

## 17. 자바 입출력

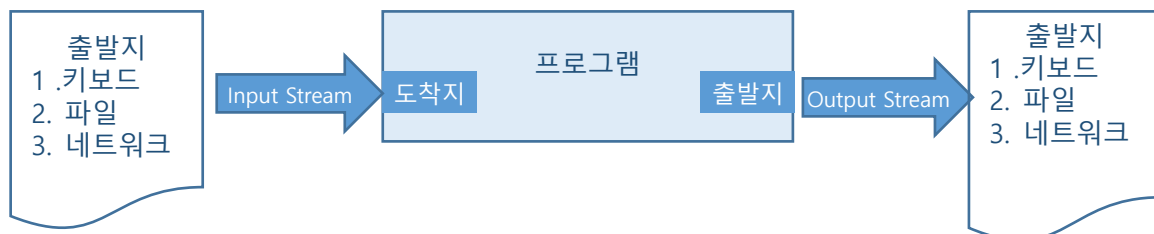
### 1. 입출력

컴퓨터로 데이터를 입력하는 것은 input 컴퓨터가 데이터를 출력하는 것은 Output이라고 한다.

자바에서 모든 입출력은 스트림(stream)을 통해 이루어지며 스트림은 단방향 통신만 가능하기 때문에 하나의 스트림으로 입력과 출력을 동시에 처리할 수 없다.

- Input : 파일 데이터를 읽는다. 키보드의 데이터를 읽는다. 네트워크상의 데이터를 읽는다(전송)
- Output : 파일에 데이터를 쓴다. 모니터에 데이터를 쓴다(출력). 네트워크상에 데이터를 쓴다(전송)

- 입출력 이미지



### 2. Java Stream API

InputStream : 1byte단위로 입력하는 API다 이미지, 동영상 등의 데이터에 주로 사용된다

OutputStream : 1byte단위로 출력하는 API다. 이미지, 동영상 등의 데이터에 주로 사용된다

- InputStream과 OutputStream는 byte를 다루기 위한 abstract 클래스라고 할 수 있으며 byte형으로 데이터를 입출력 하기 때문에 한글이나 한문등 한 글자에 2byte를 요구하는 문자는 깨진다..
- 보조 스트림(하위 클래스)으로는 대표적으로 DataInputStream / DataOutputStream가 있다.

Reader : 2byte단위로 입력하는 API다. 문자열에 주로 사용된다

Writer : 2byte단위로 출력하는 API다. 문자열에 주로 사용된다

- Reader와 writer은 char 기반의 문자단위를 처리하기 위한 클래스이며 둘 다 abstract클래스이다..
- 보조 스트림으로는 BufferedReader / BufferedWriter / PrintWriter가 있다.

## 1) InputStream

먼저 inputStream의 로직을 단계별로 구분해 보면 아래와 같다.

1. 파일(Stream객체 생성)을 연다 -> 2. 파일의 데이터를 읽는다(필요한 만큼 반복) -> 3. 파일을 닫는다

자주 사용하는 메소드로는

- read() : 객체의 데이터를 1 byte단위로 읽어온다. 매개변수에 byte[] 형식으로 괄호에 읽어올 크기의 값을 지정한 byte배열을 넣으면 지정한 byte값만큼 읽어온다. 읽어올 값이 없으면 -1을 리턴한다.
- close() : 모든 입출력 클래스에 있는 메소드이다. 스트림을 닫아주지 않았을 때 생기는 예상치 못한 에러를 방지하기 위함으로 해당 함수로 명시적으로 닫아주는 것을 권장하고 있다.

Ex)

```
1 import java.io.FileInputStream;
2 import java.io.FileNotFoundException;
3 import java.io.IOException;
4 import java.io.InputStream;
5 // (1번).파일을 연다 (Stream객체 생성) (2번).데이터를 읽는다. (3번).파일을 닫는다(Stream을 닫는다).
6 public class Ex1_InputStream {
7     public static void main(String[] args) {
8         InputStream is = null; //InputStream 객체를 예외문 밖에 null값으로 선언하였다
9         //예외문안에서 정의하고 예외가 발생하면 아래의 예외문에서 선언되지않은 예외 발생하기때문
10        try { //매개변수로 입력한 파일이 없을 수 있을 예외 발생
11            is = new FileInputStream("tstFile/inTest.txt");
12            //InputStream은 추상클래스라 하위클래스인 FileInputStream로 생성
13            // (1번). 매개변수로 입력한 파일이 있으면 객체생성, 한 바이트씩읽는다
14            while (true) {
15                int i = is.read(); // (2번).read()메소드를 데이터르 읽는다 //1바이트씩 H e l l ...
16                if (i == -1) //파일의 끝이라는 코드값이 -1
17                    break; // i의 값이 -1일때 종료
18                System.out.print((char) i);
19                System.out.print(" "+i+""); // H의 아스키코드값이나옴
20            }
21
22            // H(72)e(101)l(108)o(111),(44)J(74)A(65)V(86)A(65)
23            // (13)
24            // (10)H(72)i(105),(44) (32)J(74)a(97)v(118)a(97) (32)!(33)
25            // (13)
26            // (10)0(48)1(49)2(50)3(51)4(52)5(53)6(54)7(55)8(56)9(57)
27            // (13)
28            // (10)¼(190)È(200)³(179)ç(231),(44) (32)À(192)Ú(218)¹(185)Û(217)~(126) // 한글은 깨져서 출력된다.
29
30            System.out.println("\n끝");
31            //예외문 안에서는 예외발생하면 stream을 닫을수 없으므로 finally 문에서 닫아준다
32
33        } catch (FileNotFoundException e) {
34            System.out.println("파일을 못 찾는 경우 예외" + e.getMessage());
35        } catch (IOException e) { //FileNotFoundException보다 상위 클래스이므로 아래에
36            System.out.println("파일을 못 읽는 경우 예외" + e.getMessage());
37        } finally {
38            // (3번). 파일을 닫는다.
39
40            try {
41                if (is != null) //is객체에 데이터 값이있음 == 예외가 발생 안했음
42                    is.close(); //객체 is에 값이 만들어 갈수도 있는 예외발생 할수있으므로 예외처리
43                //InputStream안에 close()함수가 있어서 객체가 생성안되면 해당 메소드 실행불가
44            } catch (IOException e) {
45                System.out.println("파일을 못 읽는 경우 예외" + e.getMessage());
46            }
47        }
```

## 2) OutputStream

OutputStream의 로직을 단계별로 구분해 보면 아래와 같다.

1. 파일(Stream객체 생성)을 연다 -> 2. 파일의 데이터를 쓴다(write 한다) -> 3. 파일을 닫는다

주요 메소드 역시 inputStream과 비슷하다.

- write() : 객체의 데이터를 1 byte단위로 출력한다. 매개변수에 byte[] 형식으로 괄호에 출력 크기의 값을  
지정한 byte배열을 넣으면 지정한 byte값만큼 출력온다. 역시 출력할 값이 없으면 -1을 리턴한다.

Ex) 이미지 복사 입출력 예제

```
1 import java.io.FileInputStream;
2 import java.io.FileNotFoundException;
3 import java.io.FileOutputStream;
4 import java.io.IOException;
5 import java.io.InputStream;
6
7 public class Ex05_filecopyStep2 {
8     //1.Stream객체(입력용, 출력용) 2. 읽고 쓴다 3. 파일 닫는다(Stream객체.close())
9     public static void main(String[] args) {
10         InputStream is = null; // 파일 읽어올 용 inputStream
11         FileOutputStream os = null; // 파일 출력할 용 OutputStream(FileOutputStream)
12
13         try {
14             is = new FileInputStream("d:\\webProDK\\T.jpg"); // 읽어올 이미지파일
15             os = new FileOutputStream("d:/webProDK/T_copied.jpg");// 출력해서 복사할 이미지파일
16             int cnt = 0;
17             byte[] bs = new byte[1024];
18             while (true) {
19                 int readByCount = is.read(bs); // 1024 byte씩 읽어온다.
20                 if (readByCount == -1) // 마지막일때 break
21                     break;
22                 os.write(bs, 0, readByCount);
23                 //배열 bs의 0번 index부터 readByCount길이만큼 쓴다
24                 cnt++;
25             }
26             System.out.println(cnt + "번 while문 실행하여 힘들게 복사 성공");
27         } catch (FileNotFoundException e) {
28             e.printStackTrace();
29         } catch (IOException e) {
30             e.printStackTrace();
31         } finally {
32             try {
33                 if(os != null)os.close(); //처음 만든 객체를 나중에 클로즈
34                 if(is != null)is.close();
35             } catch (IOException e) {
36
37             }
38         }
```