

K G 아 이 티 뱅 크

J A V A

J A V A

Stream Class

## Stream Class

### ❖ 입력과 출력을 추상화하여 중개하기 위한 추상클래스

- 입력과 출력은 출발지와 목적지가 존재하는 구조
- 이를 통일된 형태로 쓰기 위해 상속되어 있음

### ❖ 기본적인 용도에 따라 File과 Data로 구분되어 있음

- File Stream Class : 파일 입출력을 위해 1 Byte를 읽음
- Data Stream Class : wrapper/decorator 클래스

```
import java.io.FileInputStream; // 파일입출력 클래스
import java.io.FileOutputStream;
import java.io.DataInputStream; // 데이터입출력 클래스
import java.io.DataOutputStream;
import java.io.FileNotFoundException; // 예외 클래스
....
// FileNotFoundException이 주로 발생함
FileInputStream fis = new FileInputStream("FILE");
// Data Stream은 File Stream에 편의성을 추가해줌
DataInputStream dis = new DataInputStream(fis);
```

## Stream Class

### ❖ File Stream의 경우 기본적으로 파일의 입출력을 담당함

- ① File Stream은 기본적인 1 Byte 파일입출력을 제공함
- ② 1 Byte의 크기만을 읽고 쓰며 넘을 경우 잘려나감
- ③ INPUT은 읽기이며, OUTPUT은 쓰기를 의미함
- ④ 이 때 자료형은 기본적으로 byte자료형을 사용함

```
try {  
    FileOutputStream fos = new FileOutputStream("FILE");  
    fos.write(1);  
    fos.close(); // 반드시 close()처리를 해줘야 함  
    FileInputStream fis = new FileInputStream("FILE");  
    byte data = fis.read();  
    fis.close(); // 입출력은 동시에 하면 안되니 주의  
}  
catch(FileNotFoundException e) {  
    System.out.println("ERROR");  
}
```

## Stream Class

### ❖ Data Stream의 경우 읽기/쓰기를 편하게 만들어서 제공함

- ① Data Stream은 자료형에 따라 읽기/쓰기 메서드가 다름
- ② 객체는 불가하고, 원시자료형에 대한 변환을 제공함
- ③ 각 자료형의 바이트 크기만큼 파일을 읽어서 인식함
- ④ String 클래스는 문자로 바뀌어서 전송해야 올바르게 작동함

```
try {  
    DataOutputStream dos = new DataOutputStream(fos);  
    dos.writeInt(13);  
    dos.close();           // dos에서 close메서드를 호출함  
    DataInputStream dis = new DataInputStream(fis);  
    int value = dis.readInt(); // EOF에 주의해야 함  
    dis.close();  
}  
catch (EOFException e) {  
    System.out.println("ERROR");  
}
```