

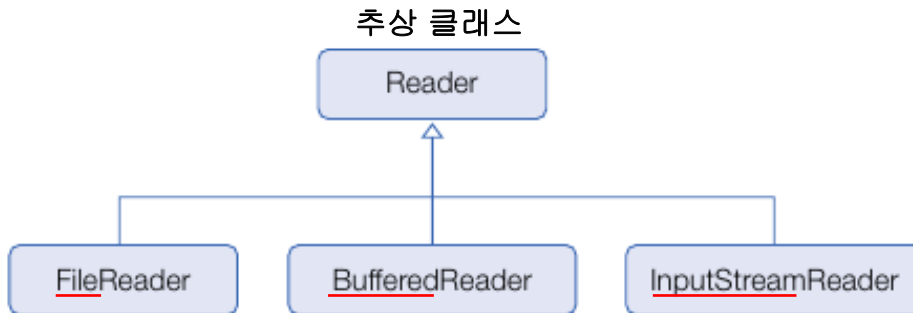
문자 입출력 스트림

▣ 문자 입출력 스트림

- 바이트 입출력 스트림인 `InputStream` 과 `OutputStream`에 대응하는 문자 입출력 스트림으로 `Reader` 와 `Writer`가 있다.
- 입출력되는 단위가 문자인 것을 제외하고는 바이트 입출력 스트림과 사용 방법이 동일하다.

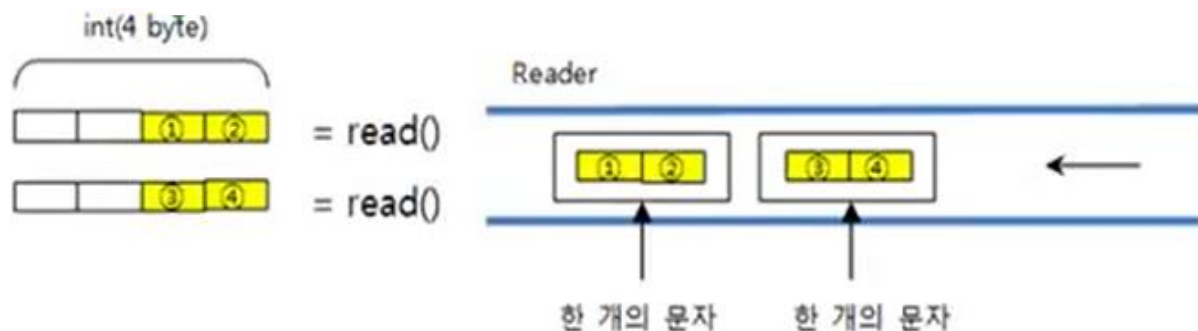
◎ 문자 읽기

- Reader는 문자 입력 스트림의 최상위 클래스로, 추상 클래스이다.
- 모든 문자 입력 스트림 클래스는 Reader 클래스를 상속받아서 만들어진다.
- Reader 클래스는 모든 문자 입력 스트림이 기본적으로 가져야 할 메소드가 정의되어 있다.



메소드		설명
int	read()	1개의 문자를 읽고 리턴
int	read(char[] cbuf)	읽은 문자들을 매개값으로 주어진 문자 배열에 저장하고 읽은 문자 수를 리턴
void	close()	입력 스트림을 닫고, 사용 메모리 해제

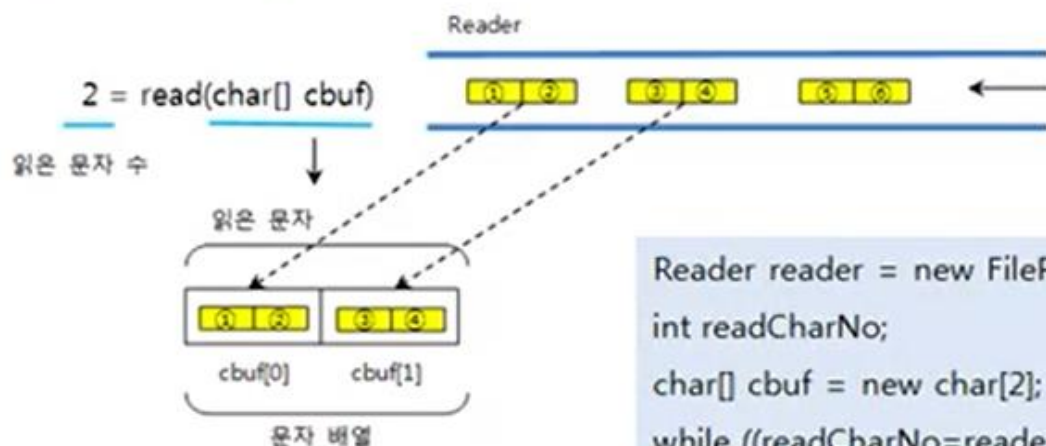
➤ read() 메소드



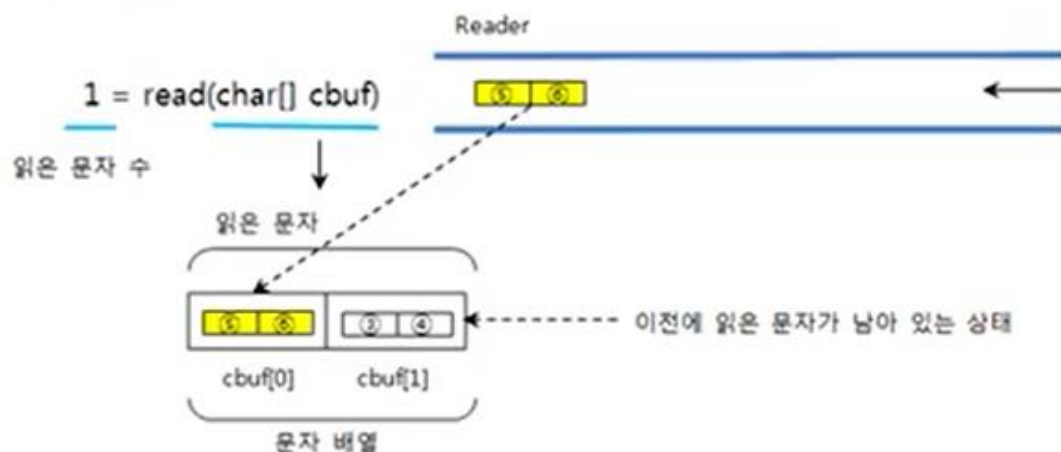
```
Reader reader = new FileReader("C:/test.txt");
int readData;
while ((readData=reader.read()) != -1) {
    char charData = (char) readData;
}
```

➤ read(char[] cbuf) 메소드

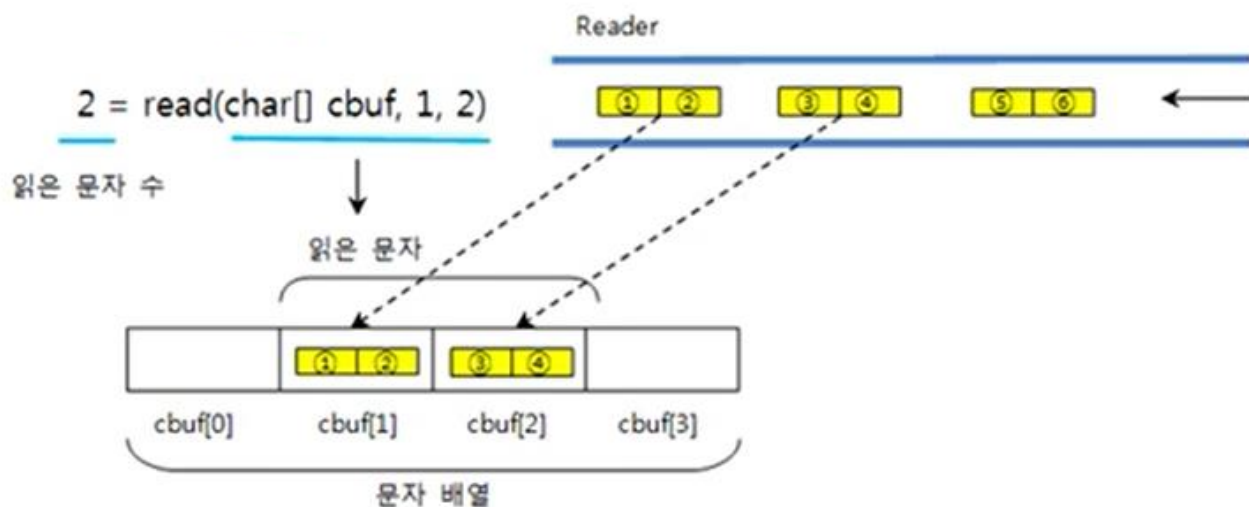
첫번째 읽을 경우



두번째 읽을 경우



➤ `read(char[] cbuf, int off, int len)` 메소드



```
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);
```

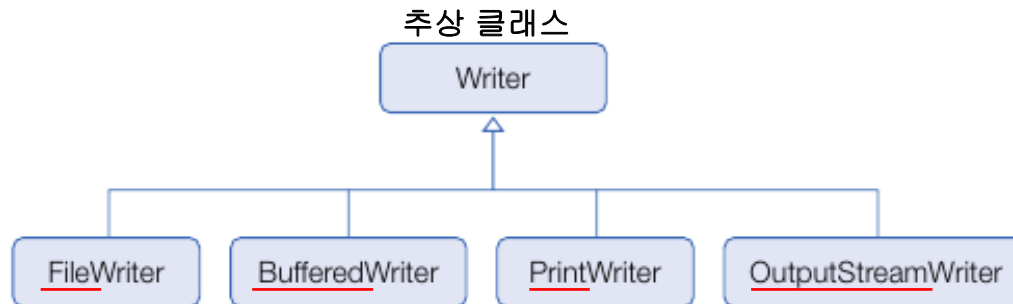
➤ `close()` 메소드

- Reader를 더 이상 사용하지 않을 경우 호출해 준다.
- Reader에서 사용했던 시스템 자원을 풀어준다.

```
reader.close();
```

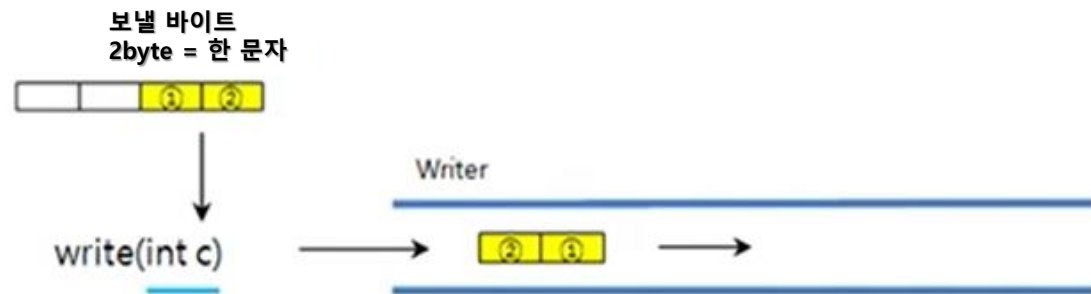
◎ 문자 출력

- Writer는 문자 출력 스트림의 최상위 클래스로, 추상 클래스이다.
- 모든 문자 출력 스트림 클래스는 Writer 클래스를 상속받아서 만들어진다.
- Write 클래스는 모든 문자 출력 스트림이 기본적으로 가져야 할 메소드가 정의되어 있다.



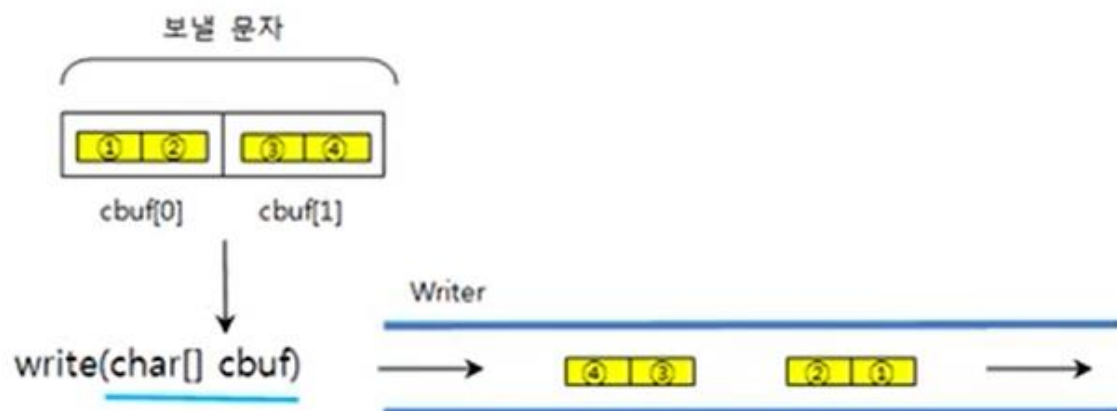
리턴 타입	메소드	설명
void	write(int c)	매개값으로 주어진 한 문자를 출력
void	write(char[] cbuf)	매개값으로 주어진 배열의 모든 문자를 출력
void	write(char[] cbuf, int off, int len)	매개값으로 주어진 배열에서 cbuf[off]부터 len개까지의 문자를 출력
void	write(String str)	매개값으로 주어진 문자열을 출력
void	write(String str, int off, int len)	매개값으로 주어진 문자열에서 off 순번부터 len개까지의 문자를 출력
void	flush()	버퍼에 잔류하는 모든 문자를 출력
void	close()	출력 스트림을 닫고 사용 메모리를 해제

➤ write(int c) 메소드



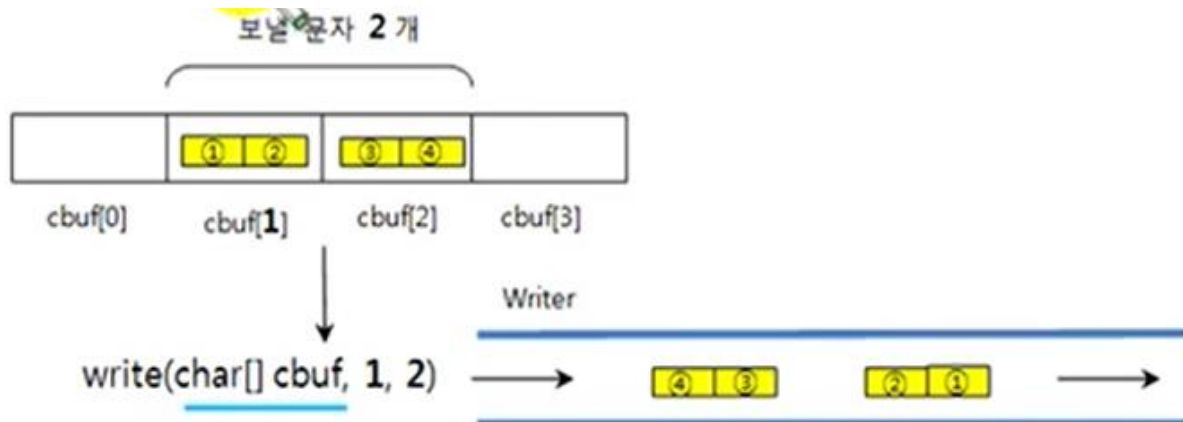
```
Writer writer = new FileWriter("C:/test.txt");  
char[] data = "홍길동".toCharArray();  
for(int i=0; i<data.length; i++) {  
    writer.write(data[i]); // "홍", "길", "동"을 하나씩 출력  
}
```


➤ **write(char[] cbuf)** 메소드



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

➤ `write(char[] cbuf, int off, int len)` 메소드



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

➤ `write(String str)` 와 `write(String str, int off, int len)` 메소드

