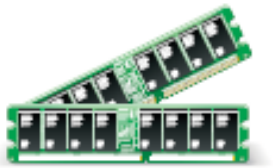
어서와 Java는 처음이지!

# 제17장 파일입출력

# 파일의 필요성

```
class Test
{
   ...
   ...
   ...
}
```

프로그램



메모리

객체는 모두 메모리에 만들어지고 이것들은 모두 전원이 꺼지면 사라진다.

하드 디스크

하드 디스크에 파일 형태로 저장하면 전원이 꺼지더라도 데이터가 보존된다.

데이터를 영구히 보관하려면 파일에 저장해야 합니다.
스트림을 사용하면 됩니다.

# 스트림(stream)

- 스트림(stream)은 "순서가 있는 데이터의 연속적인 흐름"이다.
- 스트림은 입출력을 물의 흐름처럼 간주하는 것이다.

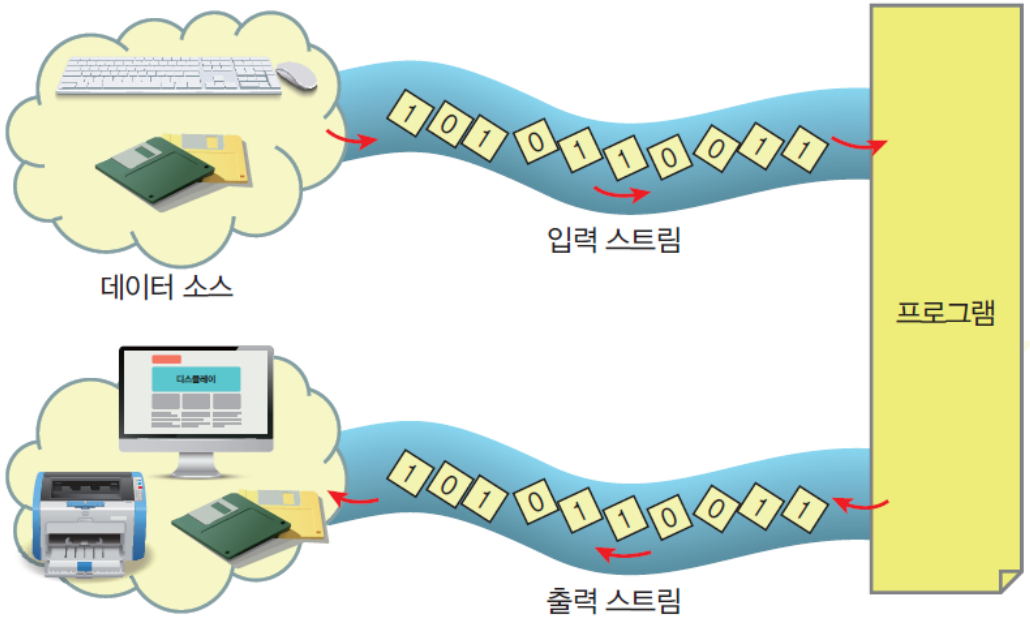



그림 17-1 • 스트림의 개념

# 스트림의 분류

- 입출력의 단위에 따라서 분류

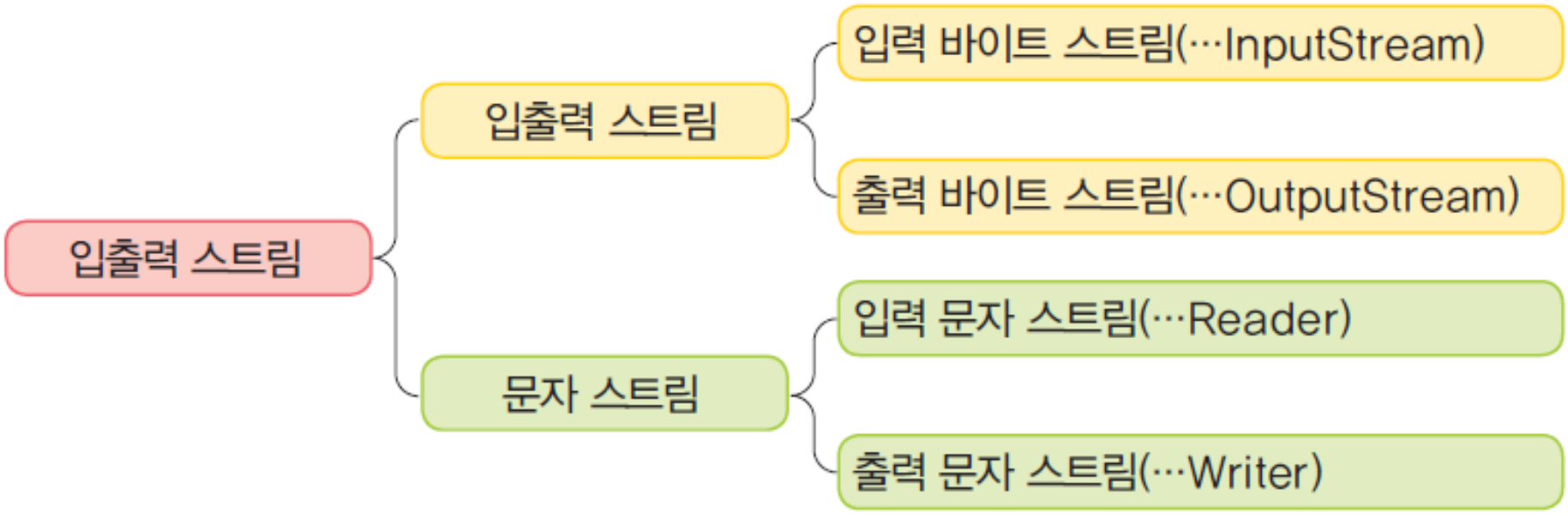



그림 17-2 • 스트림의 분류

# 바이트 스트림

- 바이트 스트림(byte stream)은 바이트 단위로 입출력하는 클래스
- 바이트 스트림 클래스들은 추상 클래스인 InputStream와 OutputStream에서 파생된다.
- 바이트 스트림 클래스 이름에는InputStream(입력)과 OutputStream(출력)이 붙는다.

# 기본적인 메소드

- InputStream 클래스
  - abstract int read() - 한 바이트를 읽어서 반환한다(0에서 255 사이의 정수).
- OutputStream 클래스
  - abstract void write(int b) - 한 바이트를 특정한 장치에 쓴다.

# 문자 스트림

- 문자 스트림(character stream)은 문자 단위로 입출력하는 클래스
- 이들은 모두 기본 추상 클래스인 Reader와 Write 클래스에서 파생된다.
- 문자 스트림 클래스 이름에는 Reader(입력)와 Writer(출력)가 붙는다.

# 기본적인 메소드

- Reader 클래스
  - abstract int read() - 한 문자를 읽어서 반환한다.
- Writer 클래스
  - abstract void write(int c) - 한 문자를 특정한 장치에 쓴다.

# 파일 문자 스트림

그림 17-4 • 문자 스크림의 개념

# 예제

```
public class CopyFile2 {
    public static void main(String[] args) throws IOException {

        FileReader inputStream = null;
        FileWriter outputStream = null;

        try {
            inputStream = new FileReader("input.txt");
            outputStream = new FileWriter("output.txt");

            int c;
            while ((c = inputStream.read()) != -1) {
                outputStream.write(c);
            }
        } finally {
            if (inputStream != null) {
                inputStream.close();
            }
            if (outputStream != null) {
                outputStream.close();
            }
        }
    }
}
```

# 스트림들은 연결될 수 있다.

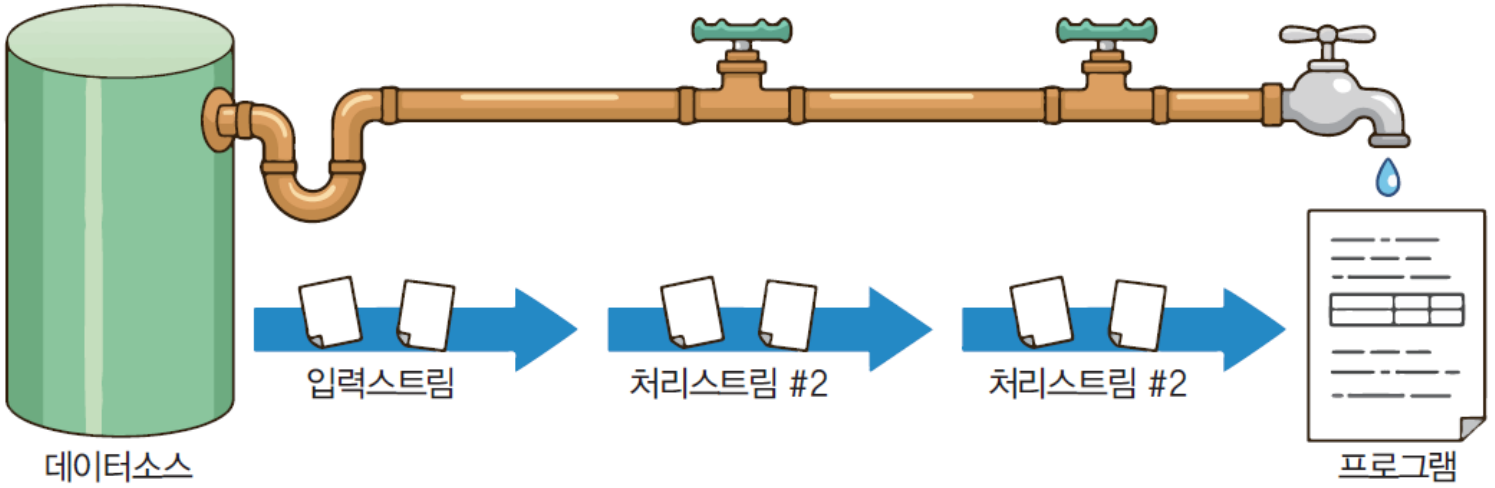
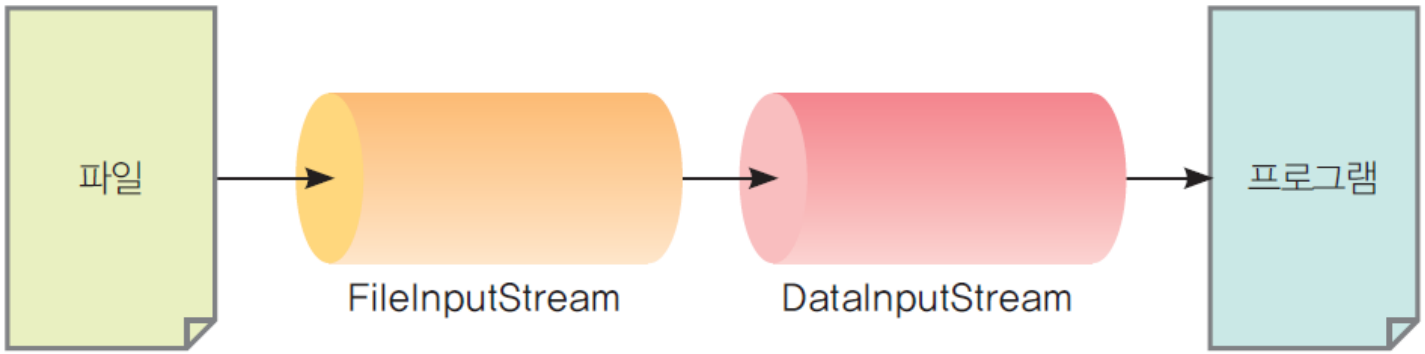



그림 17-5 • 스트림은 연결될 수 있다.

# 예제

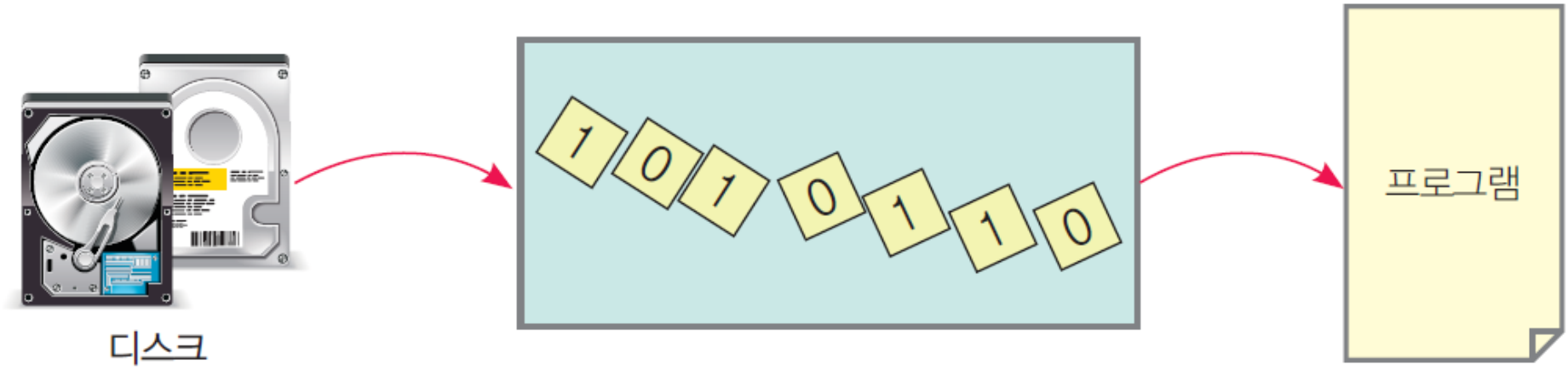
```
FileInputStream fileSt = new FileInputStream("sample.dat");
DataInputStream dataSt = new DataInputStream(fileSt);
int i = dataSt.readInt();
```



# 버퍼 스트림

한번 읽을 때 왕창
읽어 놓습니다.

하나씩 꺼내 씁니다.
다 떨어지면 다시
디스크에서 읽습니다.

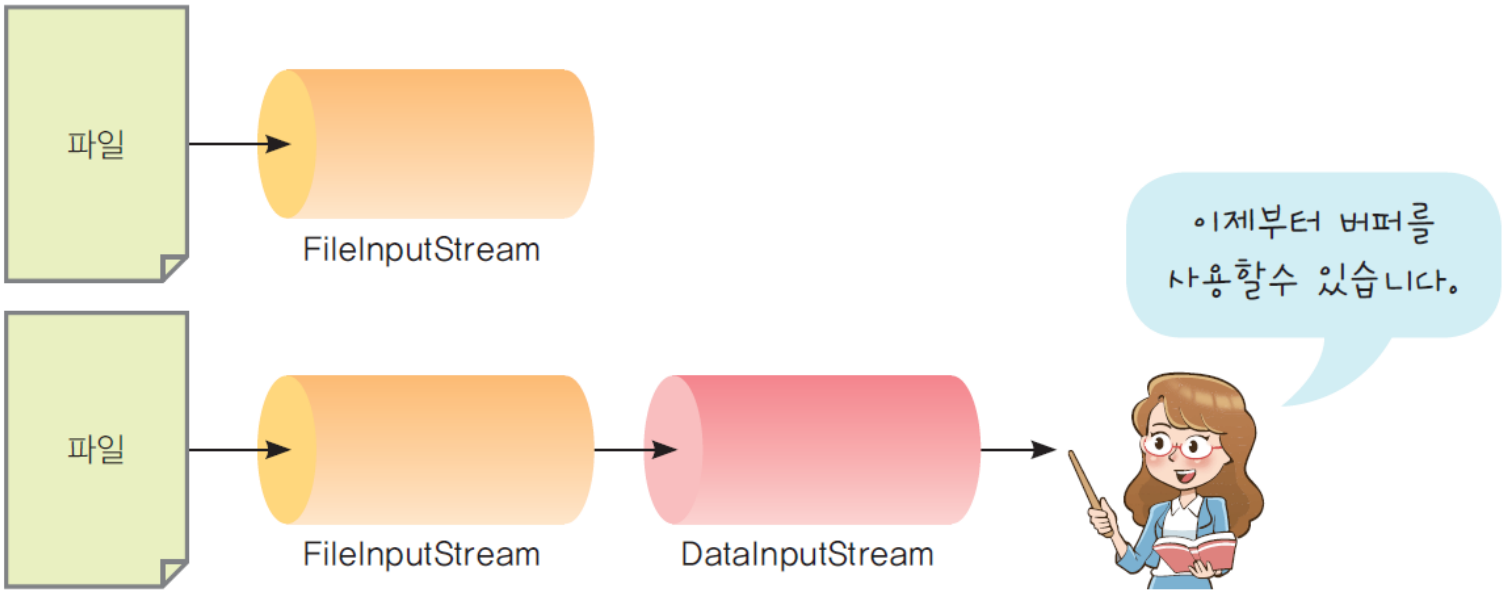



**버퍼(buffer)**
데이터를 저장하는 창고라고
할 수 있습니다.

그림 17-6 • 버퍼 스트림의 개념

# 예제

```
inputStream = new BufferedReader(new FileReader("input.txt"));
outputStream = new BufferedWriter(new FileWriter("out put.txt"));
```



# 브릿지 스트림

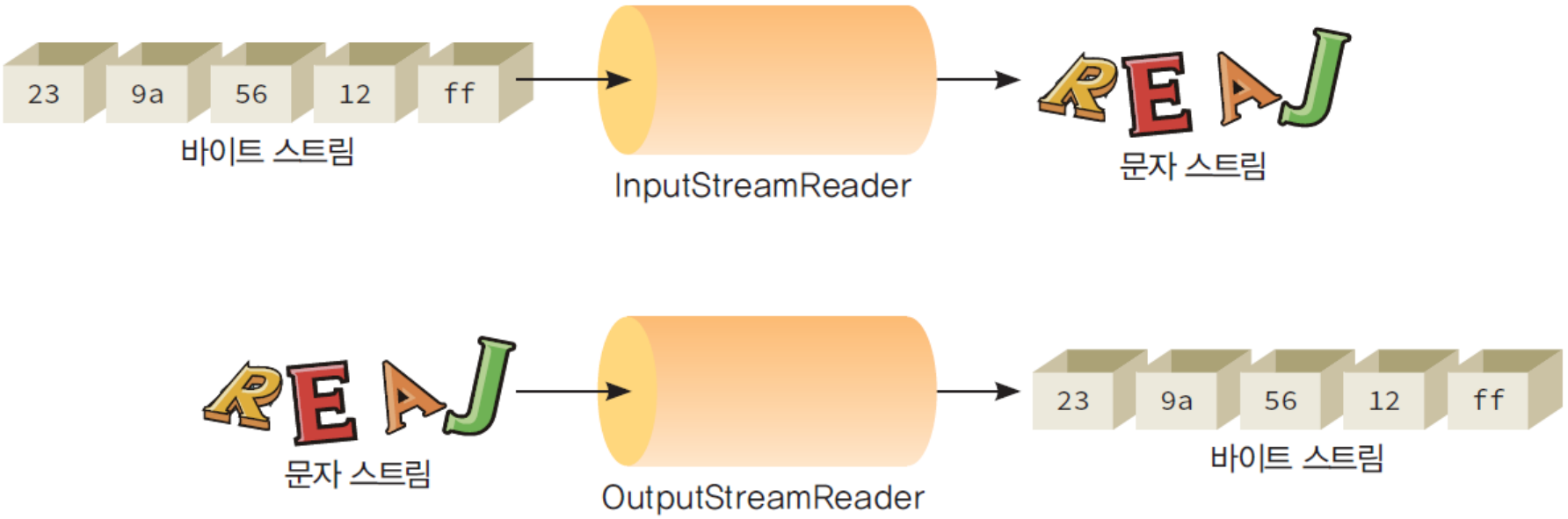



그림 17-7 • 브릿지 스트림

# 파일 정보를 얻으려면

- Path 클래스는 경로를 나타내는 클래스로서 "/home/work" 와 같은 경로를 받아서 객체를 반환한다.
- (예) Path workDirectory = Paths.get("C:₩home₩work");

- File 객체는 파일이 아닌 파일 이름을 나타낸다.
- (예) File file = **new** File("data.txt");