy.data 를 읽어서... 파일 copy.data 로 저장(출력)하는 프로그램

```
1 import java.io.*;
2 public class ByteExam6 {
3     public static void main(String[] args) throws IOException{
4         InputStream is = new FileInputStream("heavy.data");
5         OutputStream os = new FileOutputStream("copy.data");
6         int i;
7         int copyByte=0;
8
9         long start = System.currentTimeMillis();
10        while(true){
11            i = is.read();
12            if(i == -1) break;
13            os.write(i);
14            copyByte++;
15        }
16        long end = System.currentTimeMillis();
17        os.close();
18        is.close();
19
20        System.out.println(end-start);
21        System.out.println("복사된 파일은 "+copyByte+"");
22    }
23 }
```



조금 오래 걸리죠?

예제7 : 파일 heavy.data 를 읽어서... 파일 copy.data 로 저장(출력)하는 프로그램

```
1 import java.io.*;
2 public class ByteExam7 {
3     public static void main(String[] args) throws IOException{
4         InputStream is = new FileInputStream("heavy.data");
5         BufferedInputStream bin = new BufferedInputStream(is);
6         OutputStream os = new FileOutputStream("copy.data");
7         BufferedOutputStream bos = new BufferedOutputStream(os);
8         int copyByte=0, i;
9
10        long start = System.currentTimeMillis();
11        while(true){
12            i = bin.read();
13            if(i==-1) break;
14            bos.write(i);
15            copyByte++;
16        }
17        long end = System.currentTimeMillis();
18
19        bos.close(); bin.close();
20        System.out.println(end-start);
21        System.out.println("복사된 파일은 "+copyByte+"바이트입니다.");
22    }
23 }
```

Buffered... 클래스를
이용해 봅시다!

Console

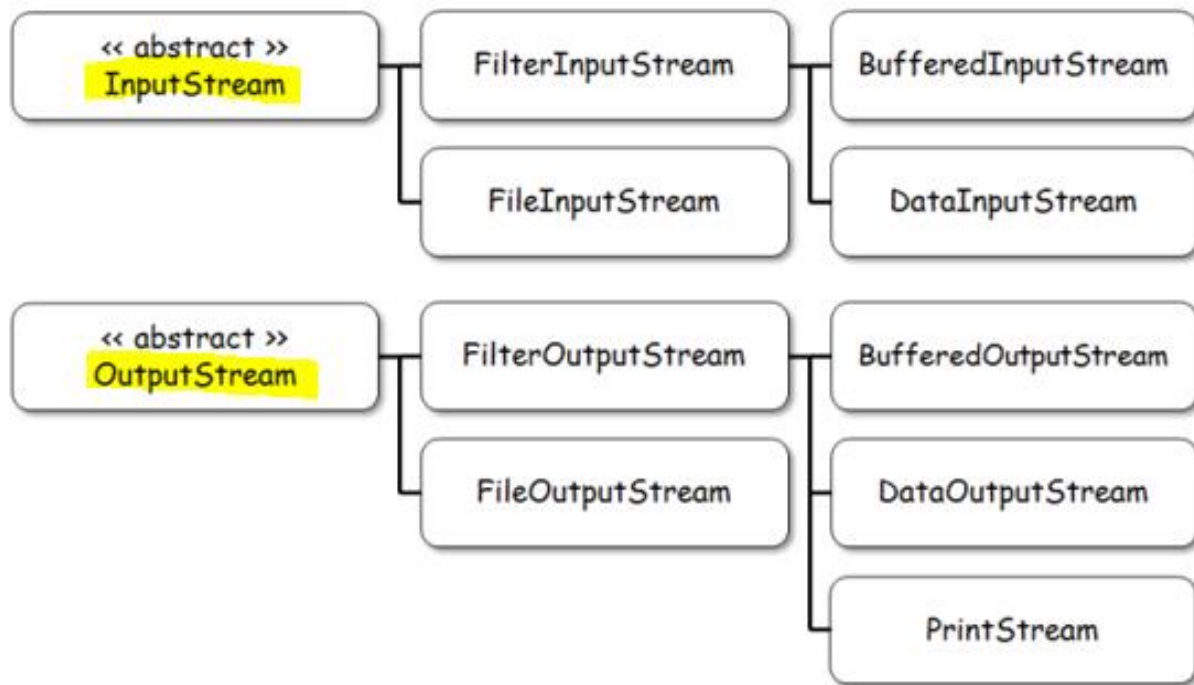
<terminated> ByteExam7 [Java Application] C:\Program File

5606

복사된 파일은 111652908바이트입니다.

정리하기

바이트(Byte) 스트림 : 1byte 단위로 읽고 쓰기 위한 스트림 : **read(), write()**



Byte 스트림

계속...

