

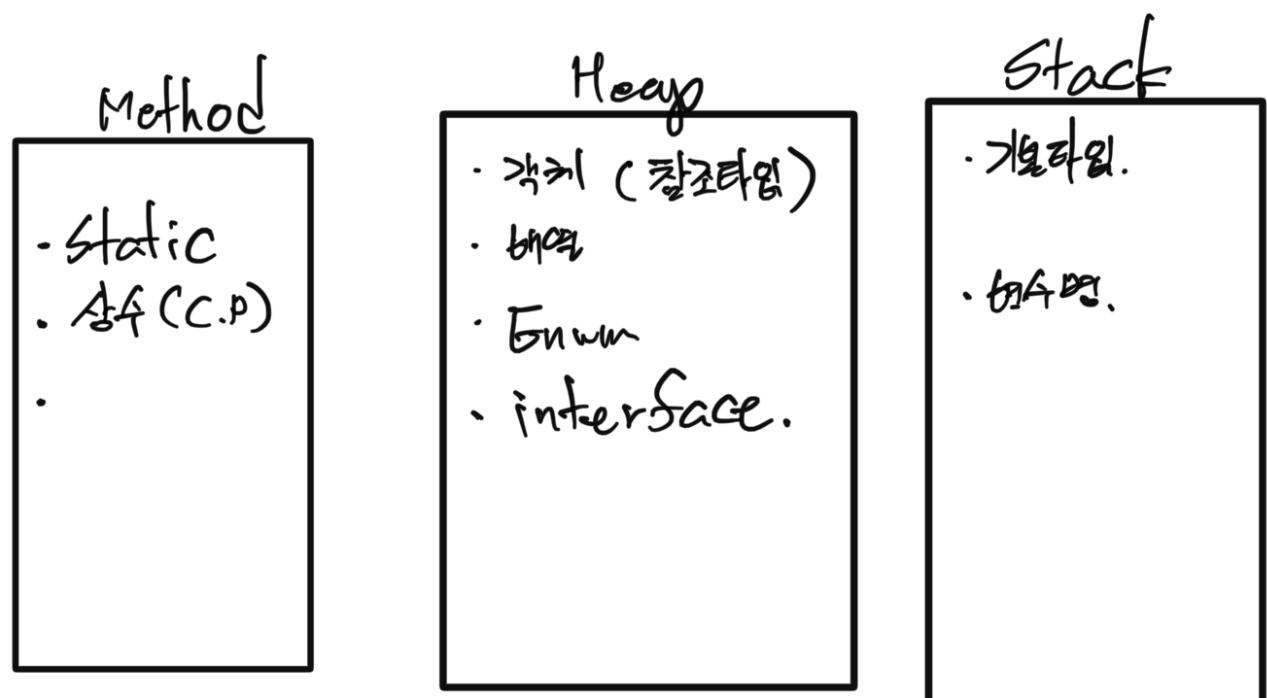
○ 컴퓨터 2진수 표현. 소수는 나누기로 표현. 그러나 소수점이 난다.

	1 byte	2 byte	4 byte	8 byte
논리	boolean			
문자		char		
정수	byte	short	int	long
실수			float	double
	8 bit	16 bit	32 bit	64 bit

1자리
 $2^1 < 1 < 2^3$

3자리
 $2^1 < 3 < 2^2$

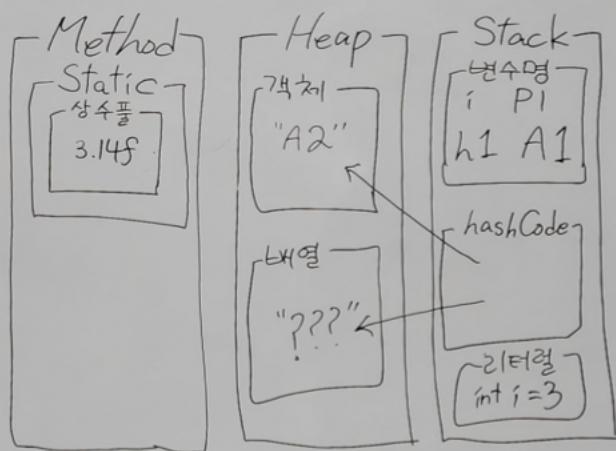
○ 양수는 0을 포함.



○ 상수는 final static으로 선언.

☞ 2번째 = 같.

```
int i = 3;  
final static float PI = 3.14f;  
String h1 = "???"  
String A1 = new String("A2");
```



- >/ 논리연산자

= 대입연산자

== 비교연산자.

&& and

|| or

/ 나누기는 예외다.

- 삼항연산자.

조건문? $\frac{\text{True} : \text{false}}{\boxed{\quad}}$

다시 삼항연산자 쓸 수 있음.

* if문을 삼항으로 바꾸는 연습.

조건문? $\frac{\text{조건문? } T : F}{T}$: $\frac{\text{조건문? } T : F}{F}$

- For & While

for(조건) { 실행문 } 조건이 숫자

while(조건) { 실행문 } 조건이 boolean.

↳ false가 되면 까지 돌아가.

: If () {

} else {

} else {

 }

◦ switch (조건문)

```
case(조건1) { 실행문 } break;  
case (조건2) {            }     "  
case (조건3) {            }     "  
case (조건4) {            } ;
```

* 해당 조건에서 시작해서 성립되면 break에서 종료

◦ 반복문 for문

◦ Enum 오일. 룰. 성능과 같이 초기화하지 않는 값을
 상수로 저장하는 것. 객체에 해당.
 객체를 생성할 수 있다.

◦ Class

<u>public</u>	<u>static</u>	<u>void</u>	<u>main</u> .
전역	메소드안에서	리턴 타입	
Global	쓰기내용	없다	

◦ 접근 제한자

◦ public 누구나 접근.

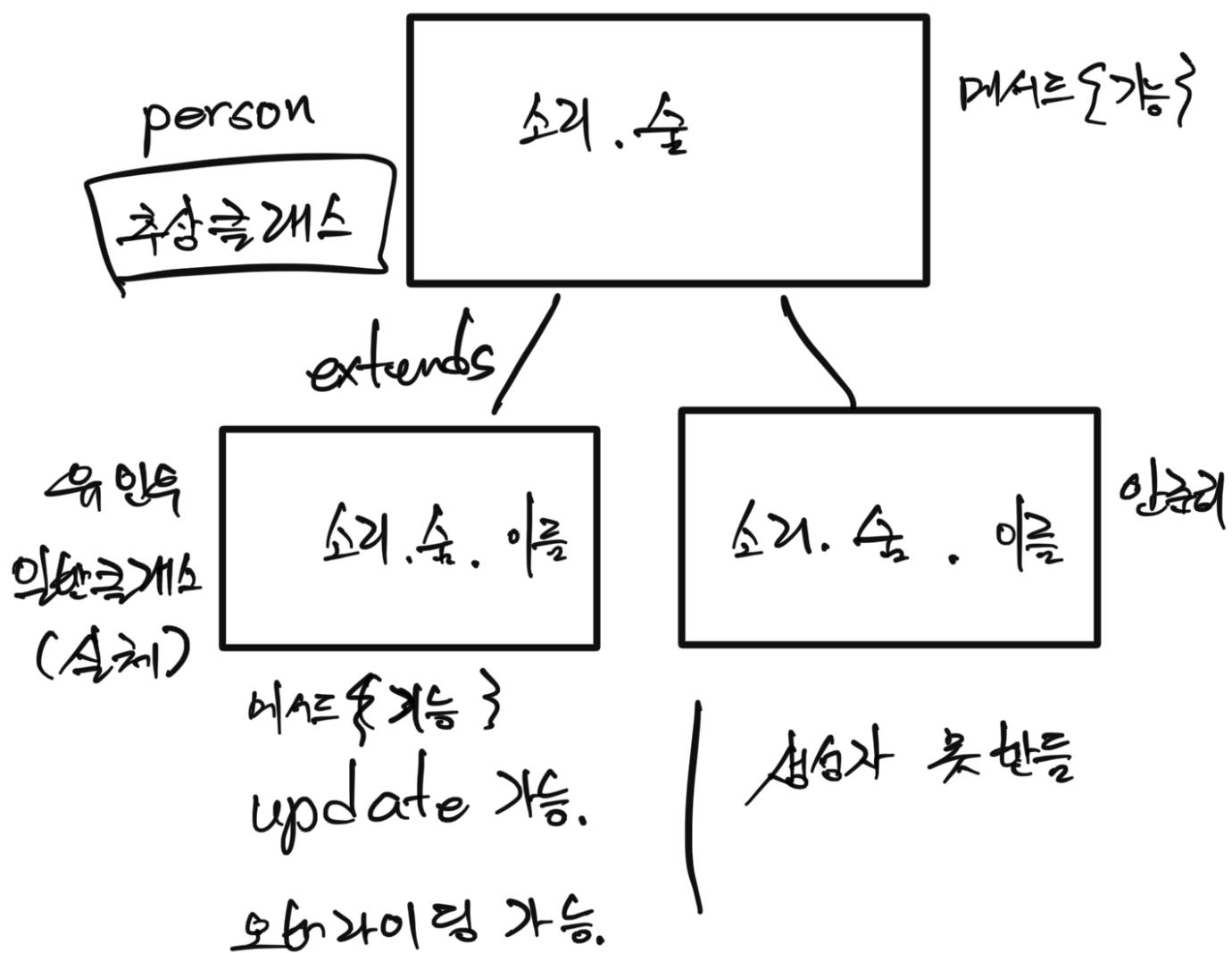
◦ protected 패키지 + 상속

private --

- `default` 패키지 내에서 접근 (외부 접근x)
- `private` 같은 패키지 내에서만 쓰는 0~ 0~

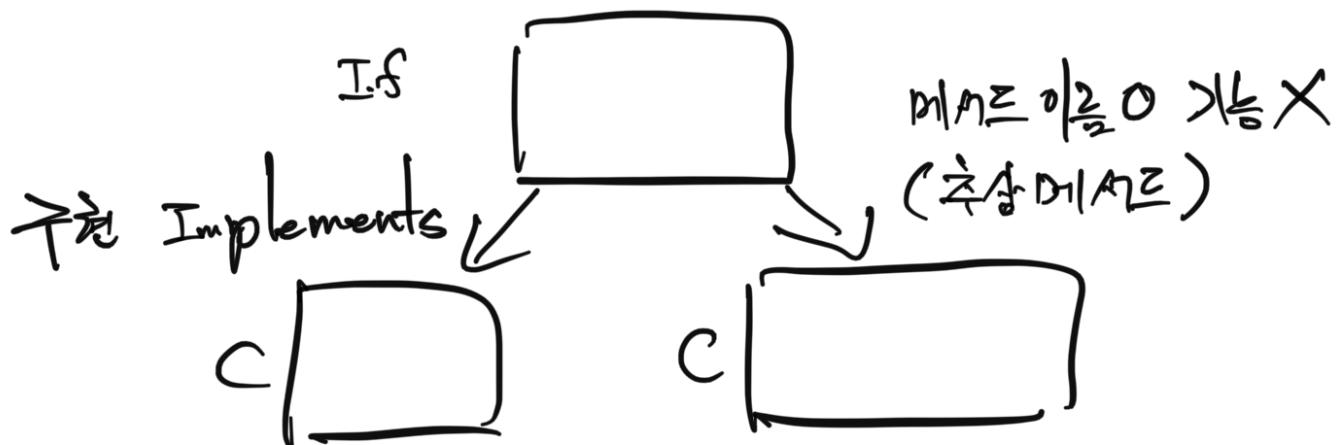


- 추상클래스. ↗ 부조건 사용.
- `Abstract`. ↗ 안에 반드시 1개 이상의 추상 메서드 보유

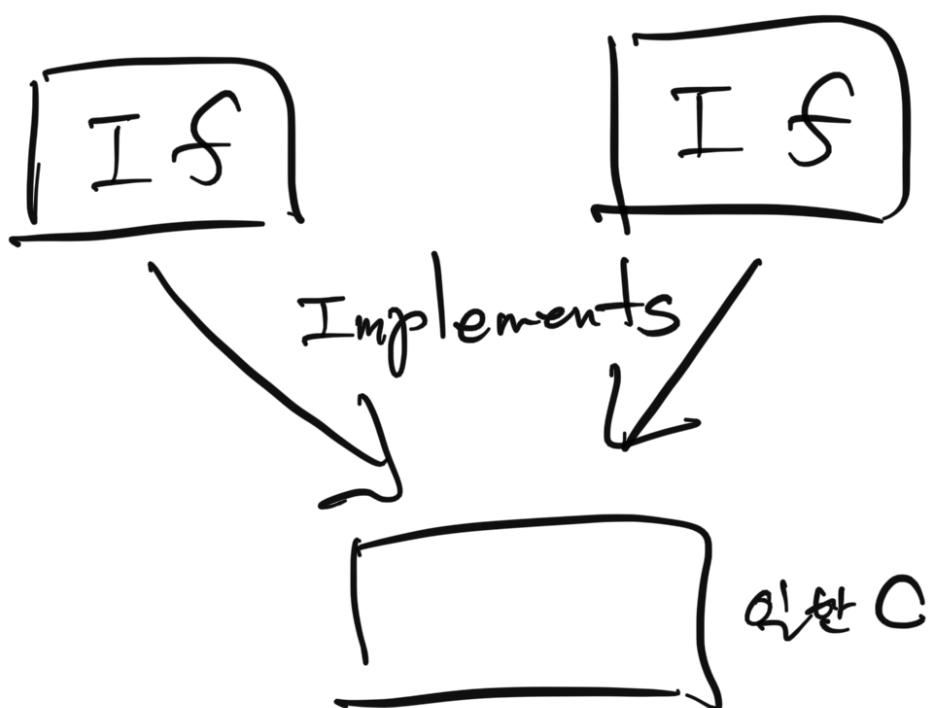


but. Interface는 상황은 X.

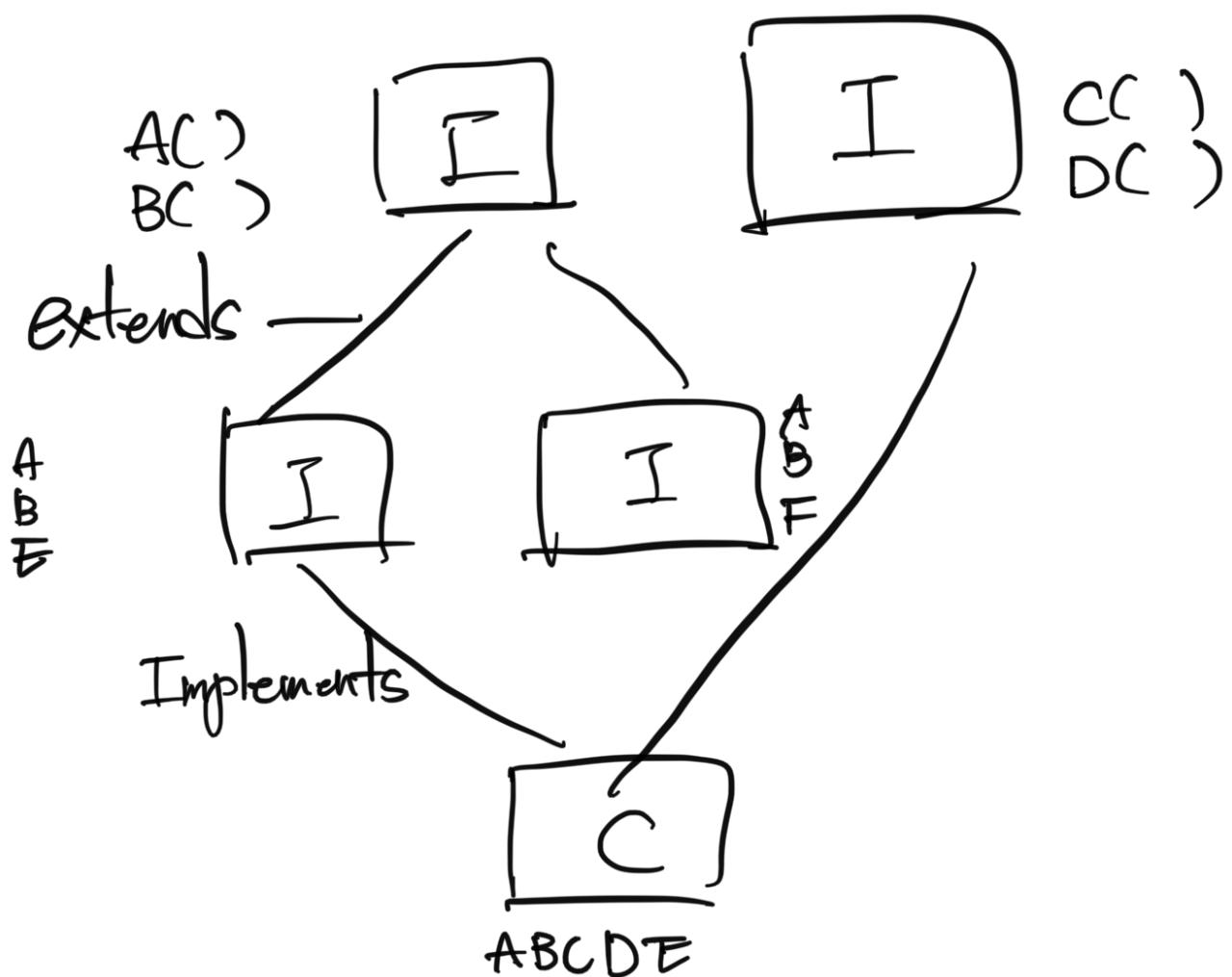
→ 가능이 없기 때문에



- 일반클래스는 단지에서 적용된 메서드의 가능을 제한의 해야함.
 - 메서드 추가 가능.
 - 생성자 못 만든다.



Interface 와 구현상속



부모	자식	Interface	class
Interface	extends	Implement	
class	X	extends	

• 익명구현객체 Why?



→ Why? 산속 같은 클래스는 Interface를
받을 수 없다.

→ New 생성자와 tf의 메서드 빙과 가능 재정의.

이란 A = New A!간 ?

<답글>

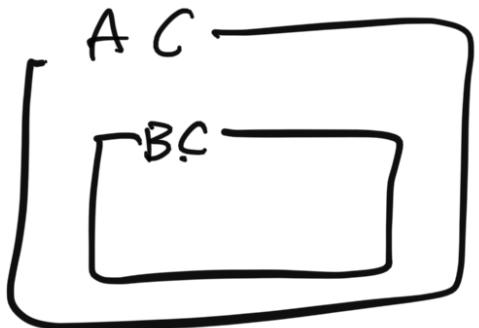
- 외부에서 접근 못하도록 private 설정
- 정적 메서드로 static A요.
- 필드/생성자는 private
- 메소드는 public

디자인 패턴(Design Pattern)이란?

디자인 패턴이란 기존 환경 내에서 반복적으로 일어나는 문제들을 어떻게 풀어나갈 것인가에 대한 일종의 솔루션 같은 것입니다. 디자인 패턴 계의 교과서로 불리는 [\[GoF의 디자인패턴\]](#)에서는 객체지향적 디자인 패턴의 카테고리를 **"생성 패턴 (Creational Pattern)"**, **"구조 패턴 (Structural Pattern)"**, **"행동 패턴 (Behavioral Pattern)"** 3가지로 구분하고 있습니다.

생성(Creational) 패턴	구조(Structural) 패턴	행동(Behavioral) 패턴
Singleton	Adapter	Command
Abstract Factory	Composite	Interpreter
Factory Method	Decorator	Iterator
Builder	Facade	Mediator
Prototype	Flyweight	Memento
	Proxy	Observer
		State
		Strategy
		Template Method

중첩 예외.



데이터 B는 AC에서 사용 X.
데이터 A는 BC에서 사용 O

예외 처리.

- 전략적 예외.

- 자동예약어 : 자바가 이미 정해 놓은 키워드.

① 예외 혹은 try 및 catch 처리 (실행) exception이 가능성을 가져 안됨.

try는 실행을 // 예외가 없을 때 finally와 그 아래 실행.
는 catch(조건)는 실행 // catch 조건 없을 때면 catch는 실행.
는 finally는 후 finally까지 실행하고 끝!
3 ↳ catch는 여러개 사용

- try catch 을 메서드로 만들어서 사용 가능
- 상위 블록이 제일 아래로

② 예외 회피 throws

- 예외가 발생해도 계속 실행되라. 그러므로 확신이 있을 때만 사용.

③ 예외 전달 throw

- 발생한 예외가 초기화 가능한 예외를繼承하여
→ 그 구현으로 이동하기 할 때 사용.

추가정보 필요!

catch or 예외 발생하고 finally를 실행

Exception

try { ① method() / ② }
Exception
catch (ArithmaticException) {

③
④

throw ex; finally { ⑤ } ⑥) ;

예외 복구 → try catch

예외 회피 → throws,,

예외 전환 → throw

method 1 = try { ① }
throw Exception; }

catch (Exception e) { ② } ③

throw Exception;)
finally { ④ } ⑤ ⑥ ,

11 23 6/15 실습.

① 추가적인 문제(예제) 풀이 필요!

스레드.

- 멀티스레드 : 동시에 여러 작업 실행
-

제네릭.

- 타입지정 (타입 Type)
- 타입을 지정해서 Read 단계에서 메모리 소모를 줄임.

wild card.

- ? 를 넣으면 모든 객체가 들어올 수 있다.

- 종속 관계에 대한 person - worker

student
└ Highschool
└ middchool.

multi 메서드.

img 불이기!

제네릭 메서드

Collection 프레임워크.

DB에서 데이터를 가지고 오는 형식.

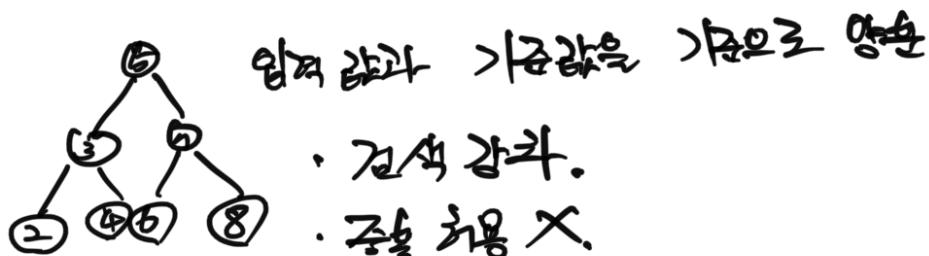
- ArrayList 배열형식의 프레임.
 - ↳ 값이 드나들면 index 값이 바뀌기 때문에 편
 - ↳ 단. 속도가 빠르다.

- LinkedList
 - ↳ 수정.삭제.추가 용이

- Vector
 - ↳ 멀티 스레드로 처리 가능함.
 - ↳ 형식은 ArrayList와 같다.

- HashSet
 - ↳ 무작위로 담긴 데이터를 이터레이터를 통해
접속할 수 있다(단. 캐싱)

- Treeset : 그린 트리 구조.



- очередь : 큐 객체의 프레임을 읽는 할 수
Queue, Array, Set을 읽고 Map은 못 읽음.

- Hash map.
 - ↳ 키: 값으로 구성. ↳ 정의되어 있지 않을 경우 null

- Tree map
 - ↳ 키: 값을 Tree 형식으로 가지고 옴.
-

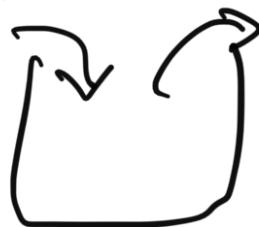
Compare.

FIFO



Queue.

LIFO



stack.

io-stream 초기 단계에 대한 부분.

stream : 흐름 : 한방향으로 in/out 을
나눠서 수행한다.

file의 이동.

- byte 단위로 파일 이동.

- ... I . I → m → n → [저장기]로 했다면

- 100 byte 를 4byte 단위로 나누어 전송 후 합성 한다.