

# 바이트 스트림

# IO 스트림

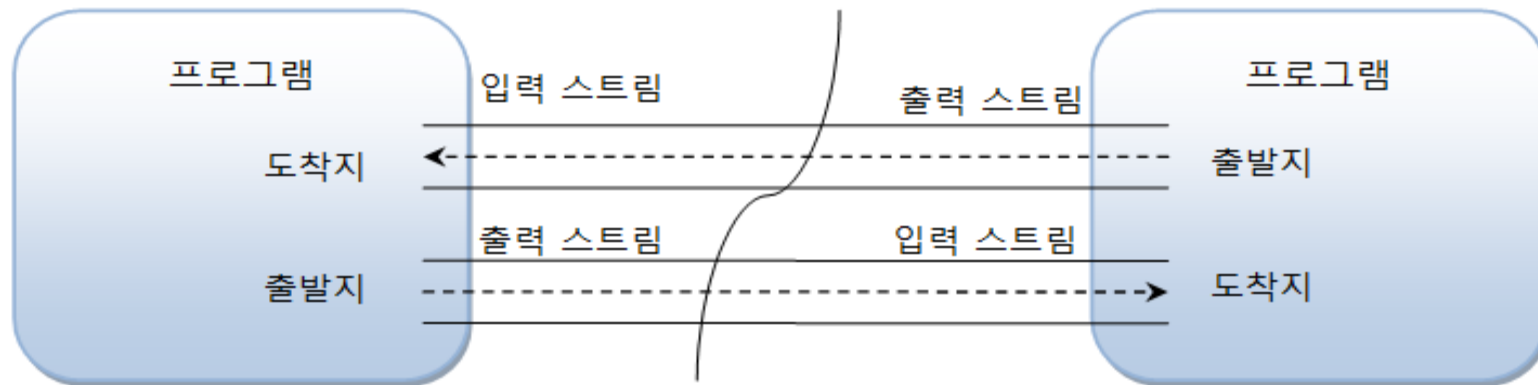
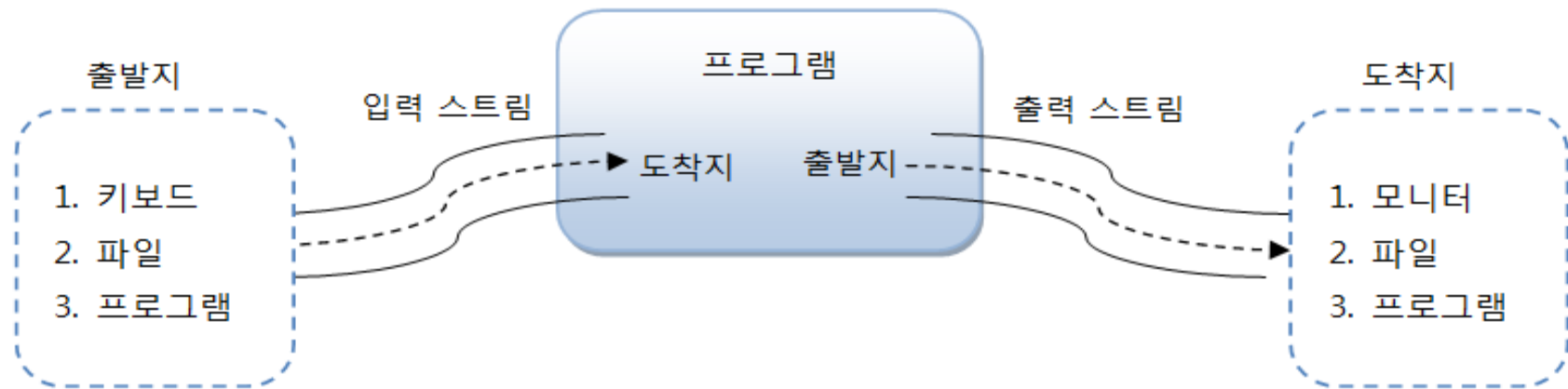
## ❖ java.io 패키지

- 자바의 기본적인 데이터 입출력(IO: Input/Output) API 제공

java.io 패키지의 주요 클래스	설명
File	파일 시스템의 파일의 정보를 얻기 위한 클래스
Console	콘솔로부터 문자를 입출력하기 위한 클래스
InputStream / OutputStream	바이트 단위 입출력을 위한 최상위 입출력 스트림 클래스
FileInputStream / FileOutputStream DataInputStream / DataOutputStream ObjectInputStream / ObjectOutputStream PrintStream BufferedInputStream / BufferedOutputStream	바이트 단위 입출력을 위한 하위 스트림 클래스
Reader / Writer	문자 단위 입출력을 위한 최상위 입출력 스트림 클래스
FileReader / FileWriter InputStreamReader / OutputStreamWriter PrintWriter BufferedReader / BufferedWriter	문자 단위 입출력을 위한 하위 스트림 클래스

# IO 스트림

## ❖ 입력 스트림과 출력 스트림의 개념

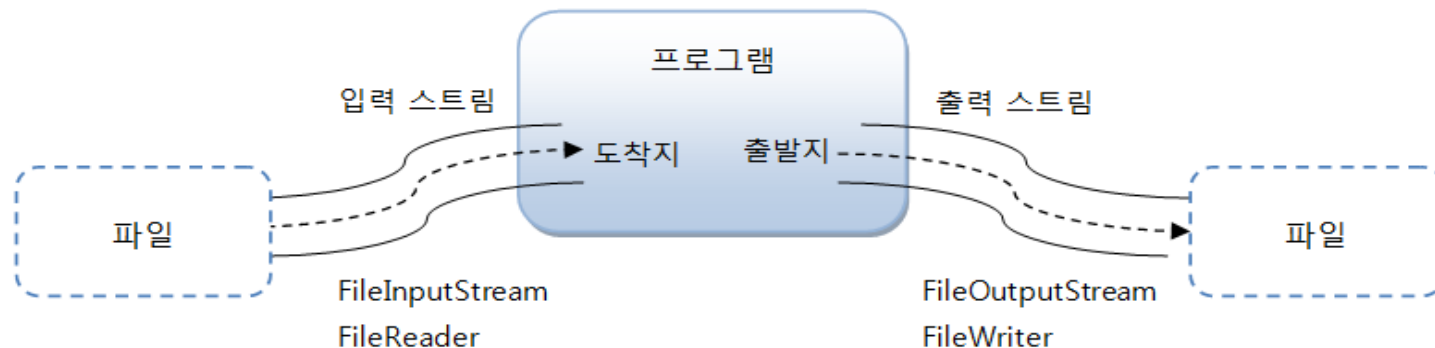


# IO 스트림

## ❖ 바이트 기반 스트림과 문자 기반 스트림

- 바이트 기반 스트림
  - 그림, 멀티미디어, 문자 등 모든 종류의 데이터를 받고 보내는 것 가능
- 문자 기반 스트림
  - 문자만 받고 보낼 수 있도록 특화

구분	바이트 기반 스트림		문자 기반 스트림	
	입력 스트림	출력 스트림	입력 스트림	출력 스트림
최상위 클래스	InputStream	OutputStream	Reader	Writer
하위 클래스 (예)	XXXpustream (FileInputStream)	XXXOutputStream (FileOutputStream)	XXXReader (FileReader)	XXXWriter (FileWriter)

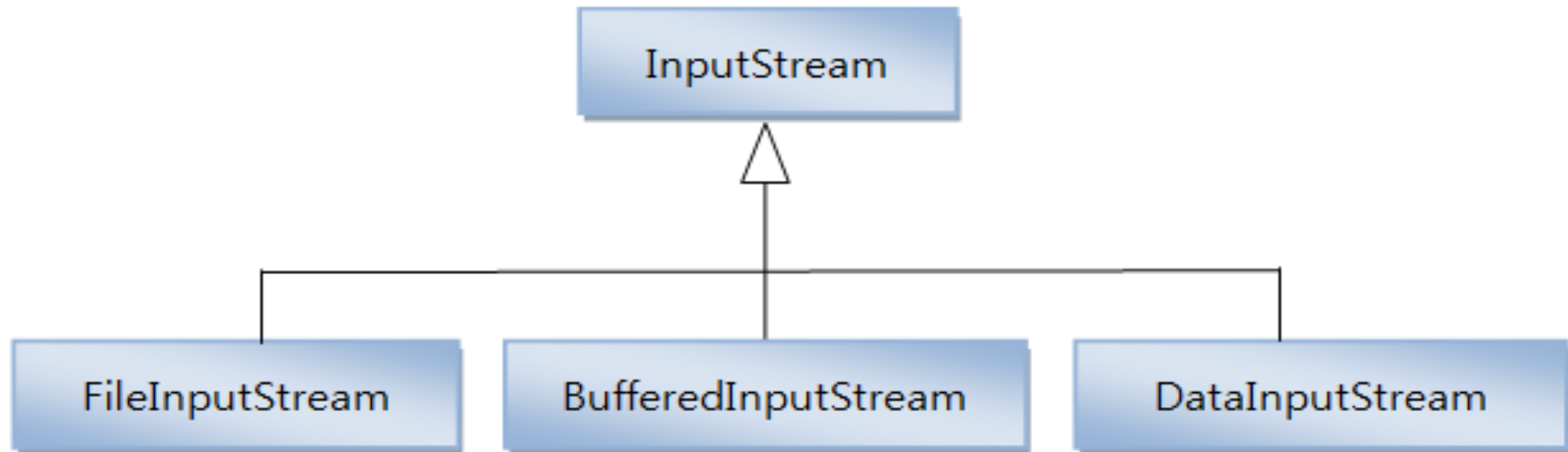


## 바이트 스트림

---

### ❖ InputStream

- 바이트 기반 입력 스트림의 최상위 클래스로 추상 클래스



## 바이트 스트림

### ❖ InputStream

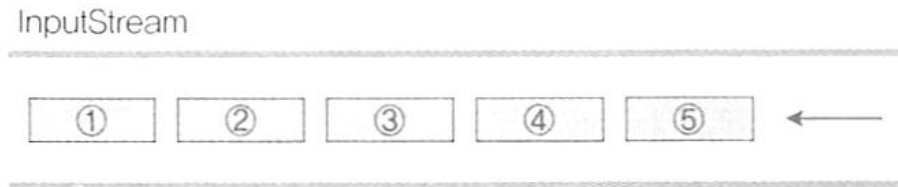
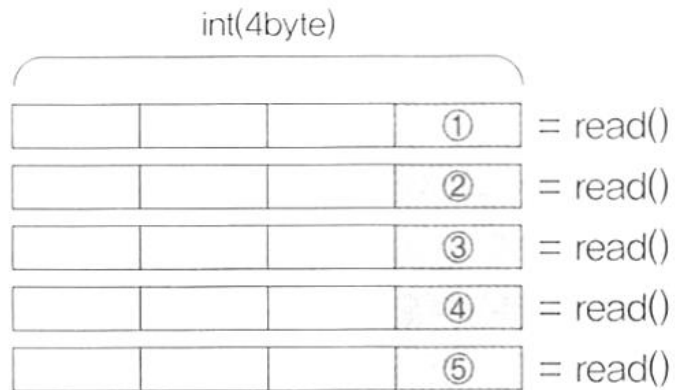
- InputStream 클래스의 주요 메소드

리턴타입	메소드	설명
int	read()	입력 스트림으로부터 1 바이트를 읽고 읽은 바이트를 리턴한다.
int	read(byte[] b)	입력 스트림으로부터 읽은 바이트들을 매개값으로 주어진 바이트 배열 b 에 저장하고 실제로 읽은 바이트 수를 리턴한다.
int	read(byte[] b, int off, int len)	입력 스트림으로부터 len 개의 바이트 만큼 읽고 매개값으로 주어진 바이트 배열 b[off] 부터 len 개까지 저장한다. 그리고 실제로 읽은 바이트 수인 len 개를 리턴한다. 만약 len 개를 모두 읽지 못하면 실제로 읽은 바이트 수를 리턴한다.
void	close()	사용한 시스템 자원을 반납하고 입력 스트림을 닫는다.

# 바이트 스트림

## ❖ InputStream

- read() 메서드



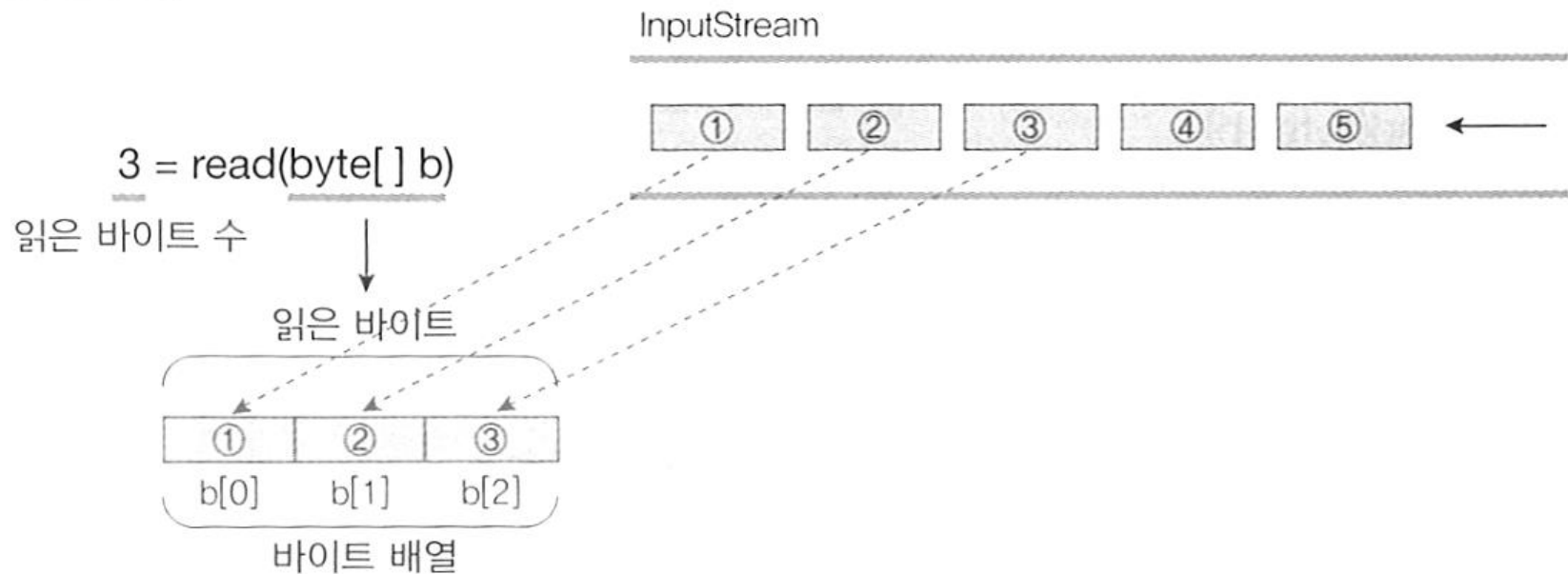
```
InputStream is = new FileInputStream("C:/test.jpg");  
int readByte;  
while ((readByte=is.read()) != -1) { ... }
```

# 바이트 스트림

## ❖ InputStream

- `read(byte[] b)` 메서드
  - 5바이트 데이터를 3바이트 배열로 읽을 때

첫 번째 읽을 경우



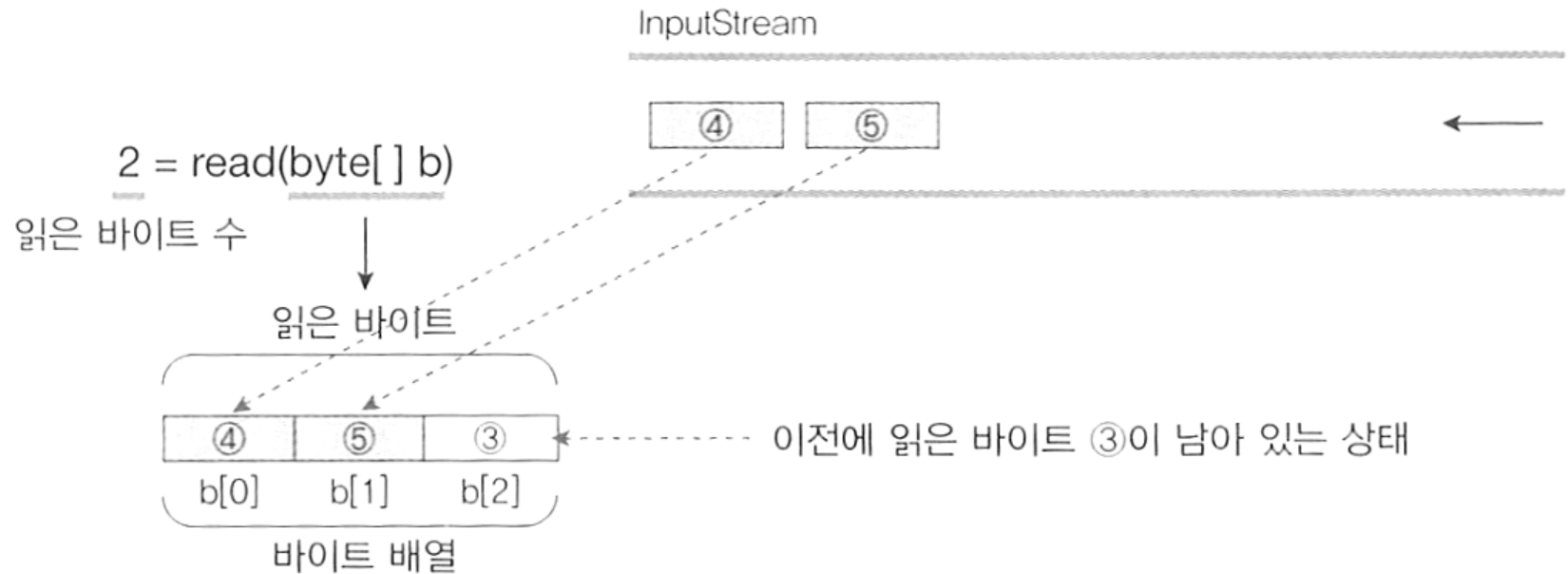


# 바이트 스트림

## ❖ InputStream

- `read(byte[] b)` 메서드
  - 5바이트 데이터를 3바이트 배열로 읽을 때

두 번째 읽을 경우



# 바이트 스트림

---

## ❖ InputStream

- read(byte[] b) 메서드

```
InputStream is = new FileInputStream("C:/test.jpg");  
int readByteNo;  
byte[ ] readBytes = new byte[100];  
while ((readByteNo=is.read(readBytes)) != -1) { ... }
```

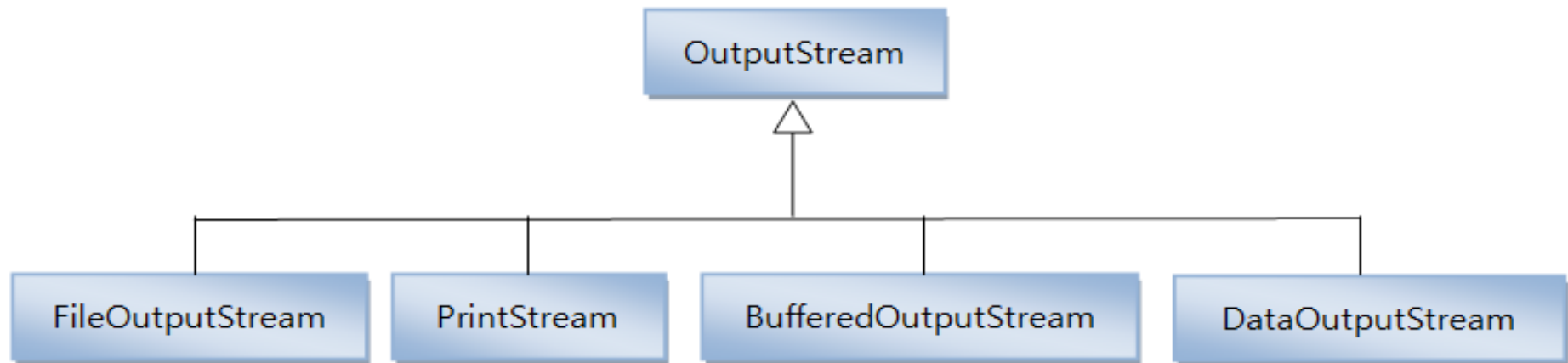
- close() 메서드
  - InputStream에서 사용했던 시스템 자원을 해제

## 바이트 스트림

---

### ❖ OutputStream

- 바이트 기반 출력 스트림의 최상위 클래스로 추상 클래스



## 바이트 스트림

### ❖ OutputStream

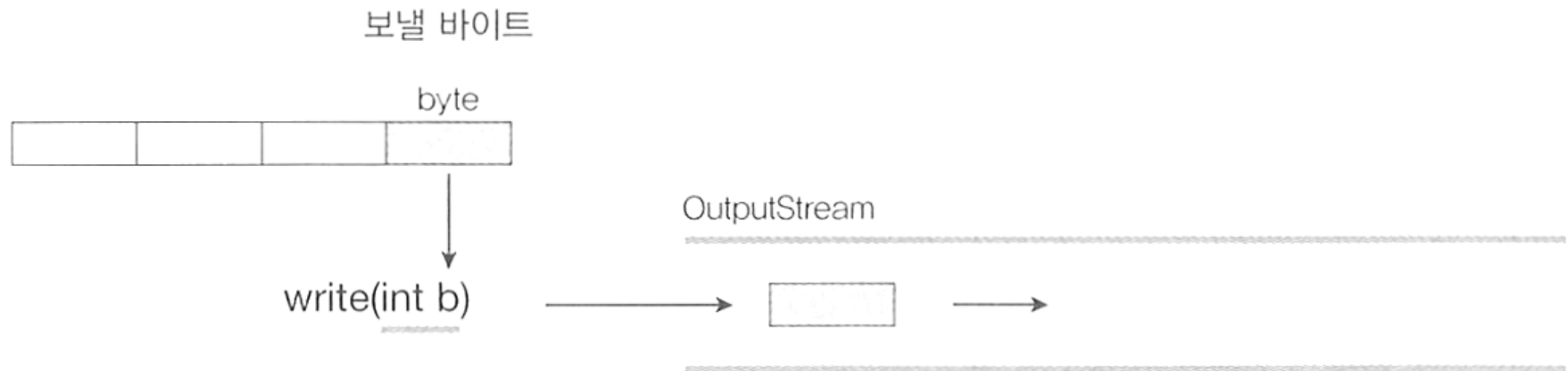
- OutputStream의 주요 메소드

리턴타입	메소드	설명
void	write(int b)	출력 스트림으로 1 바이트를 보낸다.
void	write(byte[] b)	출력 스트림에 매개값으로 주어진 바이트 배열 b 의 모든 바이트를 보낸다.
void	write(byte[] b, int off, int len)	출력 스트림에 매개값으로 주어진 바이트 배열 b[off] 부터 len 개까지의 바이트를 보낸다.
void	flush()	버퍼에 잔류하는 모든 바이트를 출력한다.
void	close()	사용한 시스템 자원을 반납하고 출력 스트림을 닫는다.

# 바이트 스트림

## ❖ OutputStream

- `write(int b)` 메서드
  - 매개변수로 주어진 `int` 값에서 끝에 있는 1바이트만 출력 스트림으로 보냄

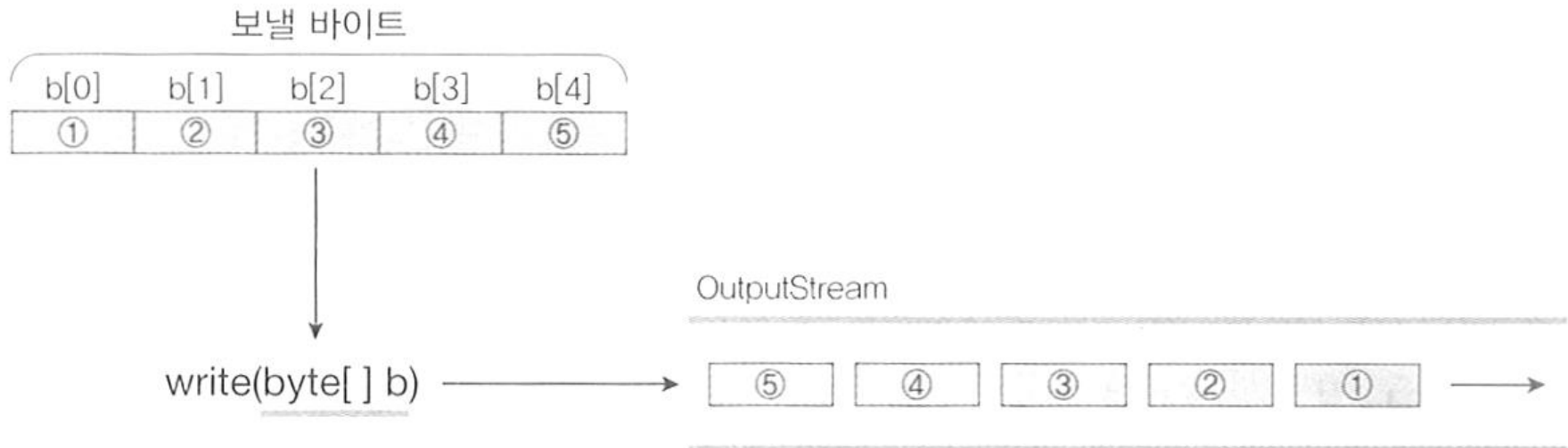


```
OutputStream os = new FileOutputStream("C:/test.txt");
byte[ ] data = "ABC".getBytes();
for(int i=0; i<data.length; i++) {
    os.write(data[i]); // "A", "B", "C"를 하나씩 출력
}
```

# 바이트 스트림

## ❖ OutputStream

- write(byte[] b) 메서드
  - 매개변수로 주어진 바이트 배열의 모든 바이트를 출력 스트림으로 보냄

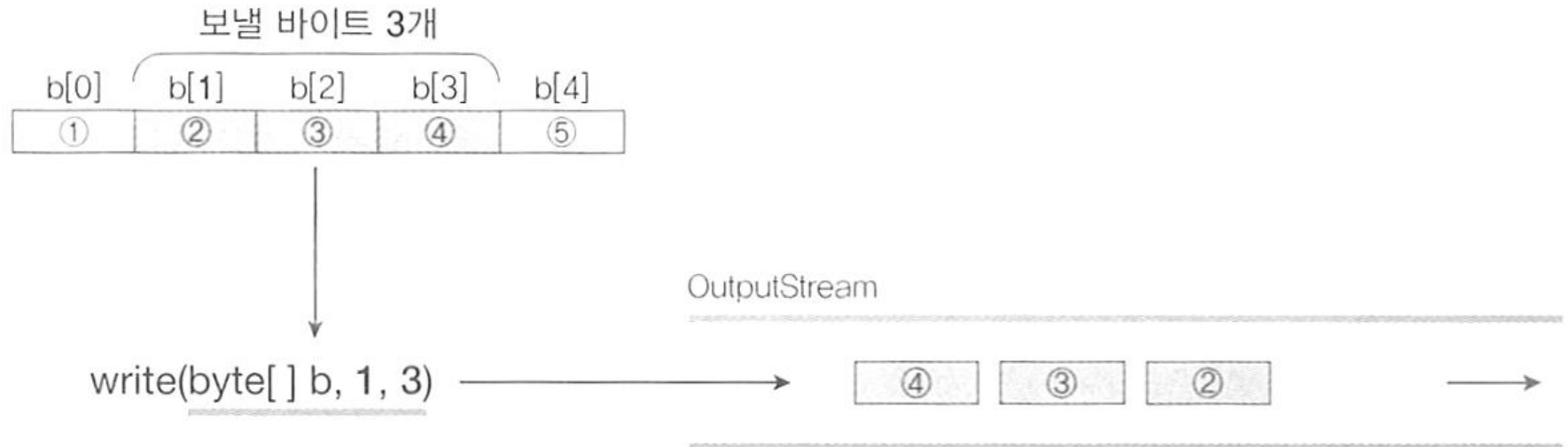


```
OutputStream os = new FileOutputStream("C:/test.txt");  
byte[] data = "ABC".getBytes();  
os.write(data); // "ABC" 모두 출력
```

# 바이트 스트림

## ❖ OutputStream

- `write(byte[] b, int off, int len)` 메서드
  - `b[off]` 부터 `len`개의 바이트를 출력 스트림으로 보냄



```
OutputStream os = new FileOutputStream("C:/test.txt");  
byte[] data = "ABC".getBytes();  
os.write(data, 1, 2); // "BC"만 출력
```

# 바이트 스트림

---

## ❖ OutputStream

- flush()와 close() 메서드
  - 버퍼에 잔류하고 있는 데이터를 모두 출력시키고 버퍼를 비움

```
OutputStream os = new FileOutputStream("C:/test.txt");  
byte[ ] data = "ABC".getBytes();  
os.write(data);  
os.flush();  
os.close();
```