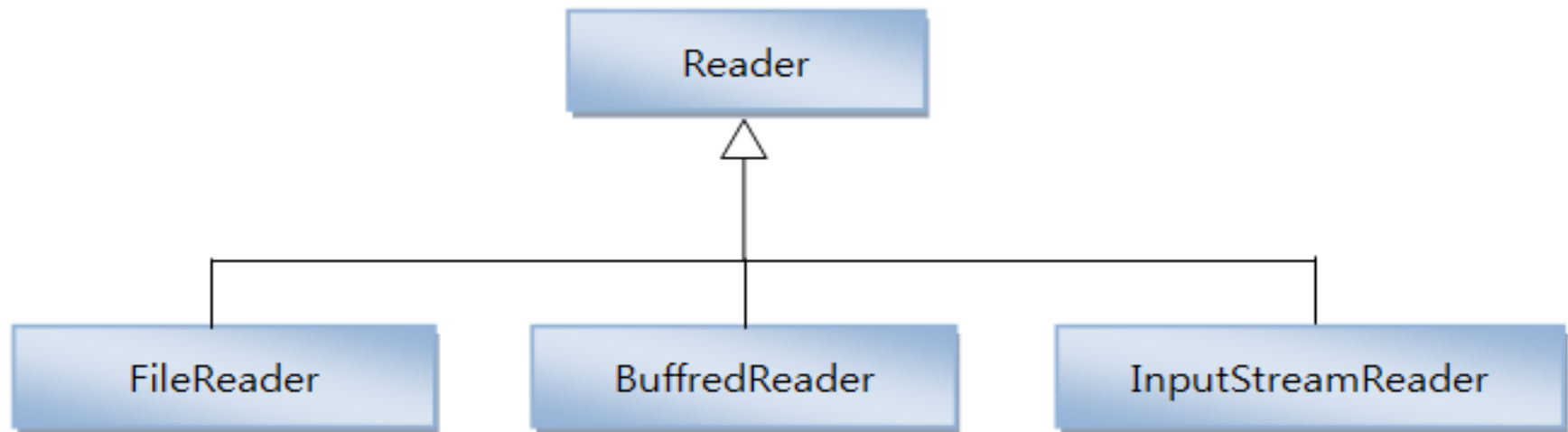


문자 스트림

문자 스트림

❖ Reader

- 문자 기반 입력 스트림의 최상위 클래스로 추상 클래스



문자 스트림

❖ Reader

○ Reader의 주요 메소드

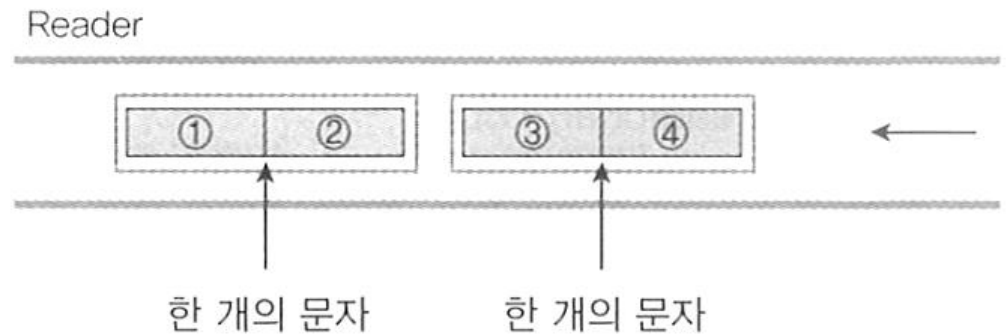
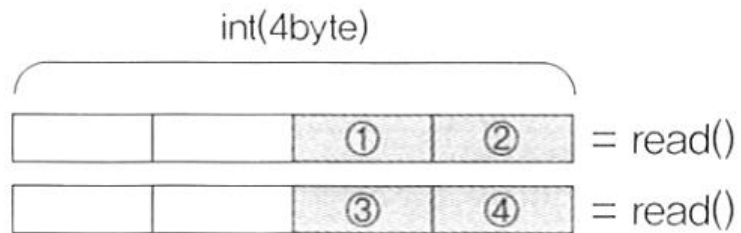
메소드		설명
int	read()	입력 스트림으로부터 한개의 문자를 읽고 리턴한다.
int	read(char[] cbuf)	입력 스트림으로부터 읽은 문자들을 매개값으로 주어진 문자 배열 cbuf 에 저장하고 실제로 읽은 문자 수를 리턴한다.
int	read(char[] cbuf, int off, int len)	입력 스트림으로부터 len 개의 문자를 읽고 매개값으로 주어진 문자 배열 cbuf[off] 부터 len 개까지 저장한다. 그리고 실제로 읽은 문자 수인 len 개를 리턴한다.
void	close()	사용한 시스템 자원을 반납하고 입력 스트림을 닫는다.

문자 스트림

❖ Reader

- read() 메서드

- 한 개의 문자(2바이트)를 읽고 4바이트 int 타입으로 리턴



```
char charData = (char) read();
```

문자 스트림

❖ Reader

- read() 메서드
 - 더 이상 읽을 데이터가 없는 경우 -1을 리턴

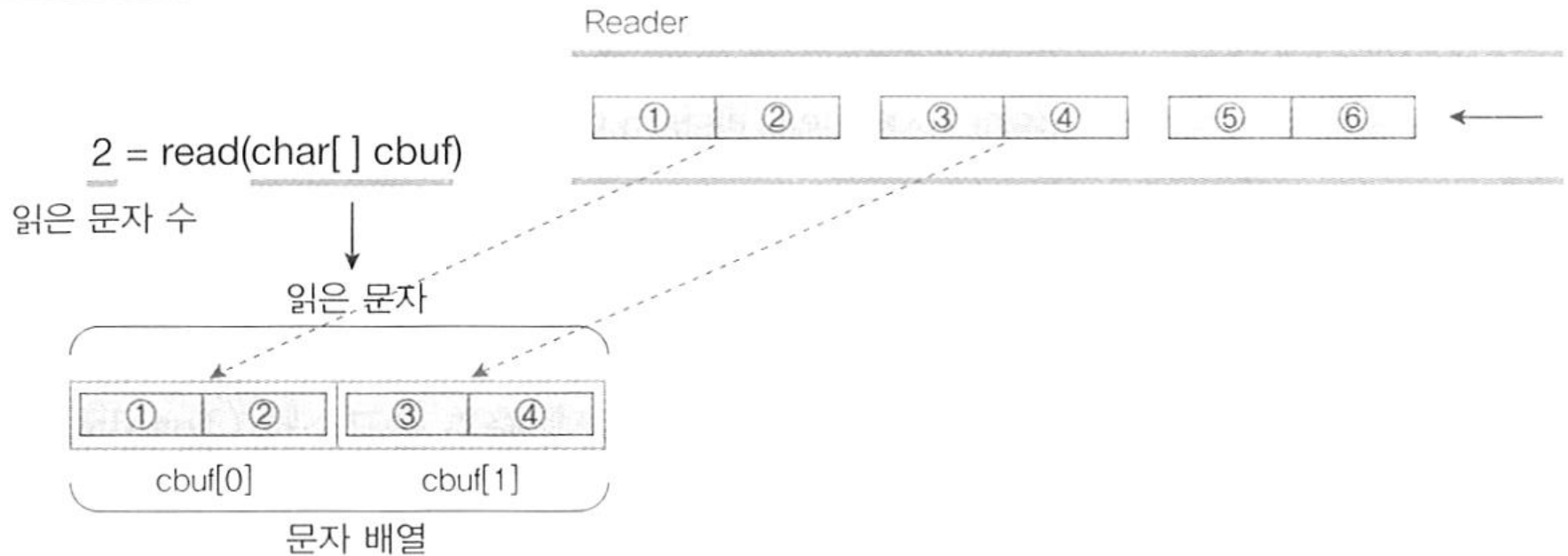
```
Reader reader = new FileReader("C:/test.txt");
int readData;
while ((readData=reader.read()) != -1) {
    char charData = (char) readData;
}
```

문자 스트림

❖ Reader

- `read(char[] cbuf)` 메서드
 - 매개값으로 주어진 문자 배열의 길이만큼 문자를 읽고 배열에 저장
 - 실제로 읽은 바이트 수를 리턴

첫 번째 읽을 경우

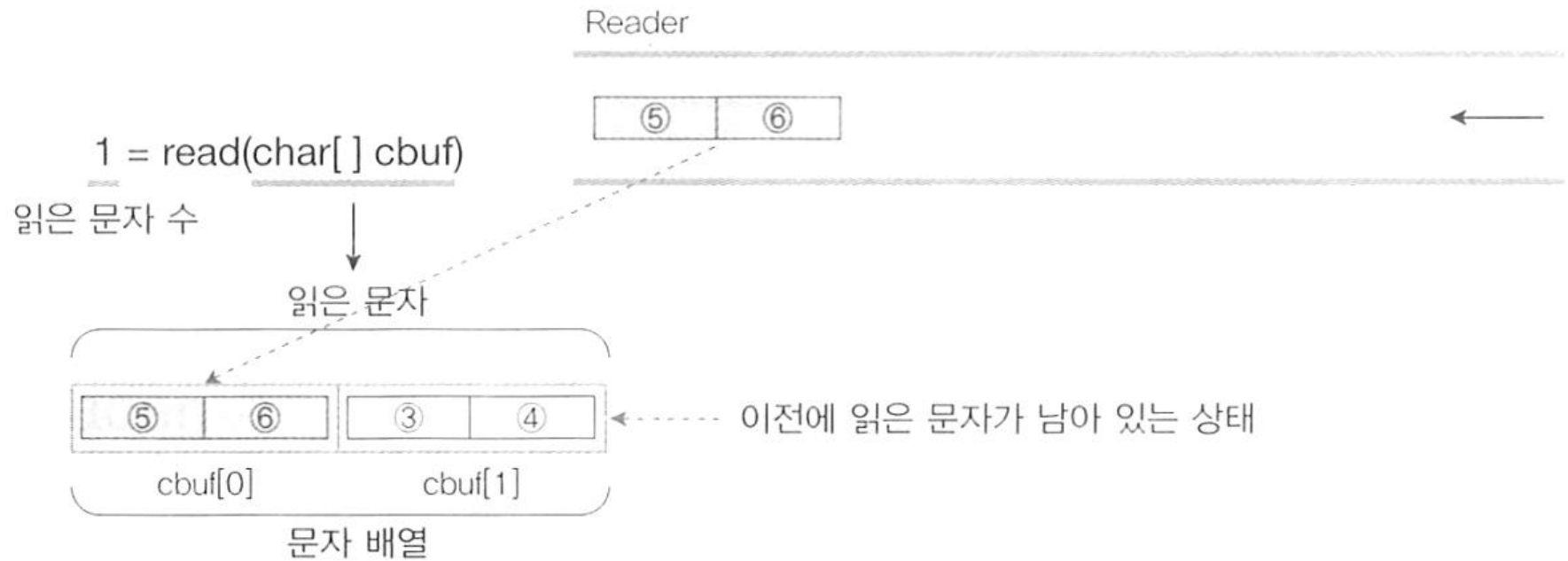


문자 스트림

❖ Reader

- `read(char[] cbuf)` 메서드

두 번째 읽을 경우



문자 스트림

❖ Reader

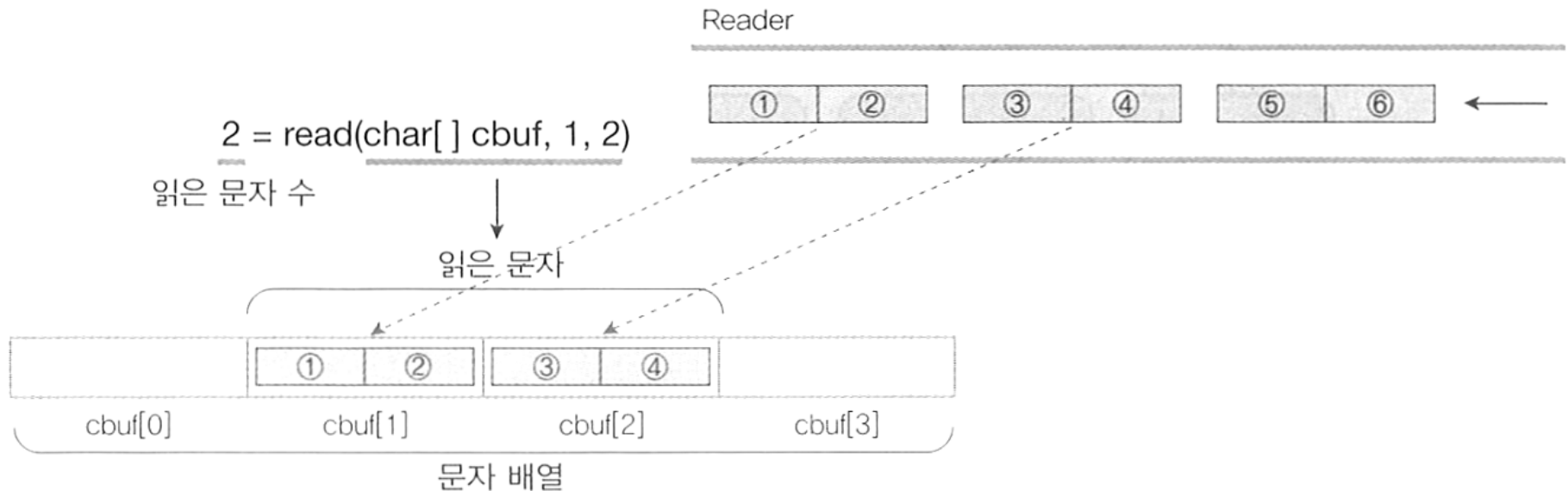
- read(char[] cbuf) 메서드

```
Reader reader = new FileReader("C:/test.txt");  
int readCharNo;  
char[ ] cbuf = new char[2];  
while ((readCharNo=reader.read(cbuf)) != -1) { ... }
```


문자 스트림

❖ Reader

- `read(char[] cbuf, int off, int len)` 메서드
 - `len`개의 문자만큼 읽고 매개값으로 주어진 문자배열 `cbuf[off]`부터 `len`개까지 저장
 - 읽은 문자 수를 리턴



문자 스트림

❖ Reader

- `read(char[] cbuf, int off, int len)` 메서드

```
Reader reader = ...;  
char[] cbuf = new char[100];  
int readCharNo=is.read(cbuf);
```

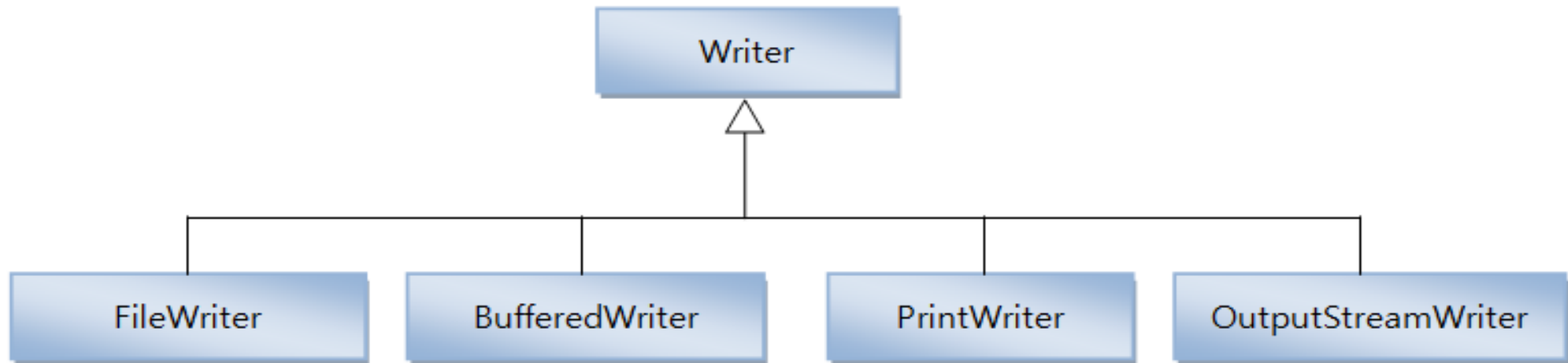
```
Reader reader = ...;  
char[] cbuf = new char[100];  
int readCharNo=is.read(cbuf, 0, 100);
```

- `close()` 메서드
 - Reader에서 사용했던 시스템 자원을 해제
`reader.close();`

문자 스트림

❖ Writer

- 문자 기반 출력 스트림의 최상위 클래스로 추상 클래스



문자 스트림

❖ Writer

○ Writer의 주요 메소드

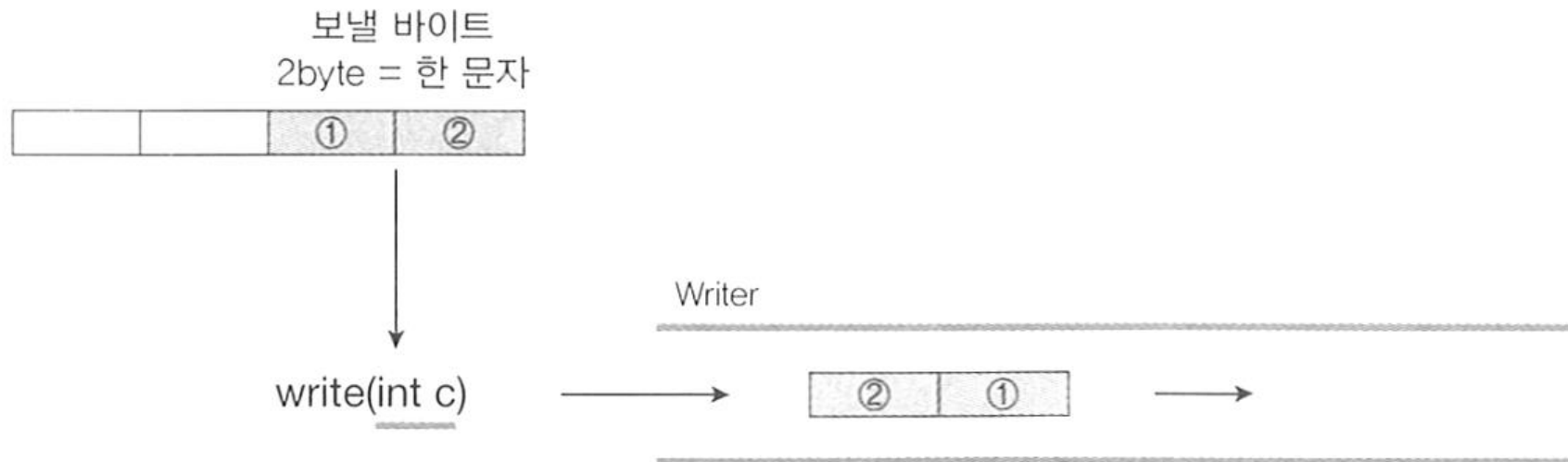
리턴타입	메소드	설명
void	write(int c)	출력 스트림으로 매개값으로 주어진 한 문자를 보낸다.
void	write(char[] cbuf)	출력 스트림에 매개값으로 주어진 문자 배열 cbuf 의 모든 문자를 보낸다.
void	write(char[] cbuf, int off, int len)	출력 스트림에 매개값으로 주어진 문자 배열 cbuf[off] 부터 len 개까지의 문자를 보낸다.
void	write(String str)	출력 스트림에 매개값으로 주어진 문자열을 전부 보낸다.
void	write(String str, int off, int len)	출력 스트림에 매개값으로 주어진 문자열 off 순번부터 len 개까지의 문자를 보낸다.
void	flush()	버퍼에 잔류하는 모든 문자열을 출력한다.
void	close()	사용한 시스템 자원을 반납하고 출력 스트림을 닫는다.

문자 스트림

❖ Writer

○ write(int c) 메서드

- 주어진 int 값에서 끝에 있는 1바이트(한 개의 문자)만 출력 스트림으로 보냄

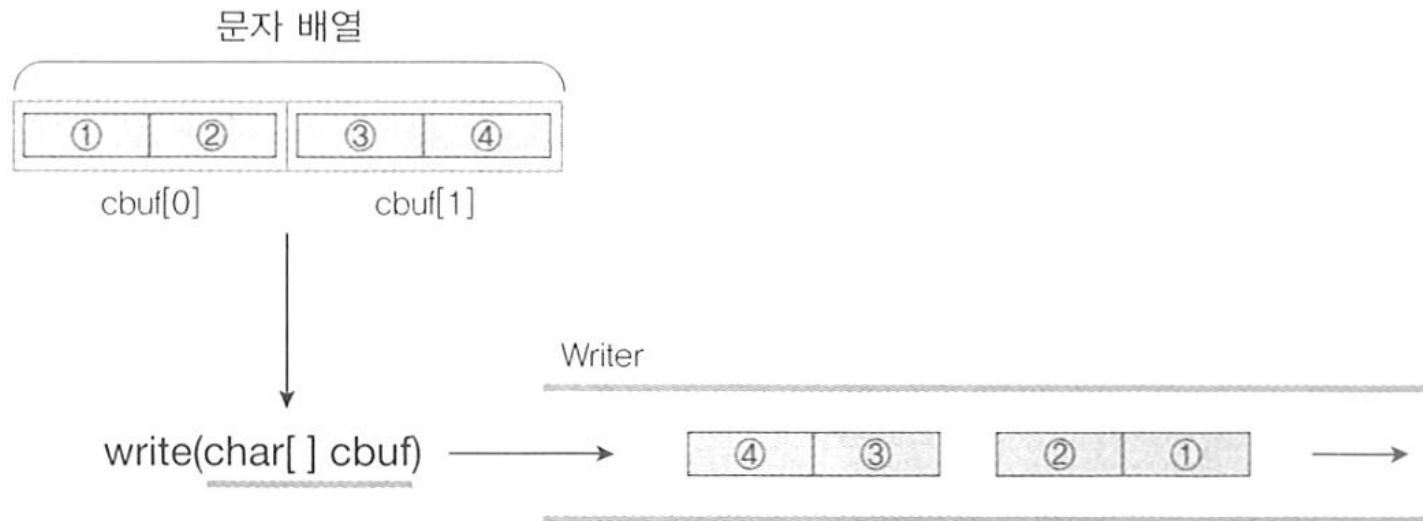


```
Writer writer = new FileWriter("C:/test.txt");  
char[ ] data = "홍길동".toCharArray();  
for(int i=0; i<data.length; i++) {  
    writer.write(data[i]); // "홍", "길", "동"을 하나씩 출력  
}
```

문자 스트림

❖ Writer

- `write(char[] cbuf)` 메서드
 - 매개값으로 주어진 `char[]` 배열의 모든 문자를 출력 스트림으로 보냄

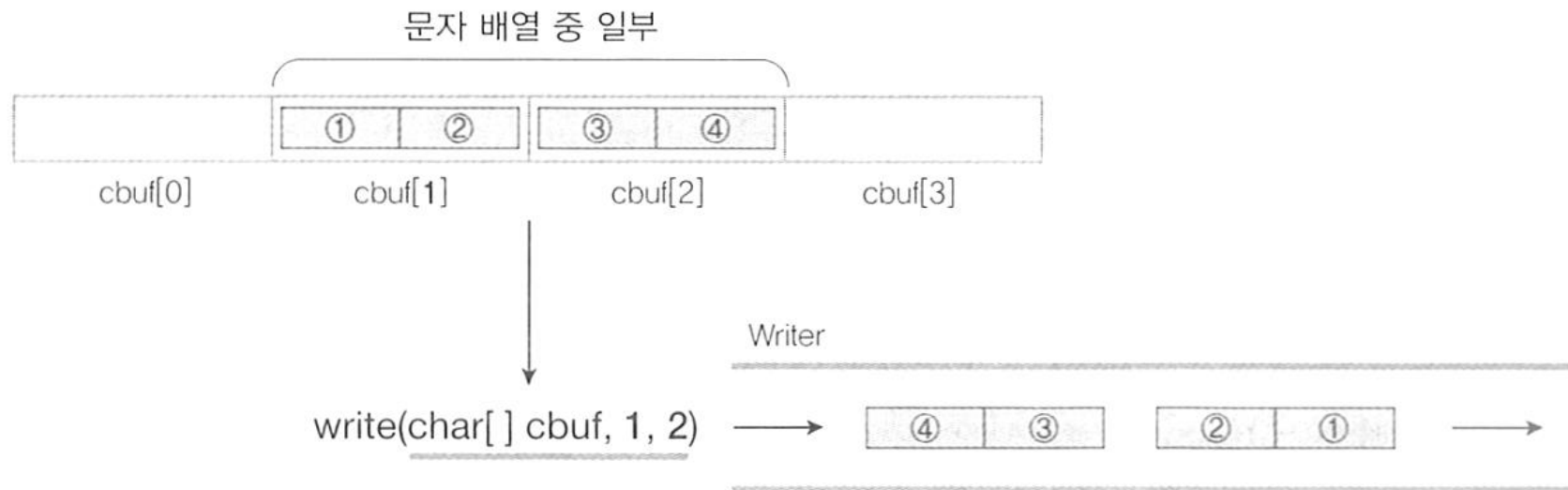


```
Writer writer = new FileWriter("C:/test.txt");  
char[] data = "홍길동".toCharArray();  
writer.write(data); // "홍길동" 모두 출력
```

문자 스트림

❖ Writer

- `write(char[] b, int off, int len)` 메서드
 - `c[off]`부터 `len` 개의 문자를 출력 스트림으로 보냄



```
Writer writer = new FileWriter("C:/test.txt");  
char[] data = "홍길동".toCharArray();  
writer.write(data, 1, 2); // "길동"만 출력
```

문자 스트림

❖ Writer

- `write(String str)`와 `write(String str, int off, int len)` 메서드
 - `write(String str)` : 문자열 전체를 출력



문자 스트림

❖ Writer

- `write(String str)`와 `write(String str, int off, int len)` 메서드
 - `write(String str, int off, int len)` : 주어진 문자열 `off` 순번부터 `len`개까지의 문자 출력



문자 스트림

❖ Writer

- `write(String str)`와 `write(String str, int off, int len)` 메서드

```
Writer writer = new FileWriter("C:/test.txt");  
String data = "안녕 자바 프로그램";  
writer.write(data);  
writer.flush();  
writer.close();
```