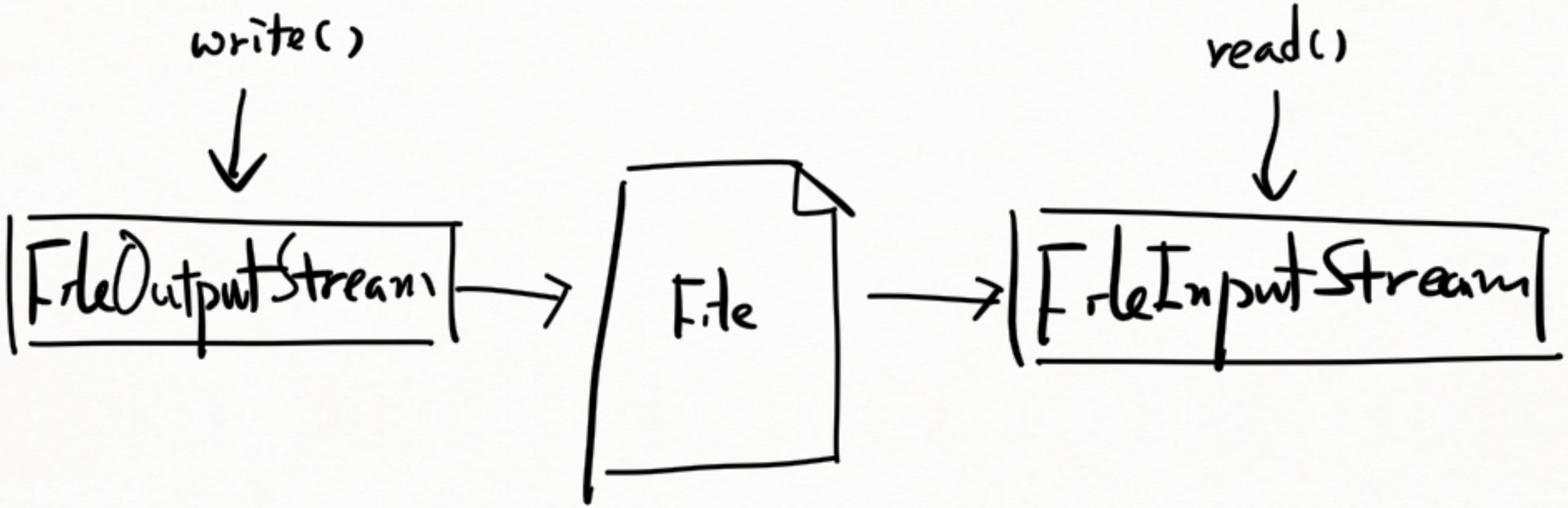
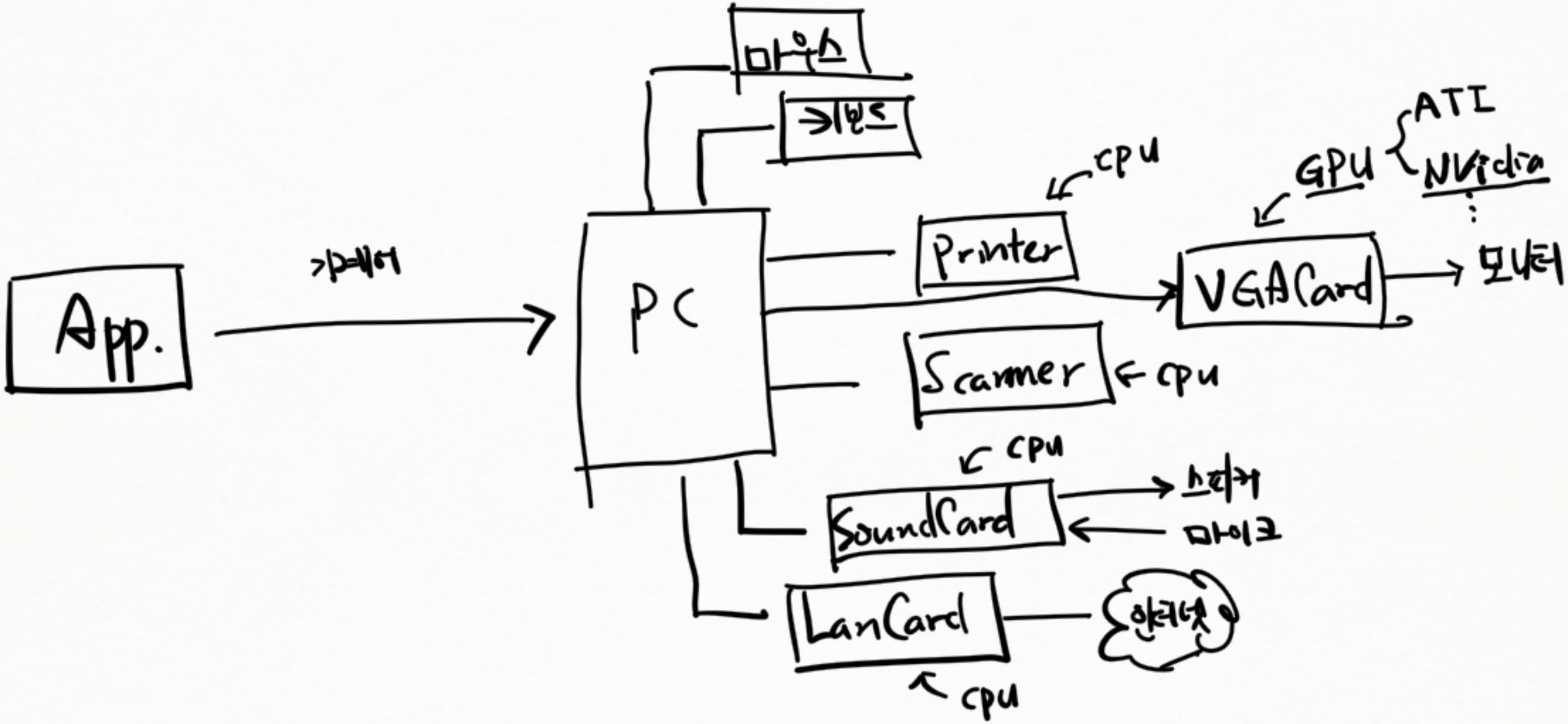
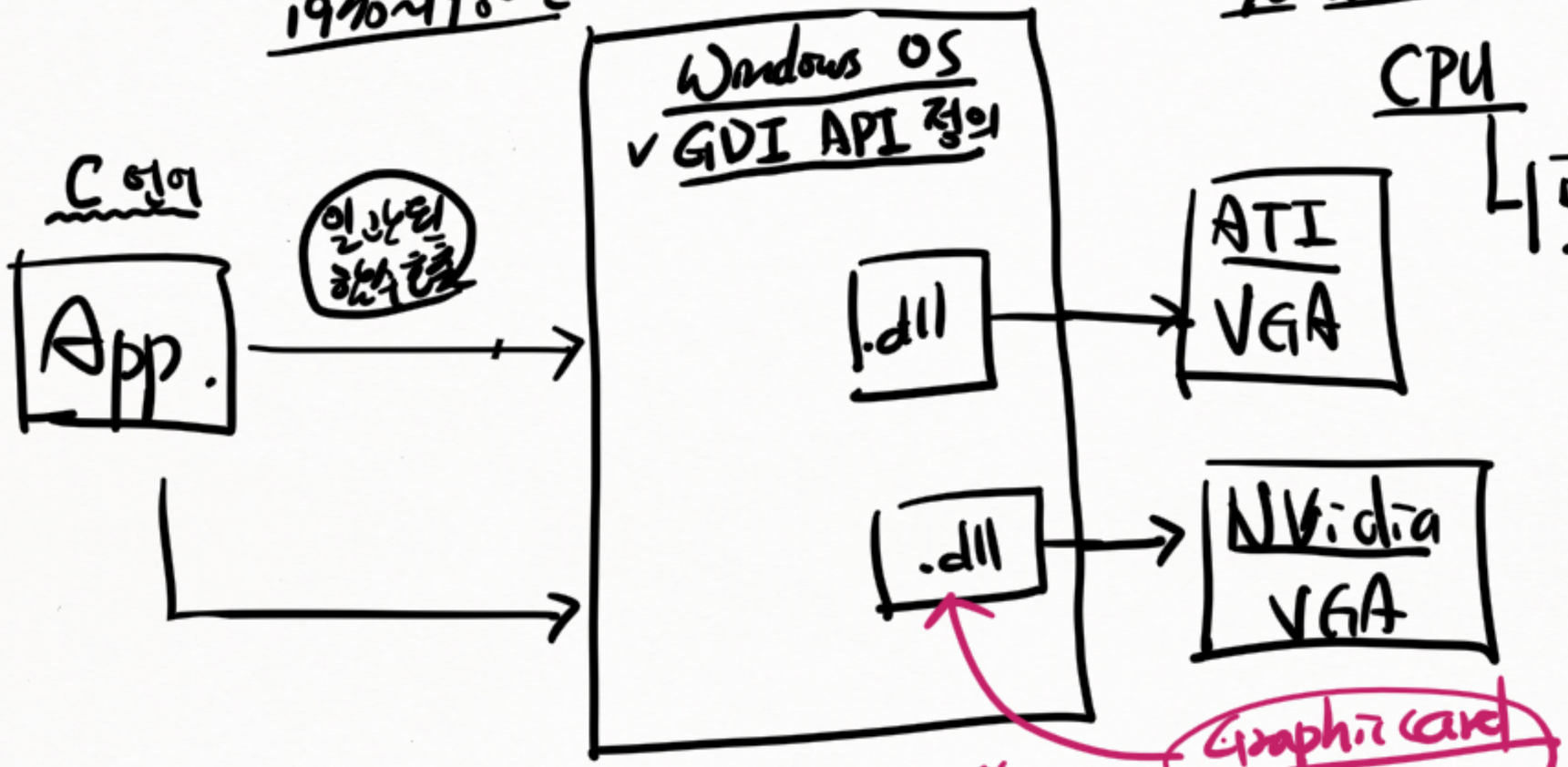

write()
read()
FileOutputStream
File
FileInputStream


App.
>프레임
PC
마우스
키보드
Printer
cpu
VGACard
GPU
ATI
NVidia
모니터
Scanner
cpu
SoundCard
cpu
스피커
마이크
LanCard
cpu
인터넷

1983 → PC → Windows 1.0, 3.0, 3.1

1990 중반

1980~1990초

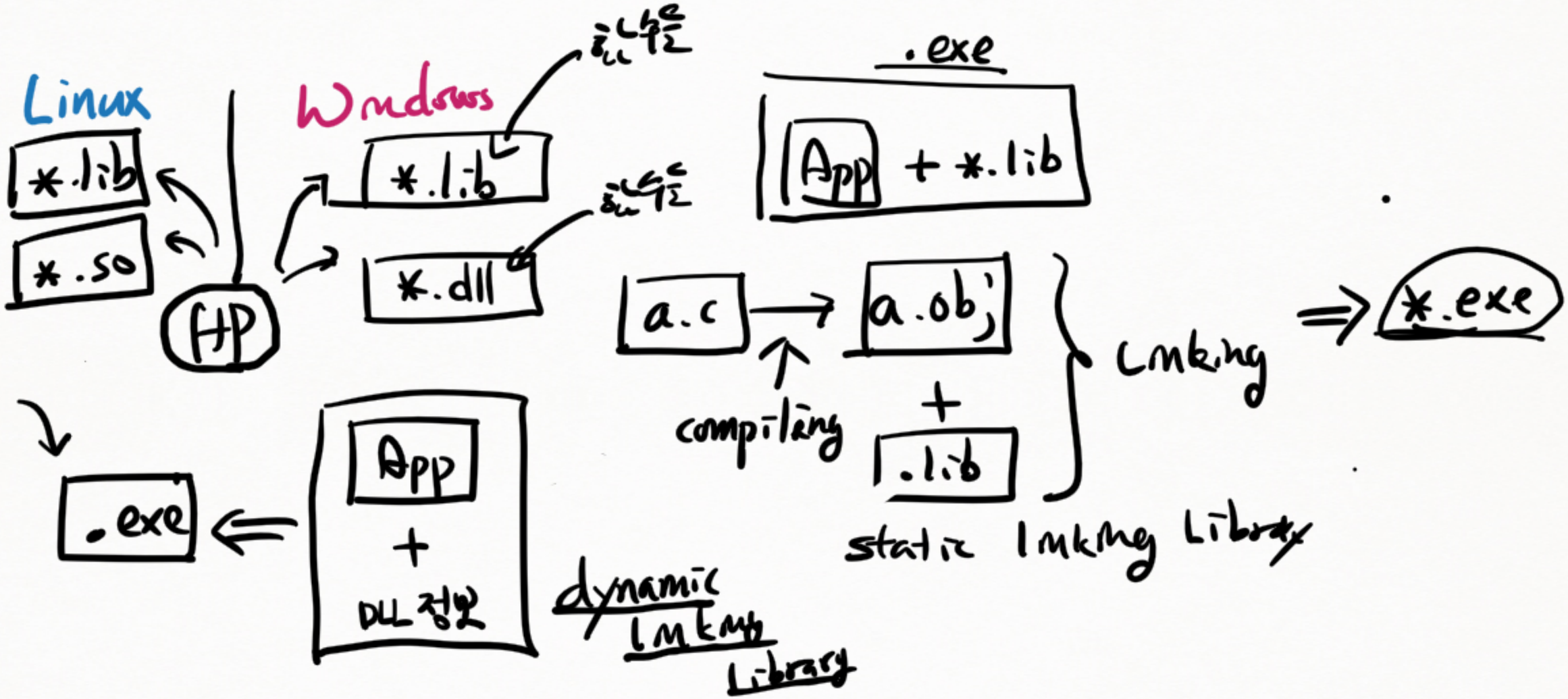


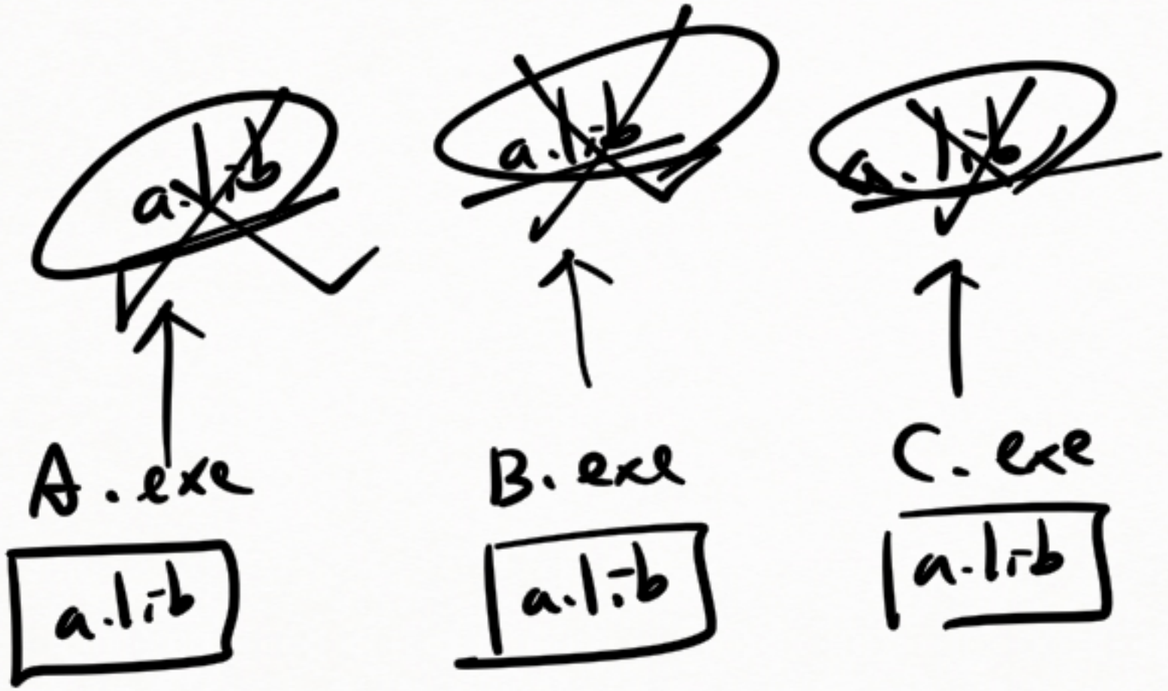
graphic card 제작

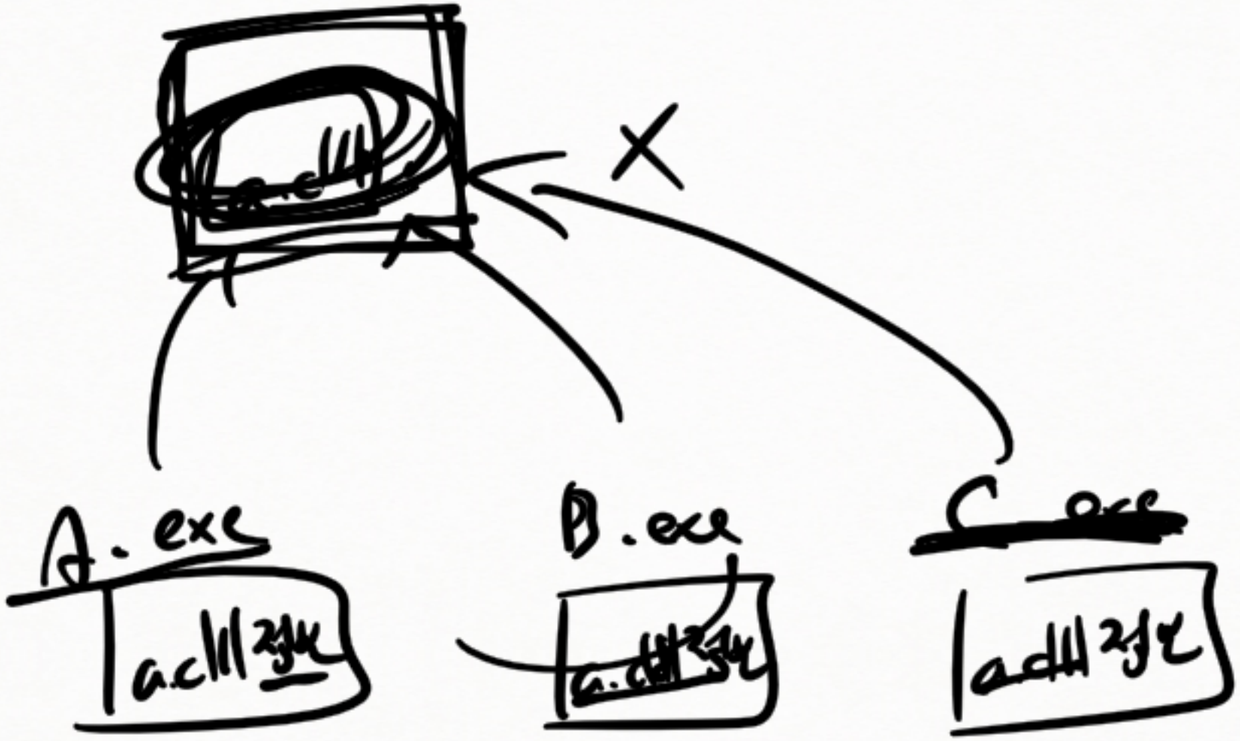
graphic "driver" 역할을 하는 S/W

CPU
RAM
명령어 + 데이터

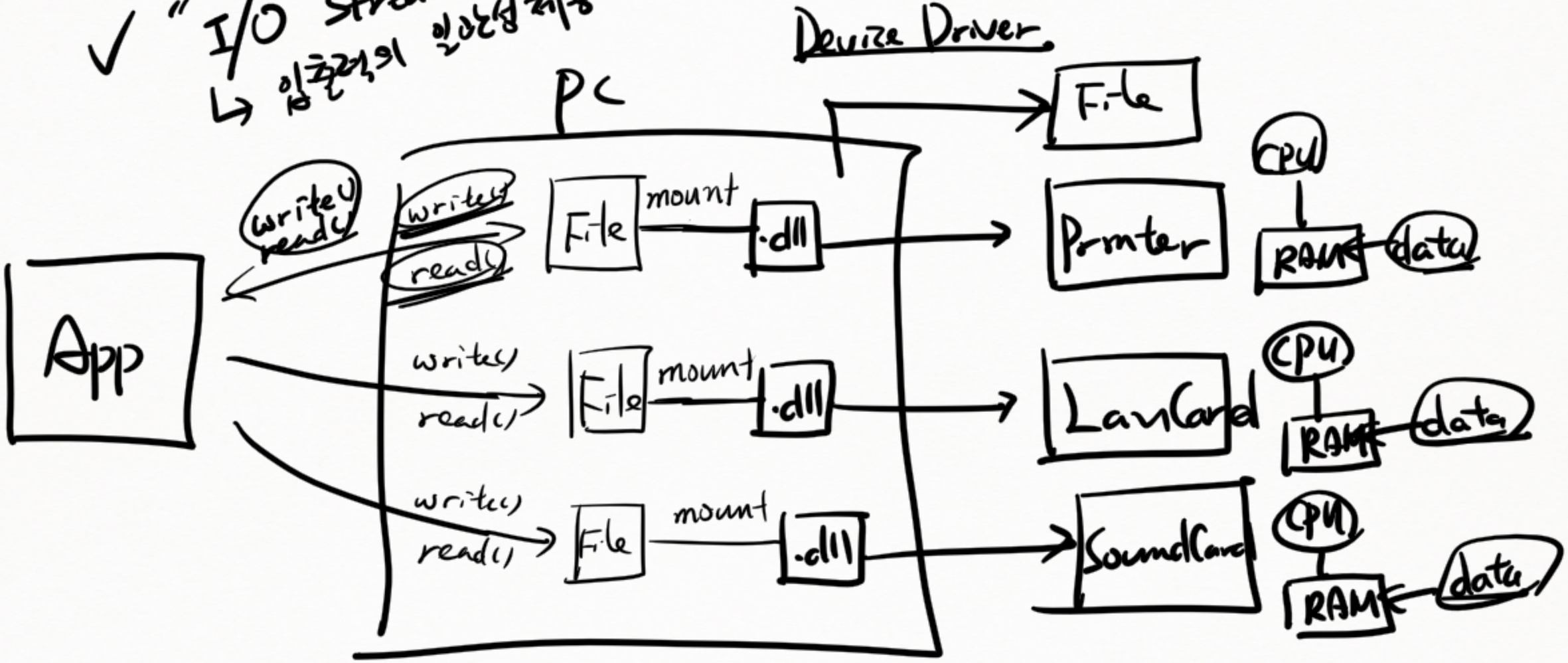
Prmter
cpu
+
RAM
양방향 데이터


Linux
*.lib
*.so
HP
Windows
*.lib
*.dll
.exe
App + *.lib
a.c
a.obj
compiling
+
.lib
Linking
*.exe
static linking Library
.exe
App
+
DLL 정보
dynamic linking Library


a.lib
a.lib
a.lib
A.exe
B.exe
C.exe
a.lib
a.lib
a.lib


a.dll
X
A.exe
a.dll 정보
B.exe
a.dll 정보
C.exe
a.dll 정보

✓ "I/O Stream API"
↳ 입출력의 일관성 제공



Java I/O Stream API

	byte stream	character stream
입력	InputStream	Reader
출력	OutputStream	Writer

Data Sink Stream Class

	Input	Output
File	FileInputStream FileReader	FileOutputStream FileWriter
Memory	ByteArrayInputStream CharArrayReader StringRead	ByteArrayOutputStream CharArrayWriter StringWriter
~~X~~ (다른 스트림) Process	PippedInputStream PippedReader	PippedOutputStream PippedWriter

ex) .grf .jpg .ppt .avi .pdf

ex) .csv .html .java .js .css .txt

JVM { '가' AC00 UCS2 (UTF-16) } → FileWriter → B0A141 (가 A)

$ java com.eomcs.io.ex03.Exam0110

MS949

EAB0 0041 (가 A)

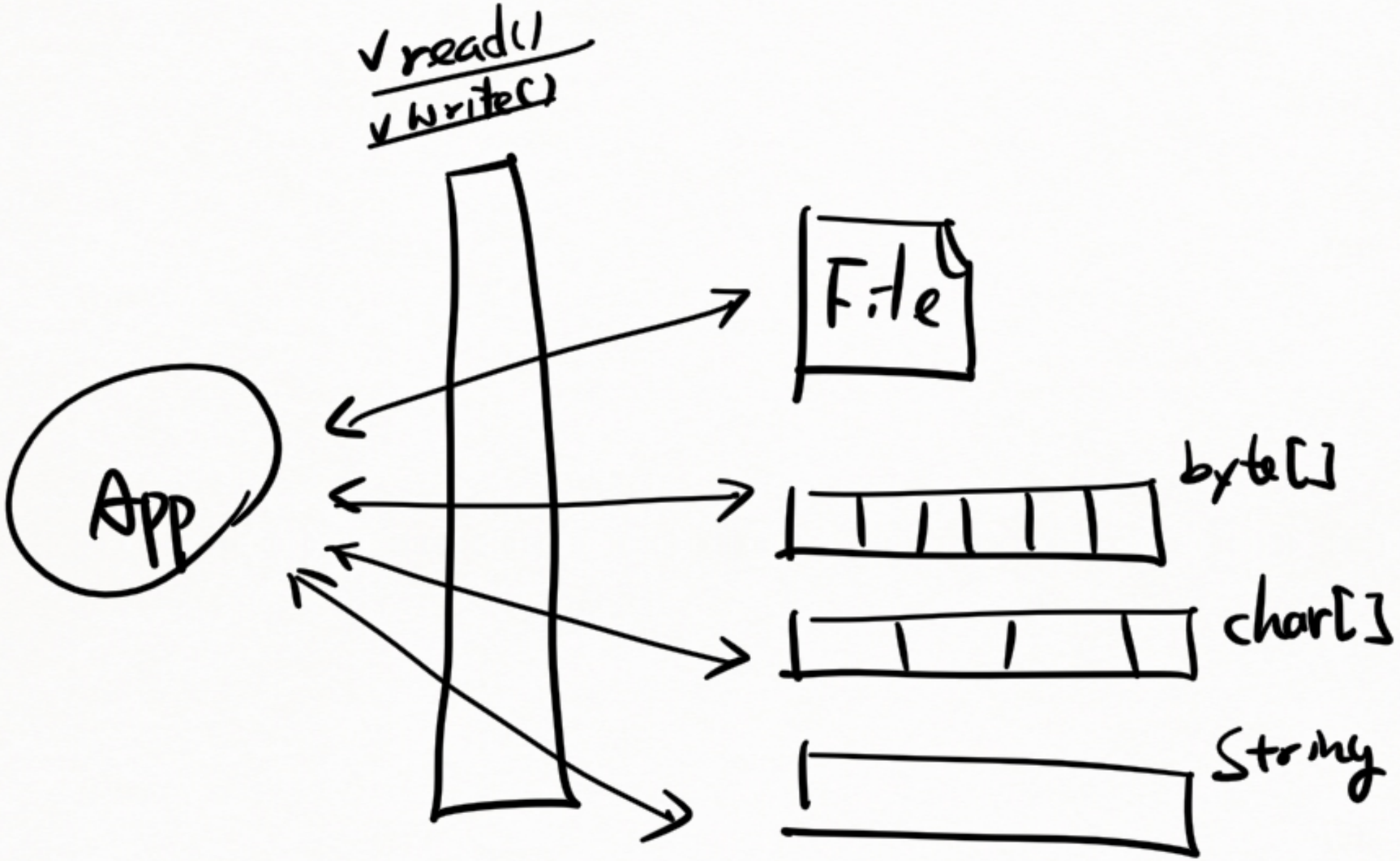
UTF-8

× OS의 기본 문자셋

⇓

✓ Windows → MS949

Linux/MacOS → UTF-8 ✓


read()
write()
App
File
byte[]
char[]
String

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

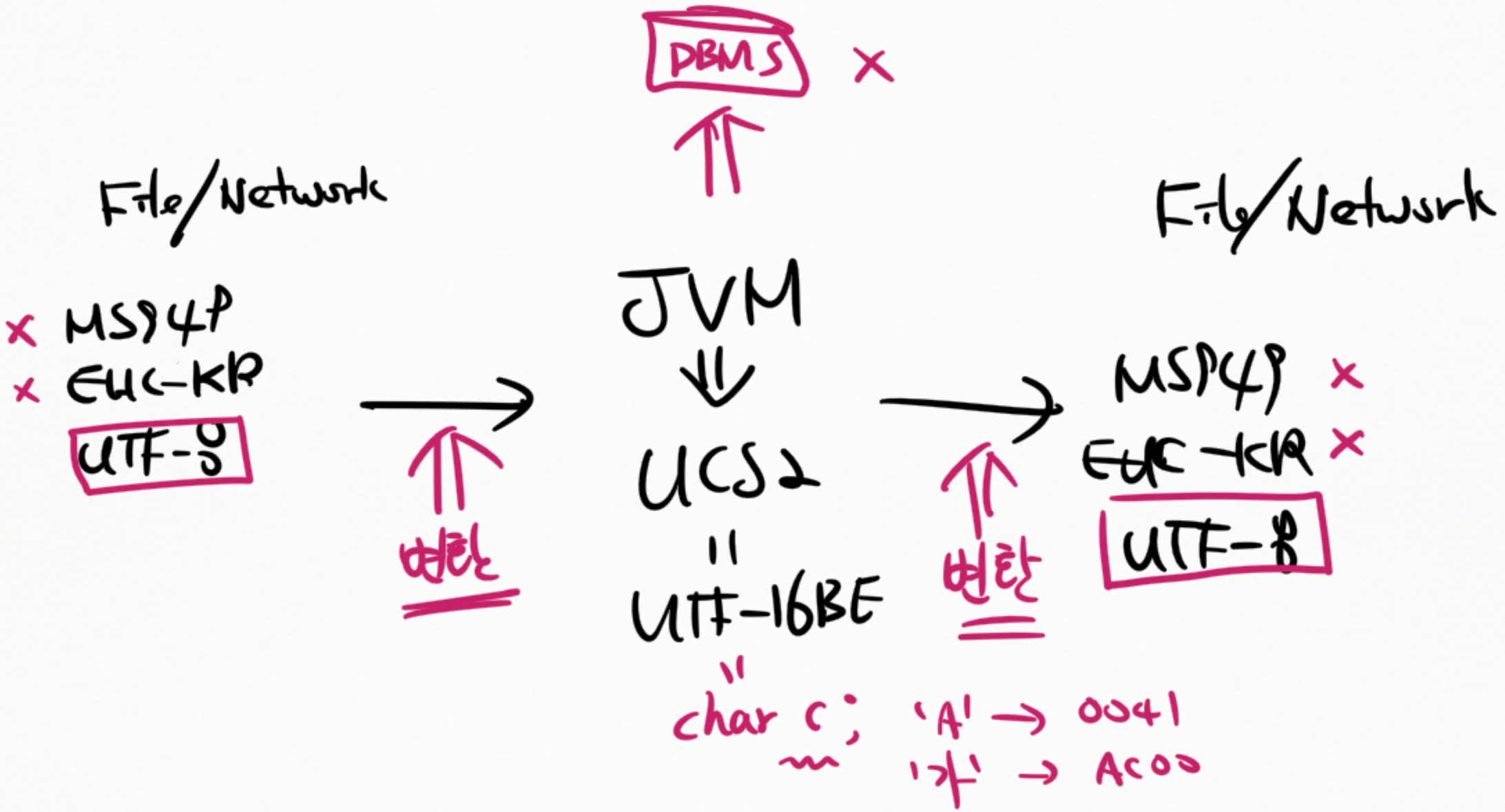
JVM

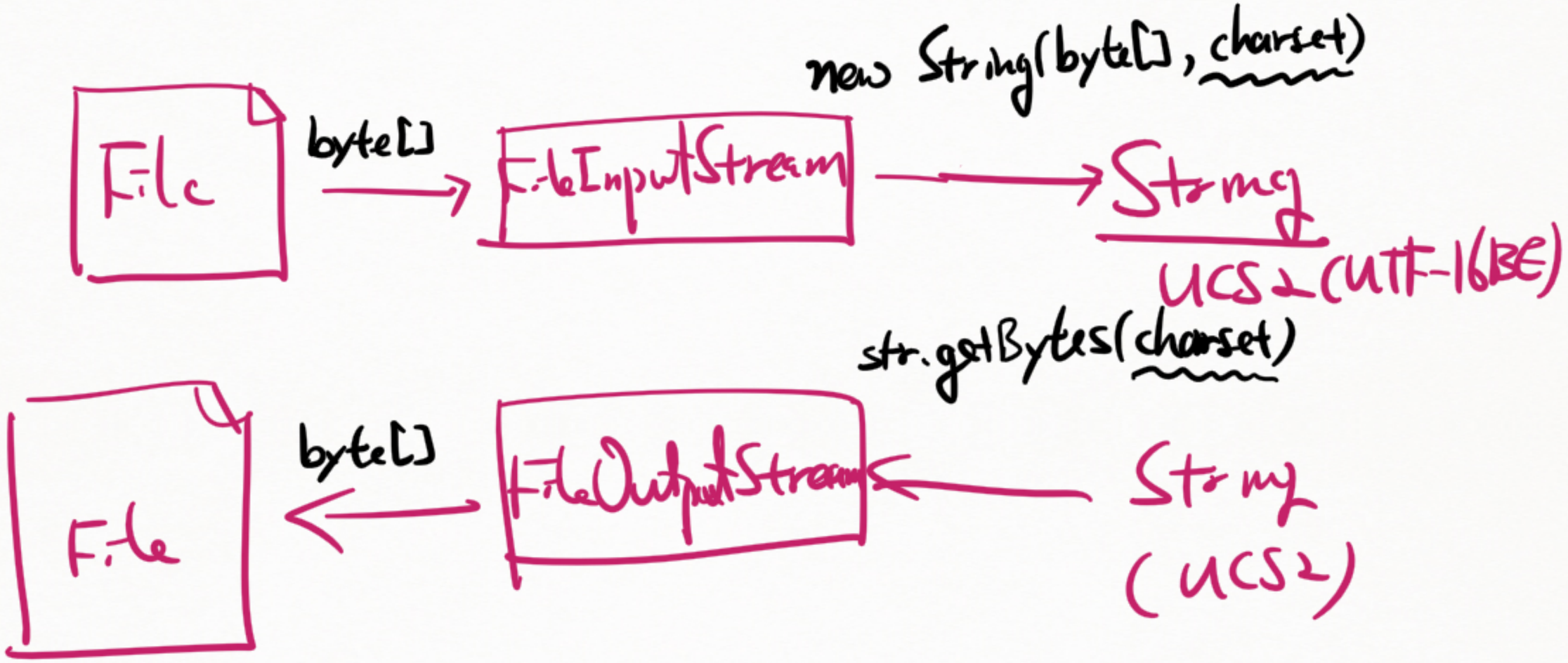
⇓

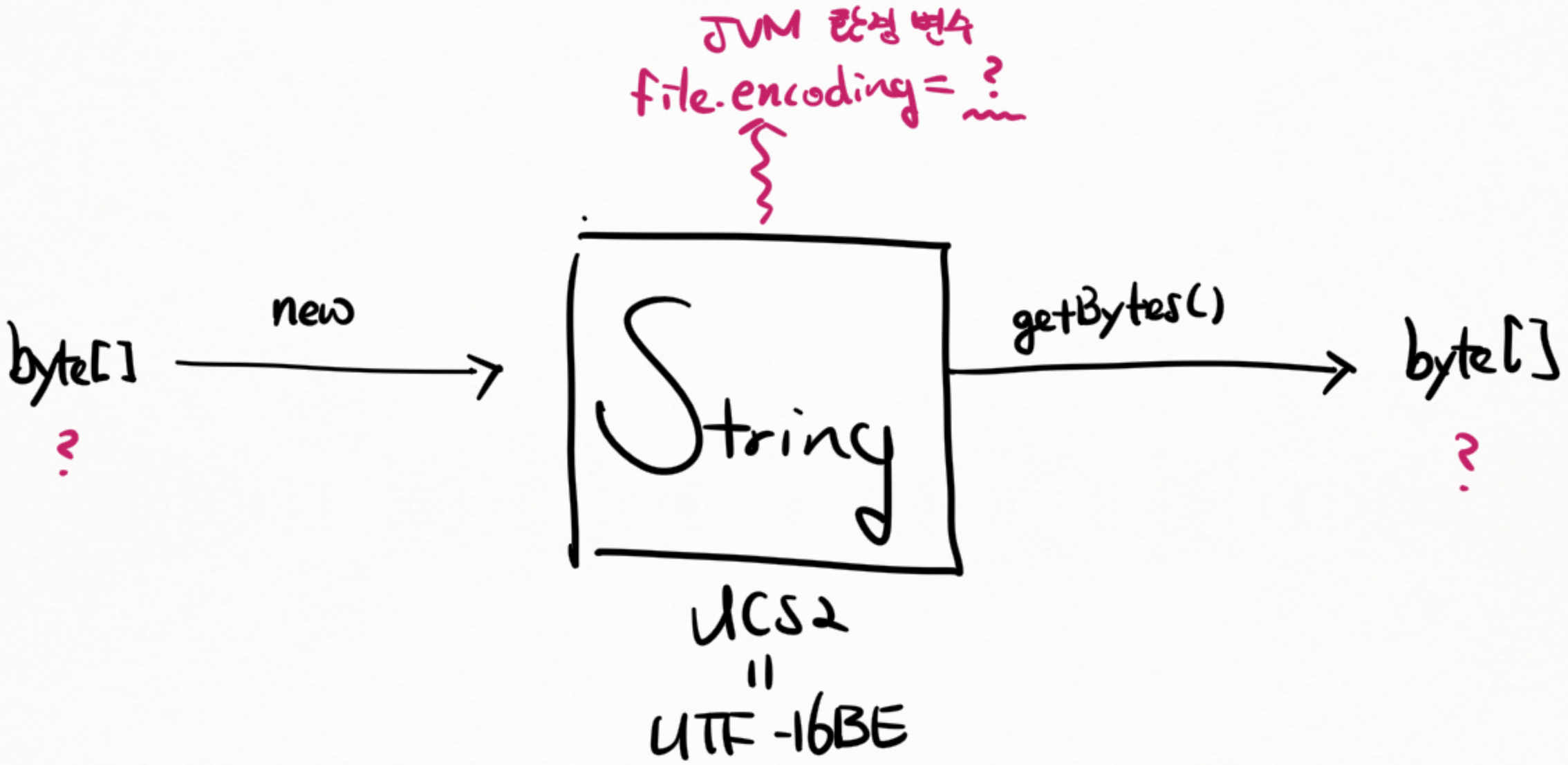
char : UCS2 (UTF-16BE)

0041	0042	AC00	AC01
A	B	가	각

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


DBMS
File/Network
MS949
EUC-KR
UTF-8
변환
JVM
UCS2
UTF-16BE
char c;
'A' → 0041
'가' → AC00
변환
File/Network
MS949
EUC-KR
UTF-8


new String(byte[], charset)
File
byte[]
FileInputStream
String
UCS2 (UTF-16BE)
str.getBytes(charset)
File
byte[]
FileOutputStream
String
(UCS2)


JVM 환경 변수
file.encoding = ?
byte[]
?
new
String
getBytes()
byte[]
?
UCS2
=
UTF-16BE

E ⇒ 3017 - 3030 ⇒ 꽝 ?

2xxx 중식

34xx - 30xx 한식

4444x - 30xx - 30xx 100 만원

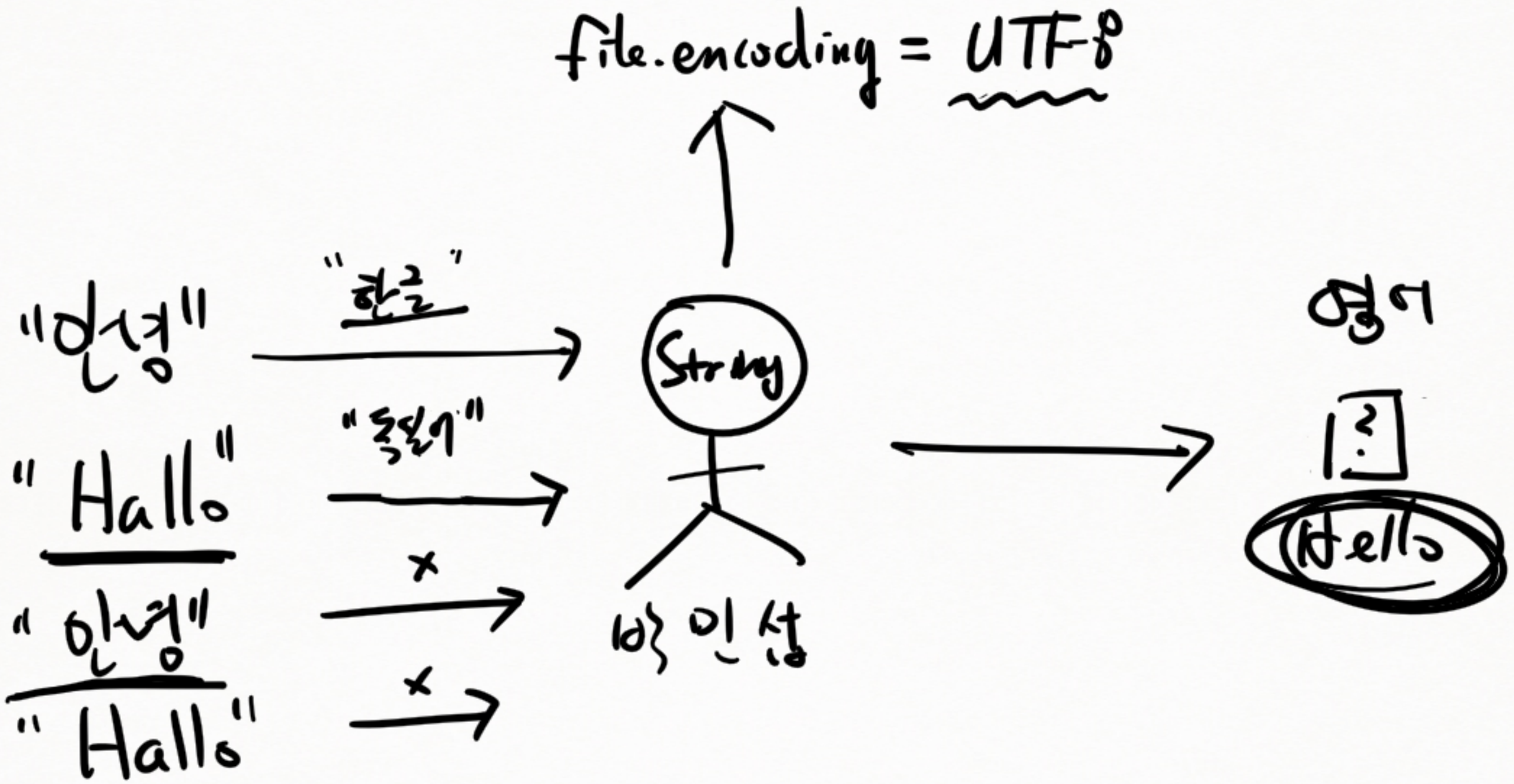
9999 + 30xx - 32xx - 30xx 1억

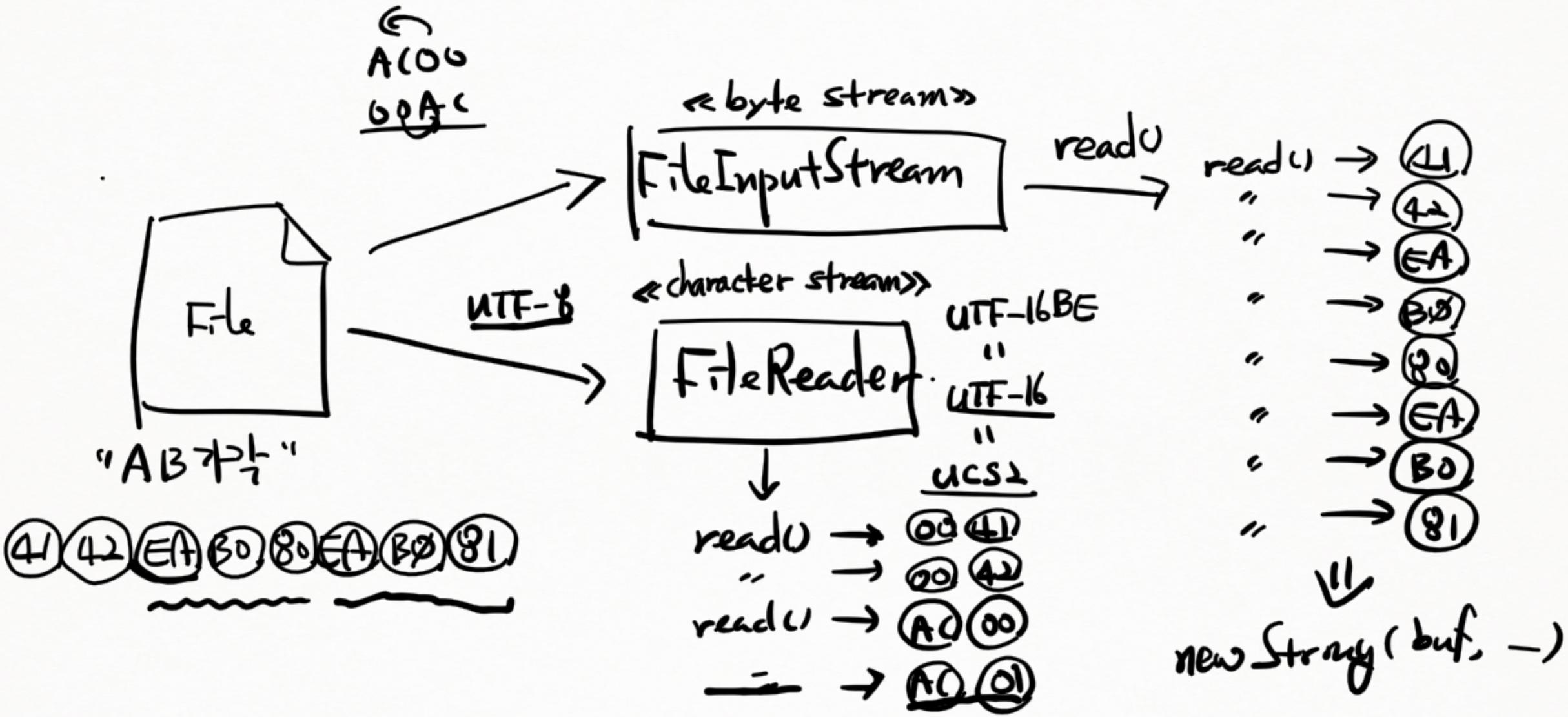
✓ A ⇒ 2234

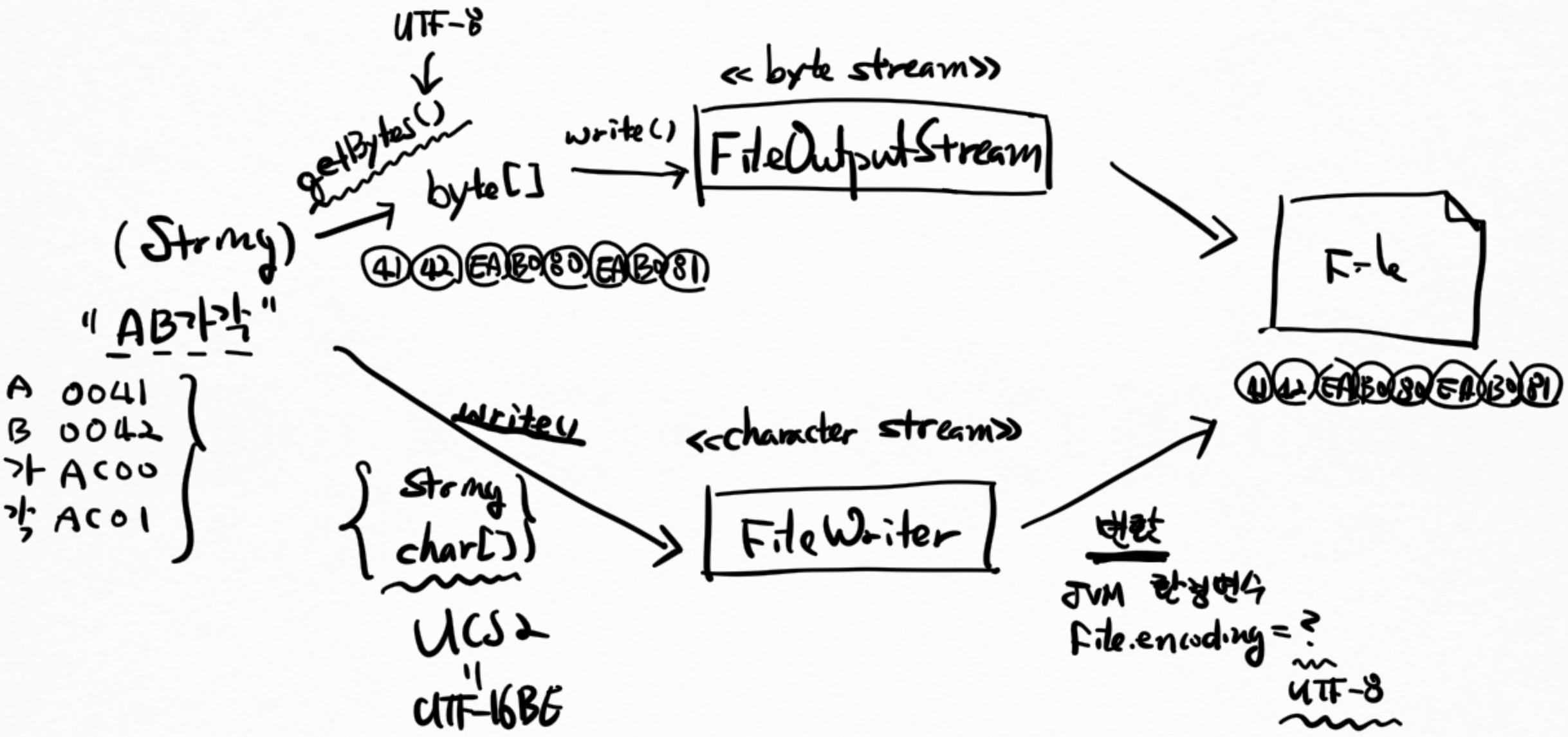
✓ B ⇒ 3478 - 3012

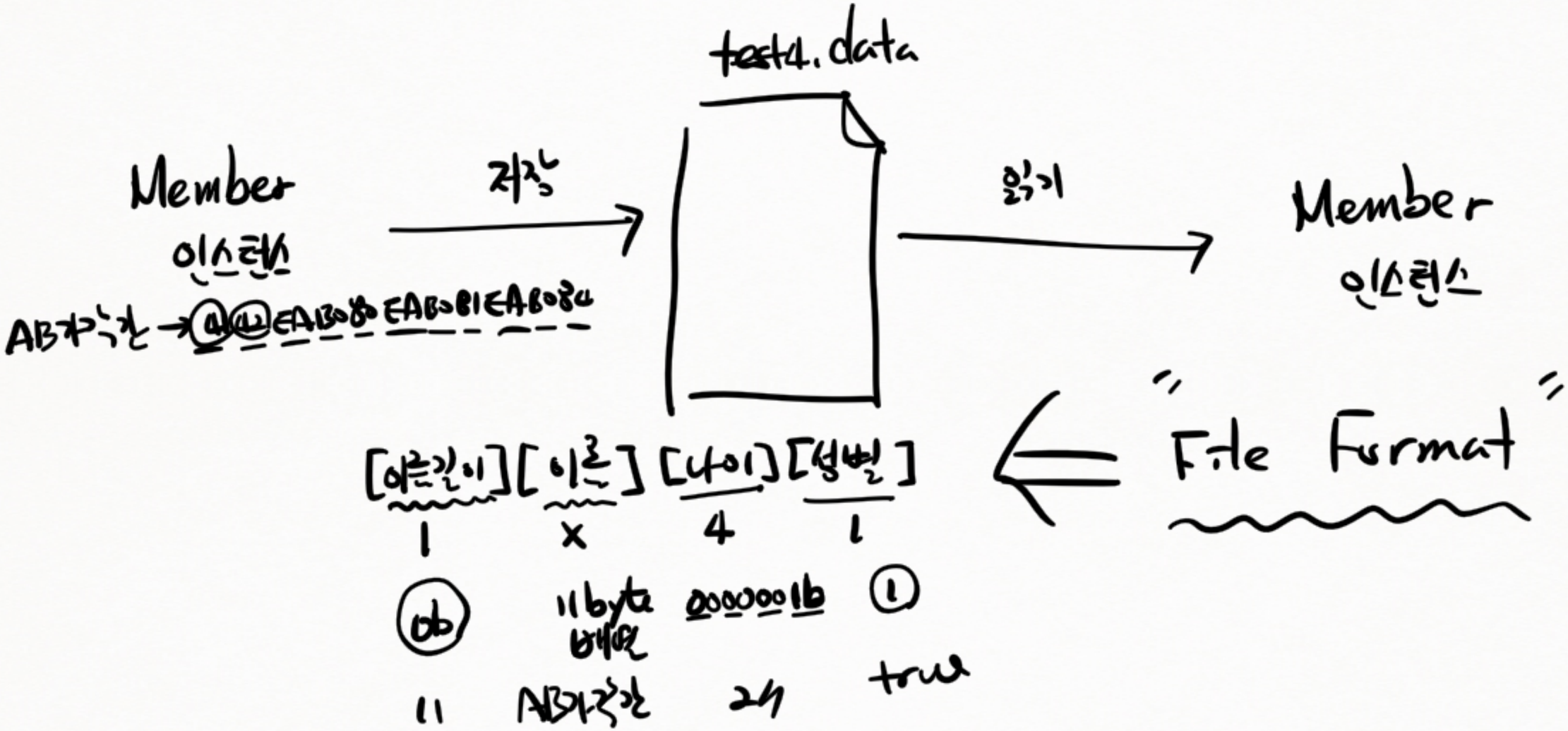
✓ C ⇒ 4444 - 3011 - 3033

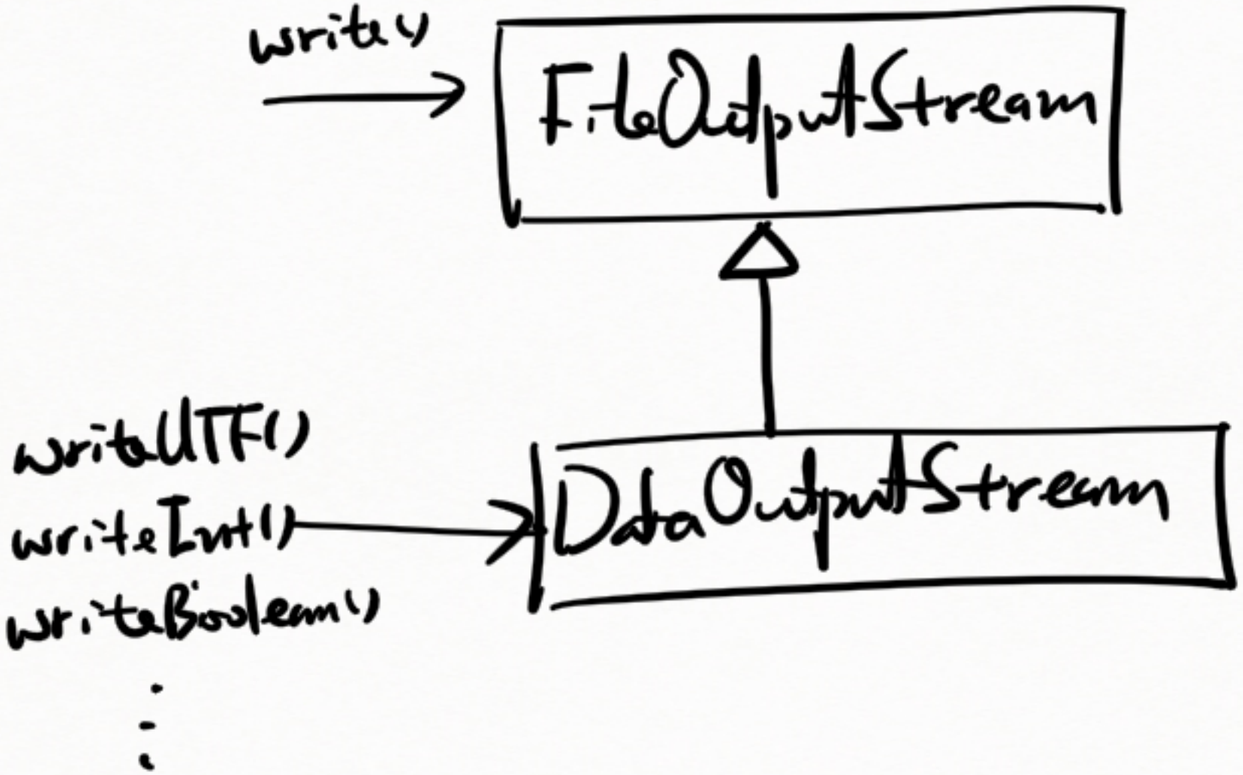
✓ D ⇒ 9999 - 3017 - 3211 - 3083

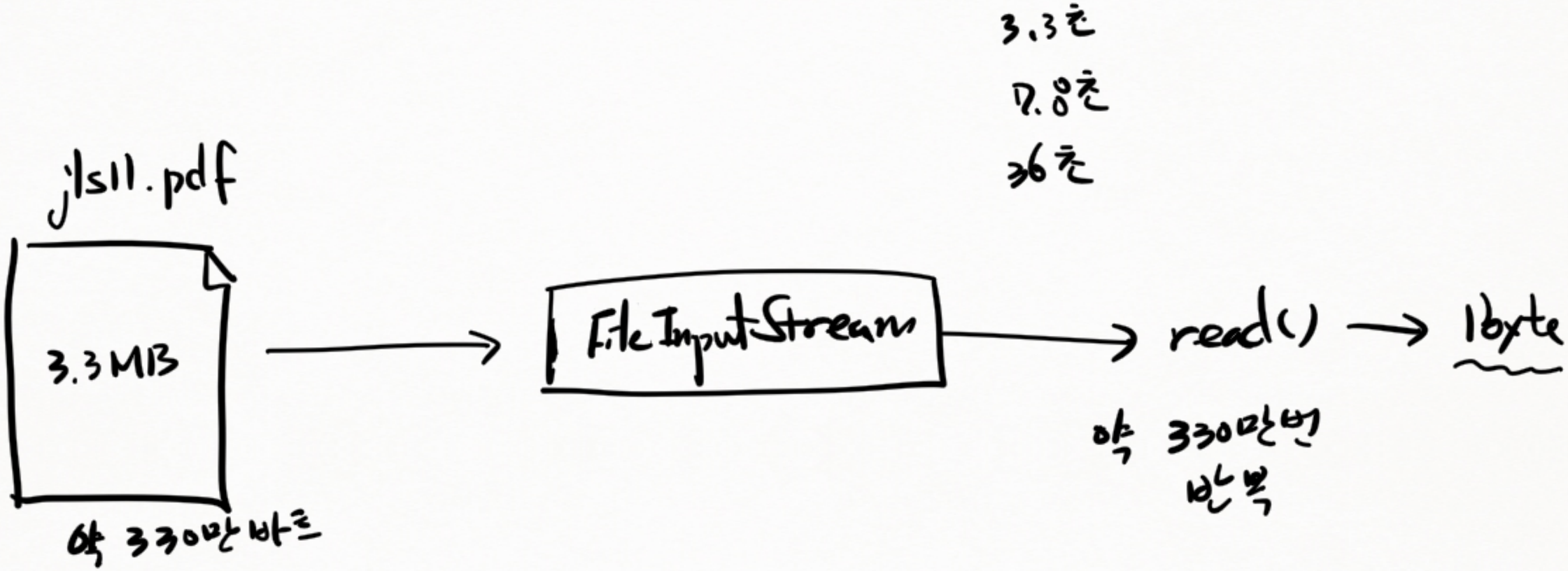

file.encoding = UTF-8
"안녕"
"한글"
"Hallo"
"독일어"
"안녕"
×
"Hallo"
×
String
바이트
영어
?
Hello

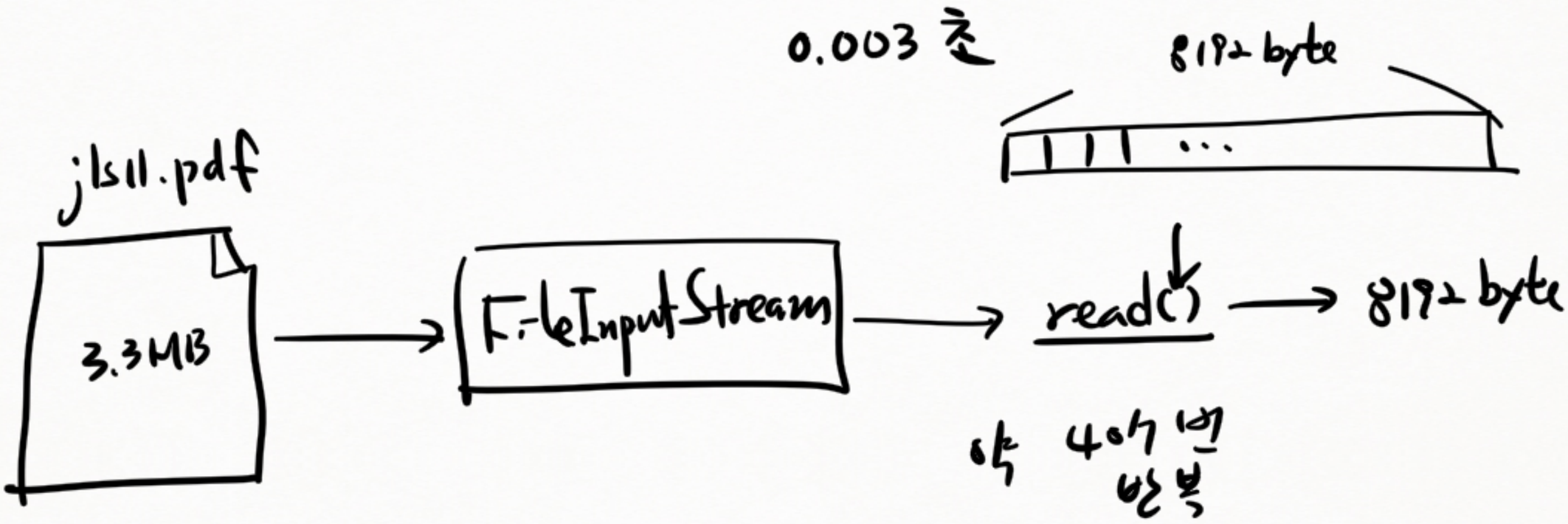

AC00
00AC
File
"AB가나"
41 42 EA B0 80 EA B0 81
UTF-8
<<byte stream>>
FileInputStream
read()
read() → 41
" → 42
" → EA
" → B0
" → 80
" → EA
" → B0
" → 81
<<character stream>>
FileReader
UTF-16BE
"
UTF-16
"
UCS2
read() → 00 41
" → 00 42
read() → AC 00
— → AC 01
new String(buf, —)

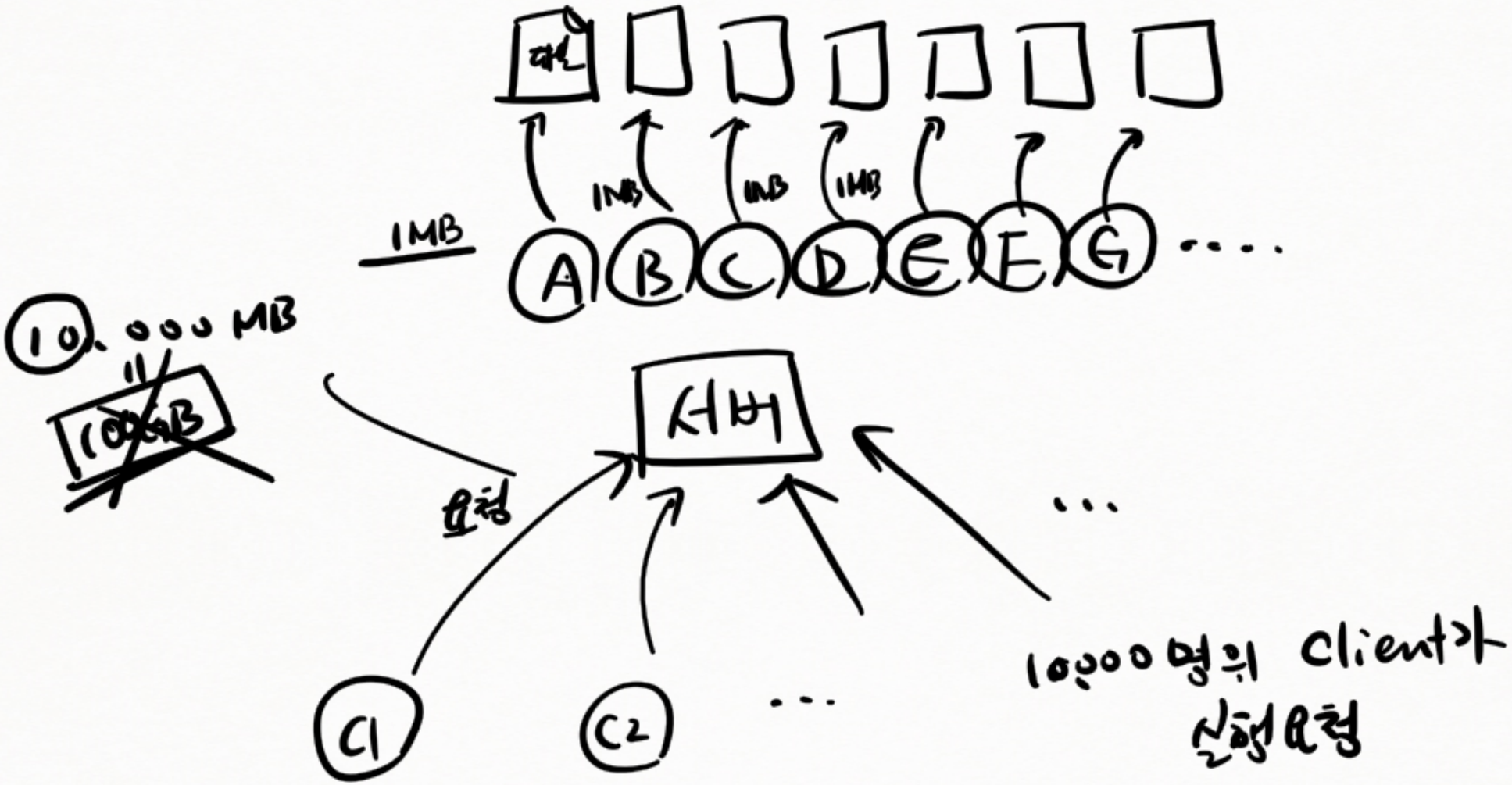

UTF-8
getBytes()
(String)
"AB가각"
A 0041
B 0042
가 AC00
각 AC01
byte[]
41 42 EA B0 80 EA B0 81
write()
<< byte stream >>
FileOutputStream
File
41 42 EA B0 80 EA B0 81
write()
String
char[]
UCS2
UTF-16BE
<<character stream>>
FileWriter
변환
JVM 환경변수
file.encoding = ?
UTF-8

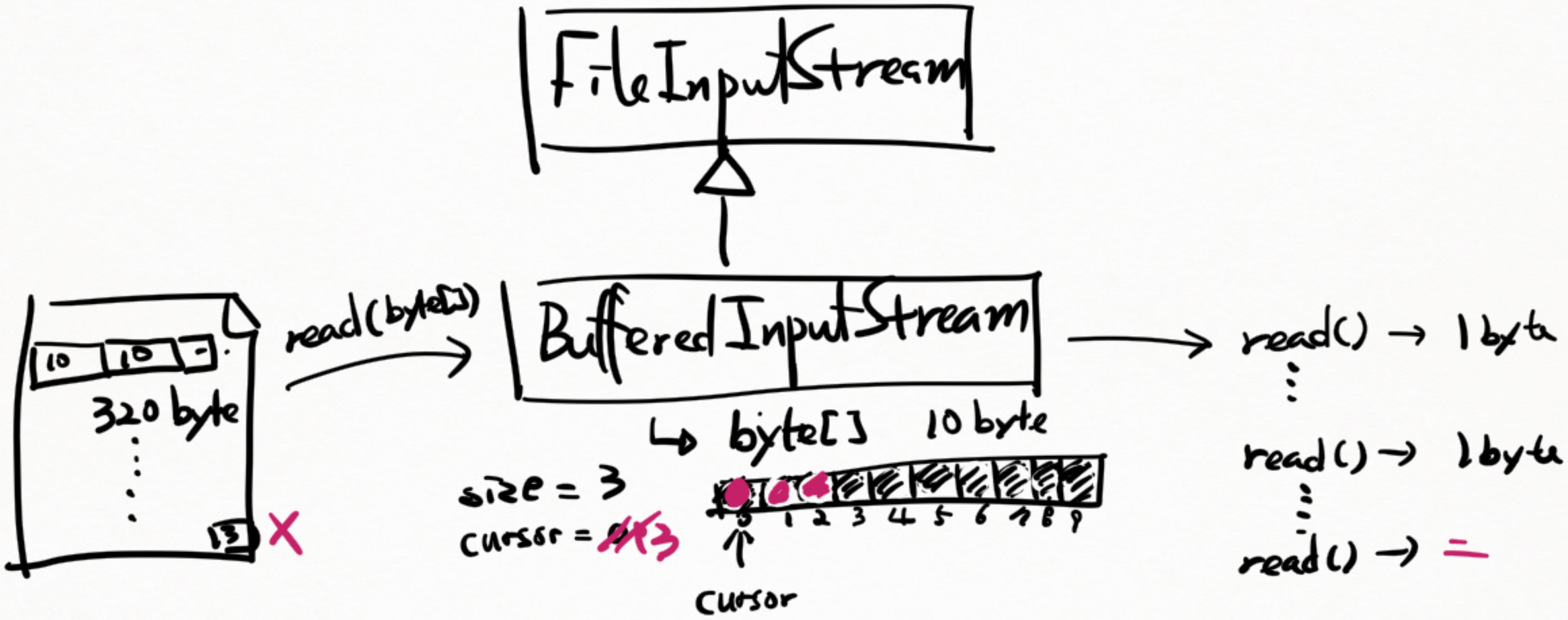

test4.data
Member
인스턴스
저장
읽기
Member
인스턴스
AB자주한
[이름길이] [이름] [나이] [성별]
1 x 4 1
1byte
배열
00000001b
true
"File Format"


write()
FileOutputStream
writeUTF()
writeInt()
writeBoolean()
DataOutputStream


3.3초
7.8초
36초
jls11.pdf
3.3MB
약 330만 바이트
FileInputStream
read()
1byte
약 330만번
반복


0.003 초
8192 byte
jls11.pdf
3.3MB
FileInputStream
read()
8192 byte
약 400번
반복


대문
1MB
1MB
1MB
1MB
A
B
C
D
E
F
G
....
10,000 MB
100GB
요청
서버
...
C1
C2
...
10,000명의 client가
실행요청


FileInputStream
BufferedInputStream
read(byte[])
10
10
320 byte
13
byte[] 10 byte
size = 3
cursor = 3
0 1 2 3 4 5 6 7 8 9
cursor
read() → 1 byte
read() → 1 byte
read() → -

'가'

buf

byte[]

EA	B0	80

byte

int

11101010

11111111	11111111	11111111	11101010

~~00000000~~ ~~11111111~~

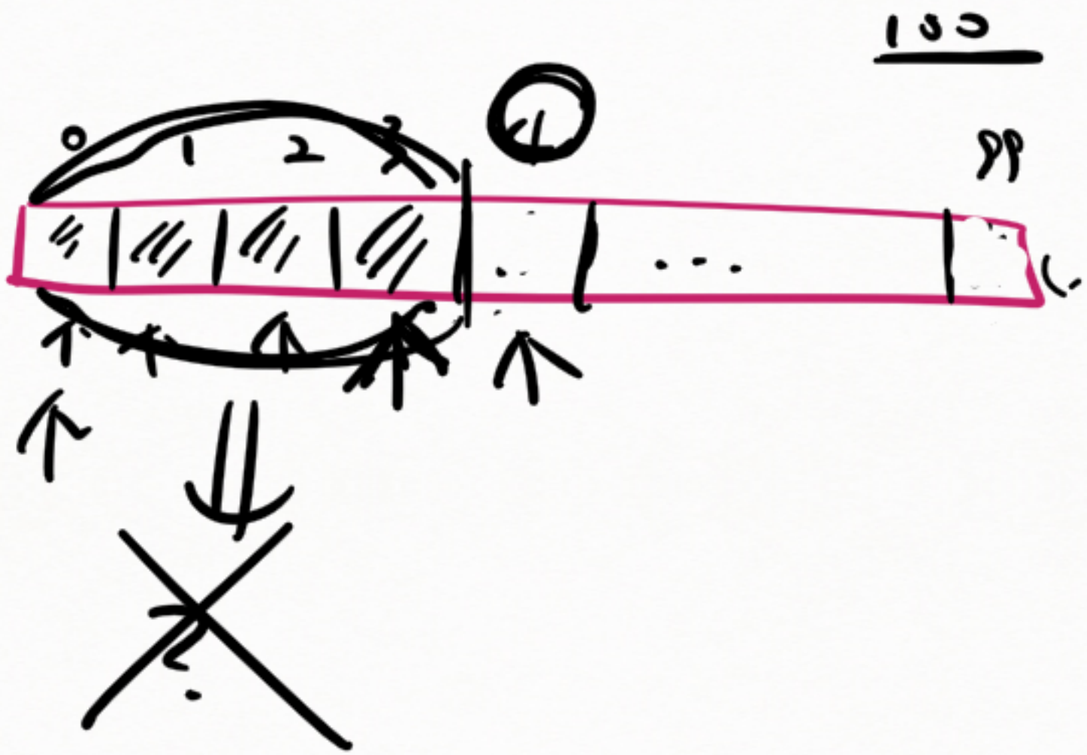
00000000 ... 11101010

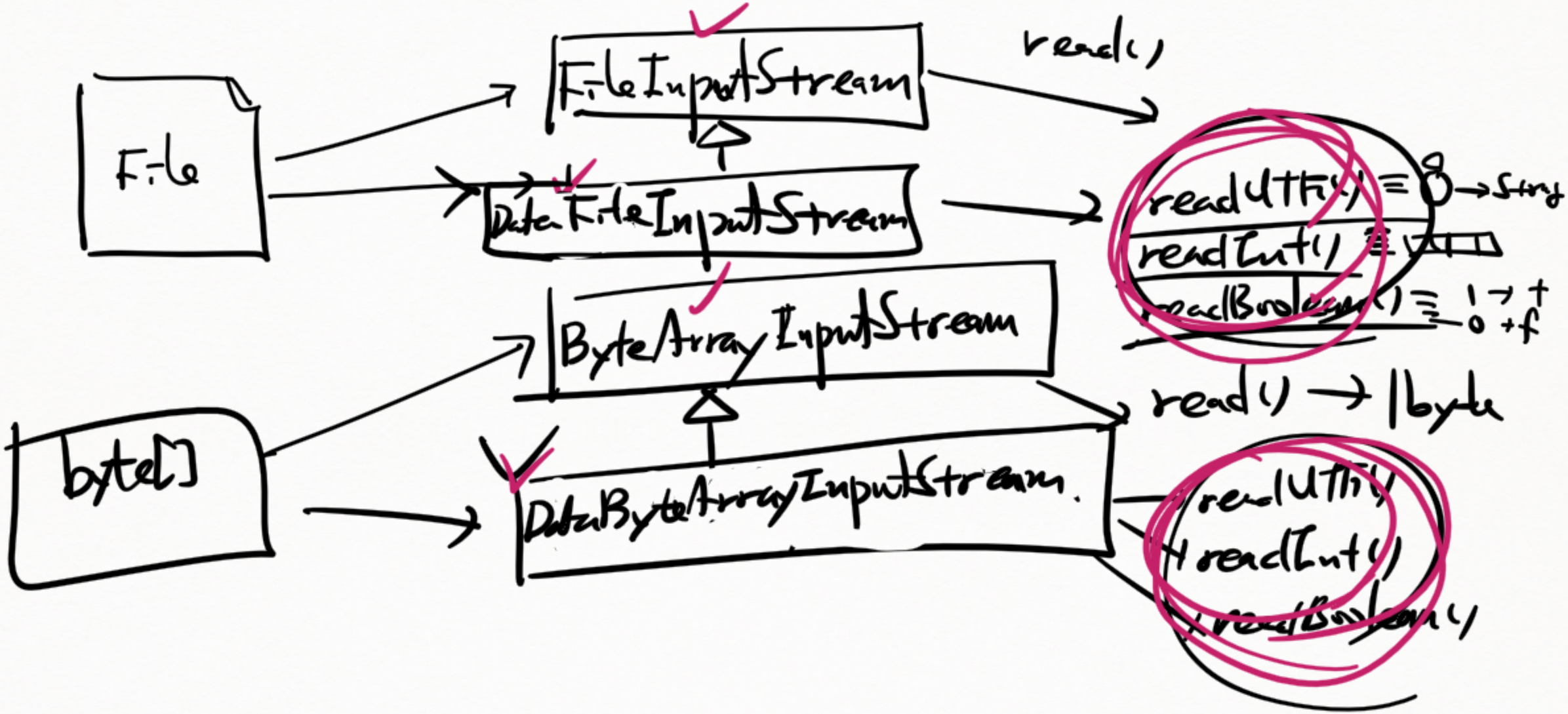
EA

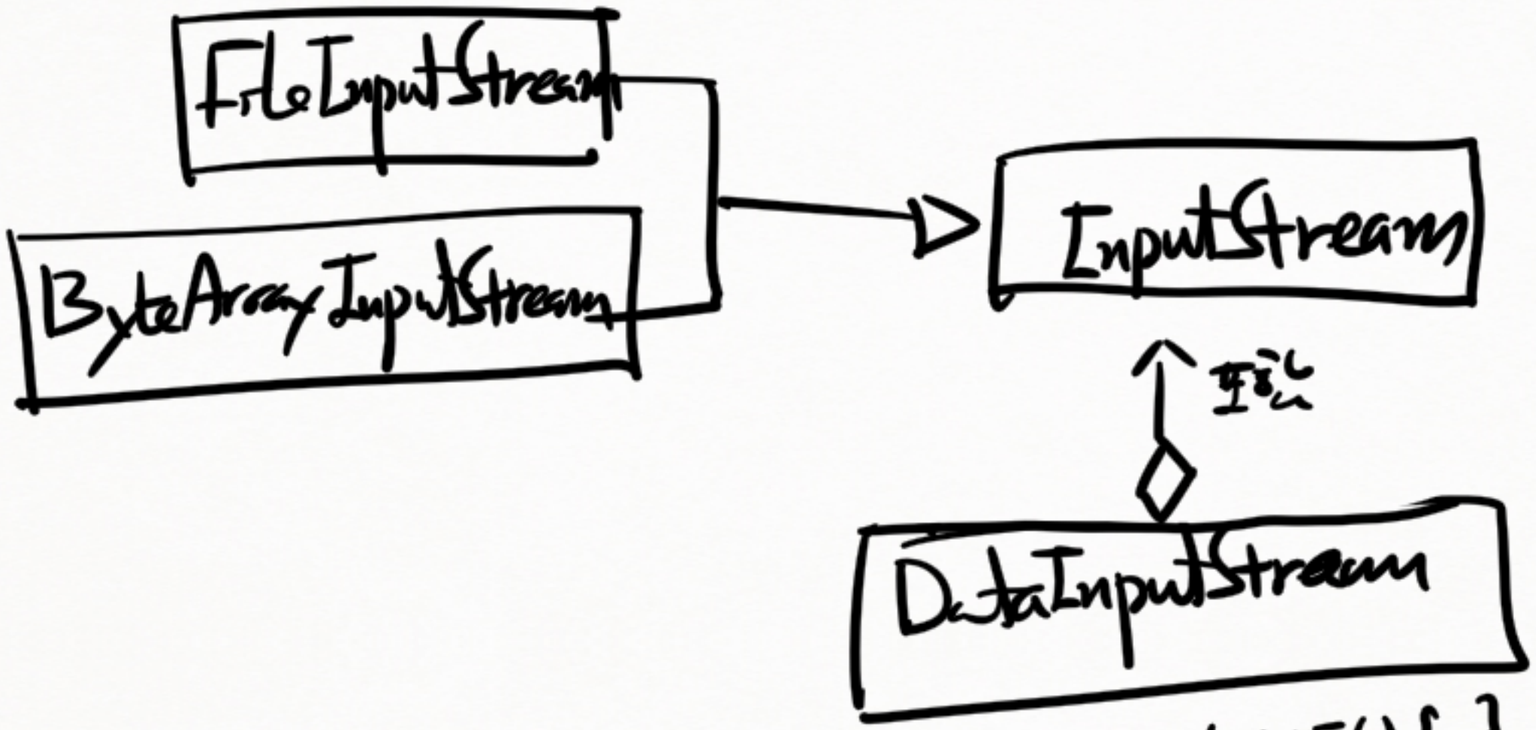
read()

00	00	00	EA

FF FF FF

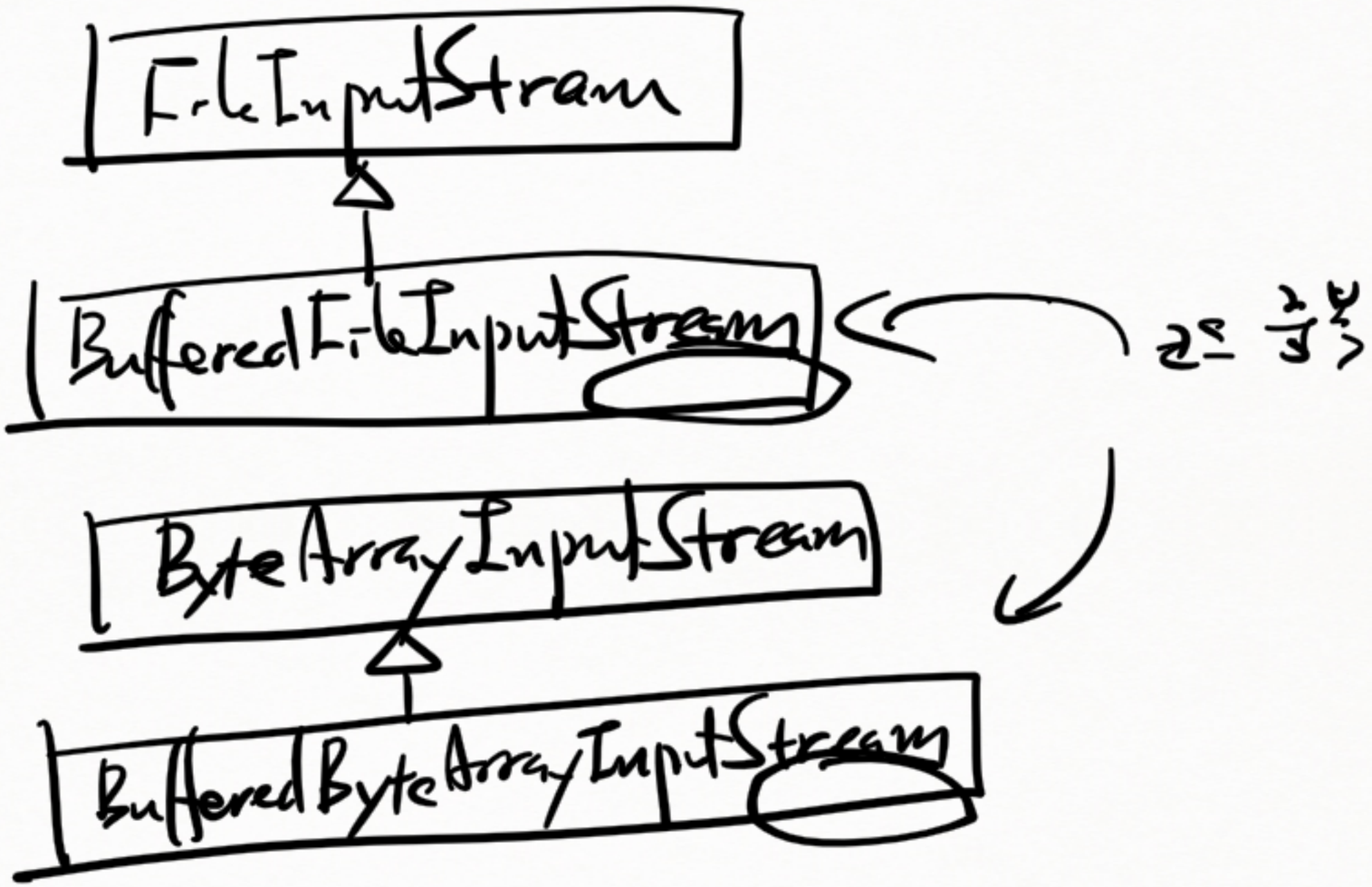

100
0
1
2
3
4
99

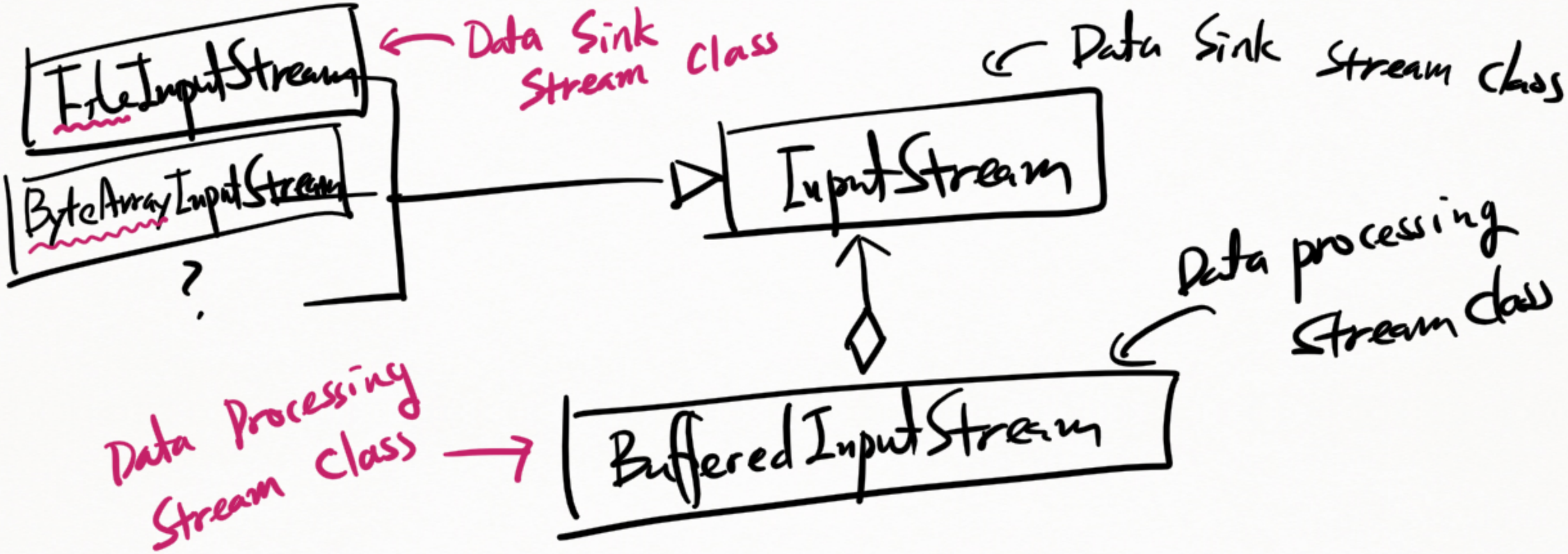

File
FileInputStream
read()
DataFileInputStream
readUTF()
readInt()
readBoolean()
ByteArrayInputStream
read() → byte
byte[]
DataByteArrayInputStream
readUTF()
readInt()
readBoolean()



장식자 (Decorator)

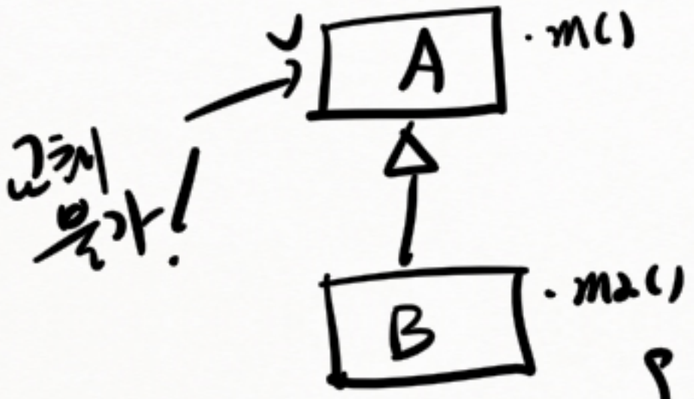
- readUTF() {-}
- readInt() {-}
- readLong() {-}
- readBoolean() {-}


FileInputStream
BufferedFileInputStream
ByteArrayInputStream
BufferedByteArrayInputStream

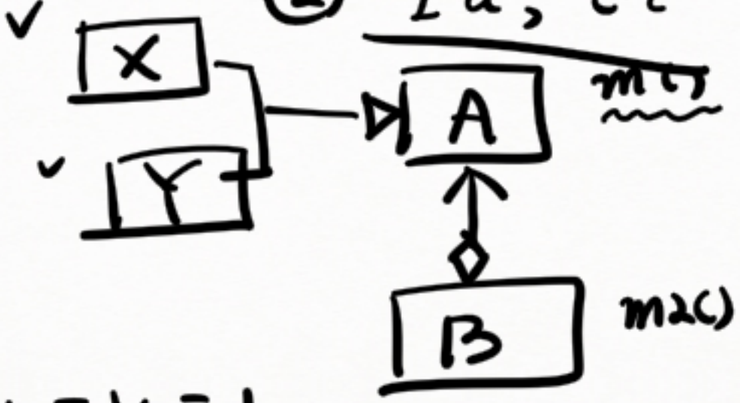

FileInputStream
ByteArrayInputStream
?
Data Sink Stream Class
Data Sink Stream class
InputStream
Data processing Stream class
Data Processing Stream Class
BufferedInputStream

기능확장

① 상속

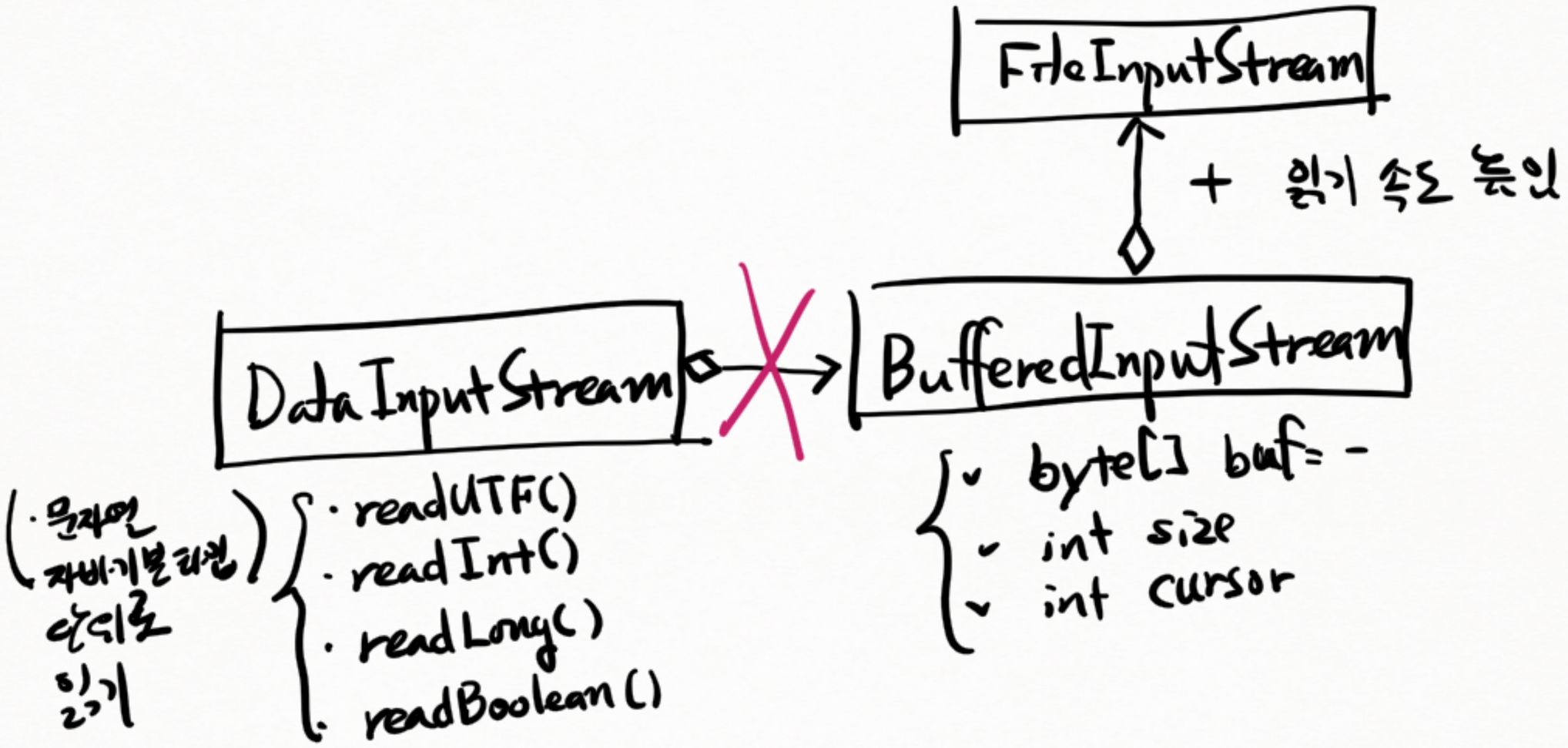


② 포함, 연관



- ✓ 중복 최소화
- ✓ 재사용
- ✓ 고치기 용이

⋮


FileInputStream
+ 읽기 속도 높인
DataInputStream
BufferedInputStream
· 문자열
자바기본타입
단위로
읽기
· readUTF()
· readInt()
· readLong()
· readBoolean()
· byte[] buf = -
· int size
· int cursor

"Decorator" 패턴

Data Sink Stream class

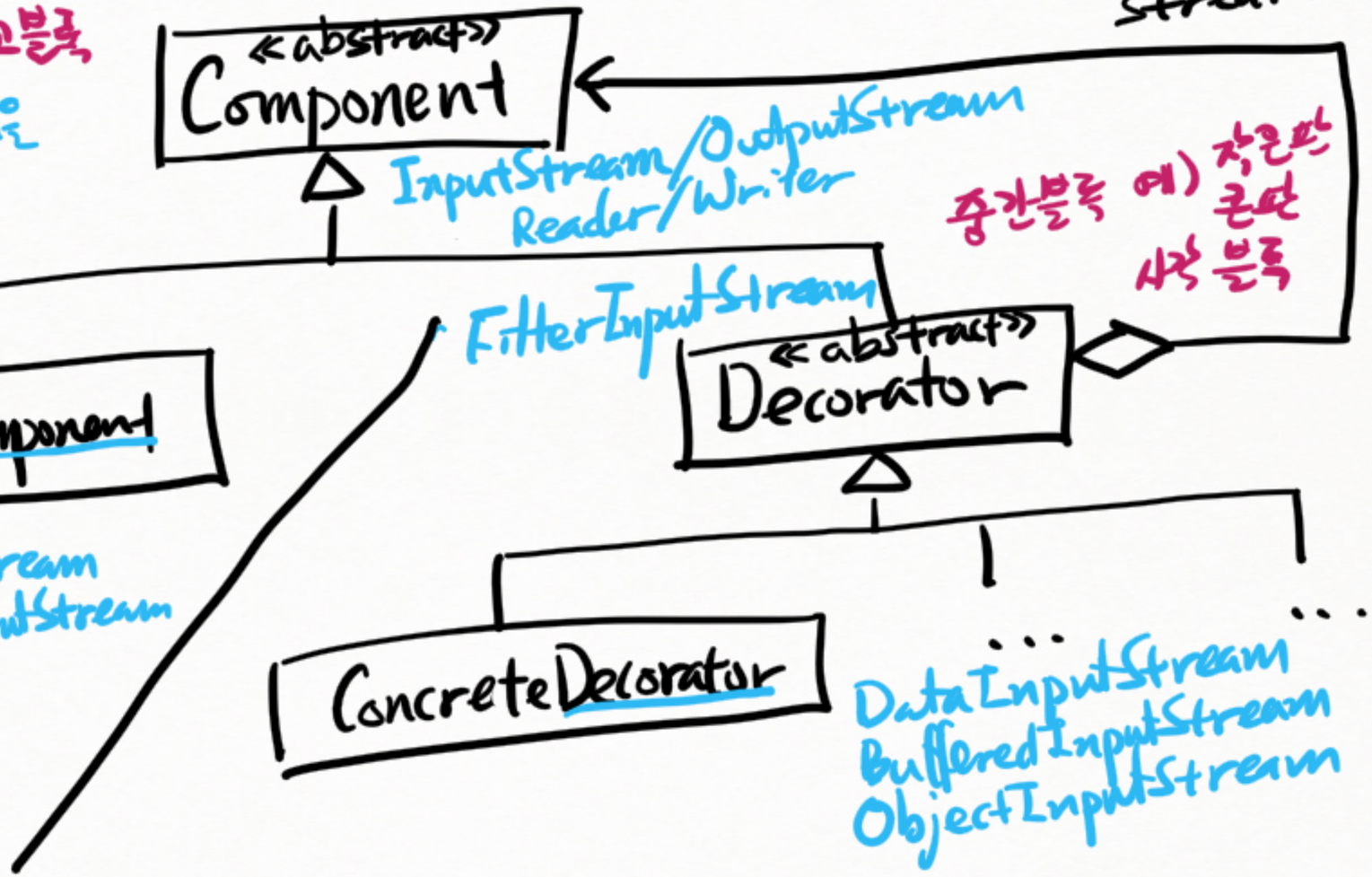
Data processing Stream class

레고블록

생성자에서 다른 InputStream을 받지 않는다-

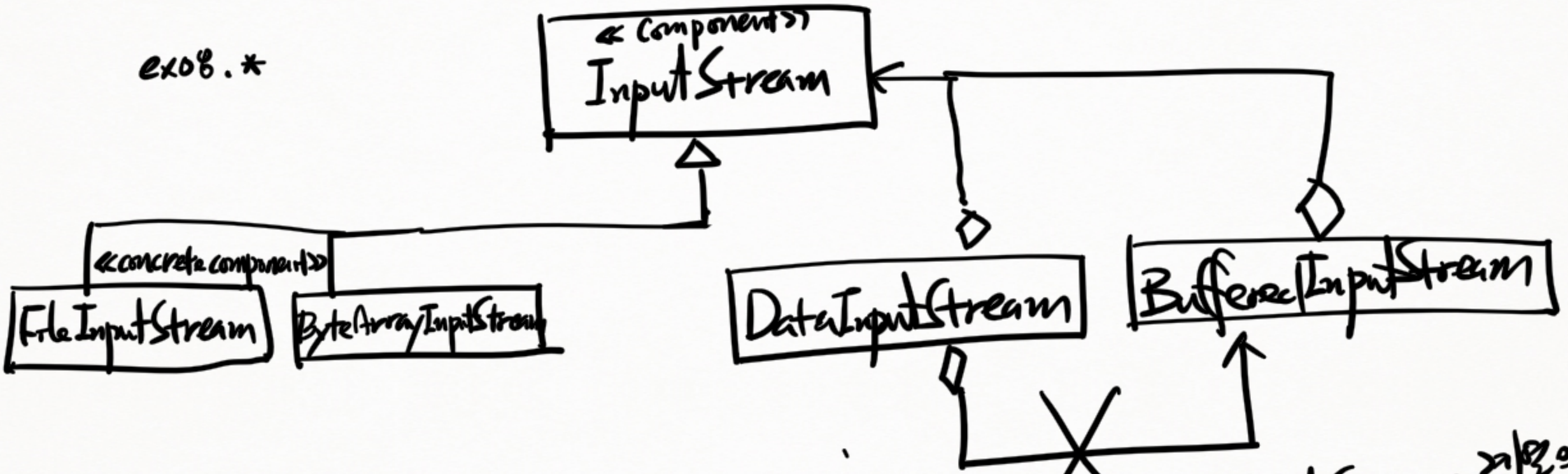
"완성품 블록"

예) 인형 집 바퀴 성

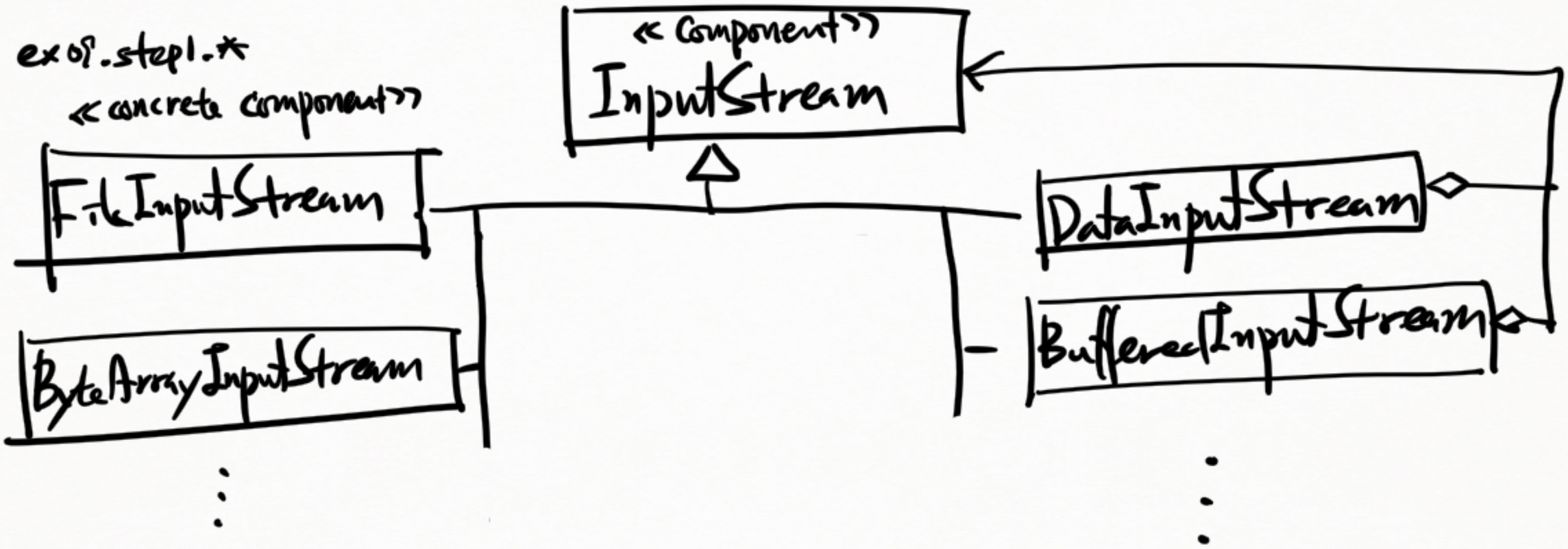


중간블록 예) 작은 콘센 사각 블록

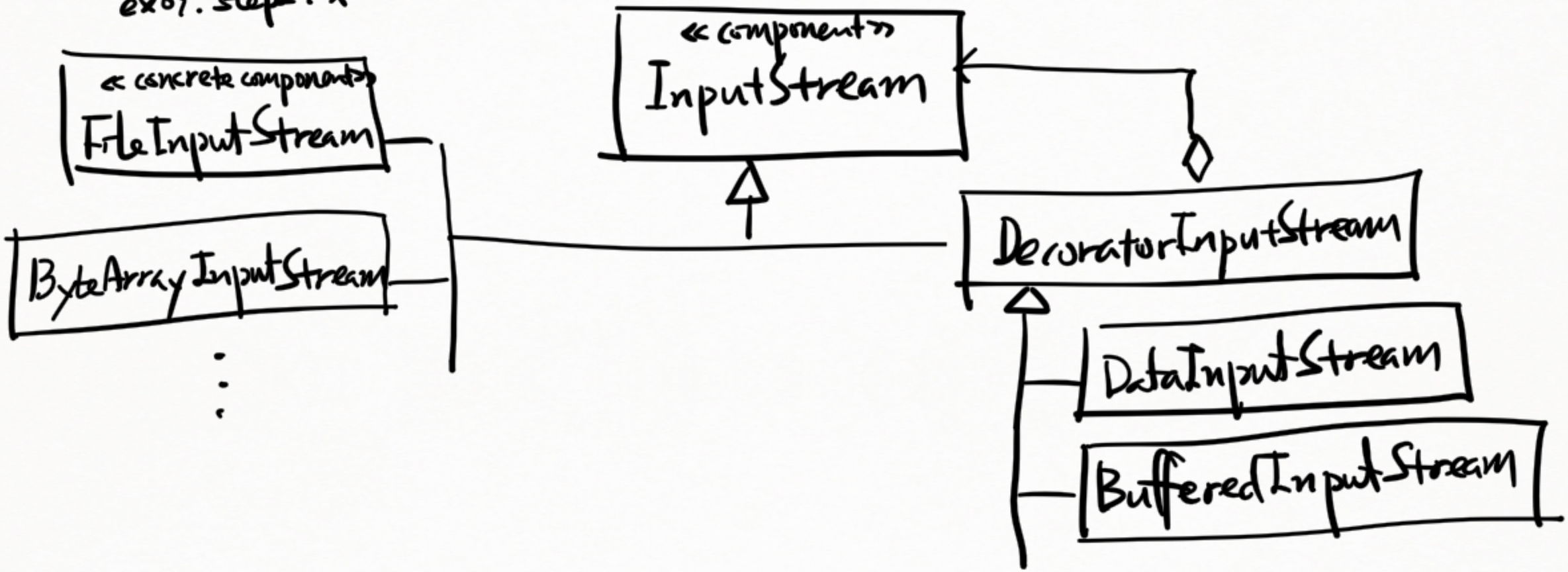
ex08.*

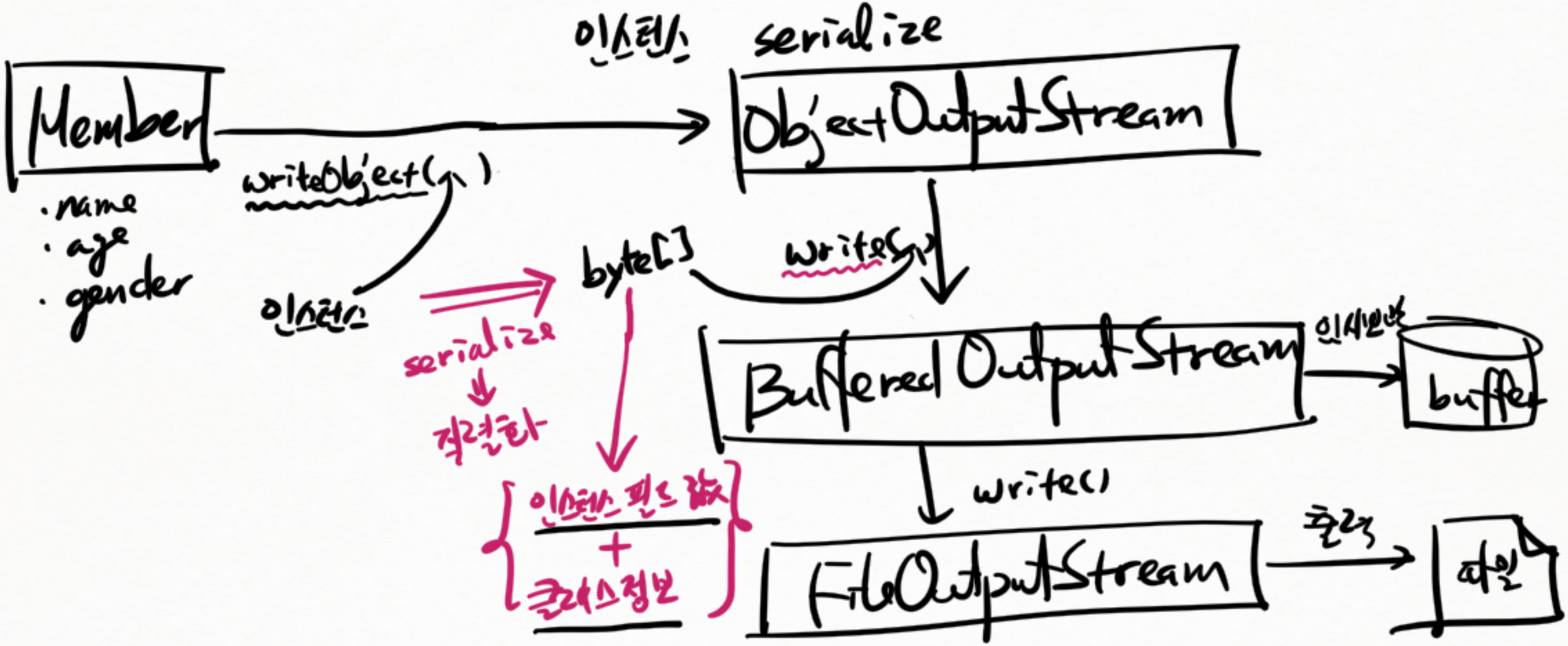


BufferedInputStream은 InputStream 기반의 객체가 아니다!
필요할까!


ex01.step1.*
«concrete component»
«Component»
InputStream
FileInputStream
ByteArrayInputStream
DataInputStream
BufferedInputStream

ex09.step2.*


«component»
InputStream
«concrete component»
FileInputStream
ByteArrayInputStream
DecoratorInputStream
DataInputStream
BufferedInputStream


인스턴스
serialize
Member
· name
· age
· gender
writeObject( )
인스턴스
ObjectOutputStream
byte[]
writes( )
serialize
직렬화
인스턴스 필드 값
+
클래스정보
BufferedOutputStream
임시보관
buffer
write()
FileOutputStream
출력
파일

(marshaling)

||

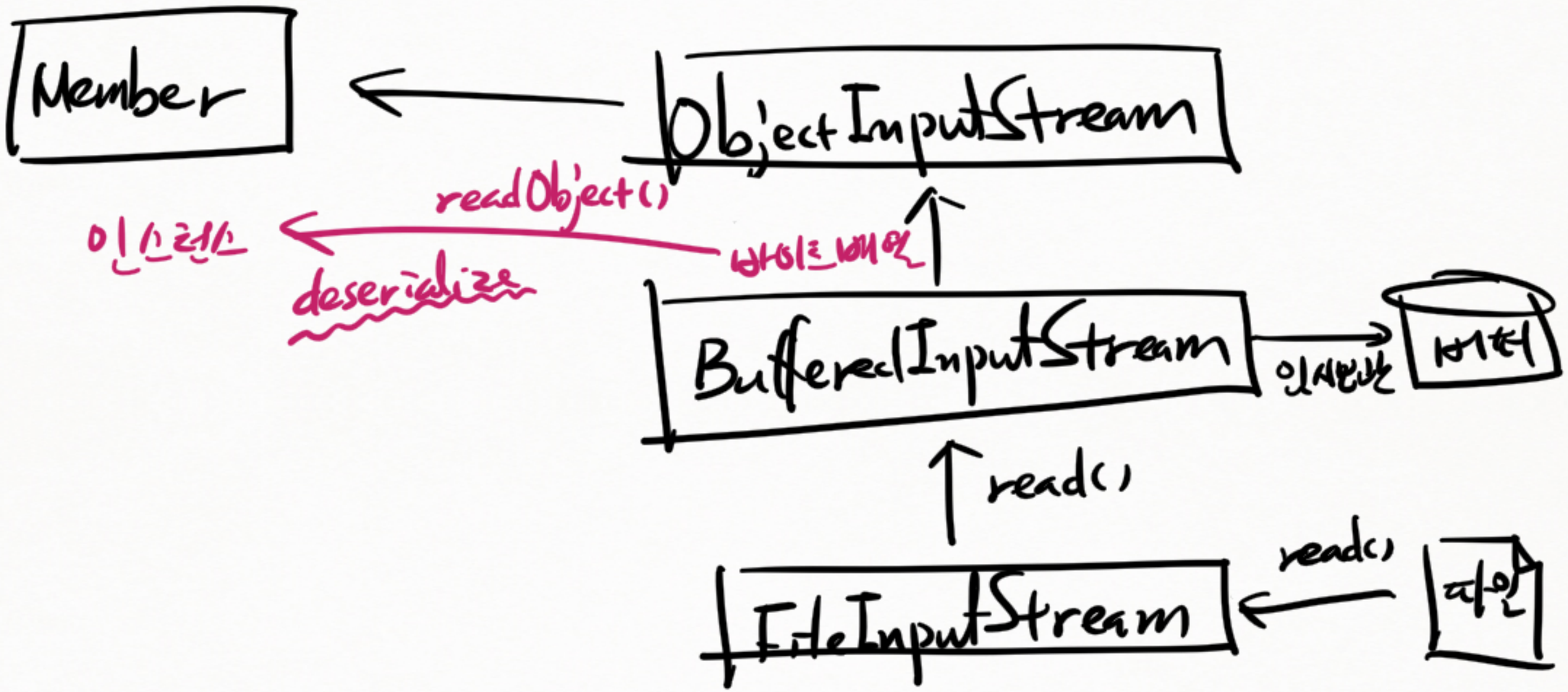
✓ serialize

인스턴스 ⇄ 바이트 배열

✓ deserialize

||

(unmarshaling)


Member
ObjectInputStream
readObject()
인스턴스
바이트배열
deserialize
BufferedInputStream
임시보관
버퍼
read()
FileInputStream
read()
파일

Origin

IDentifier

* serialVersionUID 변수를 선언하지 않으면 자동으로 추가되는, 이의 값을 찾는다 → 클래스가 변경되면 이 값도 바뀐다.

Member

writeObject()

ObjectOutputStream

{ · name
· age
· gender } + 클래스 고유번호 ⇓ serialVersionUID ← java.io.Serializable 구현 클래스인 경우

static long

· tel 추가

toString() 변경

파일 (클래스정보) 필드값

ObjectInputStream

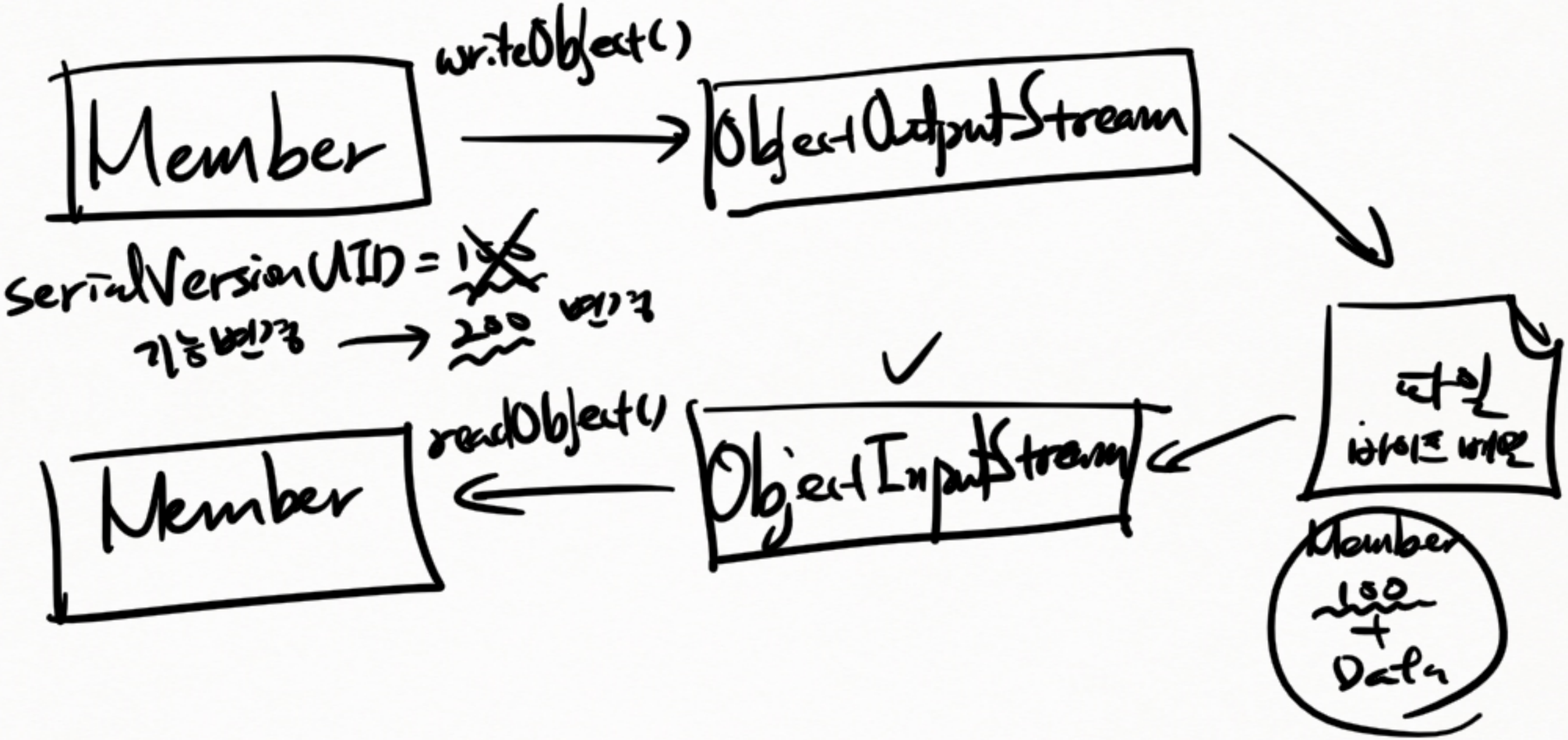
readObject()

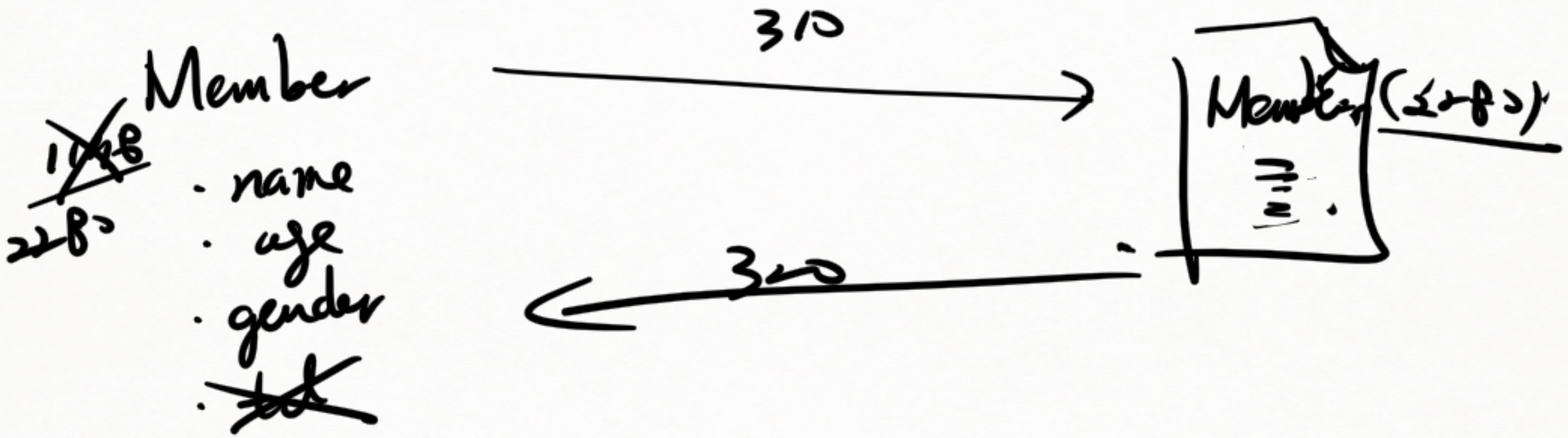
Member

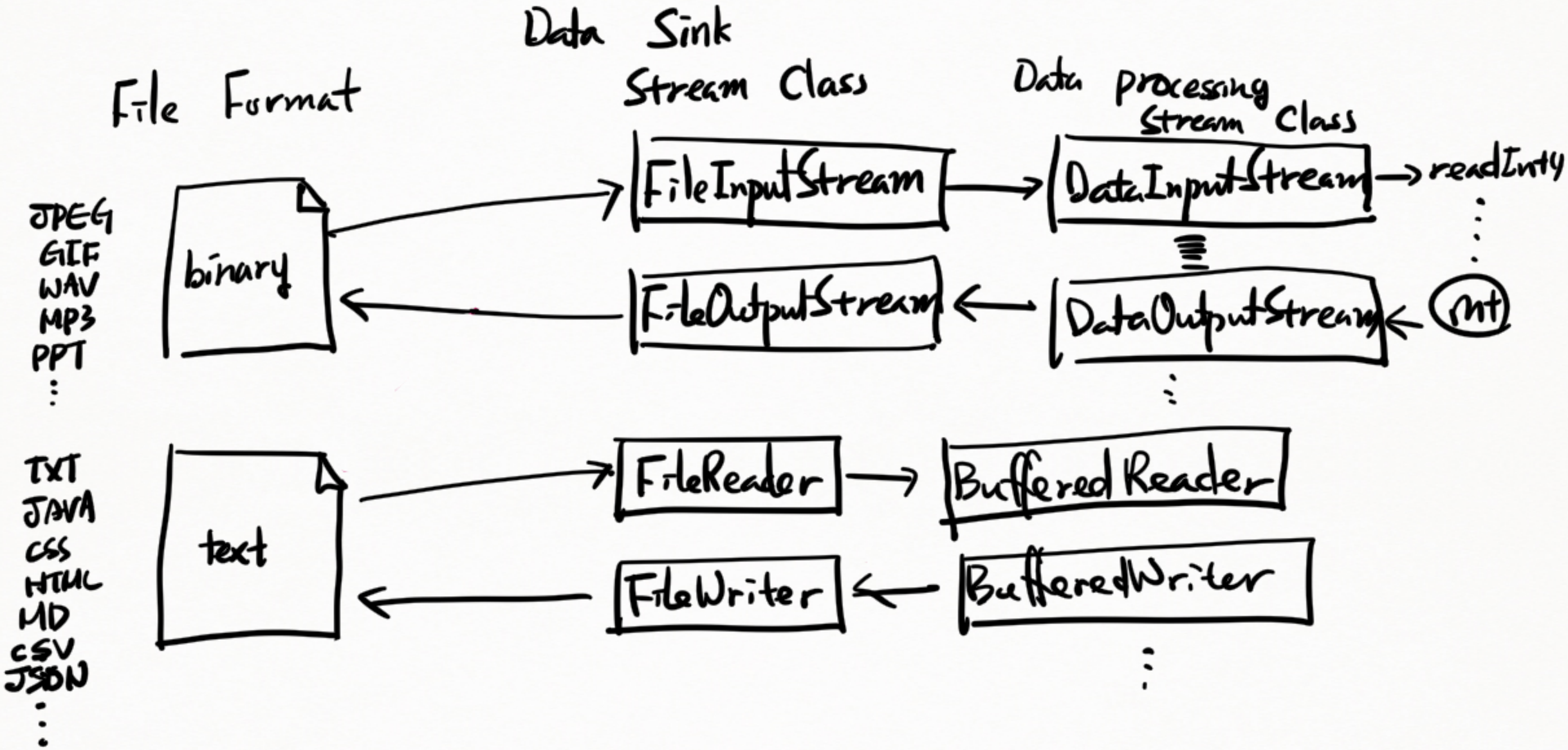
InvalidClassException

X

파일에 저장되어 있는 클래스의 serialVersionUID값과 현재 존재하는 클래스의 serialVersionUID 값이 같은지 비교한다.


Member
writeObject()
ObjectOutputStream
serialVersionUID = 100
기능변경 → 200 변경
파일
바이트 배열
Member
100
+
Data
Member
readObject()
ObjectInputStream


Member
~~10KB~~
22B
· name
· age
· gender
· ~~tel~~
310
320
Member
(22B)


Data Sink
File Format
Stream Class
Data processing Stream Class
JPEG
GIF
WAV
MP3
PPT
binary
FileInputStream
DataInputStream
readInt()
FileOutputStream
DataOutputStream
int
TXT
JAVA
CSS
HTML
MD
CSV
JSON
text
FileReader
BufferedReader
FileWriter
BufferedWriter

- 30-a : FileInputStream / FileOutputStream
- 30-b : DataInputStream / DataOutput/Stream
- 30-c : BufferedInputStream / BufferedOutputStream

} file format을 알아야 읽고 쓸 수 있다

- 30-d : ObjectInputStream / ObjectOutputStream ← 다른 언어와 호환이 안된다

30-e : 리팩터링

31-a : FileReader / FileWriter text 파일 포맷

31-b : BufferedReader / BufferedWriter

31-c : 리팩토링 I

31-d : 리팩토링 II

4byte

2byte 정수

0111111 1111111 = 32767

FFFF

[글개수: 4][번호][제목][2][내용][2][작성자][작성일:10][조회수:4]

int a = 65535;

write(a >> 8);

00 00 FF FF

write(a);

00 00 FF

00 00 00 00 FF FF FF FF

F F F F

목표는

[FF FF] [...] 00 00 00 00 FF

목표대로 mk

00000000 00000000

00 00 FF FF = 65535

65535

65535-자

65535-16비트

2byte

0

XML
HTML

(일반)tag = markup (출판계)

↑

data를 설명 (meta-data)

끝태그

<no> data </no>

시작태그

컨텐트

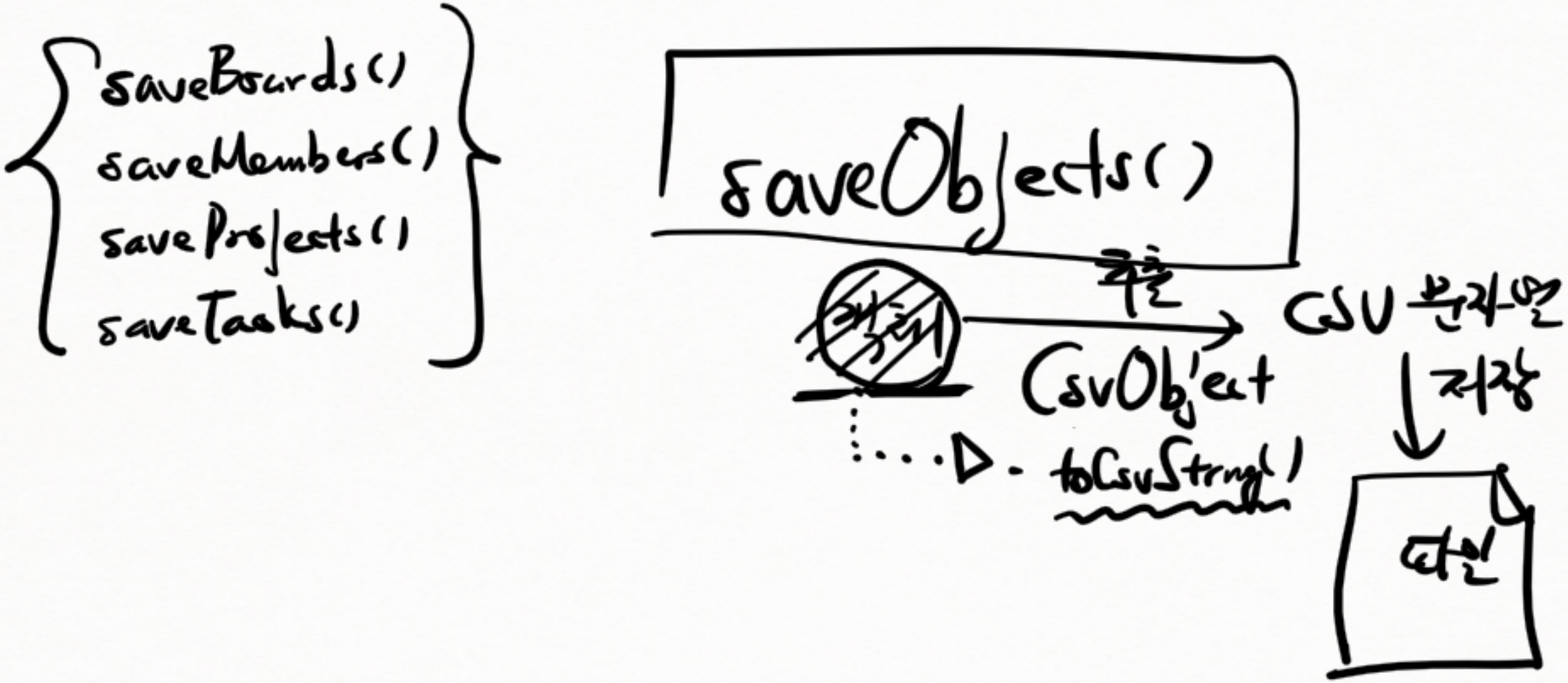
↑

필요한 Data

element (개발자)

<h1> 제목 </h1>

<b> aaa </a>


saveBoards()
saveMembers()
saveProjects()
saveTasks()
saveObjects()
추출
CsvObject
toCsvString()
CSV 문자열
저장
파일

· Data 교환

- Text
- Data 구조

↓

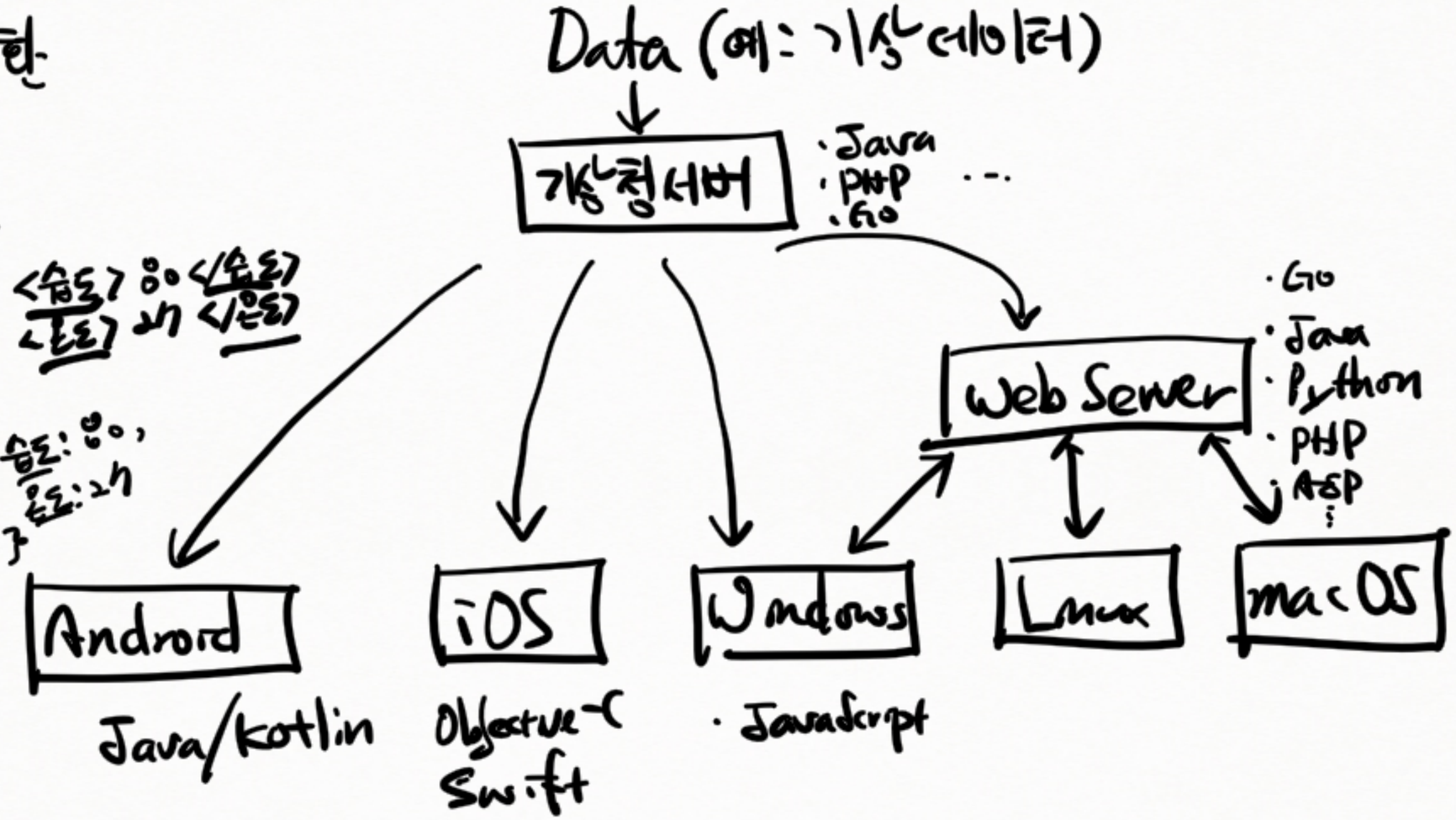
✓ XML

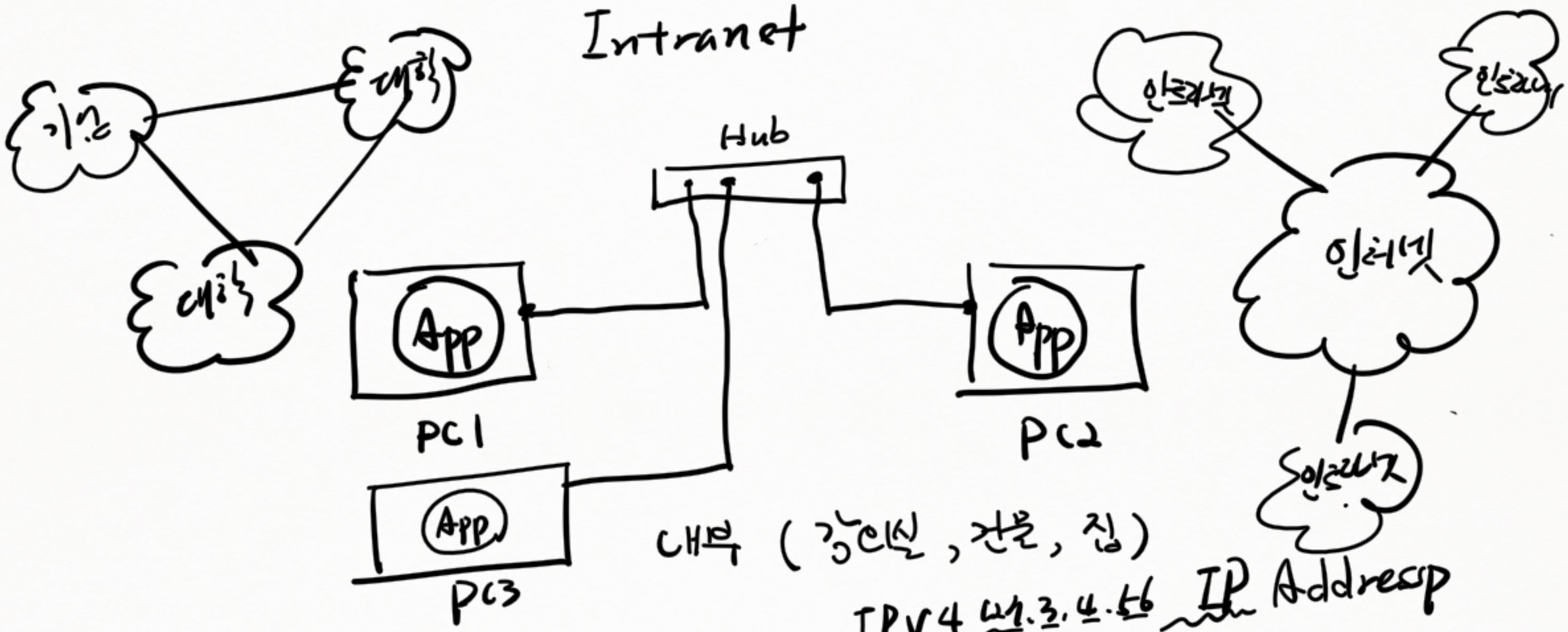
✓ JSON

JavaScript
Object
Notation

<습도> 80 </습도>
<온도> 27 </온도>

{
습도: 80,
온도: 27
}





내부 (강의실, 건물, 집)

IPv4 255.3.4.56 IP Address

IPv6

0

약 42억

010-1234-4444

0.0.0.0 = 4byte

255.255.255.255

1byte 1byte

127.354.117

221.522.143

-128~127

0~255

14.12.43.11

0D.34.2B.37

02001-122123x

01234...

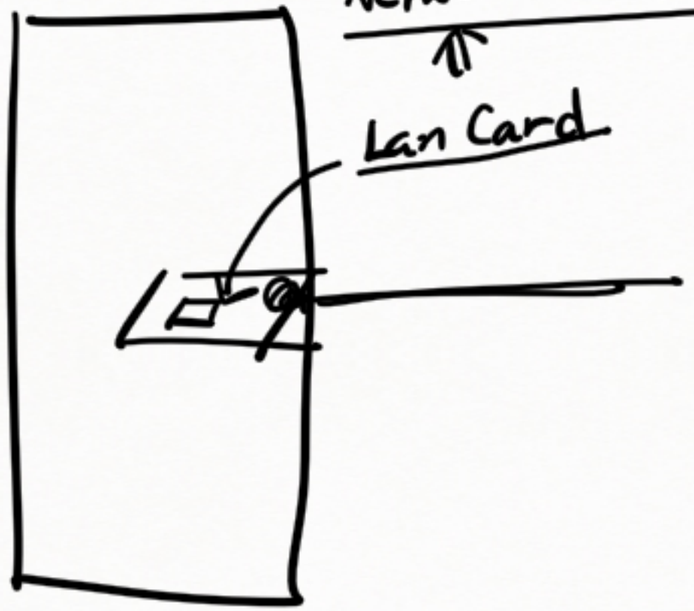
221522143 423

v4

IP Address

4Byte = 약 42억 대의 컴퓨터

컴퓨터



(NIC)

Network Interface Controller

Lan Card

Lan Card 고유번호

=

NIC Address

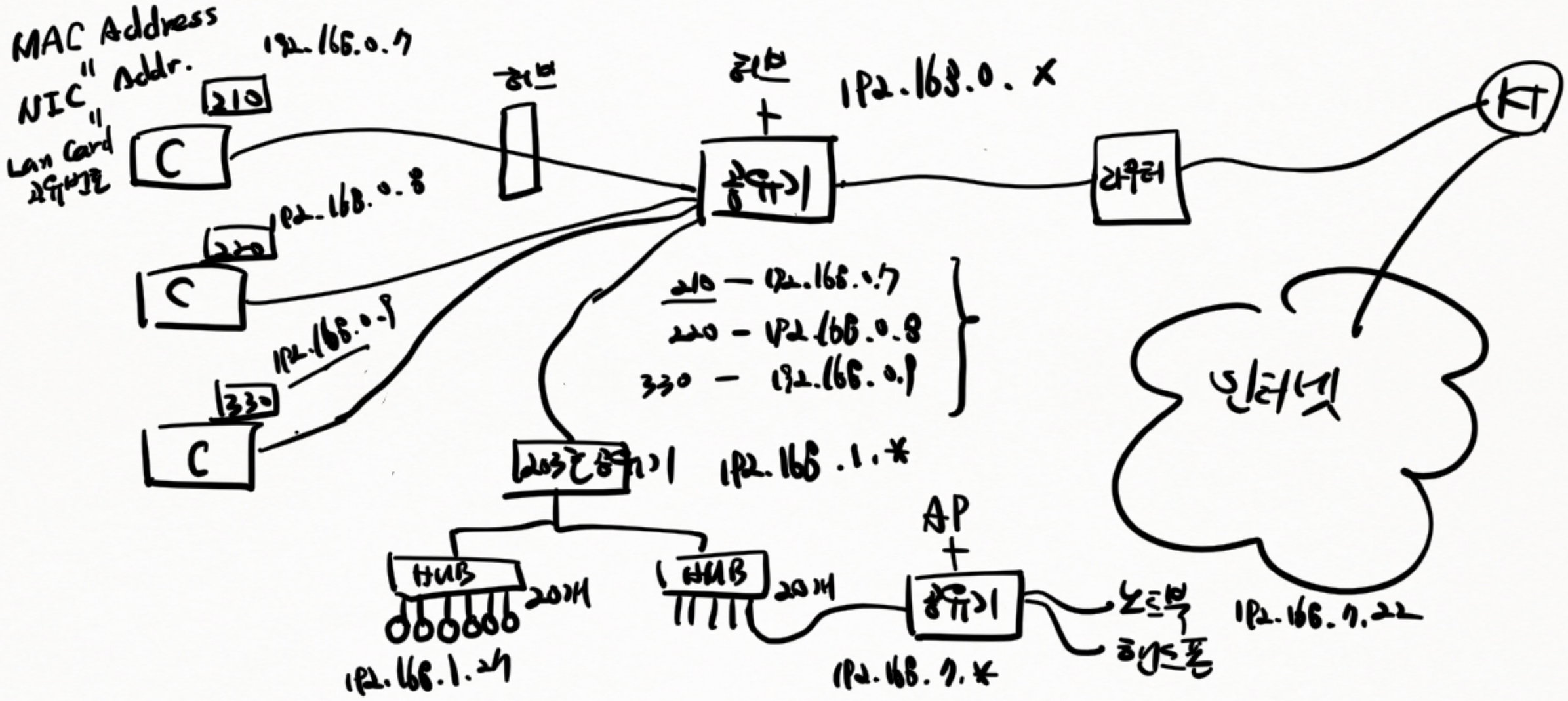
⇓

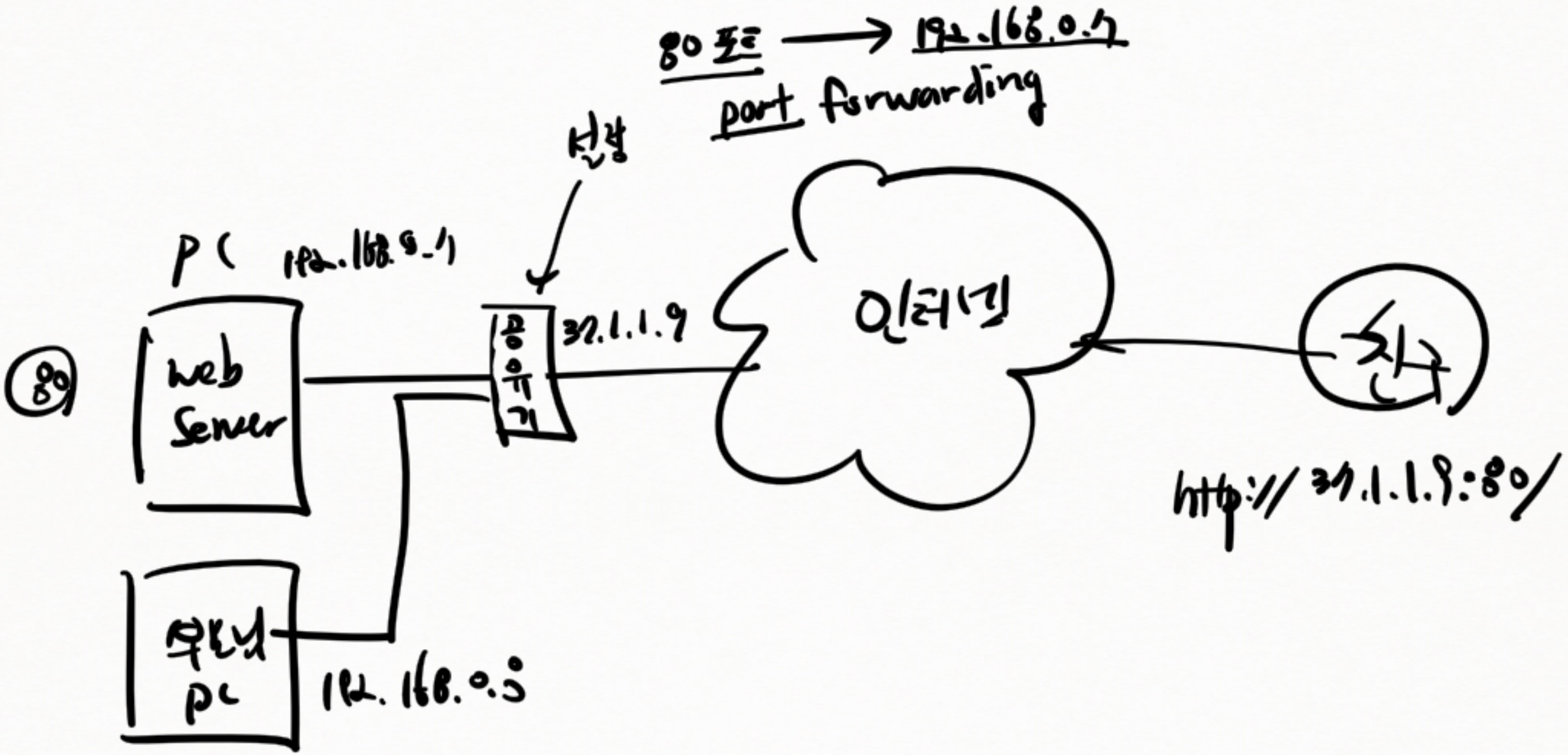
MAC Address

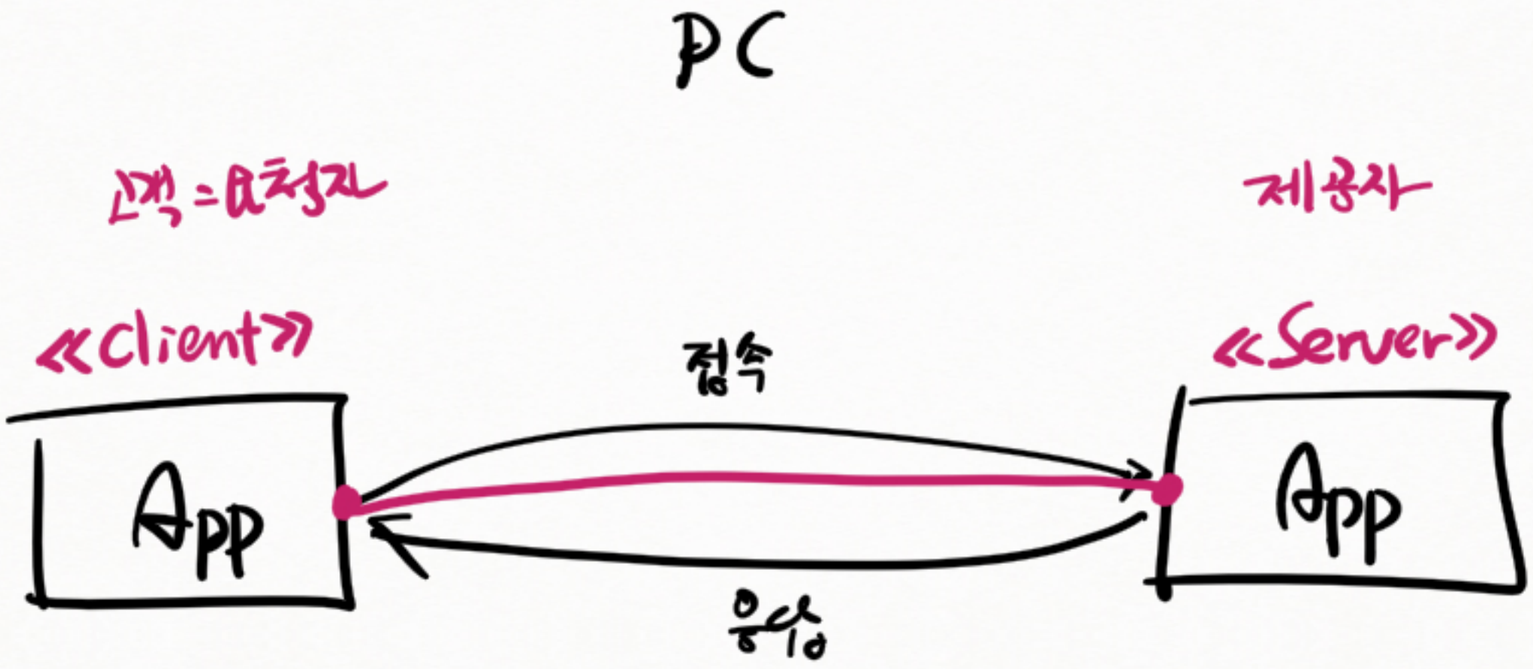
Media

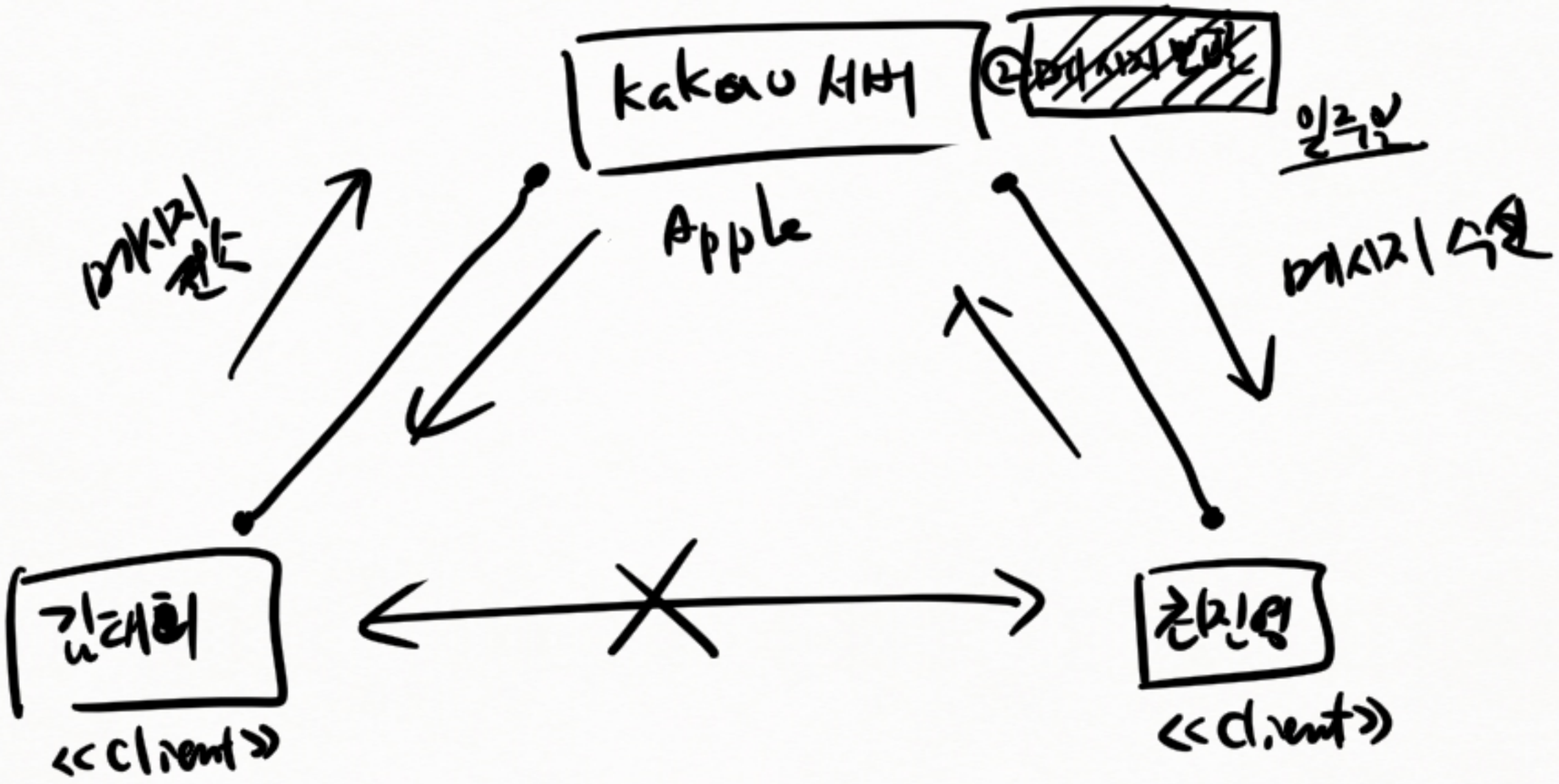
Access

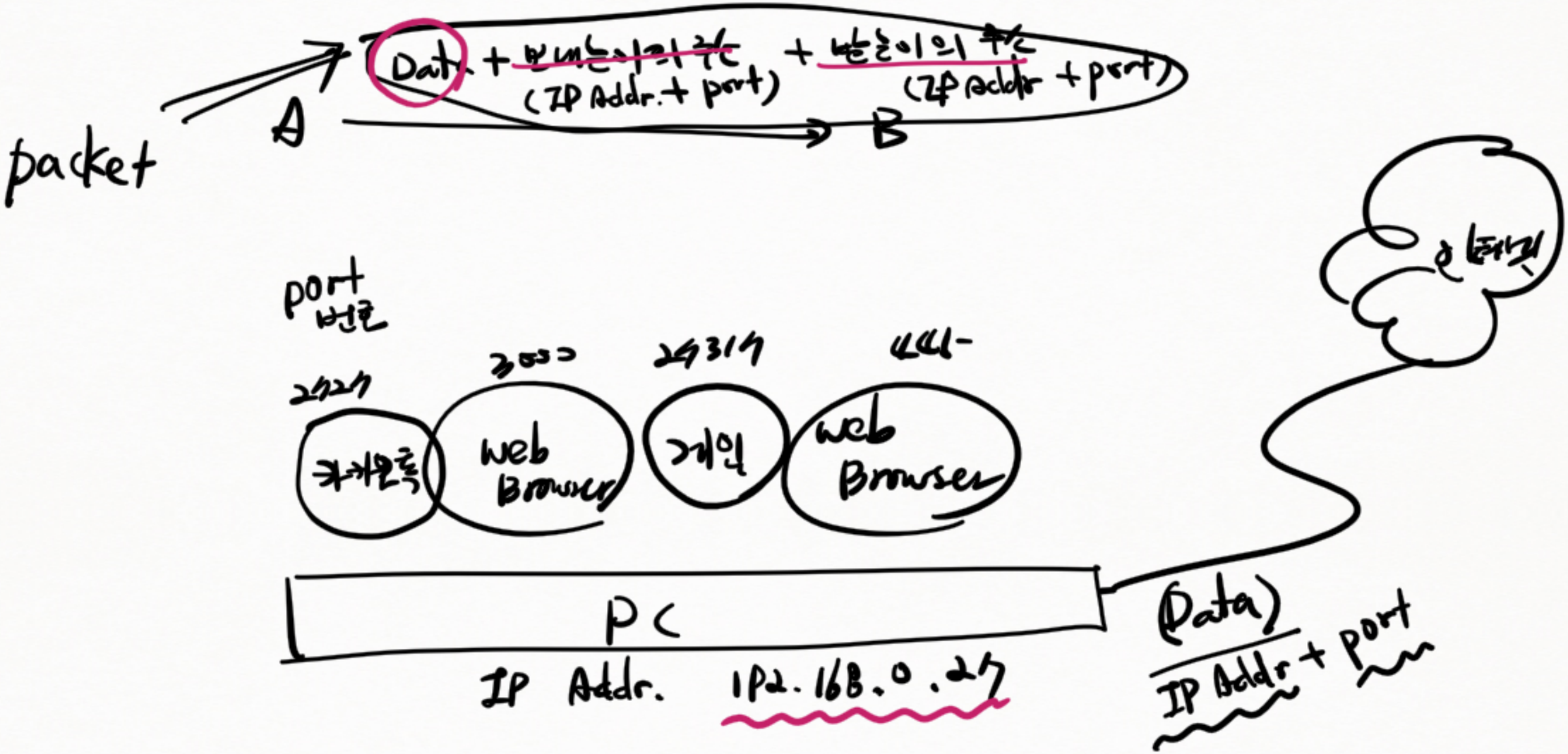
Control

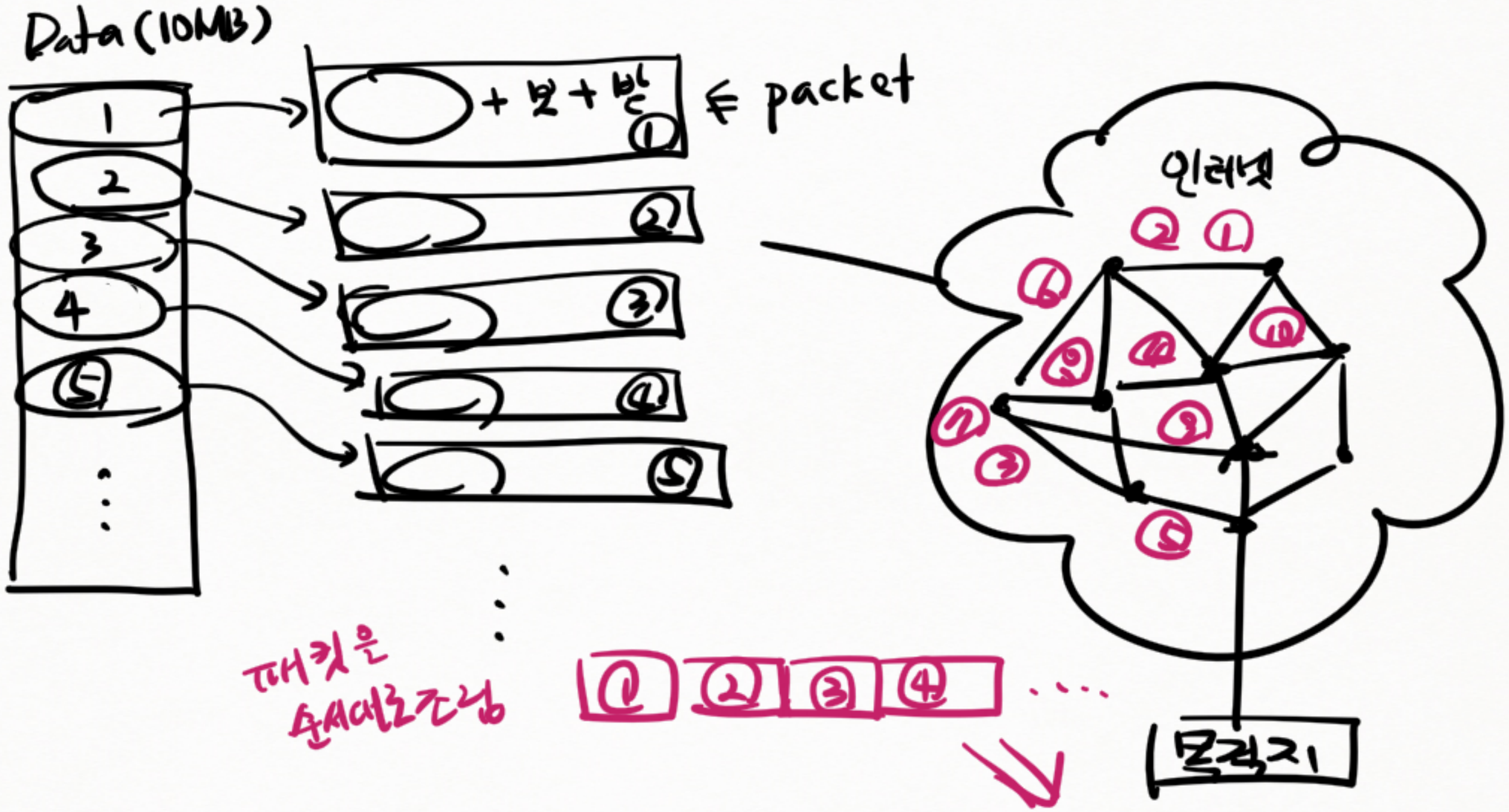

MAC Address
NIC " Addr.
Lan Card 고유번호
192.168.0.7
210
C
허브
공유기
192.168.0.X
라우터
KT
192.168.0.8
220
C
192.168.0.9
330
C
210 — 192.168.0.7
220 — 192.168.0.8
330 — 192.168.0.9
인터넷
203호 공유기
192.168.1.*
HUB
20개
192.168.1.21
HUB
20개
AP
공유기
192.168.7.*
노트북
핸드폰
192.168.7.22

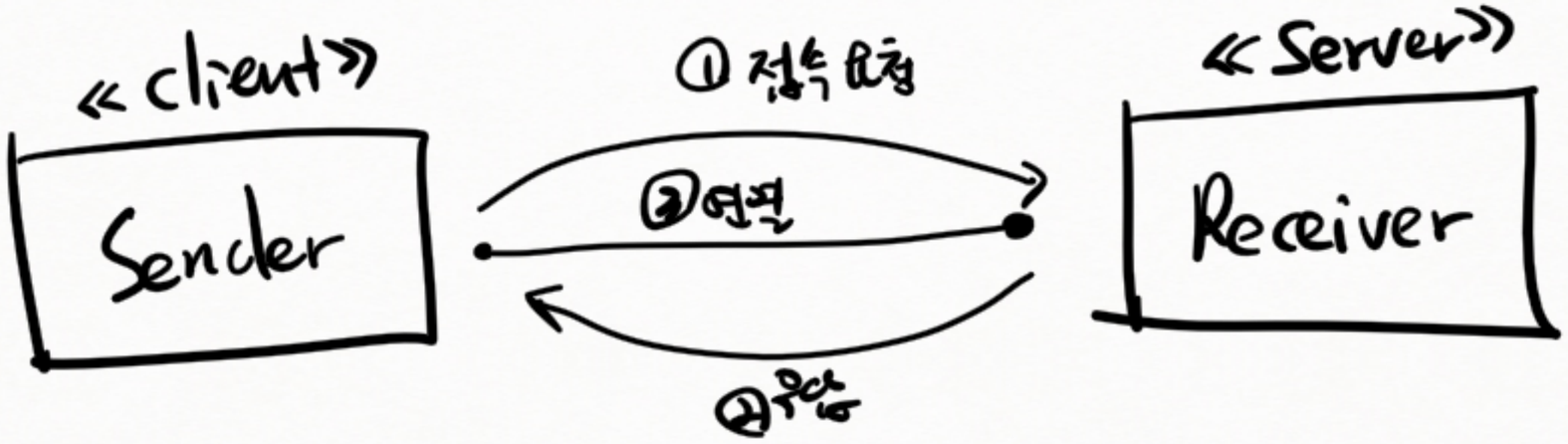

80 포트 → 192.168.0.7
port forwarding
선생
PC 192.168.0.7
80
web Server
공유기
37.1.1.9
인터넷
친구
http://37.1.1.9:80/
무선 PC
192.168.0.3


PC
고객 = 요청자
제공자
«Client»
«Server»
접속
App
App
응답


kakao 서버
②메시지 보관
Apple
일주일
메시지 수신
메시지 전달
김대희
<<client>>
천진영
<<client>>


packet
A
Data + 보내는이의 주소 (IP Addr. + port) + 받는이의 주소 (IP Addr + port)
B
port 번호
2323
카카오톡
3000
web Browser
2431
게임
4444
web Browser
PC
IP Addr. 192.168.0.27
(Data)
IP Addr + port


Data (10MB)
1
2
3
4
5
+ 보 + 받
packet
인터넷
목적지
패킷을 순서대로 조립

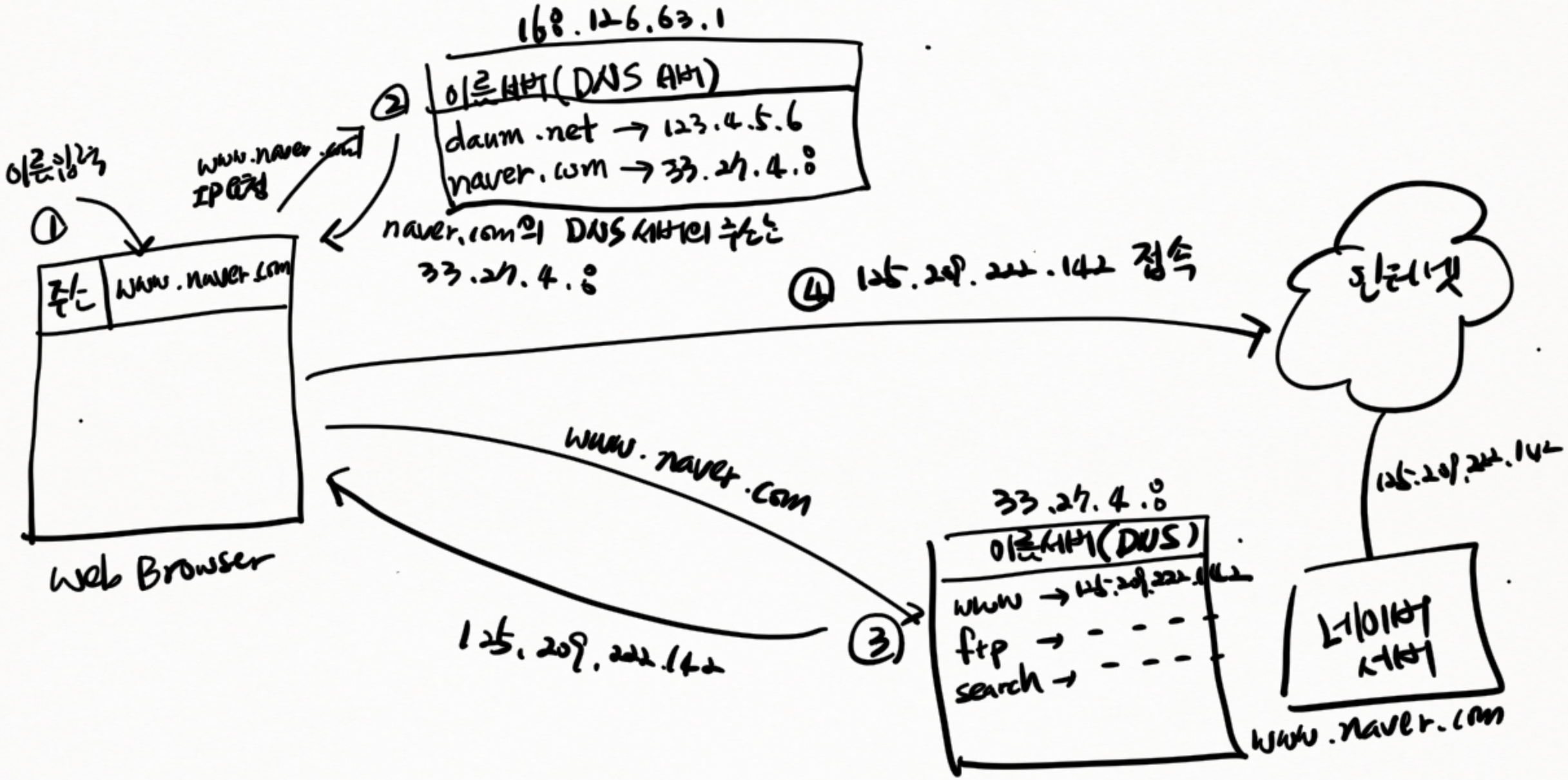

«client»
Sender
① 접속 요청
② 연결
③ 수락
«Server»
Receiver

127.0.0.1 ⇒ 현재컴퓨터

new Socket(IP Address , port)

Domain Address

localhost ⇒ 현재 컴퓨터

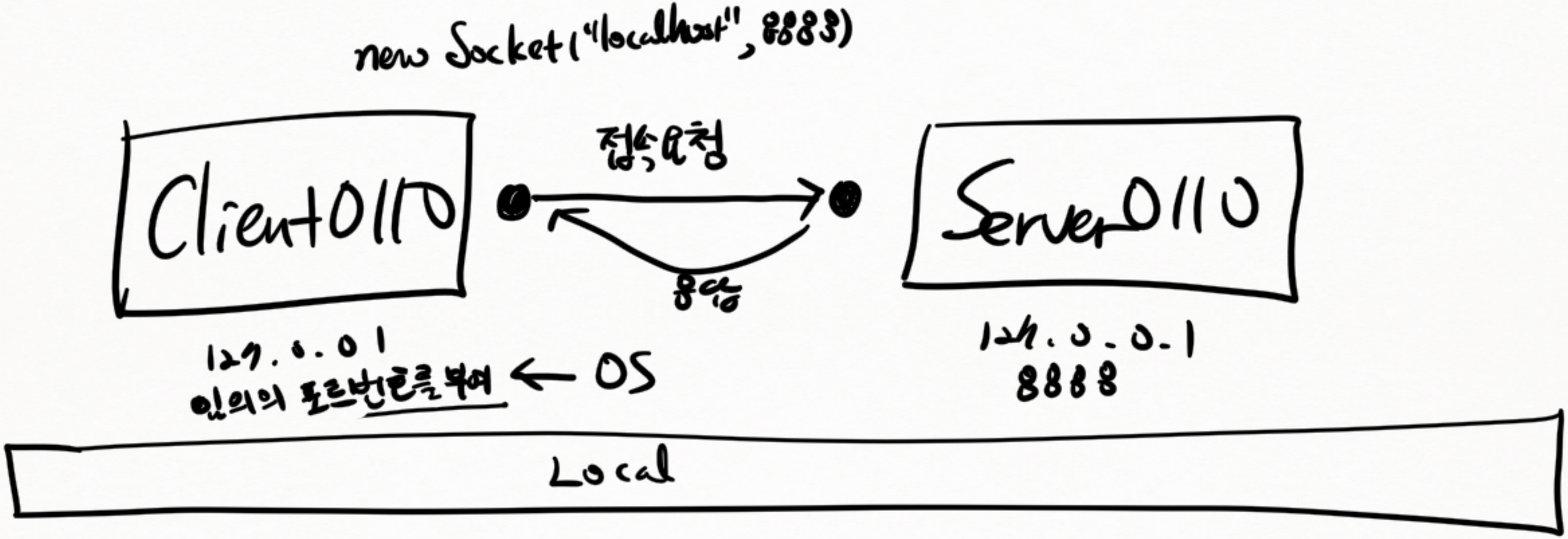

168.126.63.1
이름서버(DNS 서버)
daum.net → 123.4.5.6
naver.com → 33.27.4.0
② www.naver.com IP 요청
이름입력
①
주소
www.naver.com
naver.com의 DNS 서버의 주소는 33.27.4.0
④ 125.209.222.142 접속
인터넷
Web Browser
www.naver.com
125.209.222.142
33.27.4.0
이름서버(DNS)
www → 125.209.222.142
ftp → - - -
search → - - -
③
125.209.222.142
네이버 서버
www.naver.com

도메인 이름
서버이름

www.naver.com

Host name

Domain name


new Socket("localhost", 8888)
Client0110
접속요청
응답
Server0110
127.0.0.1
임의의 포트번호를 부여
← OS
127.0.0.1
8888
Local