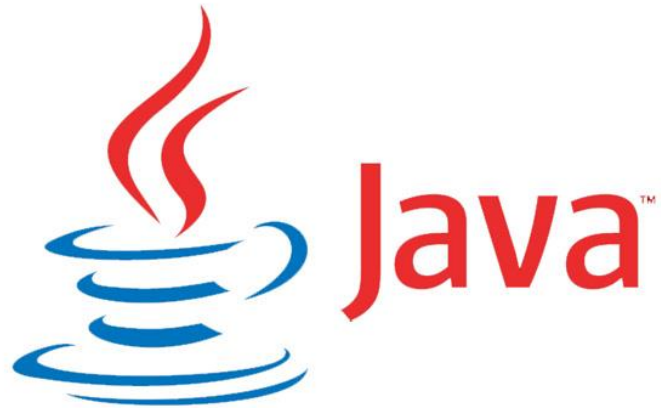


## 제9장 데이터 입출력



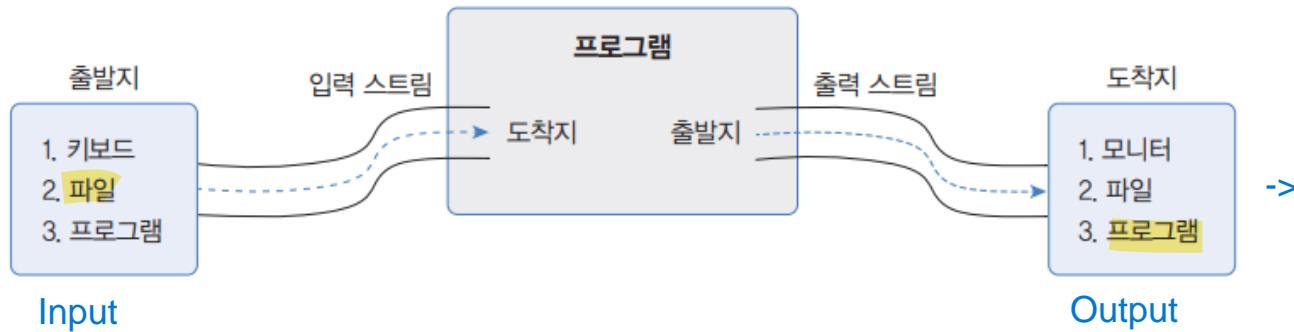
# 목차

---

1. 입출력 스트림
2. 바이트 스트림과 문자 스트림
3. 보조 스트림
4. 직렬화와 역직렬화
5. 파일 관리

# 1. 입출력 스트림


- 파일<sup>File</sup>은 컴퓨터 저장매체에 저장되는 데이터를 읽고, 쓰기 위한 데이터 저장 매체
- 프로그램에 데이터를 입출력을 위해 스트림<sup>Stream</sup> 사용
- **스트림**은 입출력(I/O) 장치와 프로그램 간 데이터 전송 통로를 의미



## 2. 바이트 스트림과 문자 스트림

1byte = 8bit

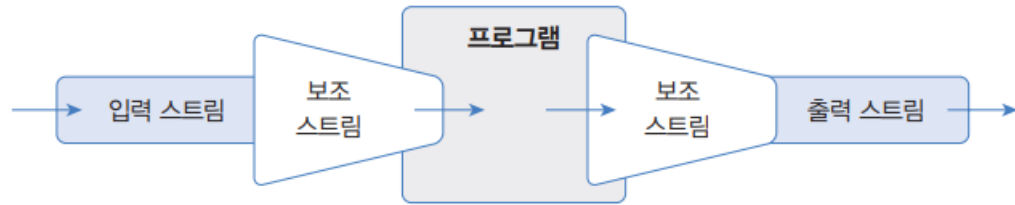
- 바이트 스트림은 데이터를 바이트 단위로 처리, 주로 이미지, 오디오, 비디오 파일과 같은 바이너리 데이터를 다룰 때 사용
- 문자 스트림은 데이터를 문자 단위로 처리, 주로 텍스트 파일을 다룰 때 사용

특징	바이트 스트림	문자 스트림
사용 목적	바이너리 데이터 입출력 (            )	텍스트 데이터 입출력 (            )
최상위 클래스	InputStream, OutputStream	Reader, Writer
대표 클래스	 FileInputStream, FileOutputStream	FileReader, FileWriter
크기	1byte (8 bit)	2 byte (16 bit)
사용 예	파일 복사, 네트워크 데이터 전송	텍스트 파일 읽기/쓰기

### 3. 보조 스트림

---

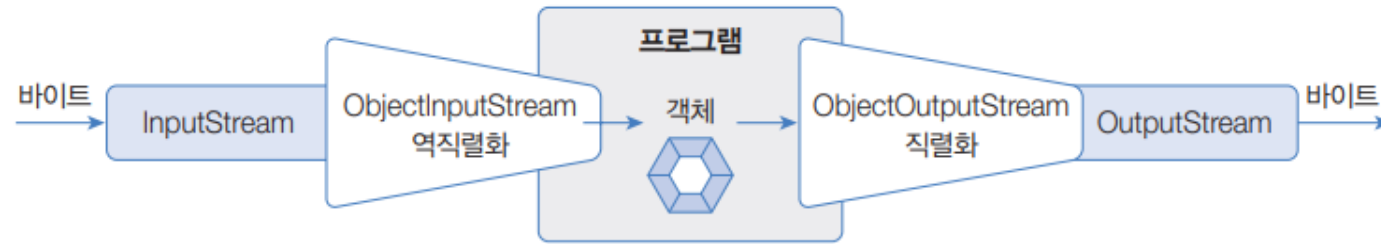
- 기본 스트림과 연결되어 다양한 기능을 제공하는 스트림
- 기본 입출력 스트림에 Buffer 보조 스트림을 연결해서 실행 성능 향상



## 4. 직렬화와 역직렬화

---

- 직렬화는 메모리에 생성된 객체를 파일 또는 네트워크로 출력하기 위해 Byte 단위로 변경
- 역직렬화는 직렬화된 Byte 단위를 객체로 복원



## 5. 파일 관리

---

- File 클래스는 파일과 디렉토리를 관리하는 클래스
- Files 클래스는 File 클래스를 개선, 더 많은 기능을 제공

주요 메서드	내용
<code>createNewFile()</code>	새 파일 생성
<code>mkdir()</code>	새 디렉토리 생성
<code>delete()</code>	파일 또는 디렉토리 삭제
<code>exists()</code>	파일 또는 디렉토리 존재 여부 확인
<code>getName()</code>	파일 또는 디렉토리 이름 리턴
<code>getPath()</code>	파일 또는 디렉토리 경로 리턴
<code>getAbsolutePath()</code>	파일 또는 디렉토리 절대 경로 리턴
<code>isFile()</code>	파일 여부 확인
<code>isDirectory()</code>	디렉토리 여부 확인