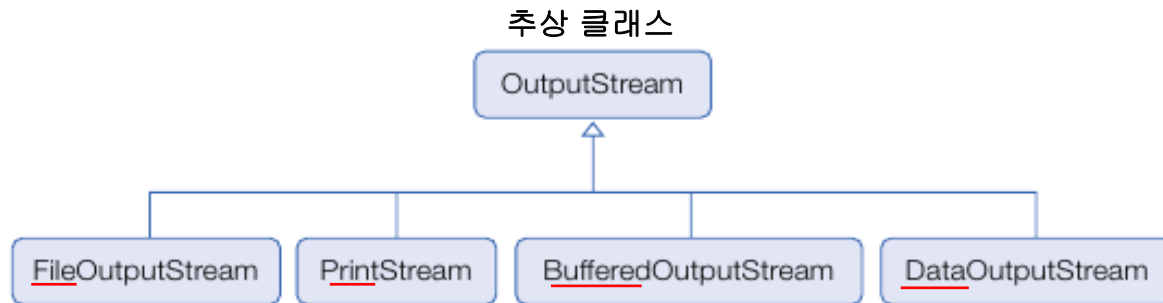


바이트 출력 스트림

## ■ 바이트 출력 스트림

- OutputStream은 바이트 출력 스트림의 최상위 클래스로 추상 클래스이다.
- OutputStream 클래스에는 모든 바이트 출력 스트림이 기본적으로 가져야 할 메소드가 정의되어 있다

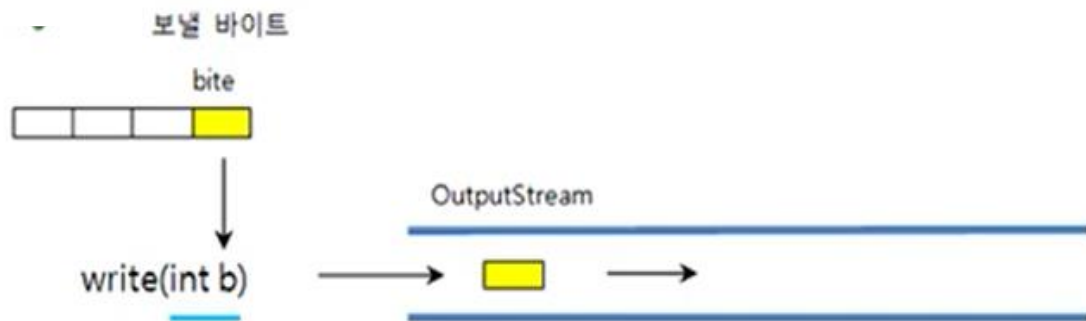


리턴 타입	메소드	설명
void	write(int b)	1byte를 출력
void	write(byte[ ] b)	매개값으로 주어진 배열 b의 모든 바이트를 출력
void	write(byte[ ] b, int off, int len)	매개값으로 주어진 배열 b[off]부터 len개의 바이트를 출력
void	flush()	출력 버퍼에 잔류하는 모든 바이트를 출력
void	close()	출력 스트림을 닫고 사용 메모리 해제

◎ 1 byte 출력

- write(int b) 메소드는 매개값 int(4byte)에서 끝 1byte만 출력한다.
- 매개변수가 int 타입이므로 4byte 모두를 보내는 것은 아니다.

➤ write(int b) 메소드

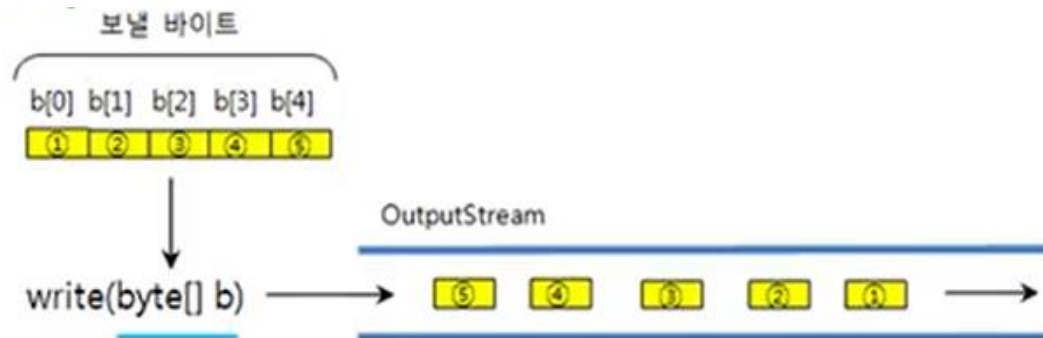


```
OutputStream os = new FileOutputStream("C:/test.txt");  
byte[] data = "ABC".getBytes();  
for(int i=0; i<data.length; i++) {  
    os.write(data[i]); // "A", "B", "C"를 하나씩 출력  
}
```

## ◎ byte 배열 출력

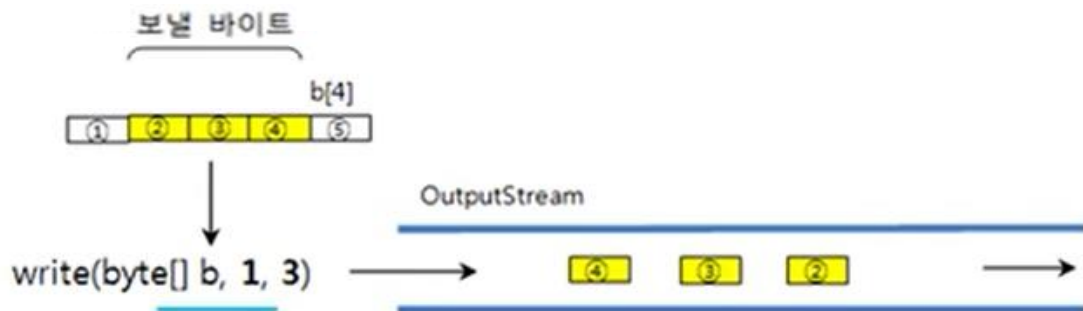
- 일반적으로 1byte를 출력하는 경우는 드물고, 보통 byte 배열을 통째로 출력하는 경우가 많다.
- `write(byte[] b)` 메소드는 매개값으로 주어진 배열의 모든 바이트를 출력한다.
- 만약 배열의 일부분을 출력하고 싶다면 `write(byte[] b, int off, int len)` 메소드를 사용하면 된다.
- 이 메소드는 `b[off]`부터 `len`개의 바이트를 출력한다.

➤ `write(byte[] b)` 메소드



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

➤ `write(byte[] b, int off, int len)` 메소드



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

➤ `flush()`와 `close()` 메소드

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

◎ 바이너리 값을 파일에 저장하는 바이트 스트림 코드

```
FileOutputStream fout = new FileOutputStream("c:\\Temp\\test.out");
```

```
byte b[] = {7,51,3,4,-1,24};
```

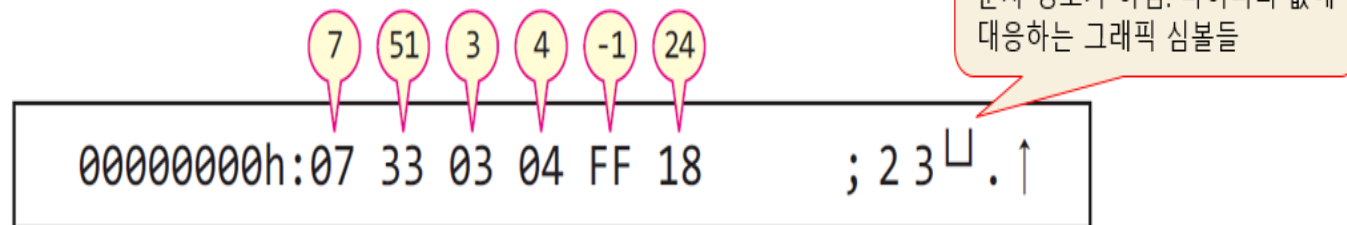
```
for(int i=0; i<b.length; i++)
```

```
    fout.write(b[i]);
```

파일에 배열 b[i]의 정수 값(바이너리)을 그대로 기록

```
fout.close();
```

스트림을 닫음. 파일도 닫힘. 더 이상 스트림으로부터 읽을 수 없음



test.out 파일의 내부