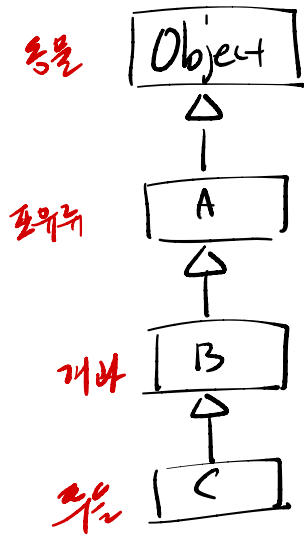


\* Object 클래스의 instanceof



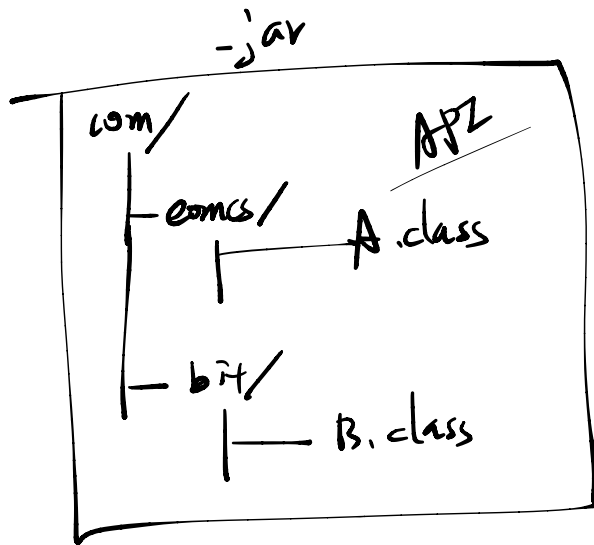
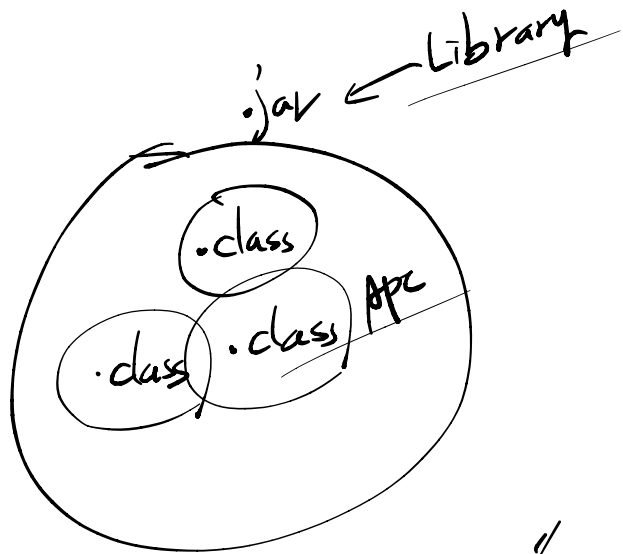
obj = new 클래스();

obj instanceof 클래스

obj " true

obj " false

obj " true



"Java Archive"

JAR  
↳ 021

# \* Hash Value

데이터

1 byte ~

Hash  
(algorithm)

수학적 공식에 따라  
계산을 수행

연산

32bit 값  
64bit 값

128bit 값

256bit 값

512bit 값

1024bit 값

⋮

\* 다른 데이터가  
같은 값을 가질 수 있다

MD4

MD5

SHA

PGP

CRC

⋮

{ A  
B }

100  
해시값

⇒ 연산량이 줄수록 같은 값을 가질 확률이 줄어든다.  
(복잡하리, 변환속도 느리고, 결과값이 커진다)

# \* Hash Value

데이터

Hash  
값

사람

간단 = 정확도↓ = 생성시간 빠르다

복잡 = 정확도↑ = 생성시간 느리다. 모든 비트 검사 ← 너무 느리다, 현실성 없다.

디지털  
서명

원본 데이터인지  
검사하기 더 쉬운 빠르다.

"디지털 지문"

지문

더 간단, 통계보다 정확성↓

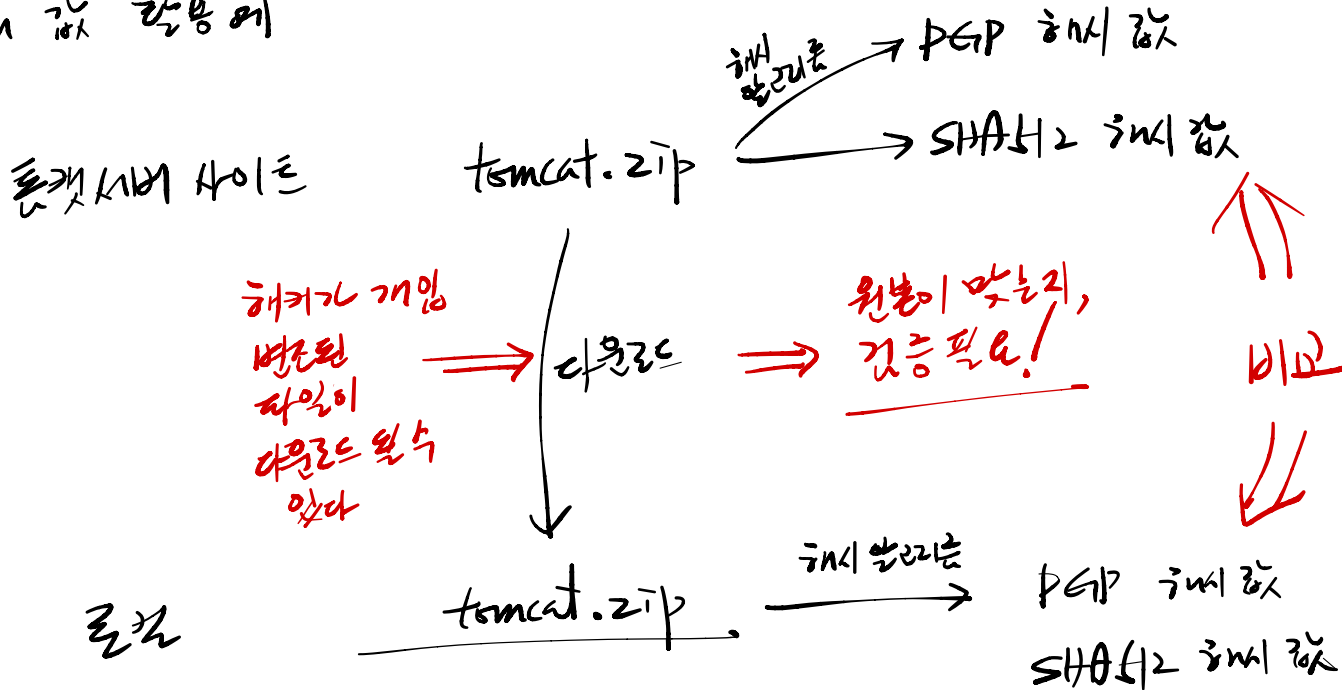
홍채

유전자를 검사하는  
정보를  
보이는  
정보를  
더 쉬운 빠르다

유전자

← 너무 느리다, 현실성 없다

\* Hash 값 활용 예



하하

A ok.avi - 1212 →

B ok.avi - 1212 →

C yes.avi - 1212 →

D ~~a. 1212~~

가장 많은 .avi = 1212

{ A의 IP주소  
B의 IP주소  
C의 IP주소  
D의 IP주소 }

my <sup>이름</sup> → A  
→ B  
→ C

A | C | B | A | C