

# Java I/O Stream API

byte stream

character stream

O ↗  
↓ ↙

InputStream

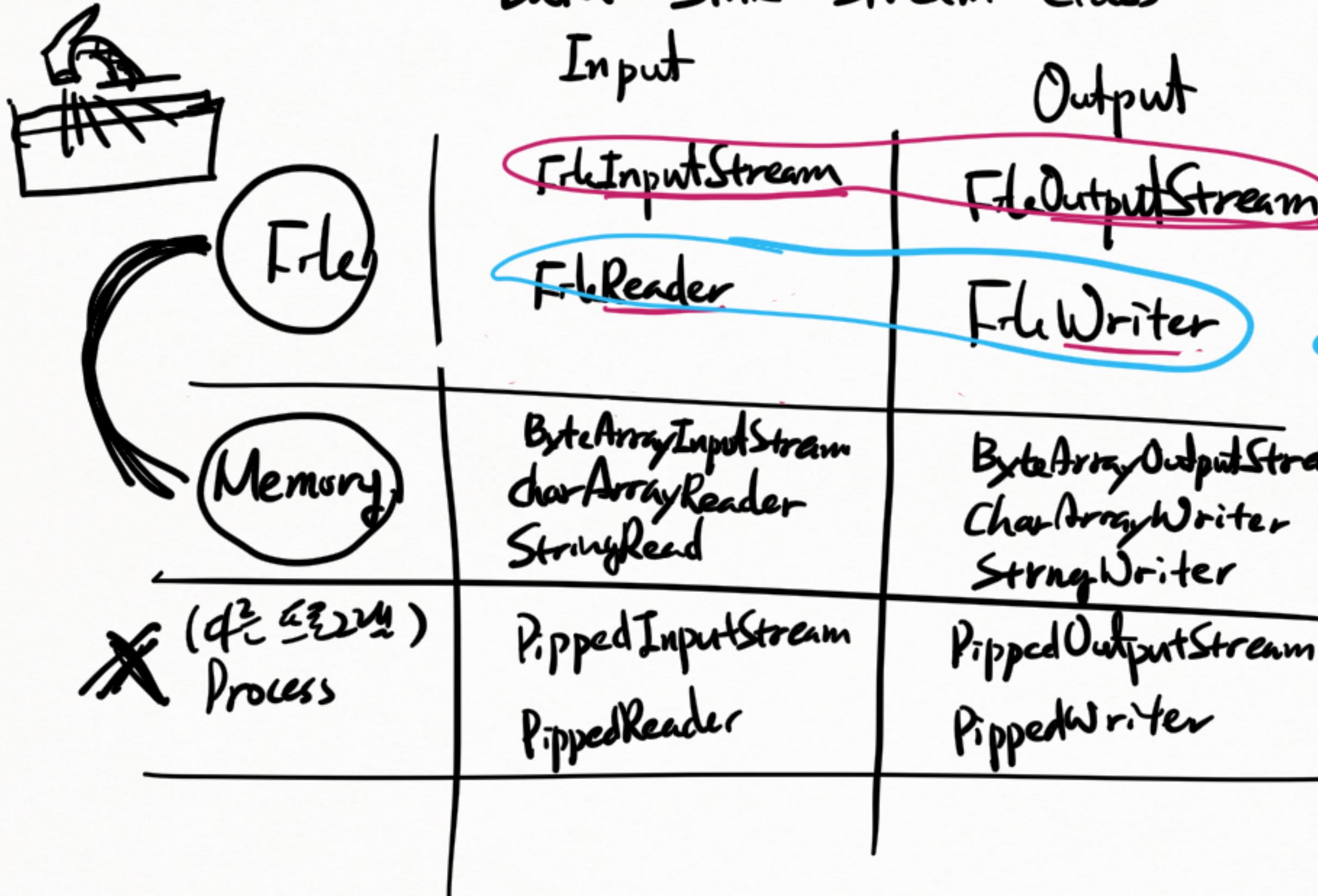
Reader

↗ I ↗  
↓ ↙

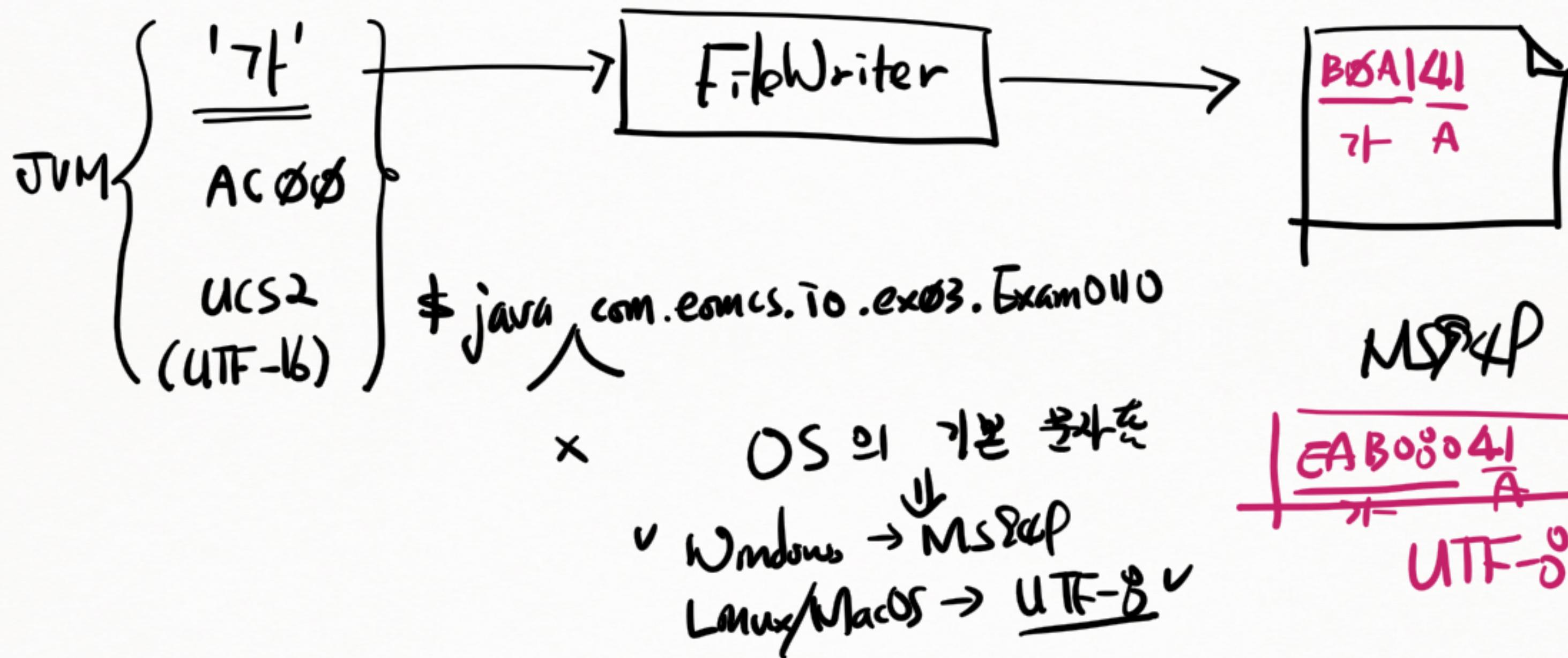
OutputStream

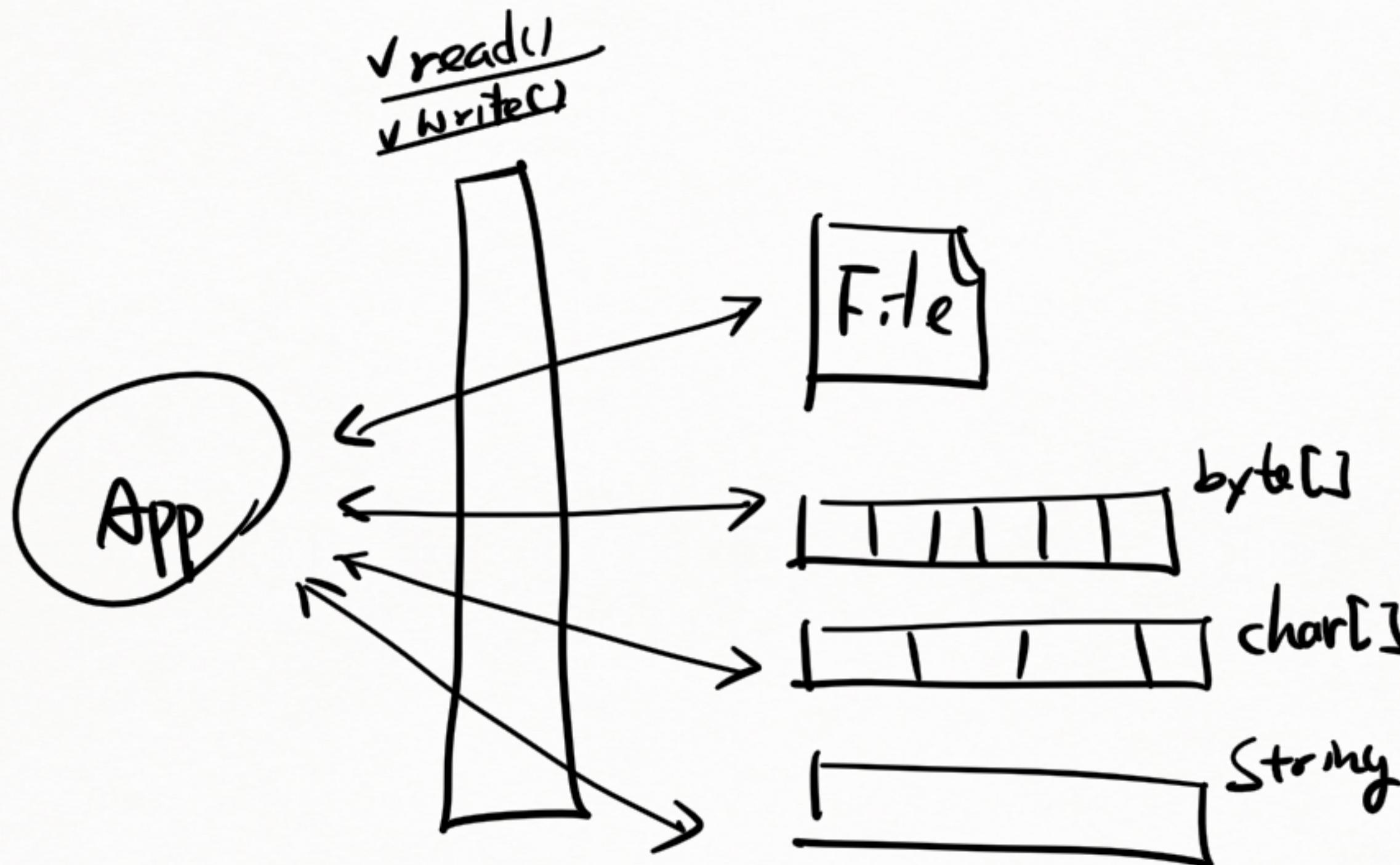
Writer

# Data Sink Stream Class



ex) .grf - jpg - ppt  
.avi - pdf  
ex) .csv .html .java  
.js .css .txt





MS949 : 41 42 B0 A1 B0 A2

UTF-8 : 41 42 EA B0 80 EA B0 81

UTF-16BE : 0041 0042 AC00 AC01

UTF-16LE : 4100 4200 00AC 01AC

JVM  
↓

char : UCS2 (UTF-16BE)

0041    0042    AC00    AC01  
A              B              𠂇              𠂇

new String( byte[] , offset , length , charset )

File/Network

✗ MS949  
✗ EUC-KR  
**UTF-8**



DBMS  
↓  
✗

JVM



UCS2

"

UTF-16BE

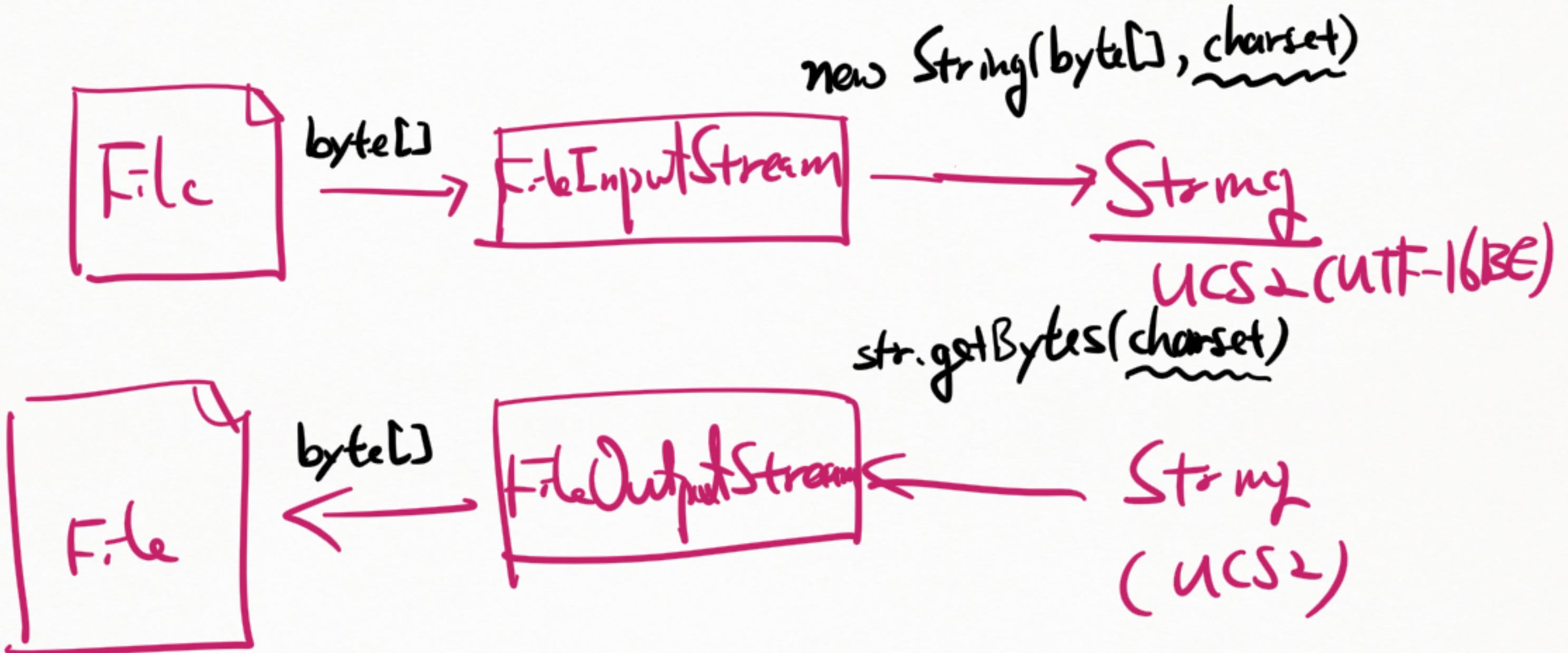
"

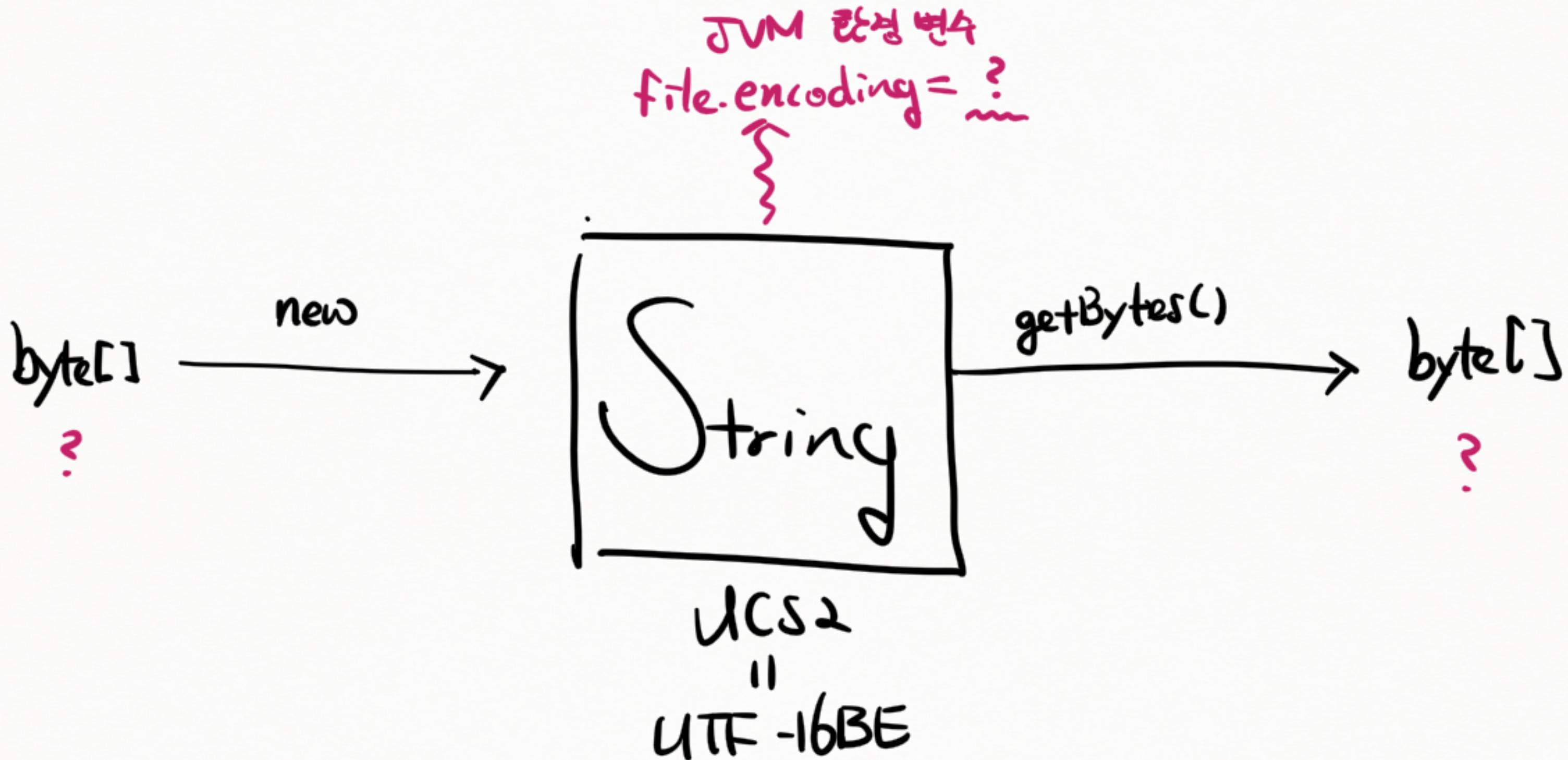
char c;  
~~~ 'A' → 0041  
'가' → Aced

File/Network

MSP430  
✗  
EUC-KR  
**UTF-8**







$$E \Rightarrow \underline{3017} - \underline{3030} \Rightarrow \text{값 } \boxed{?}$$

2×XX

중식

$34\text{XX} - 30\text{XX}$  합식

$$\begin{array}{r} 411X - 30XX - 30XX \quad 190 \text{ 만원} \\ 9999 + 30XX - 12XX - 30XX \quad 13 \end{array}$$

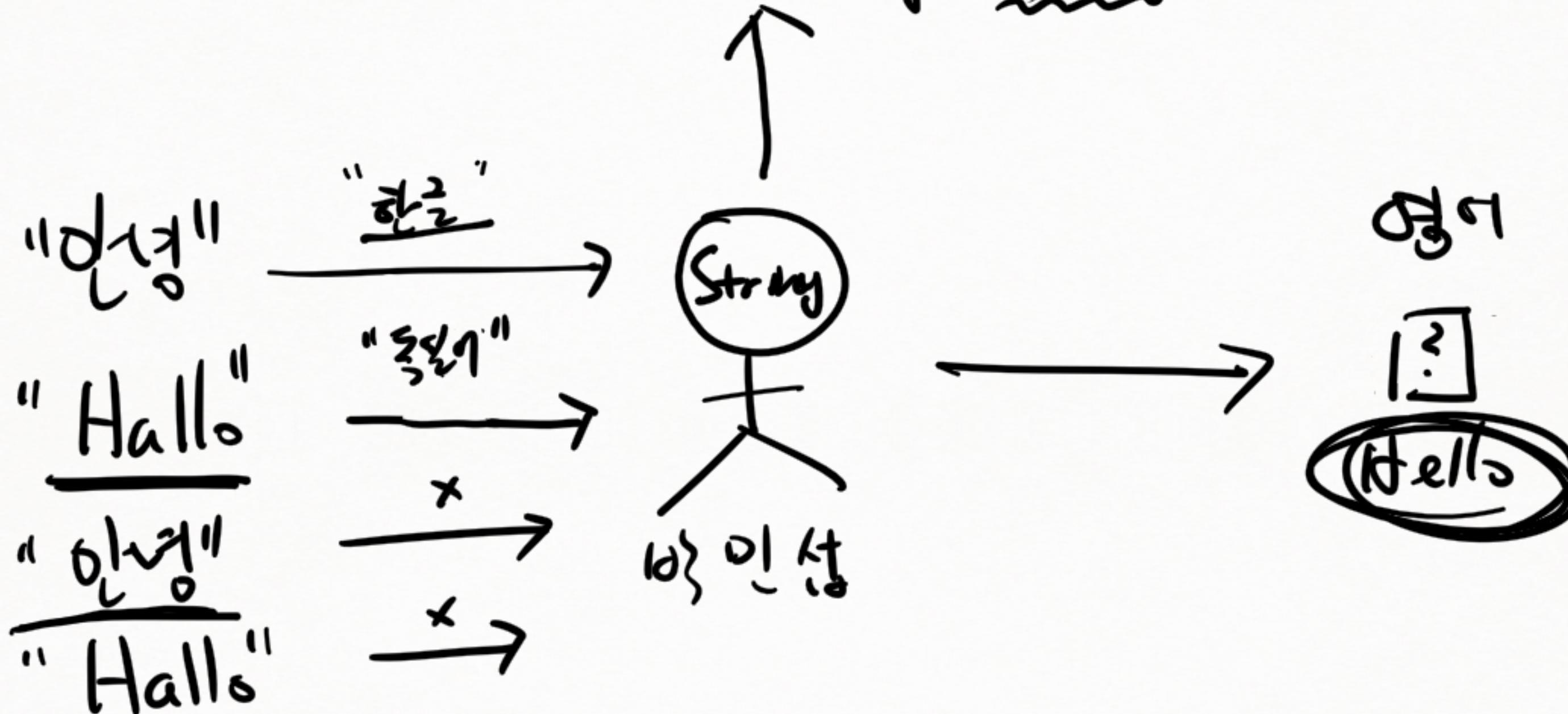
✓ A  $\Rightarrow \begin{array}{r} 2134 \\ 3418 - 3012 \\ \hline 12 \end{array}$

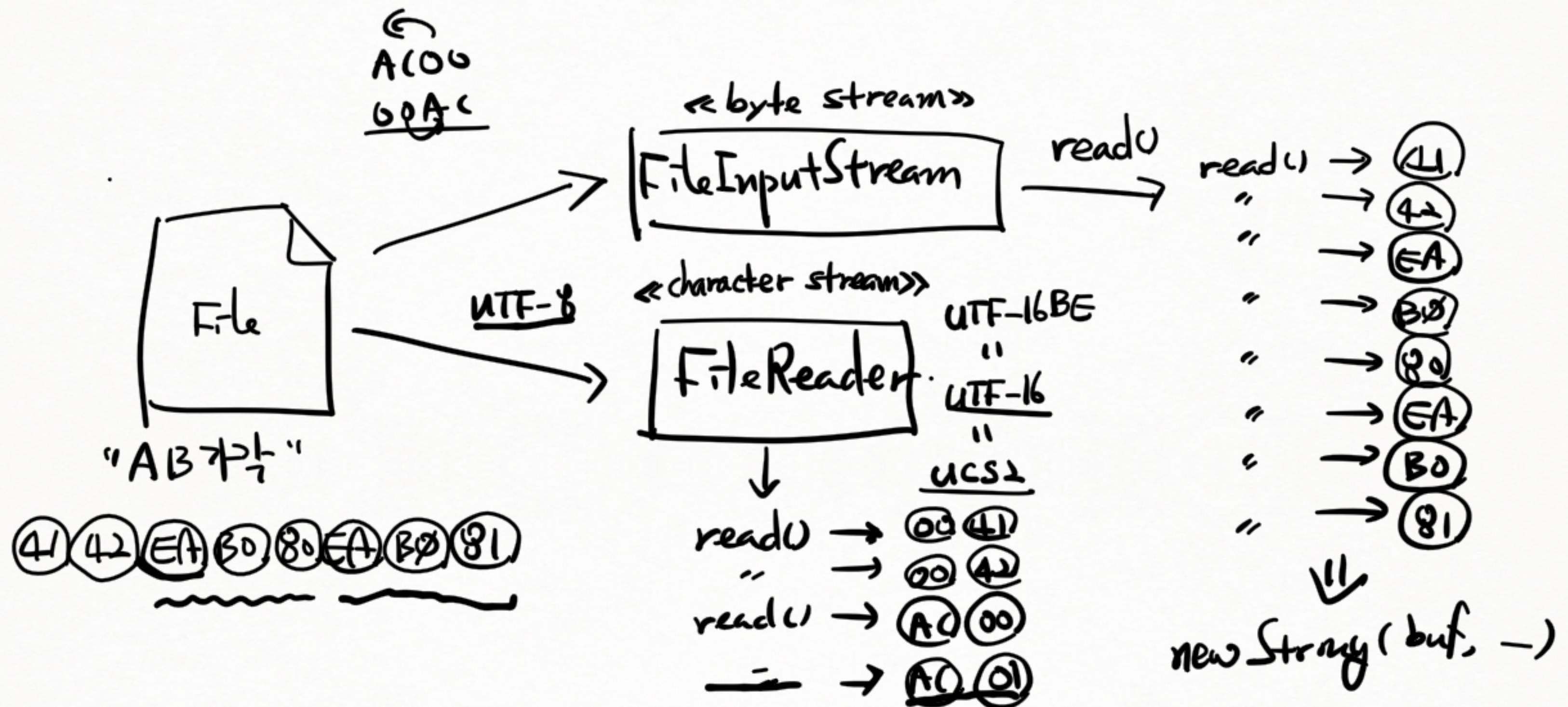
✓ B  $\Rightarrow \begin{array}{r} 3418 - 3011 \\ 4119 - 3011 \\ \hline 12 \end{array}$

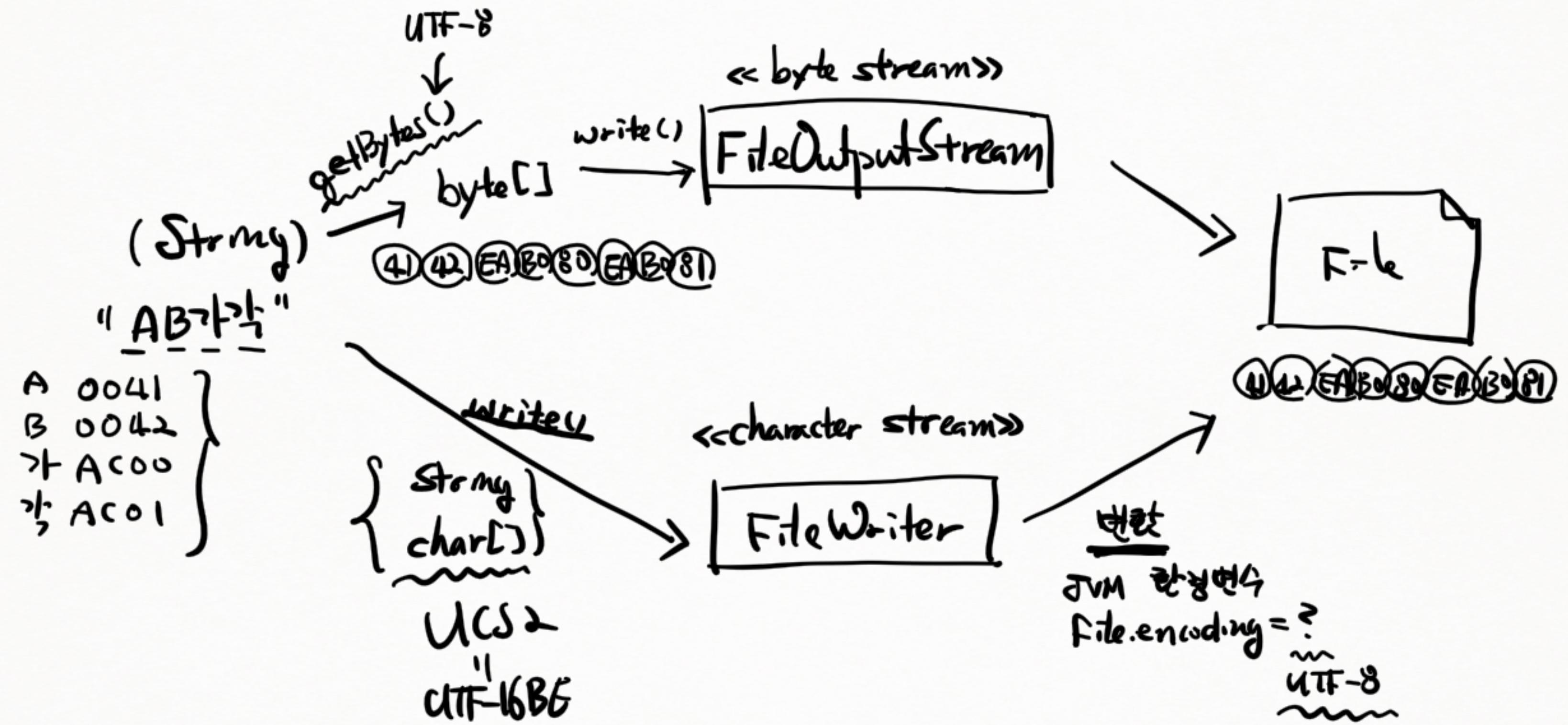
✓ C  $\Rightarrow \begin{array}{r} 4133 \\ 9999 - 3017 \\ \hline 12 \end{array}$

D  $\Rightarrow \begin{array}{r} 3083 \\ 9999 - 3017 \\ \hline 12 \end{array}$

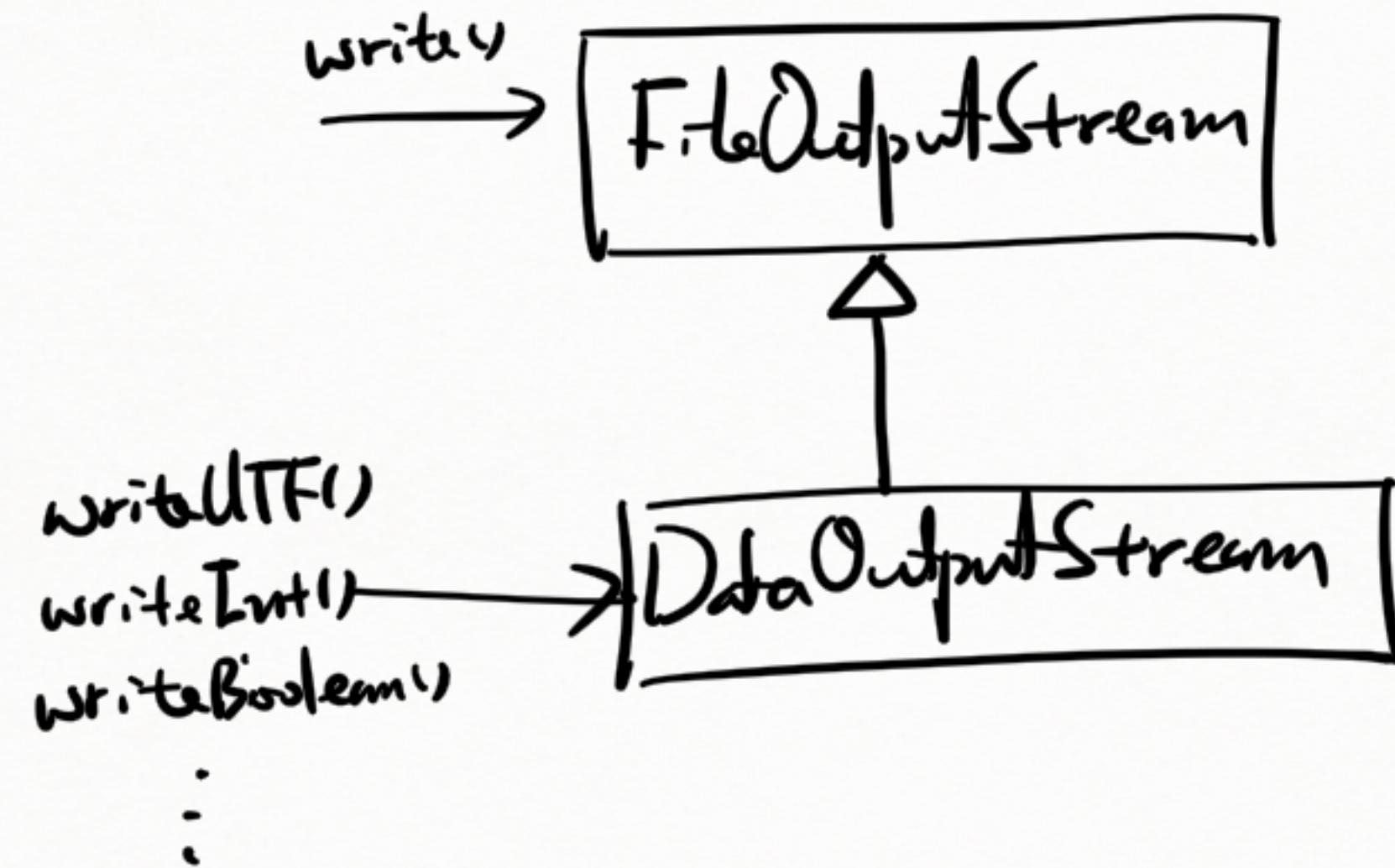
file.encoding = UTF-8

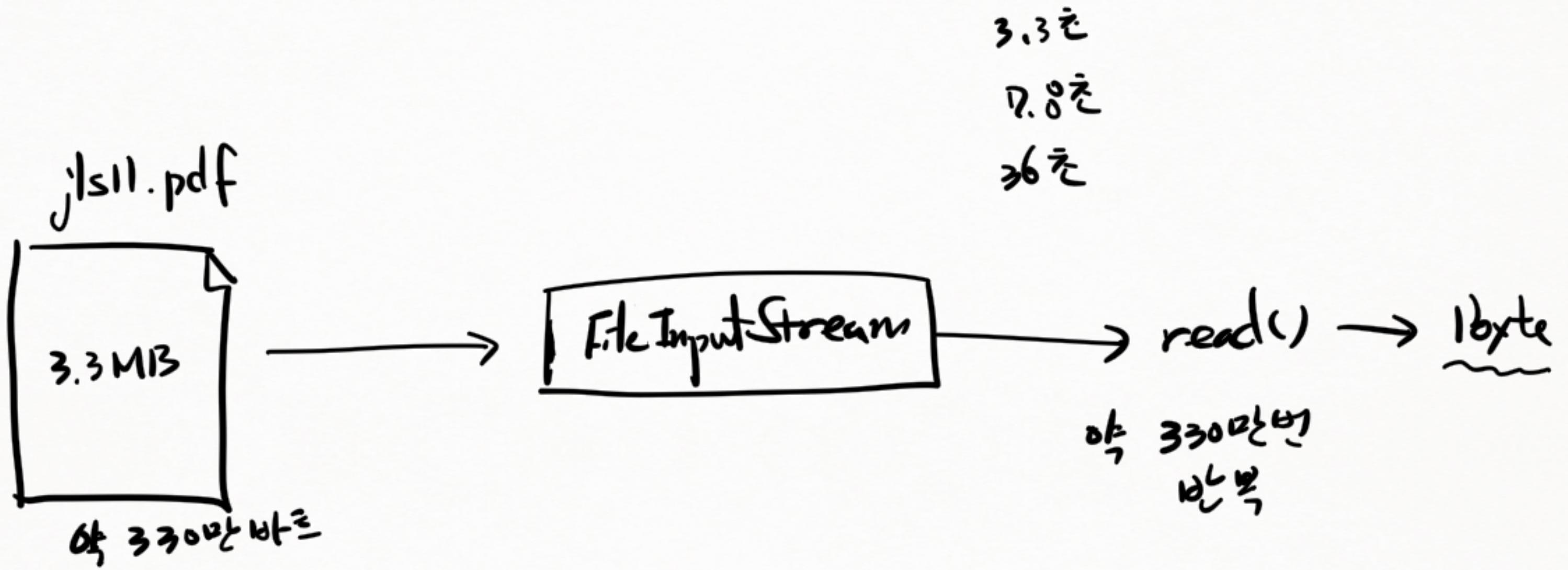


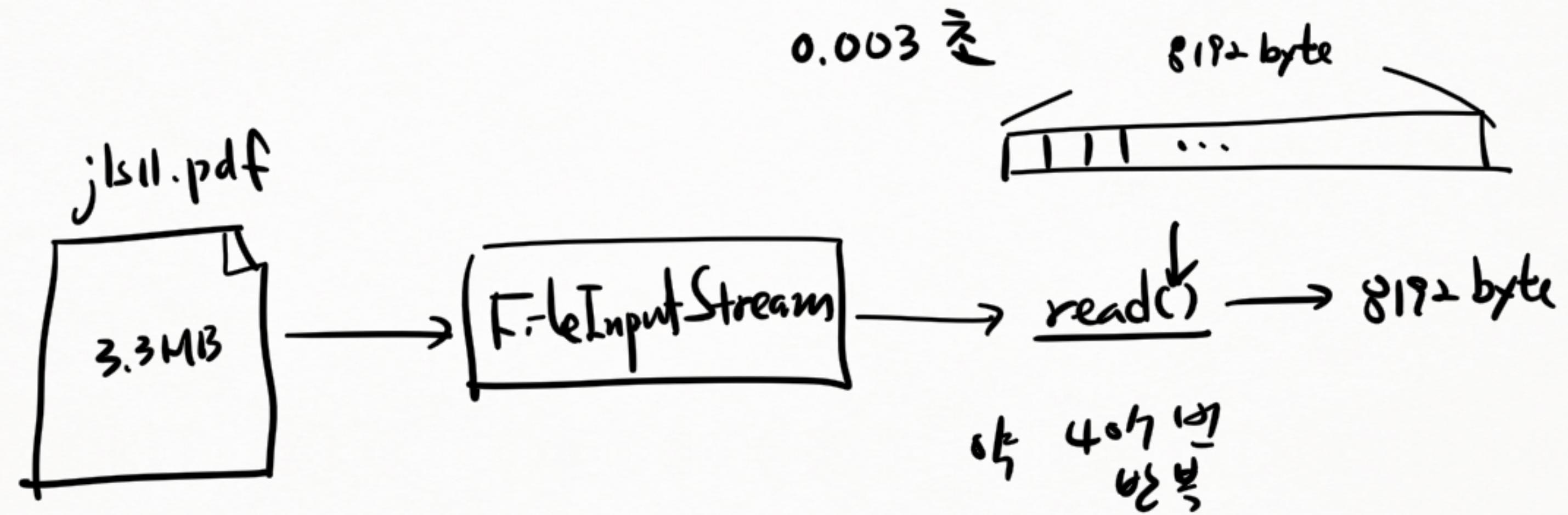


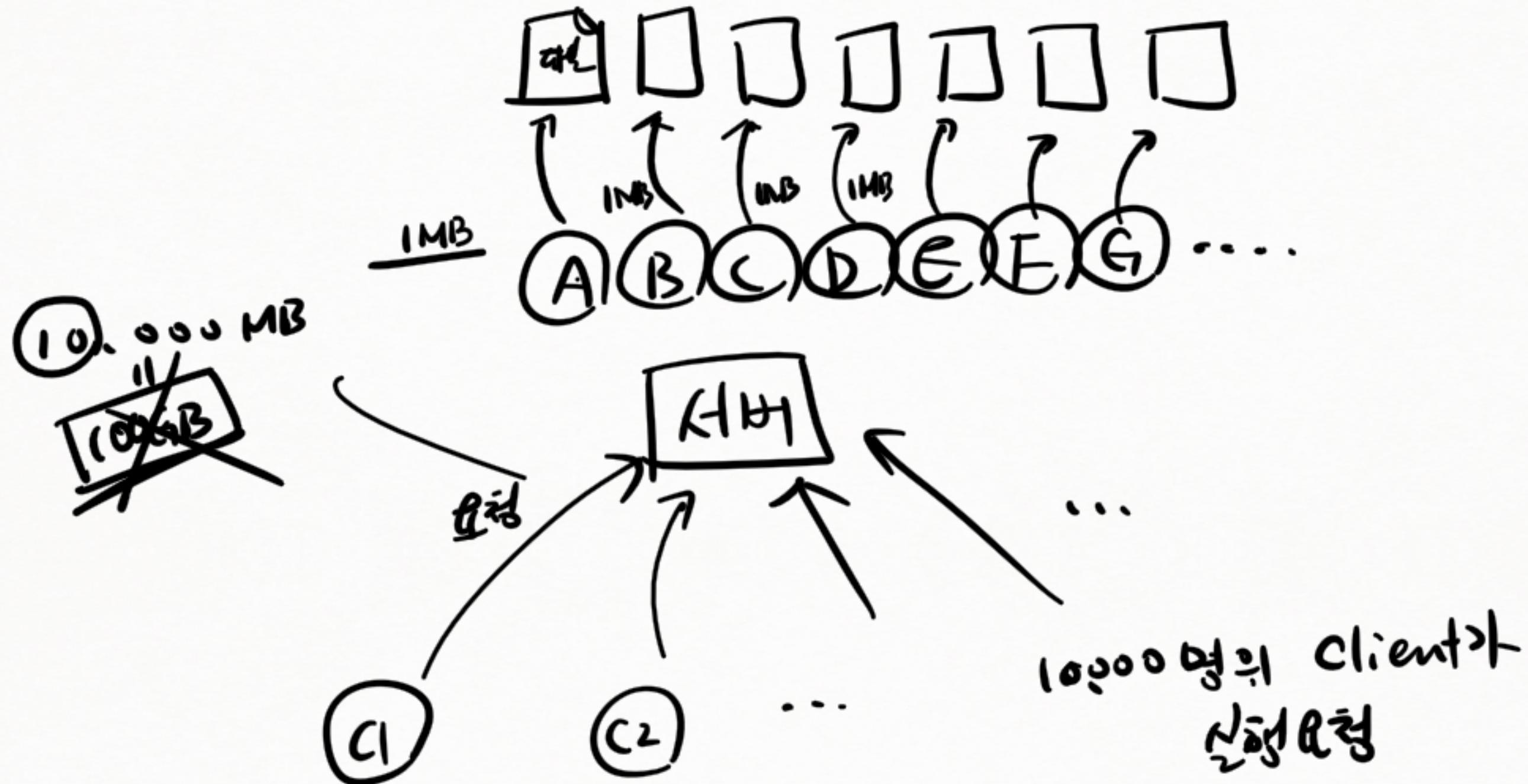


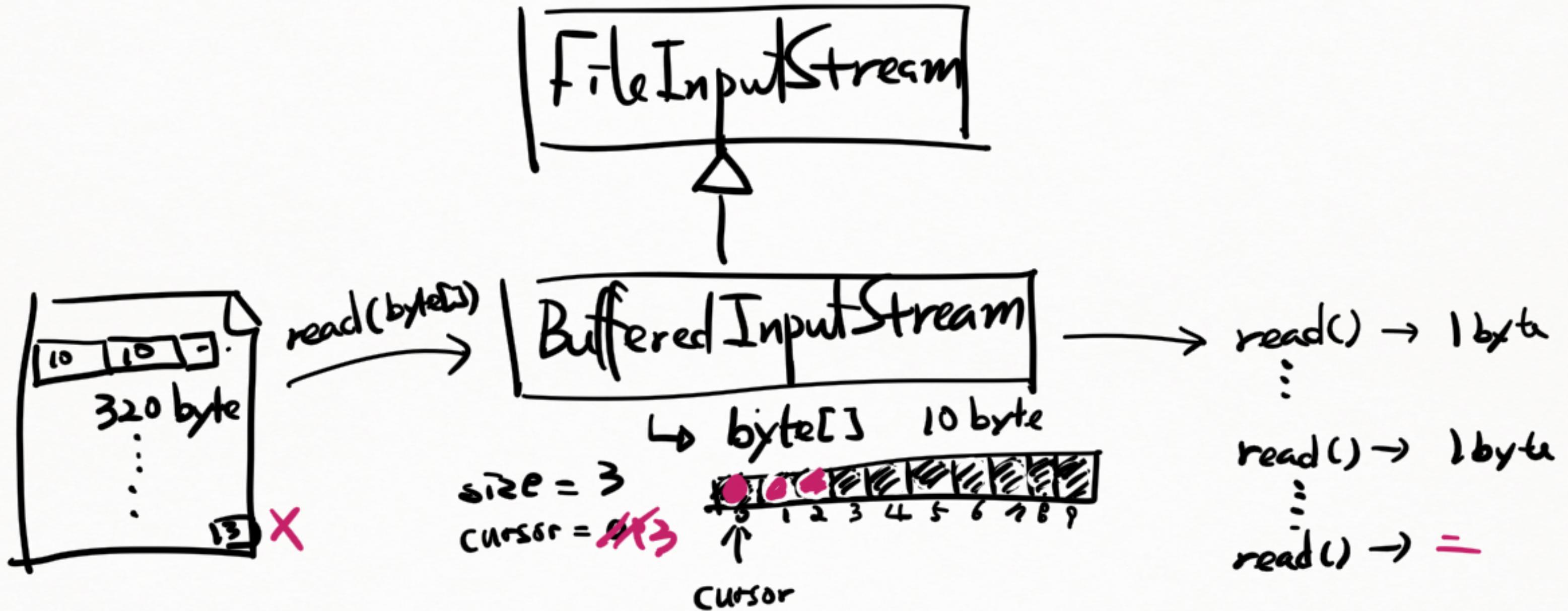


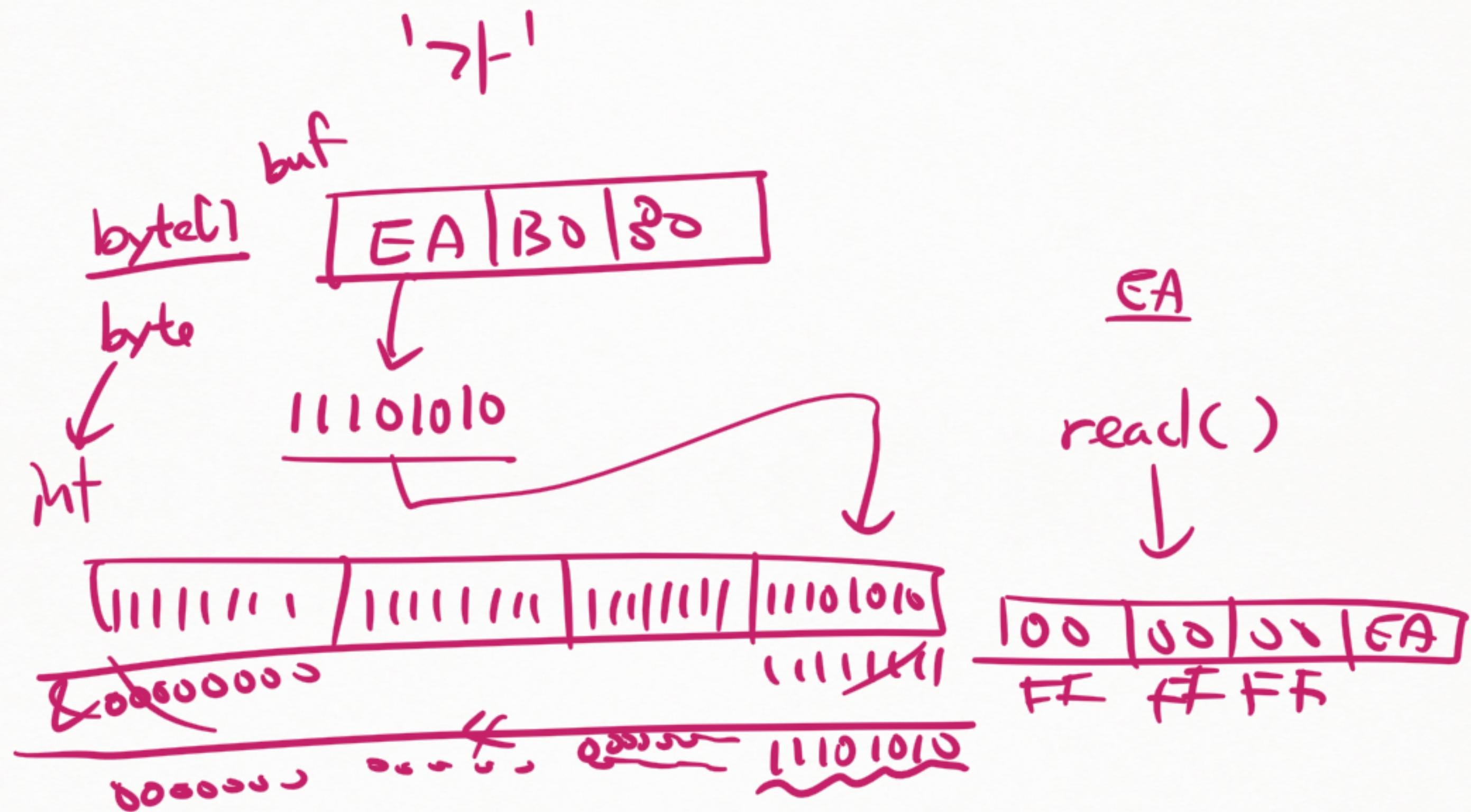


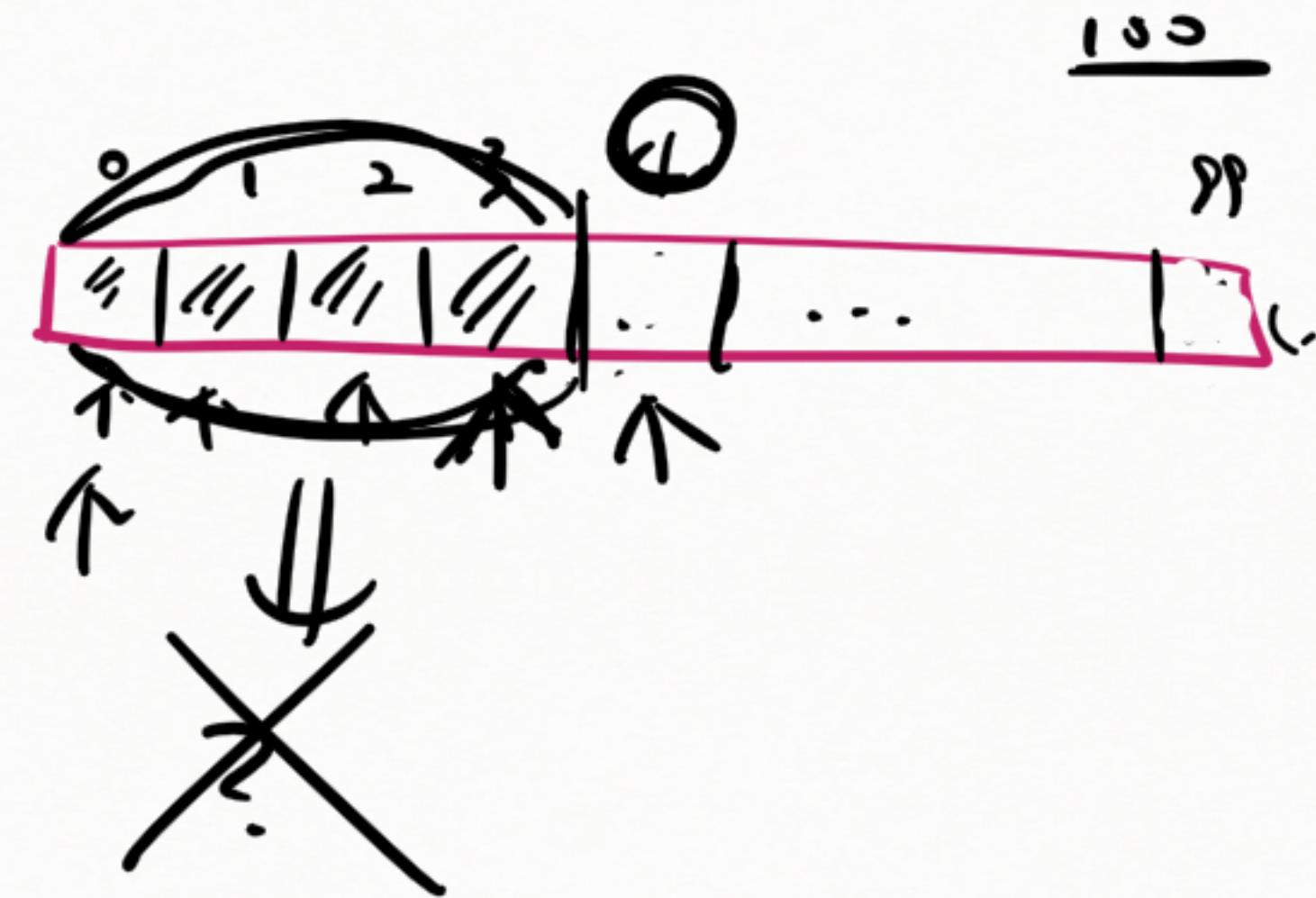


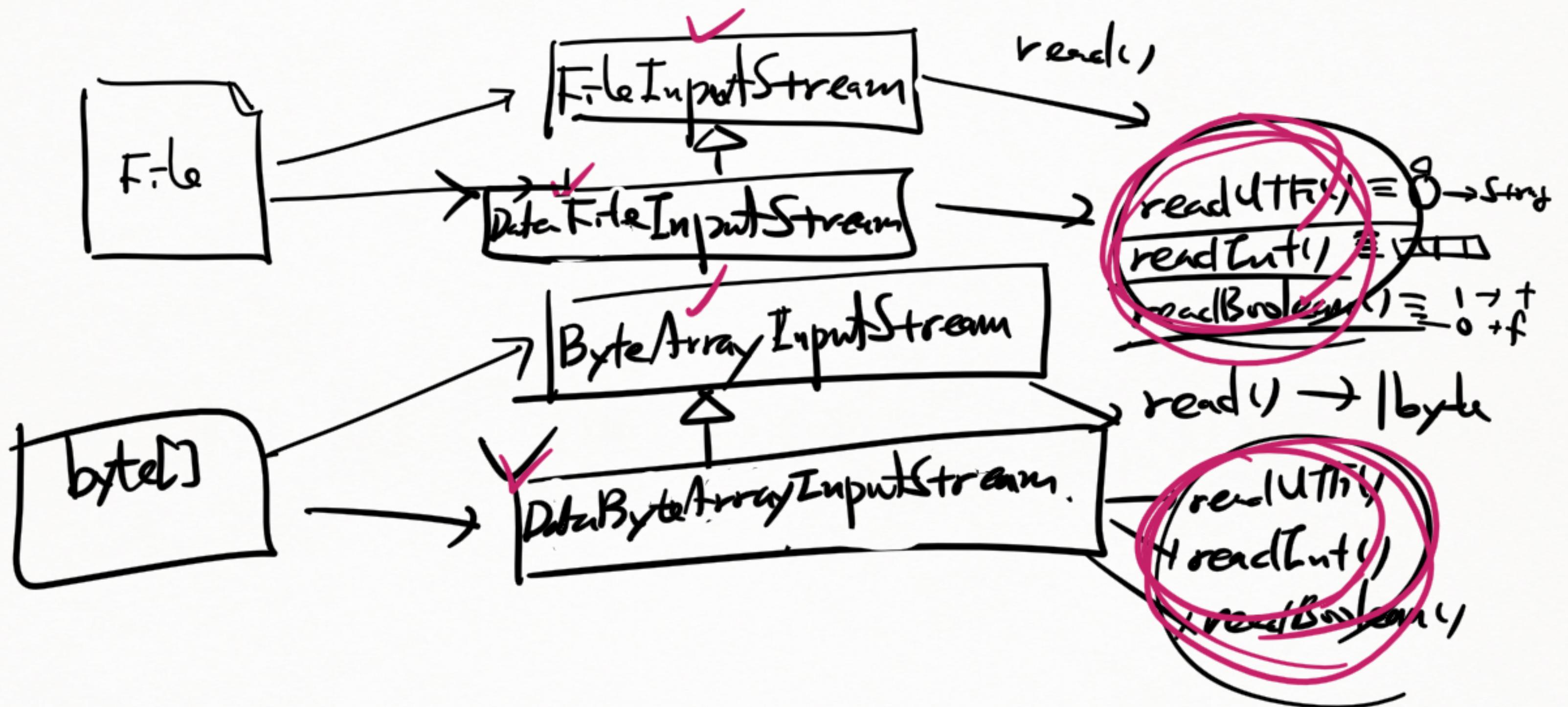


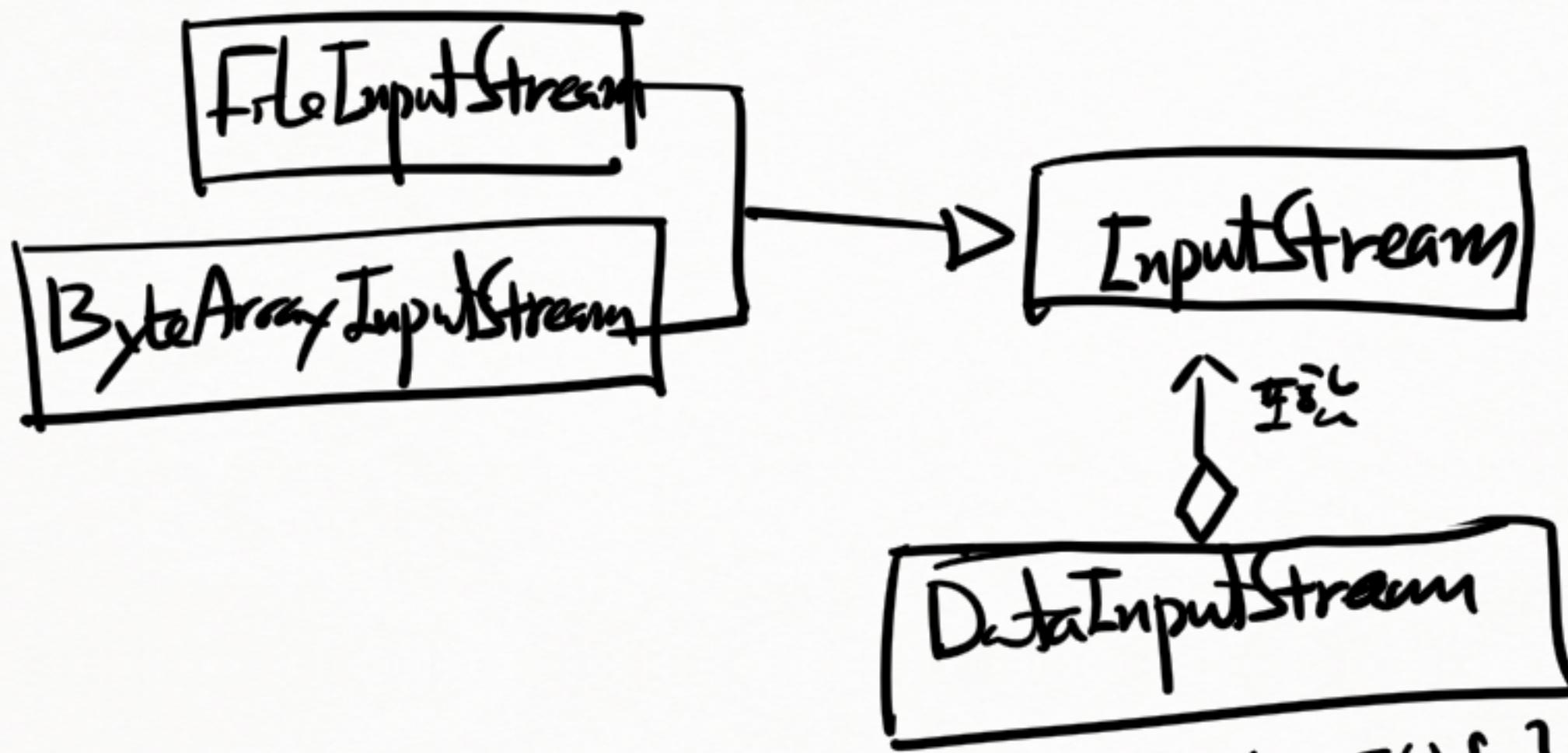






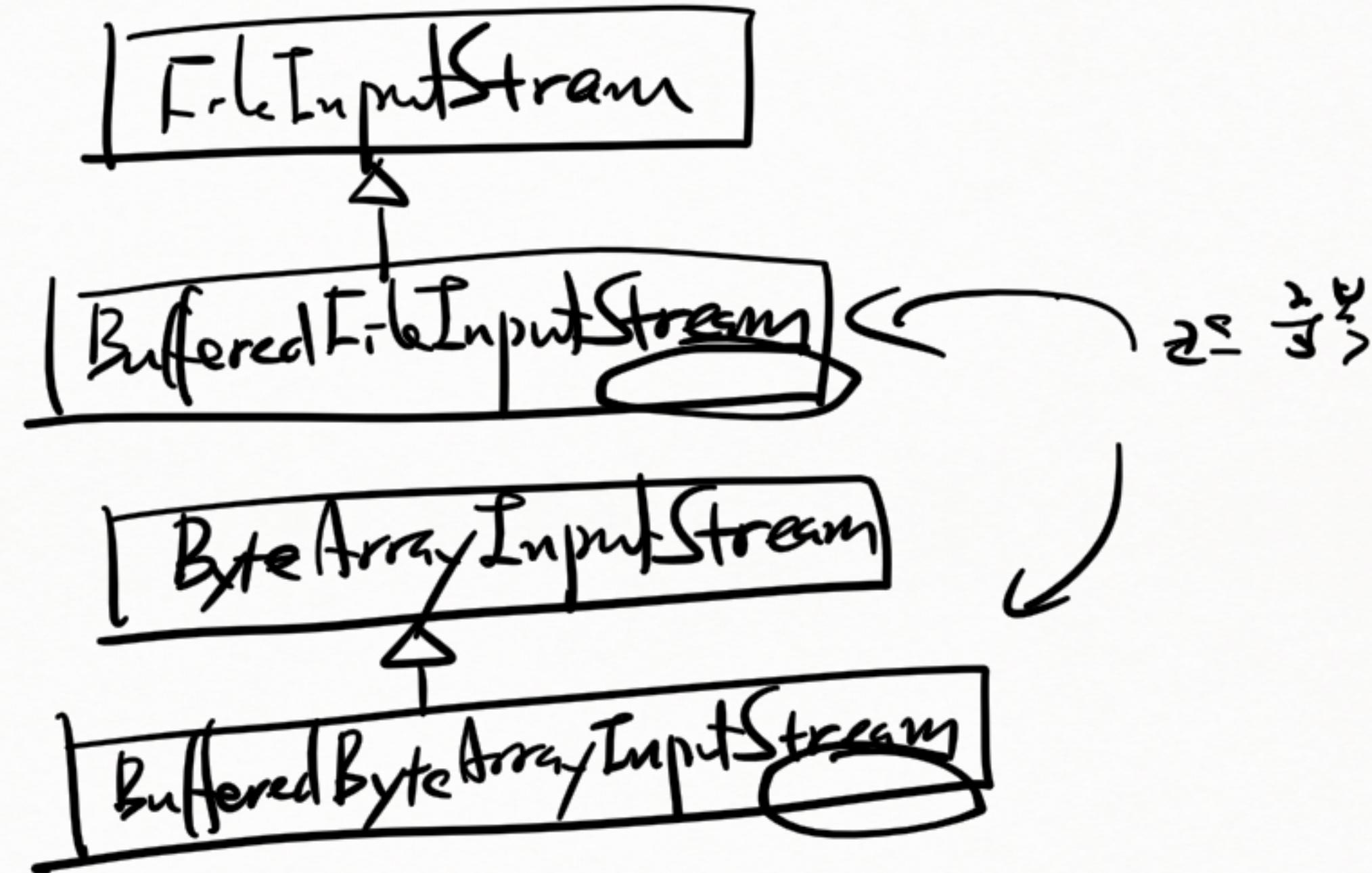


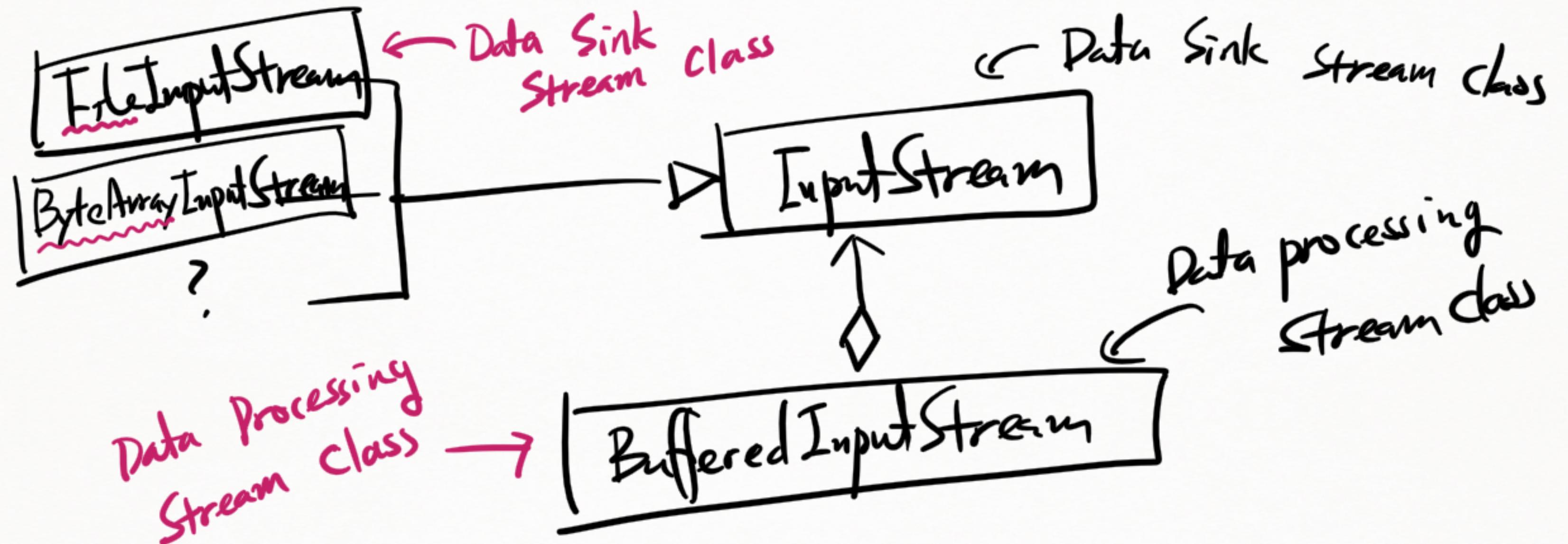




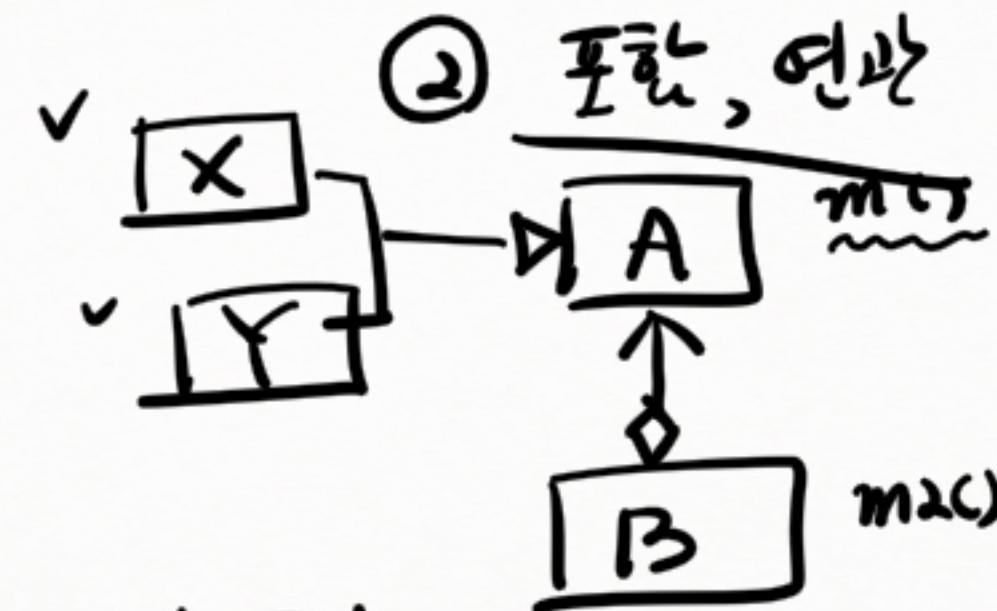
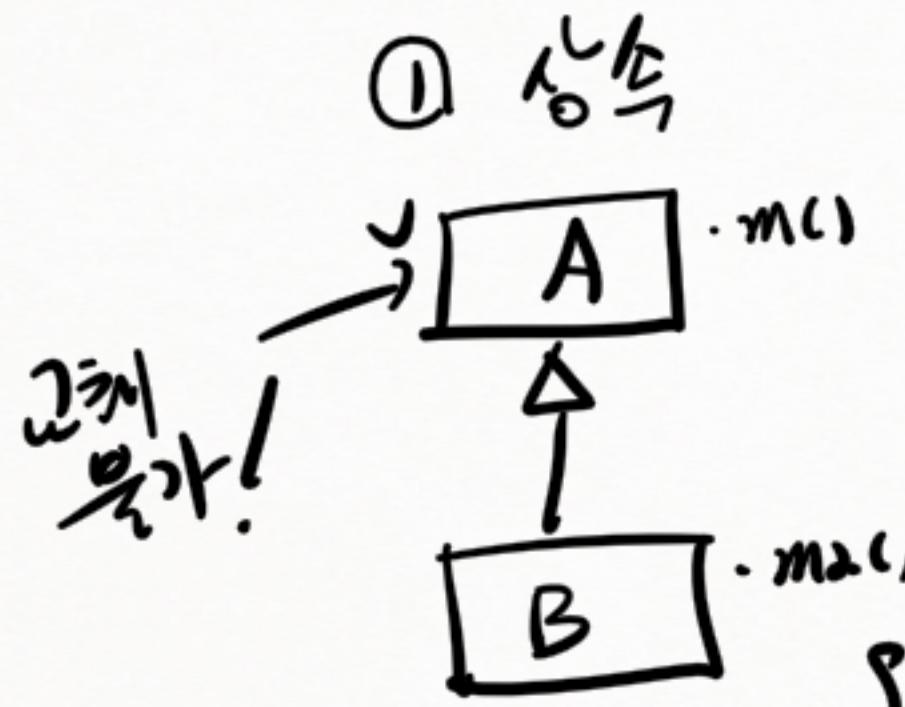
장식구 (Decorator)

- readUTF() {-}
- readInt() {-}
- readLong() {-}
- readBoolean() {-}

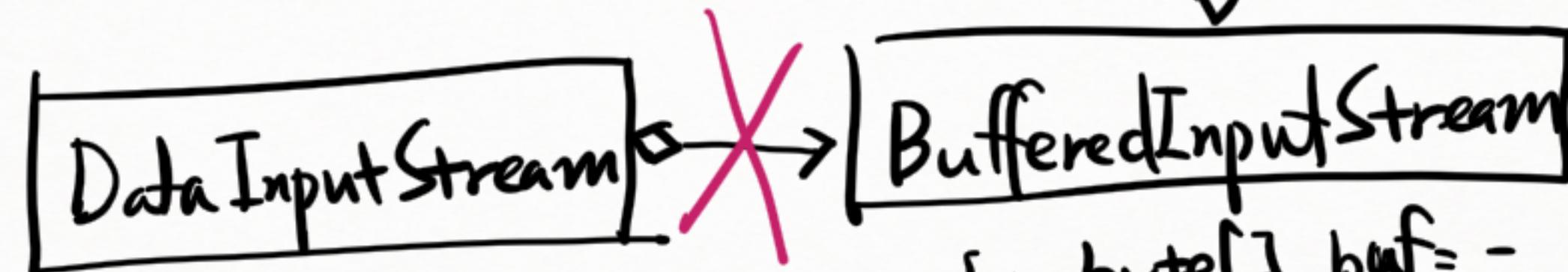




## 기능학장



- }    ✓ 중복 처리화  
      ✓ 재사용  
      ✓ ① 초기화-용이



(  
· 문자열  
· 자바 기본 타입  
· byte[]로  
· 쓰기  
· 읽기  
· readUTF()  
· readInt()  
· readLong()  
· readBoolean()

{  
· byte[] buf = -  
· int size  
· int cursor

Data Sink  
Stream class

레고블록

생성자에 다른 InputStream을  
받지 않는데  
"

"완성품 블록"  
예) 인형  
집  
마카  
상

FileInputStream  
ByteArrayInputStream  
...

"Decorator" 패턴

Component

InputStream/OutputStream  
Reader/Writer

FilterInputStream

Decorator

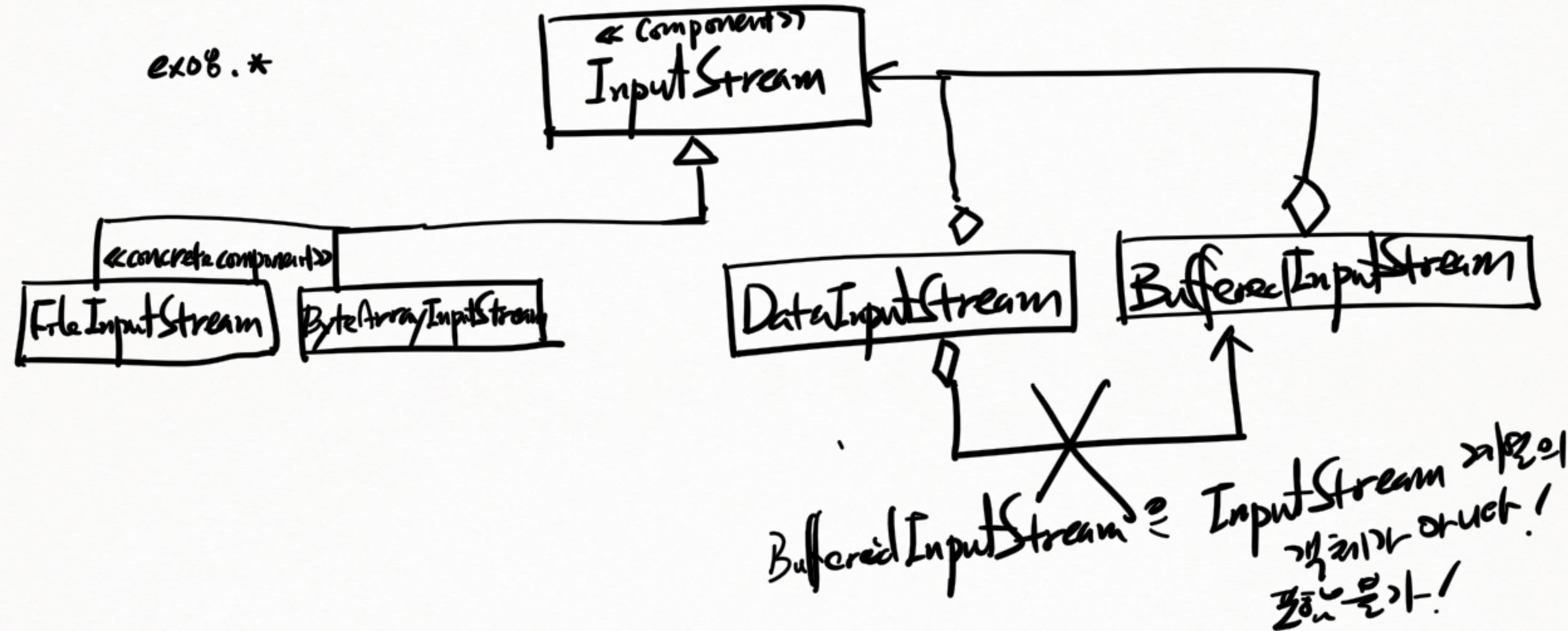
Data processing  
Stream class

중간블록 예) 짧은  
큰화  
나눌 블록

DataInputStream  
BufferedInputStream  
ObjectInputStream

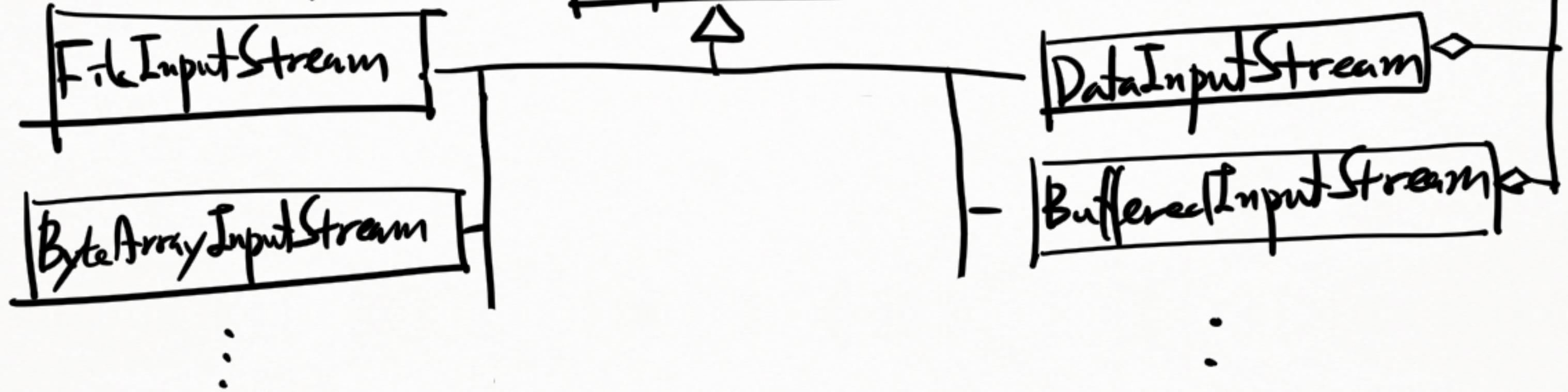
ConcreteDecorator

ex08.\*

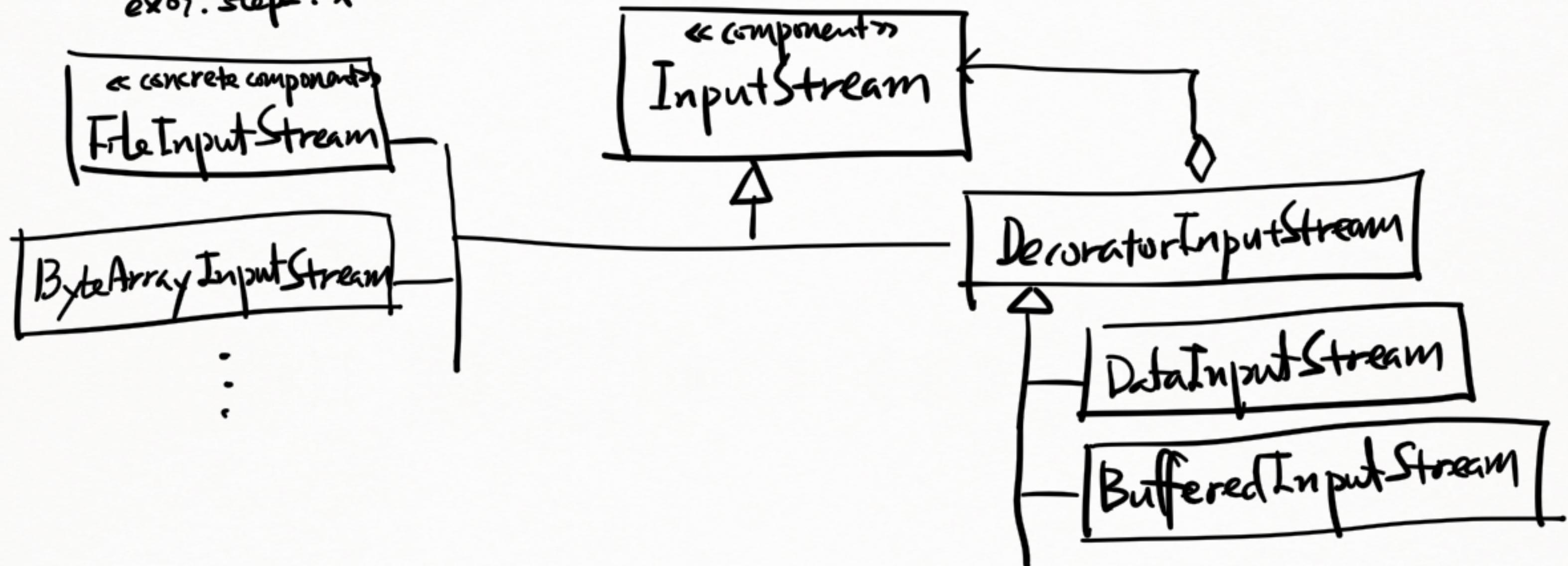


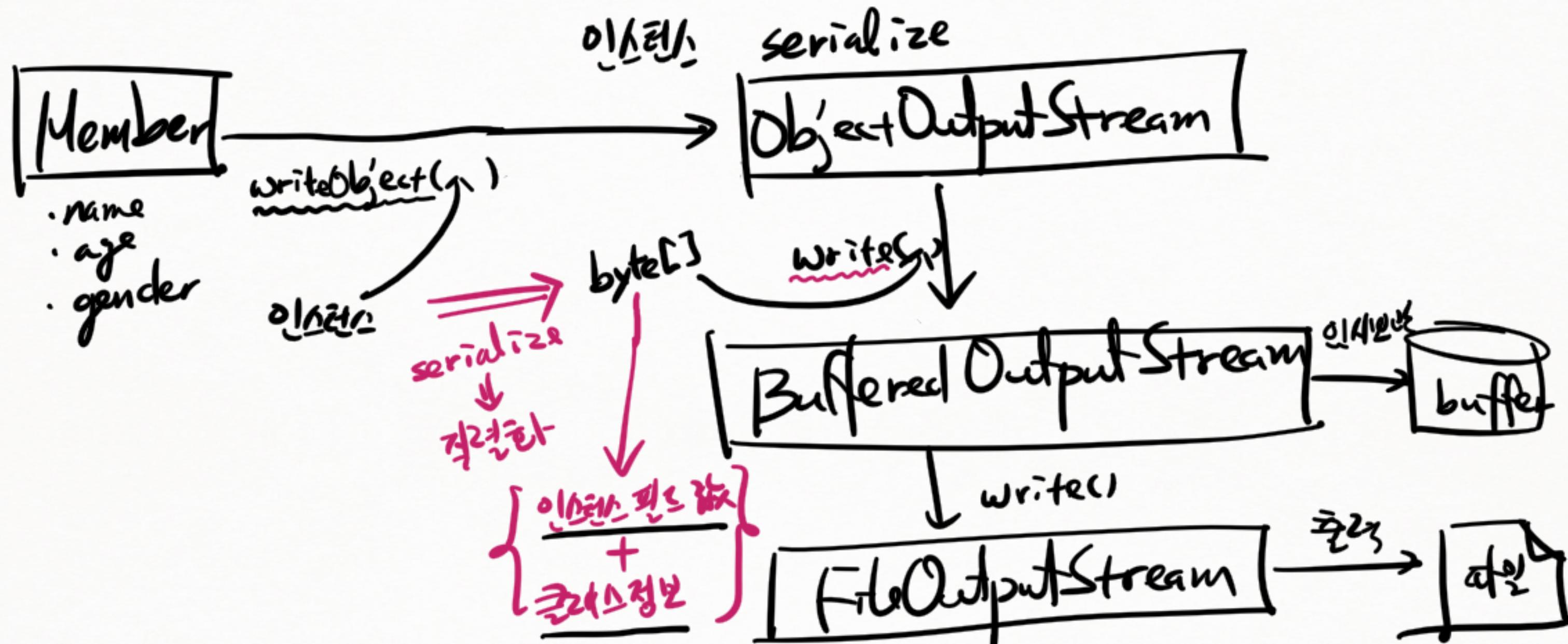
ex of step 1. \*

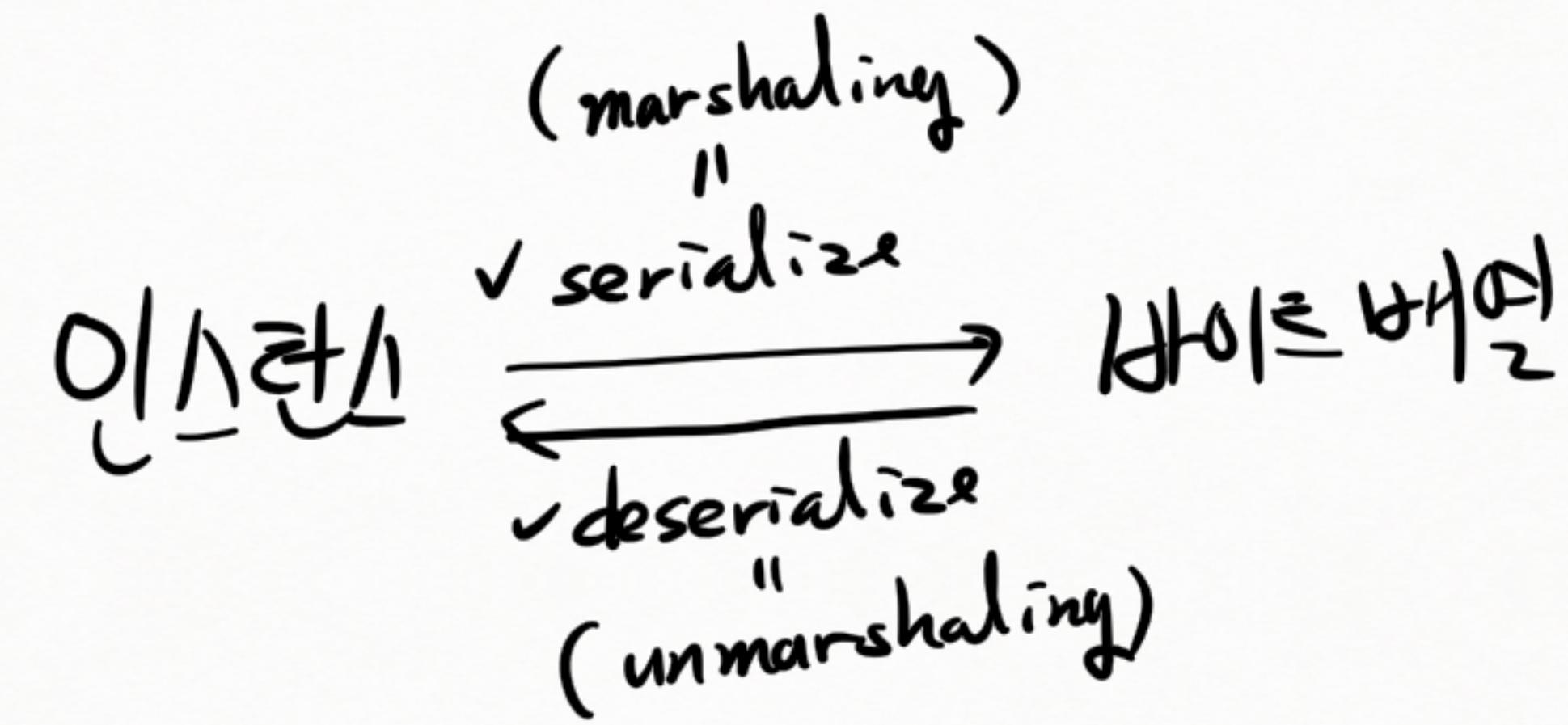
« concrete component »

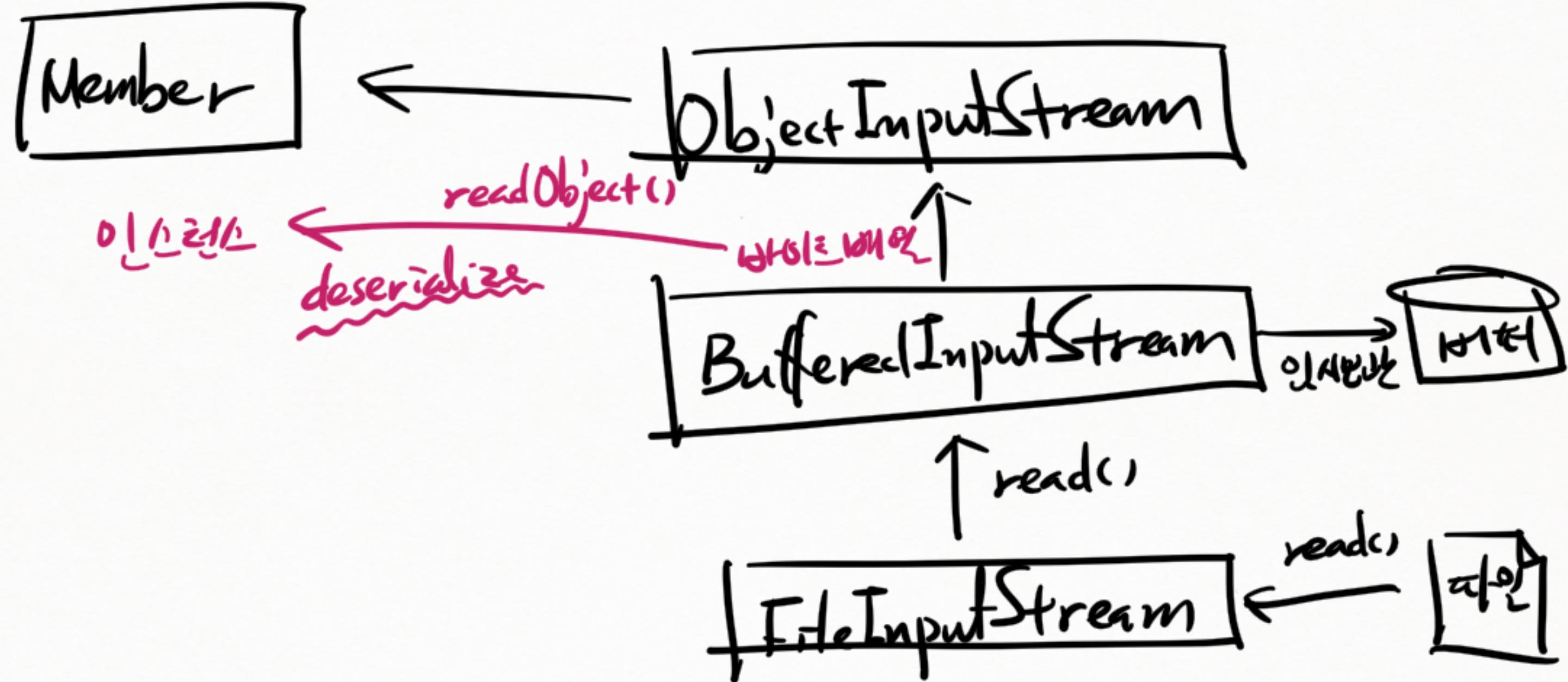


ex09. step2. \*





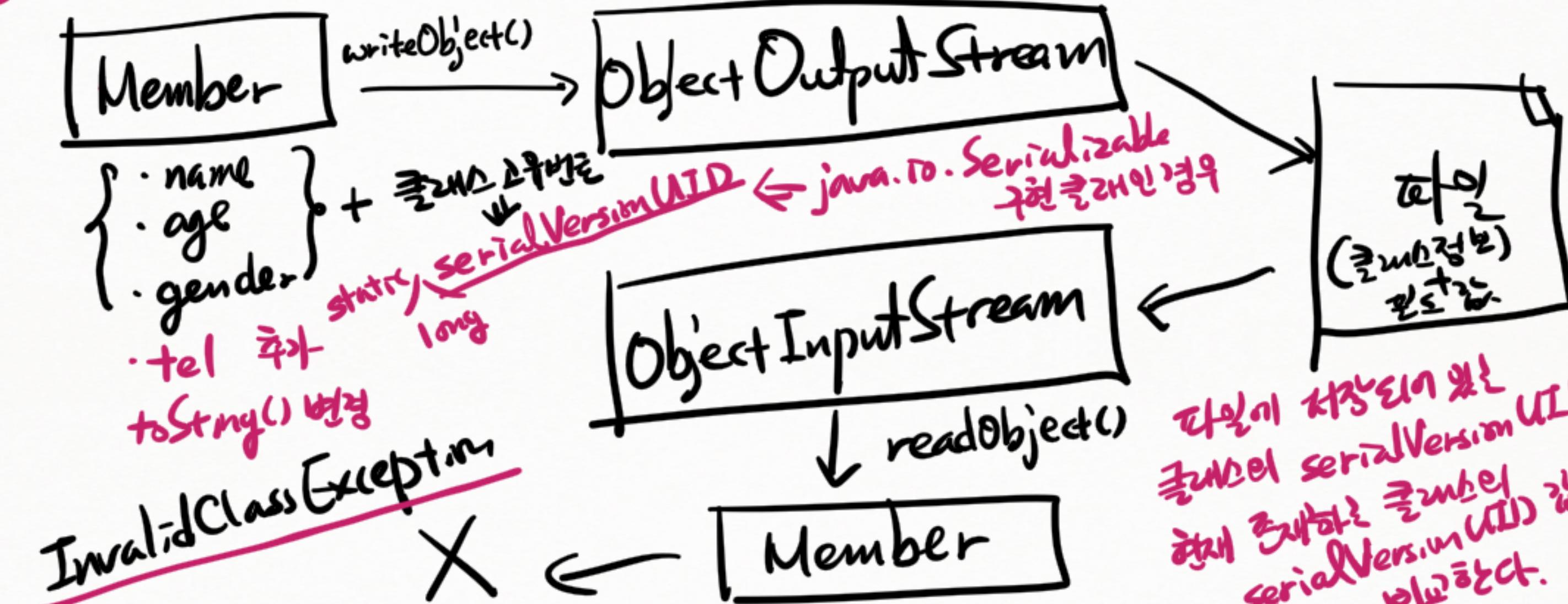




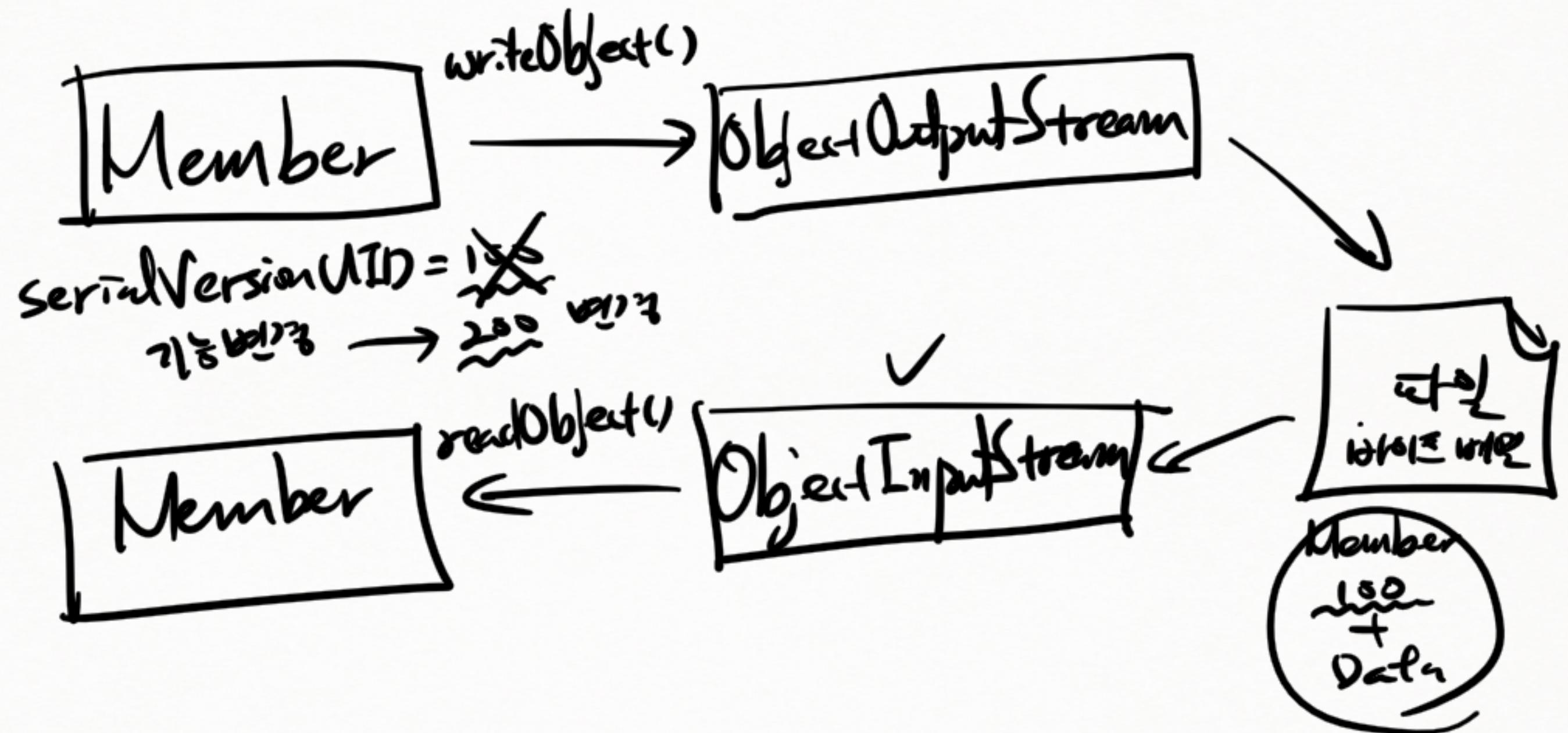
\* serialVersionUID  
값은 고정하지 않고 자동으로 추가된다.  
이유의 경우를 찾는다 → 클래스가 변경되었을  
때마다 값은 바뀐다.

① unique

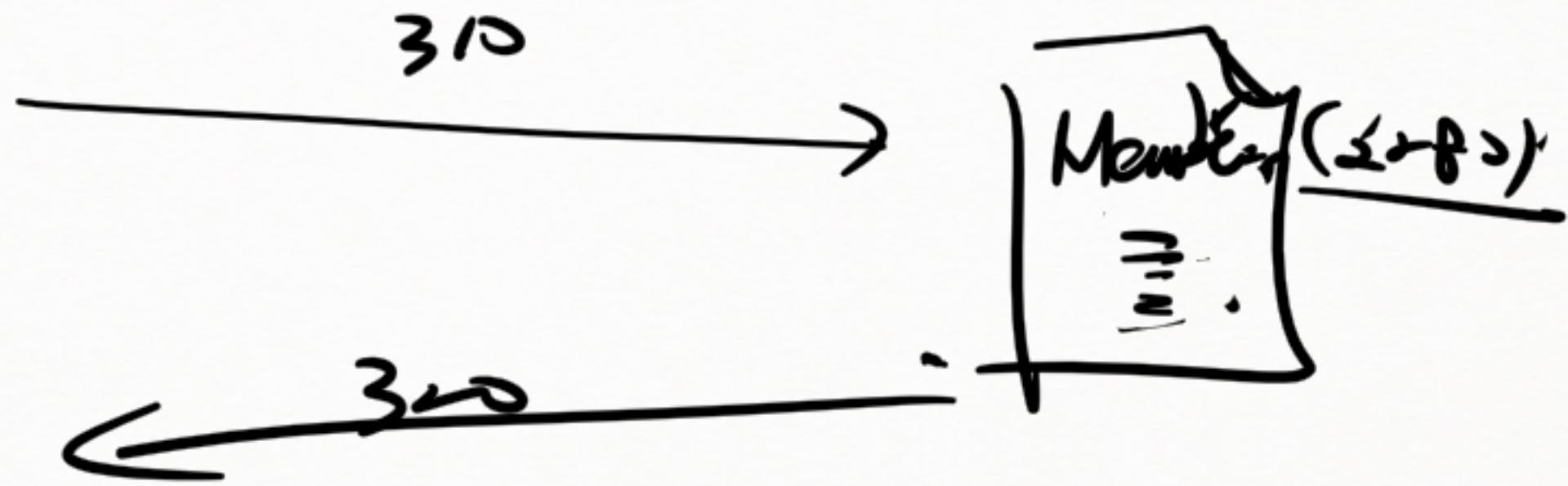
② IDentifiable



파일이 저장되어 있는  
클래스의 `serialVersionUID`와  
현재 클래스를 동일한  
`serialVersionUID` 값이  
같은지 비교한다.



~~1018~~ Member  
~~2283~~  
- name  
- age  
- gender  
- ~~test~~



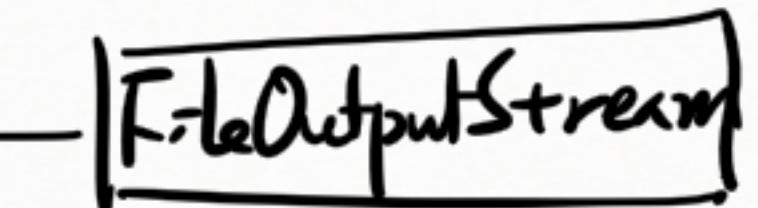
## Data Sink

### File Format

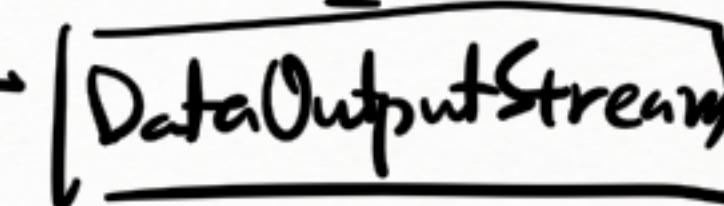
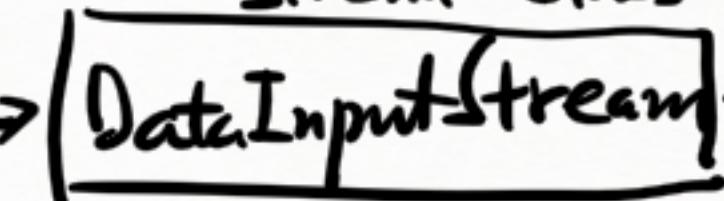
JPEG  
GIF  
WAV  
MP3  
PPT  
:



### Stream Class



### Data processing Stream Class



→ `readInt()`

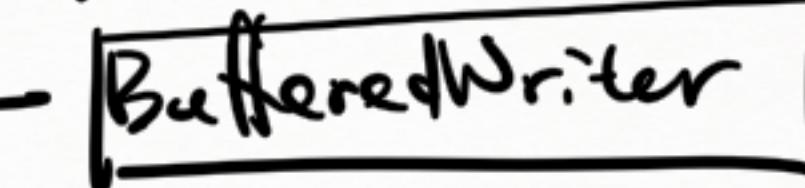
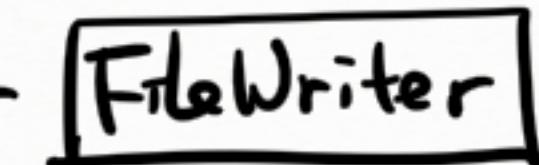
:

⋮

← `write()`

⋮

TXT  
JAVA  
CSS  
HTML  
MD  
CSV  
JSON  
⋮



⋮

- 30-a : FileInputStream / FileOutputStream
  - 30-b : DataInputStream / DataOutputStream
  - 30-c : BufferedInputStream / BufferedOutputStream
  - 30-d : ObjectInputStream / ObjectOutputStream
- } file format을 알아야  
알고는 쓰기 가능
- ← d는 인터프리터 안에서

30-a → 30-e : FileReader / FileWriter : CSV 파일

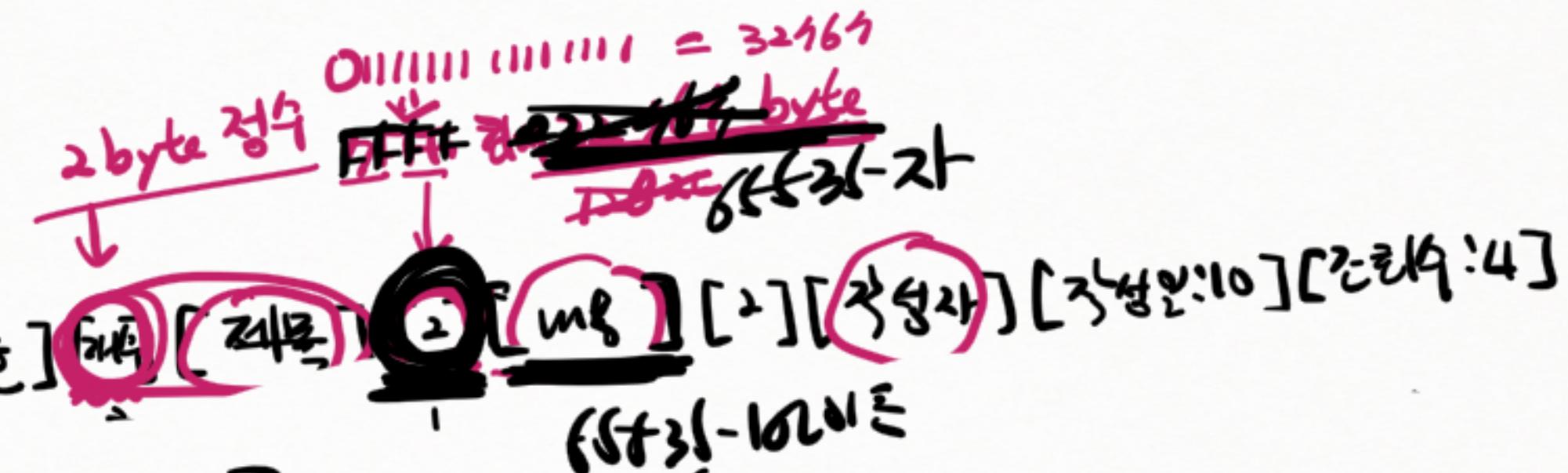
○ 30-f : BufferedReader / BufferedWriter : 입출력 속도 개선

30-b → 30-g : 2차원 배열 I

30-c → 30-h : " II

4 bytes

[길이수: 4]



int <sup>len</sup> a = 65535-;

write(a >> 8);  
00 00 FF |

write(a);  
00 00 FF

