



# 제26장

## IO기반의 입출력-Part\_2

# 3. 파일 입출력

## ❖ FileInputStream

- 파일로부터 **바이트 단위**로 읽어 들일 때 사용
  - 그림, 오디오, 비디오, 텍스트 파일 등 모든 종류의 파일을 읽을 수 있다.

### ■ 객체 생성 방법

//첫 번째 방법

```
FileInputStream fis = new FileInputStream("C:/image.gif");
```

//두 번째 방법

```
File file = new File("C:/image.gif");
```

```
FileInputStream fis = new FileInputStream(file);
```

- FileInputStream 객체가 생성될 때 파일과 **직접 연결**
  - 만약 파일이 존재하지 않으면 **FileNotFoundException** 발생
  - **try-catch문**으로 예외 처리를 해야 한다.
- 
- **InputStream** 하위 클래스이기 때문에, 사용 방법이 **InputStream**과 동일

# 3. 파일 입출력

## ❖ **FileOutputStream**

- 파일에 **바이트 단위**로 데이터를 저장할 때 사용
  - **그림, 오디오, 비디오, 텍스트 등 모든 종류의 데이터를 파일로 저장**

### ■ 객체 생성 방법

//첫 번째 방법

```
FileOutputStream fos = new FileOutputStream("C:/image.gif");
```

//두 번째 방법

```
File file = new File("C:/image.gif");
```

```
FileOutputStream fos = new FileOutputStream(file);
```

- **파일이 이미 존재할 경우, 데이터를 출력하게 되면 파일을 덮어쓰는 단점이 존재한다.**
- **기존 파일 내용 끝에 데이터를 추가할 경우에는 생성자에 true매개값을 주면 된다.(중요)**

```
FileOutputStream fis = new FileOutputStream("C:/Temp/data.txt", true);
```

```
FileOutputStream fis = new FileOutputStream(file, true);
```

- **OutputStream 하위 클래스이기 때문에 사용 방법이 OutputStream과 동일하다.**

# 3. 파일 입출력

## ❖ FileReader

- **텍스트 파일**로부터 데이터를 읽어 들일 때 사용
  - 문자 단위로 읽기 때문에, 텍스트가 아닌 그림, 오디오, 비디오 등의 파일은 읽을 수 없다.

- 객체 생성 방법

//방법 1

```
FileReader fr = new FileReader("C:/Temp/file.txt");
```

//방법 2

```
File file = new File("C:/Temp/file.txt");
```

```
FileReader fr = new FileReader(file);
```

- **FileReader** 객체가 생성될 때 파일과 직접 연결
  - 만약 파일이 존재하지 않으면 **FileNotFoundException** 발생
  - **try-catch**문으로 예외 처리를 해야 한다.
- **Reader** 하위 클래스이므로 사용 방법 Reader와 동일하다.

# 3. 파일 입출력

## ❖ **FileWriter**

- **텍스트 파일에 문자 데이터를 저장할 때 사용**
  - 텍스트가 아닌 그림, 오디오, 비디오 등의 데이터를 파일로 저장할 수 없다.

- **객체 생성 방법**

```
//첫 번째 방법  
FileWriter fw = new FileWriter("C:/file.txt");
```

```
//두 번째 방법  
File file = new File("C:/file.txt");  
FileWriter fw = new FileWriter(file);
```

- **파일이 이미 존재할 경우, 데이터를 출력하게 되면 파일을 덮어쓰는 단점이 존재한다.**
- **기존 파일 내용 끝에 데이터를 추가할 경우에는 생성자에 true매개값을 주면 된다.(중요)**

```
FileWriter fw = new FileWriter("C:/Temp/file.txt", true);  
FileWriter fw = new FileWriter(file, true);
```

- **Writer 하위 클래스이므로 사용 방법이 Writer와 동일하다.**

## 4. 보조 스트림

### ❖ 보조 스트림

- 다른 스트림과 연결이 되어 여러가지 편리한 기능을 제공해주는 스트림
  - 문자 변환, 입출력 성능 향상, 기본 데이터 타입 입출력, 객체 입출력 등의 기능을 제공



예) 입력스트림으로 문자가 들어온다면 InputStream으로 받는것 보다 보조스트림 즉 InputStreamReader를 연결해서 사용하는 것이 훨씬 효율적일 것이다.

### ■ 보조 스트림 생성

```
보조스트림 변수 = new 보조스트림(연결스트림)
```

```
InputStream is = System.in; //InputStream이 주입력스트림이 된다.(키보드로 입력)  
InputStreamReader reader = new InputStreamReader(is); //보조스트림(문자)을 연결함
```

### ■ 보조 스트림 체인 - 다른 보조 스트림과 연결되어 역할 수행



예) 주입력스트림으로 들어오는 것이 문자라면 문자스트림을 보조스트림으로 연결하고, 속도 및 성능 향상을 위해서 스트림을 연결할 수 있다.(대용량데이터)

## 4. 보조 스트림

### ❖ 문자 변환 보조 스트림

- 소스 스트림이 바이트 기반 스트림이지만 **데이터가 문자일 경우 사용함.**
  - Reader와 Writer는 문자 단위로 입,출력하기 때문에 바이트 기반 스트림 보다 편리하다.
  - 문자셋의 종류를 지정할 수 있기 때문에 다양한 문자를 입출력 가능하다.
- **InputStreamReader(InputStream을 Reader로 변환한다)**

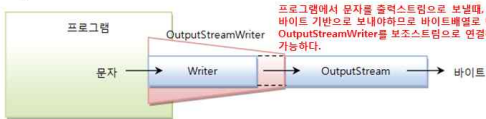


```
InputStream is = System.in; //콘솔로부터 얻는 InputStream이라면, 문자만 입력받는다.  
// (강제캐스팅이나 문자배열로 바꿔야 하는 작업이 생략될 수 있다.)  
Reader reader = new InputStreamReader(is); // InputStreamReader를 연결함.(보조스트림)
```

```
FileInputStream fis = new FileInputStream("C:/file.txt"); //위와 동일하다. 차이는 파일인 것이다.  
Reader reader = new InputStreamReader(fis);
```

## 4. 보조 스트림

### ■ OutputStreamWriter



프로그램에서 문자를 출력스트림으로 보낼때, OutputStream으로 보낼려면, 바이트 기반으로 보내야하므로 바이트배열로 변환하는 등의 작업이 필요하지만, OutputStreamWriter를 보조스트림으로 연결해주면 그러한 작업이 생략 가능하다.

//문자파일에 저장하는 **FileOutputStream**을 생성했지만 바이트기반이다. 하여 아래와 같이 **//OutputStreamWriter**을 보조스트림으로 연결해주면 된다.

```
FileOutputStream fos = new FileOutputStream("C:/file.txt");  
Writer writer = new OutputStreamWriter(fos);
```



## 4. 보조 스트림

### ❖ 성능 향상 보조 스트림

#### ■ 입,출력 성능에 영향을 미치는 입출력 소스

- 하드 디스크 : 속도가 무지하게 느리다.(ex. 낮은 r(revolutions)pm수를 가진 HDD)
- 느린 네트워크(ex. 10Mbps)

\* 프로그램이 아무리 빠른 성능을 지니고 있다고 해서, 환경이 안 좋으면 안 좋은 쪽으로 맞춰지기 때문에 최대한 성능이 빠른 버퍼와 작업하는 것이 좋다.

#### ■ 버퍼를 이용한 해결

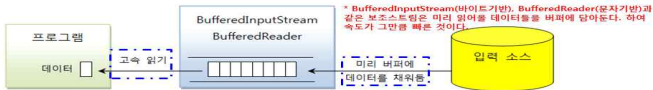
- 입출력 소스와 직접 작업하지 않고 버퍼(buffer)와 작업하므로, 실행 성능 향상



- 프로그램은 쓰기 속도 향상
- 버퍼 차게 되면 데이터를 한꺼번에 하드 디스크로 보내 출력 횟수를 줄여줌

## 4. 보조 스트림

### ■ BufferedInputStream, BufferedReader



```
BufferedInputStream bis = new BufferedInputStream (바이트입력스트림);  
BufferedReader br = new BufferedReader(문자입력스트림);
```

### ■ BufferedOutputStream과 BufferedWriter



```
BufferedOutputStream bos = new BufferedOutputStream (바이트출력스트림);  
BufferedWriter bw = new BufferedWriter(문자출력스트림);
```

감사합니다.

