• 07/15

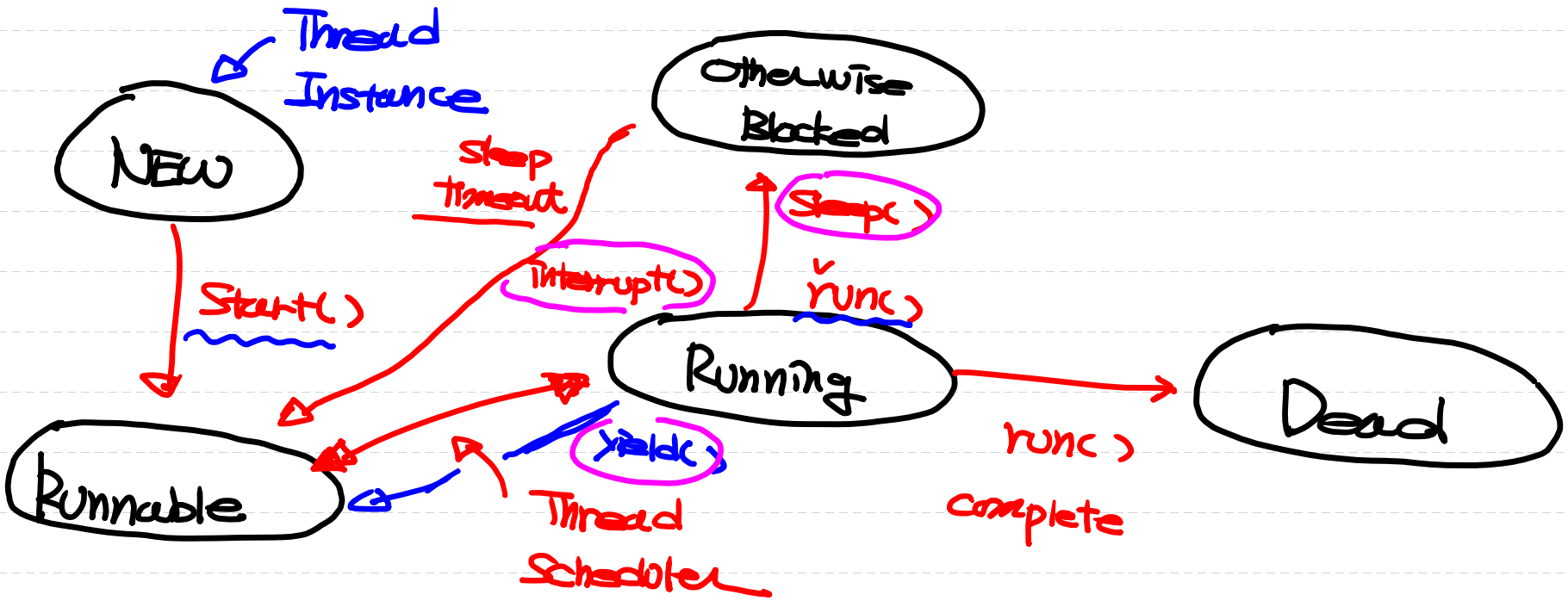
② Thread

"Thread Class"

① 상속

② Runnable Interface를 구현한 Class를 작성

→ Instance를 생성 → 이 객체를 이용하여 Thread 생성

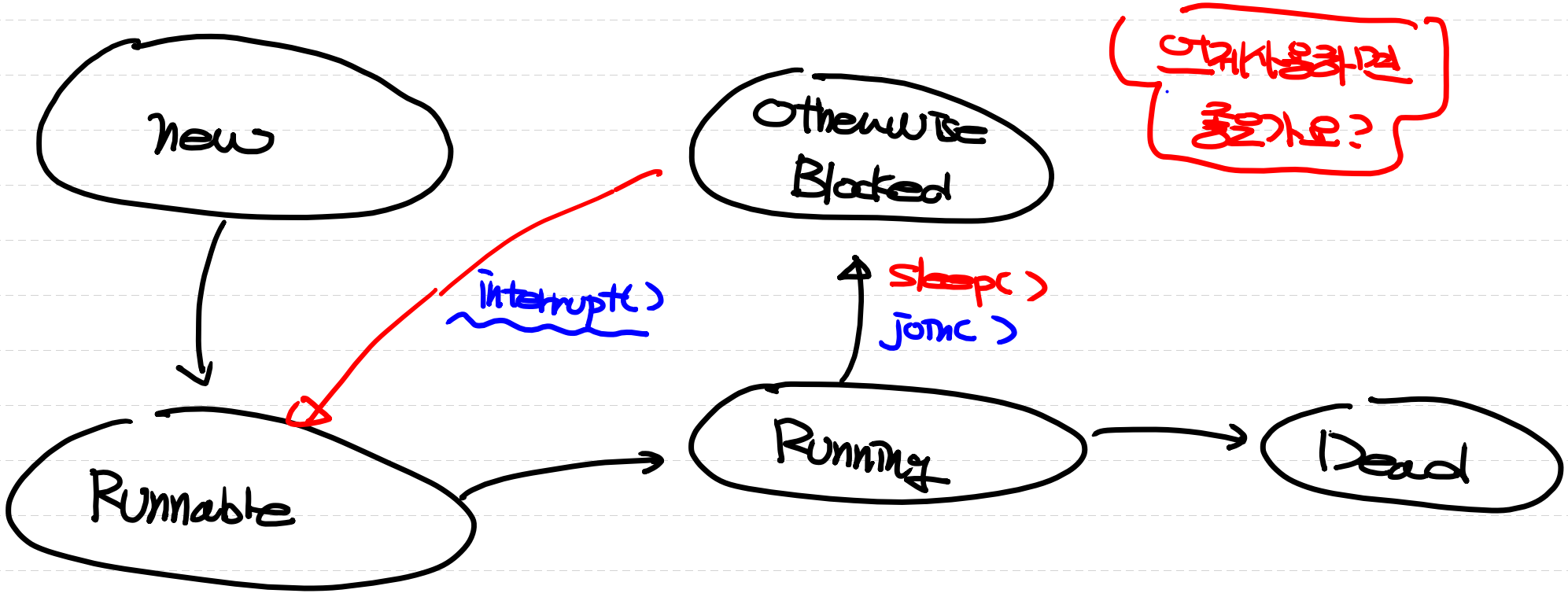


join()

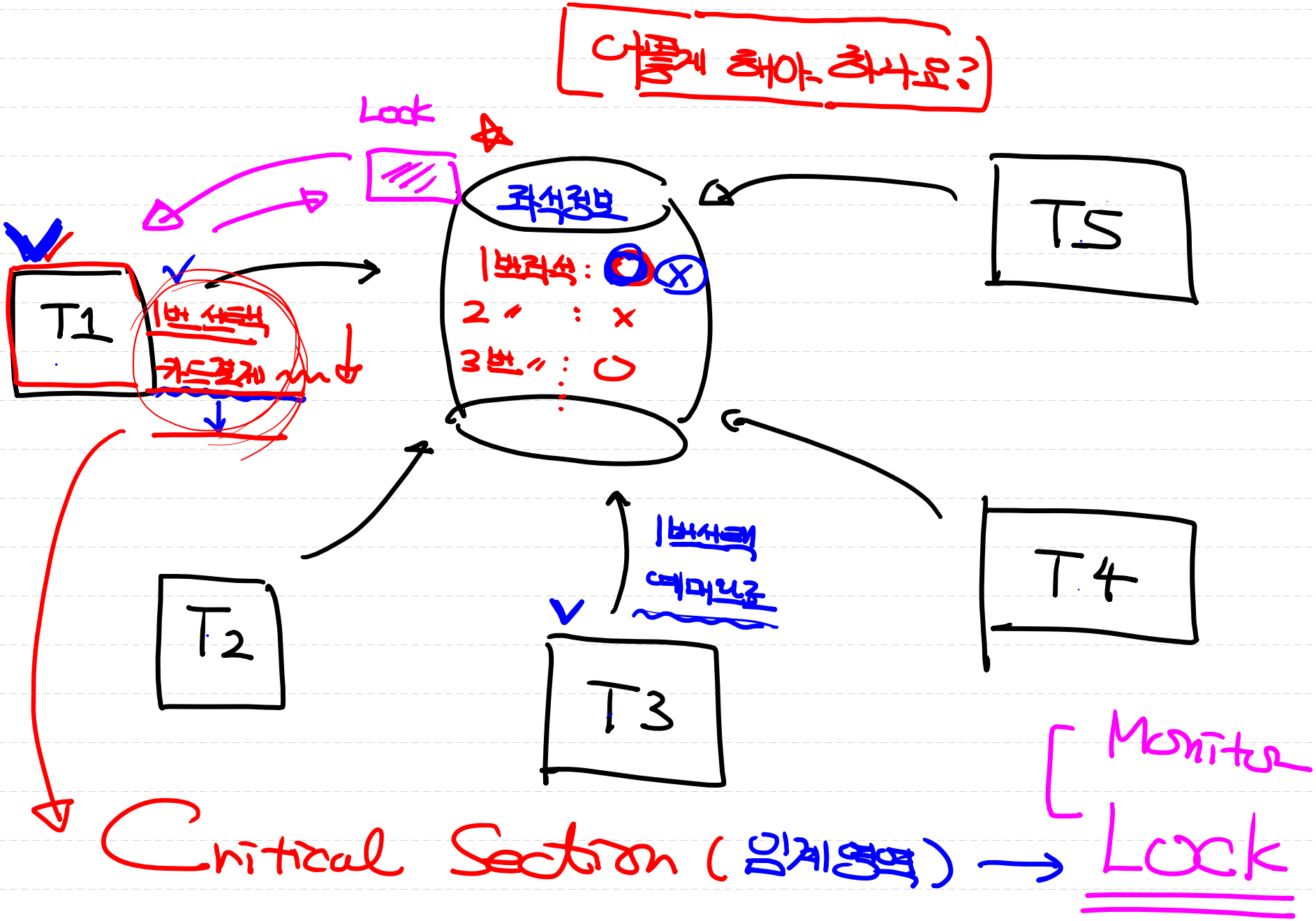
join()
join(long 초)

Sleep → Static method
Join → Instance method

"코드로 알아보아요"



Thread의 동기화


어떻게 해야 하나요?
Lock
좌석정보
1번 : O X
2번 : X
3번 : O
T1
1번 선택
카드결제 ing
T2
T3
1번선택
예매완료
T4
T5
Critical Section (임계영역) → Lock
Monitor

Java에서 "Lock"을 쓰는 암시적록을 설정하려면

⇒ synchronized keyword

↳ method 동기화

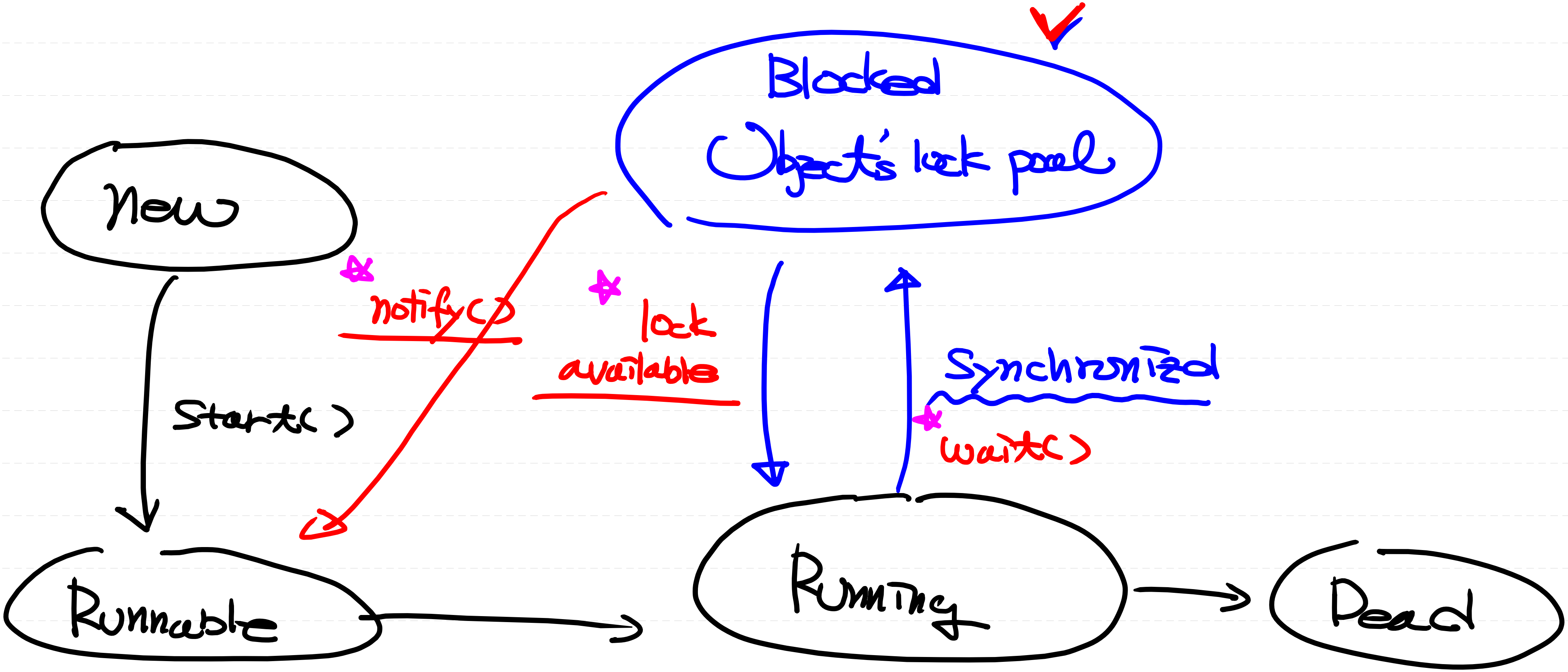
동기화 block을 생성

"Synchronized"를 이용하여 공유데이터를 보호

Thread가 공유자원에 대한 Lock을 획득할수 없으면 일정시간을 보내요..

→ 문제가 되요 [wait() → Lock을 놓고 대기
notify() → 대기상태에 있는 Thread가 실행될 수 있도록 block을 해제

"동기화 keyword"



New
Start( )
Runnable
notify( )
lock available
Blocked
Object's lock pool
Synchronized
wait( )
Running
Dead

⑤ 연습문제

" 1초마다 자신의 이름을 출력하는 Thread를 2개생성 "

↳ 이름을 출력할때 교대로 출력하는걸 보장하는 코드를 작성해 보아요!!

② Thread가 3개 이상일때 순서대로 출력!!

③ 1/2 → 1초 → 1/2 → 1초 → 1/2

구현해 보면 좋을거 같아요

"?"

Java 입출력

- IO (Input/output) → 최적화
  - ↓ 개선
  - NIO (New I/O) Java 4 ~ Java 7 향상 → 어려움

★

Java IO

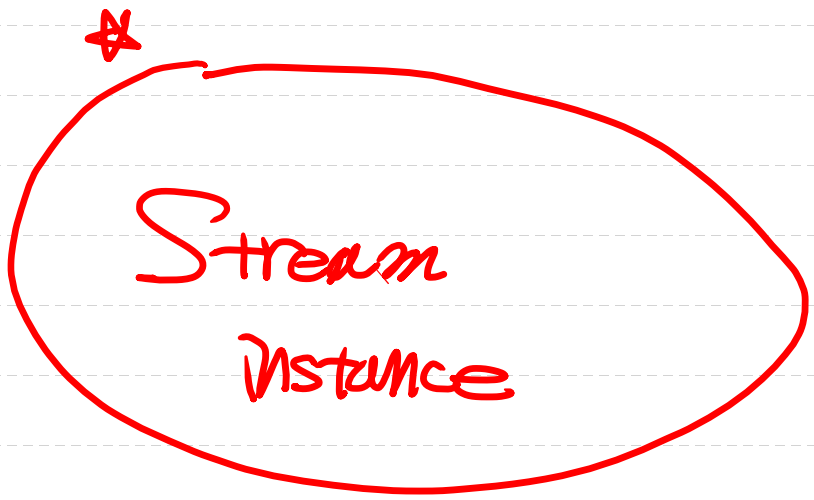
→ "java.io package로 제공"

→ Stream

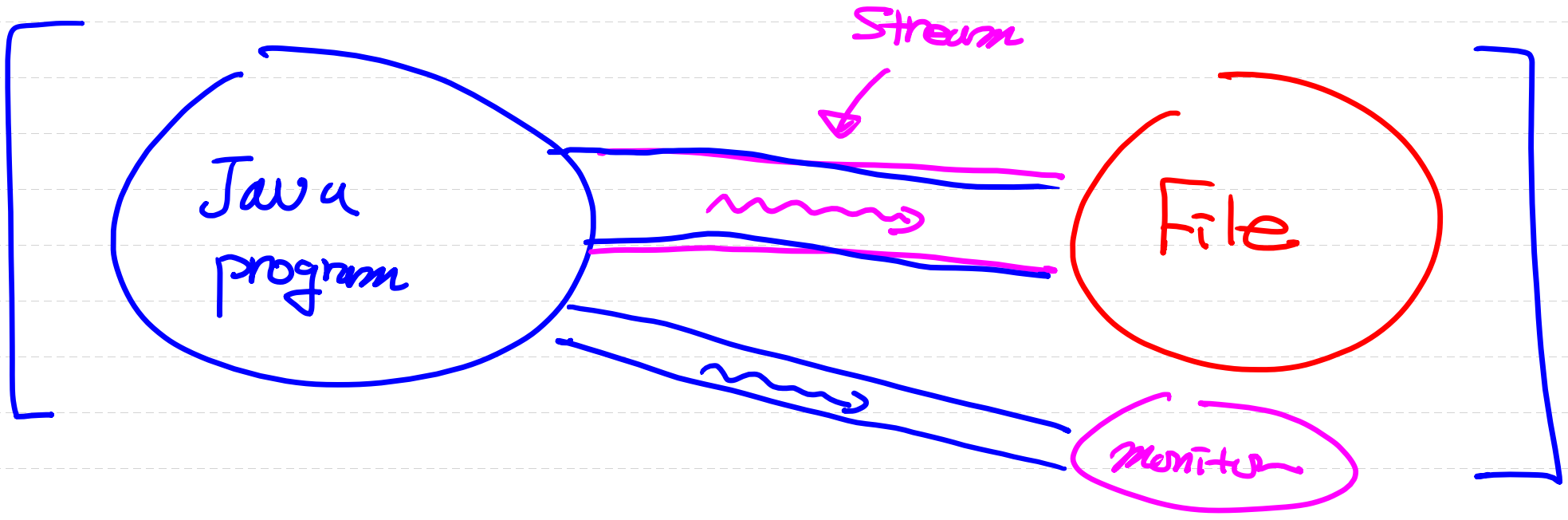
표준입력 → keyboard → System.in

표준출력 → monitor ← System.out

Stream이라는 객체를 이용해서 입/출력

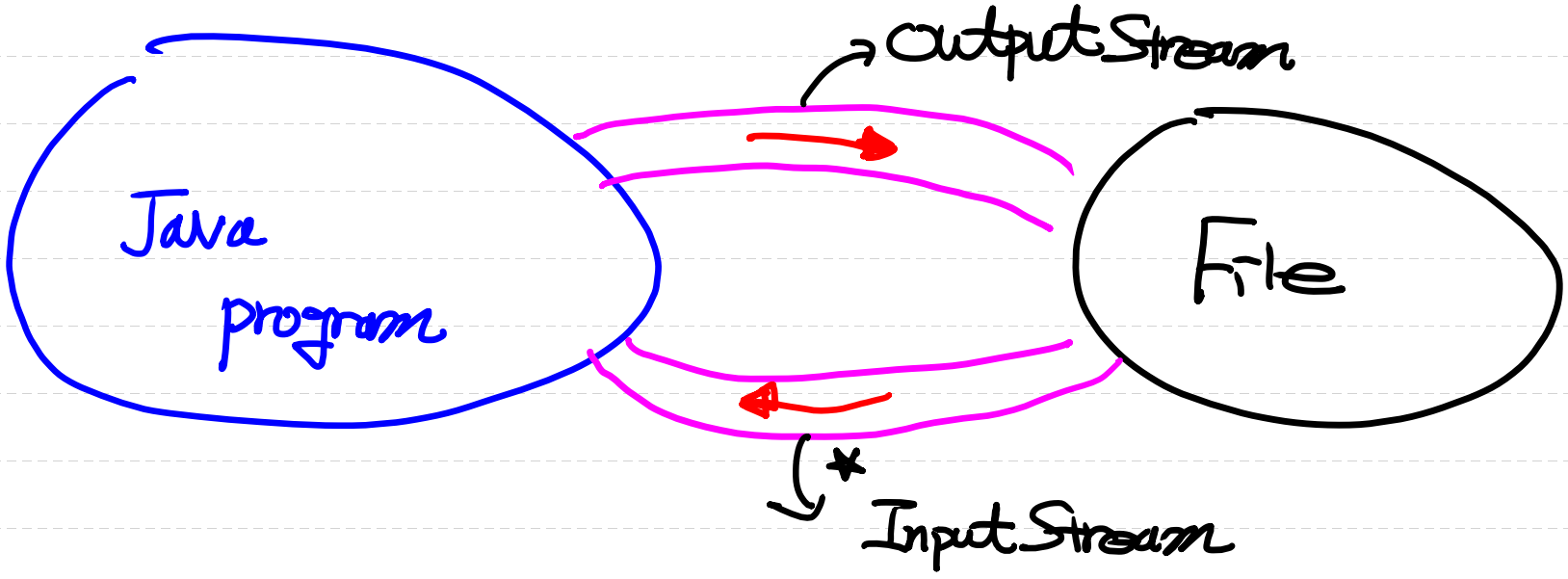


→ Java에서 특정 장치에서 data를 읽거나 특정장치로 data를 보낼때 사용하는 매개 객체

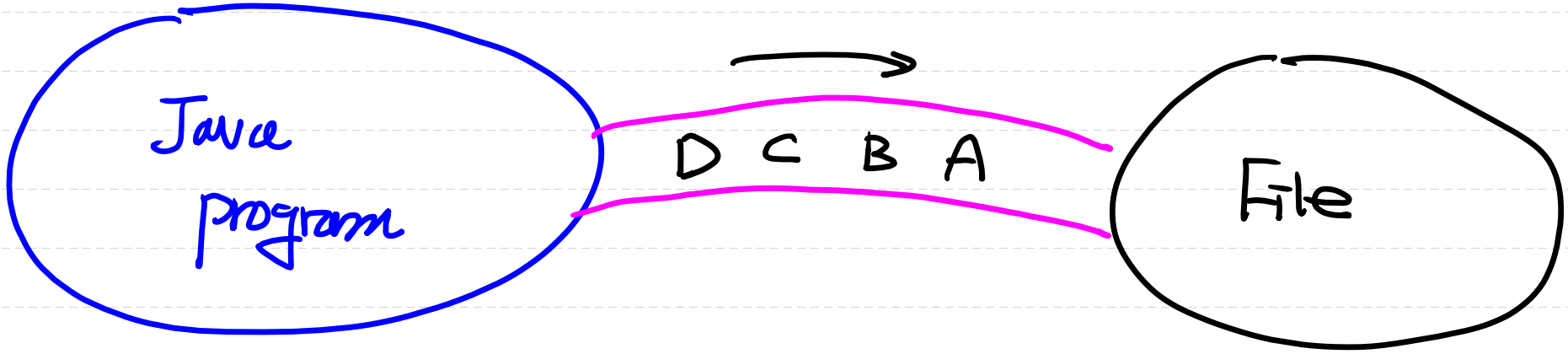


⑪ Stream 객체의 특징 - ① 단방향 *

Stream 생성할 때 Stream의 종류 결정 방향 결정



Stream 객체의 특징 - ② FIFO 구조 * → 먼저들어간 데이터가 먼저 나온다



⑩ Stream의 객체의 특징 - ③ 결합이 가능.

사용하기 편한 Stream을 만들어서 사용

Stream 구분 (Input / output
Byte / 문자 Stream

→ 써 보아요

Object Stream을 통해서 객체(Instance)도 전달할 수 있어요.

그 모든 객체가 다 되는건 아니에요 !!

⇒ 만약 Instance를 생성한 Class가

Serializable Interface를 구현하고 있으면

가능 !!

[IO 이런것만 얘기해요] → Network