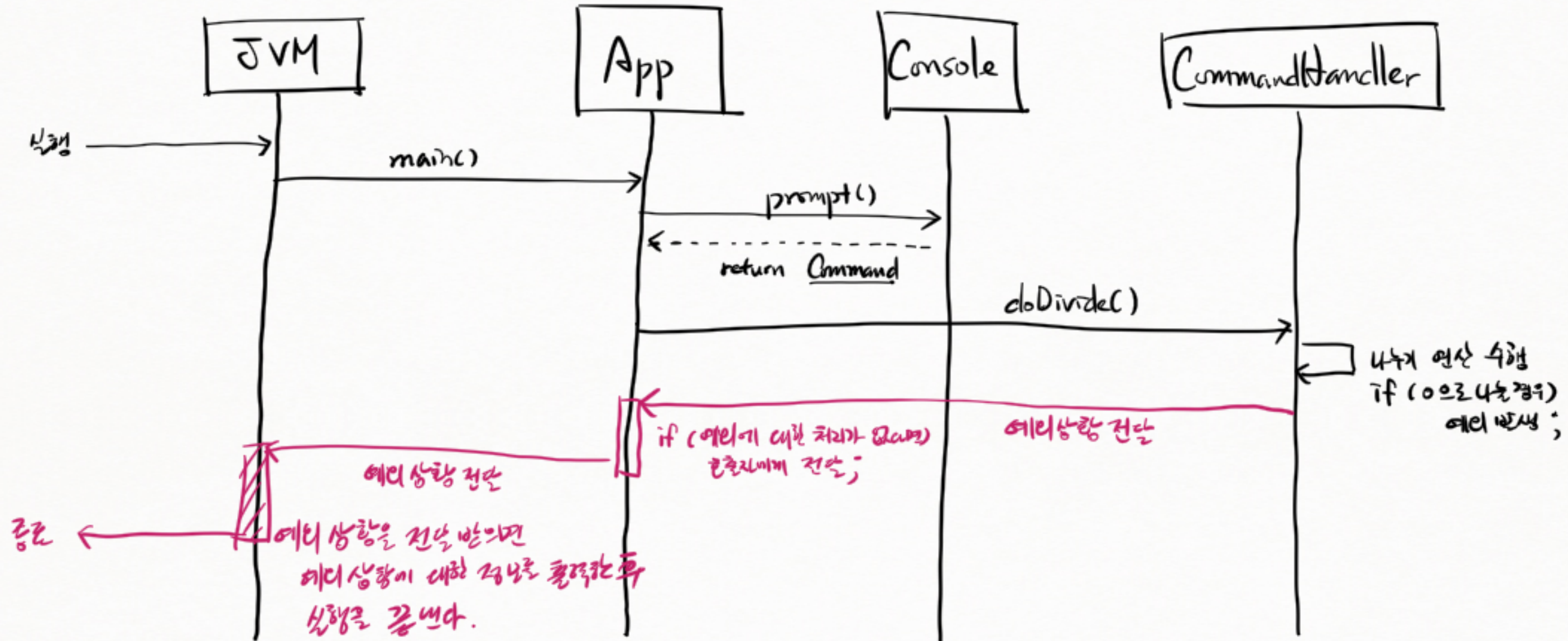
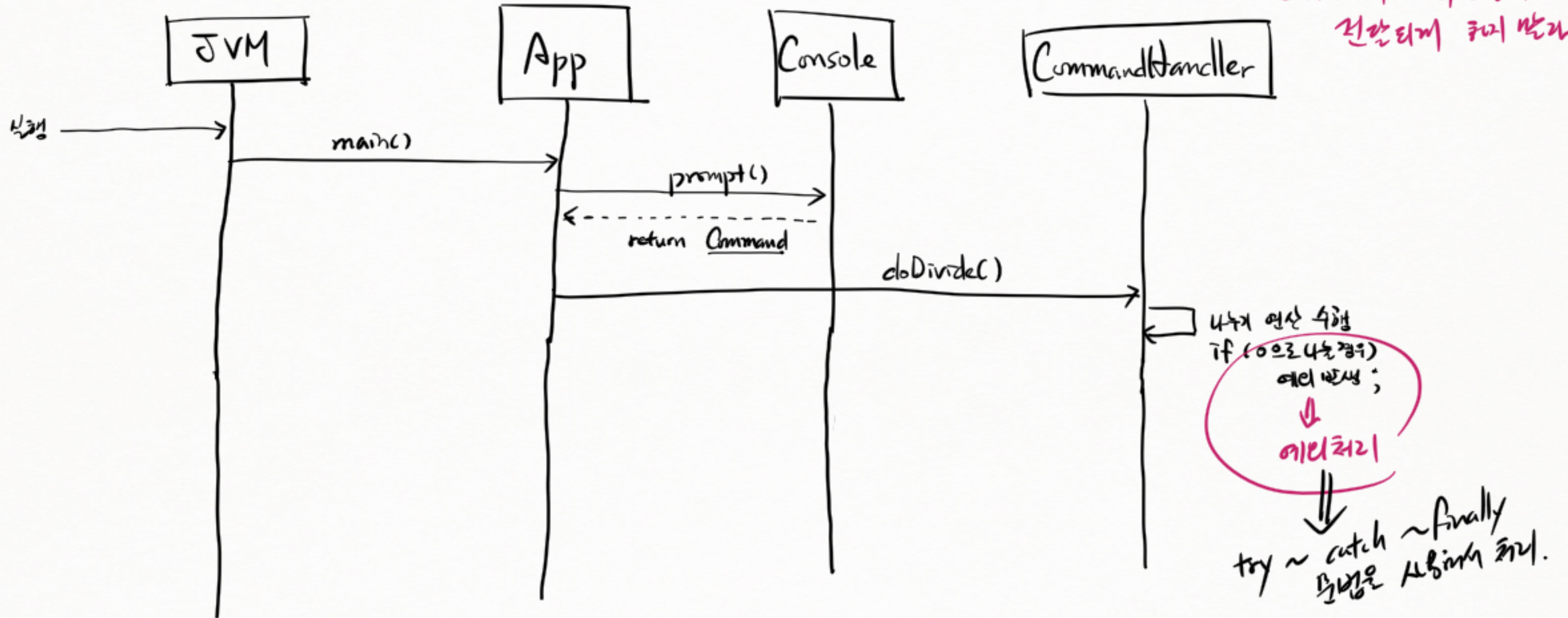


* JVM의 예외 처리

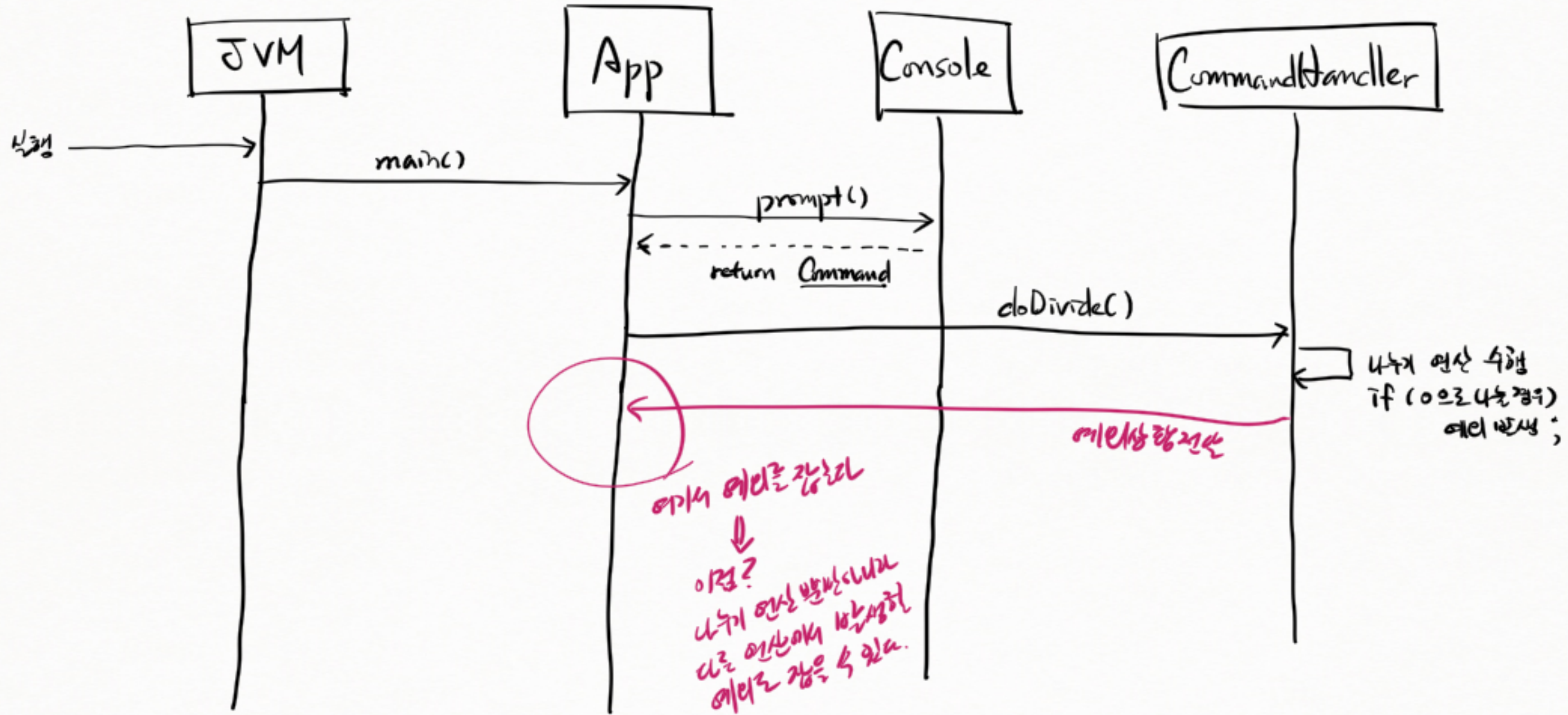


* 예외가 발생했더라도 JVM을 멈추지 않는 방법

예외를
 → 예외가 발생한 곳에서 처리하라! → JVM까지 예외 상황이 전파되지 않게 처리하라!



* 예외 발생 구조도 JVM은 만족하지 못하는 방법
 ↳ 예외처리 ②



* 데이터 처리 방법의 존재 이유

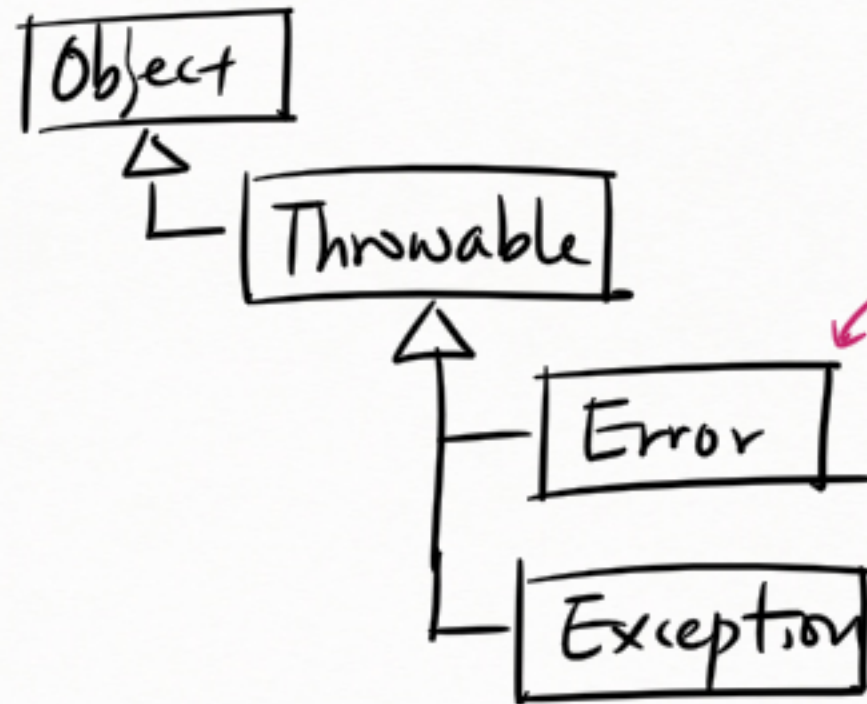


데이터 성능이 발생하는데 시스템은 번거로움에 상관없이 방법

→ 적절한 조치를 취할 수 있는 방법을 제공한다.

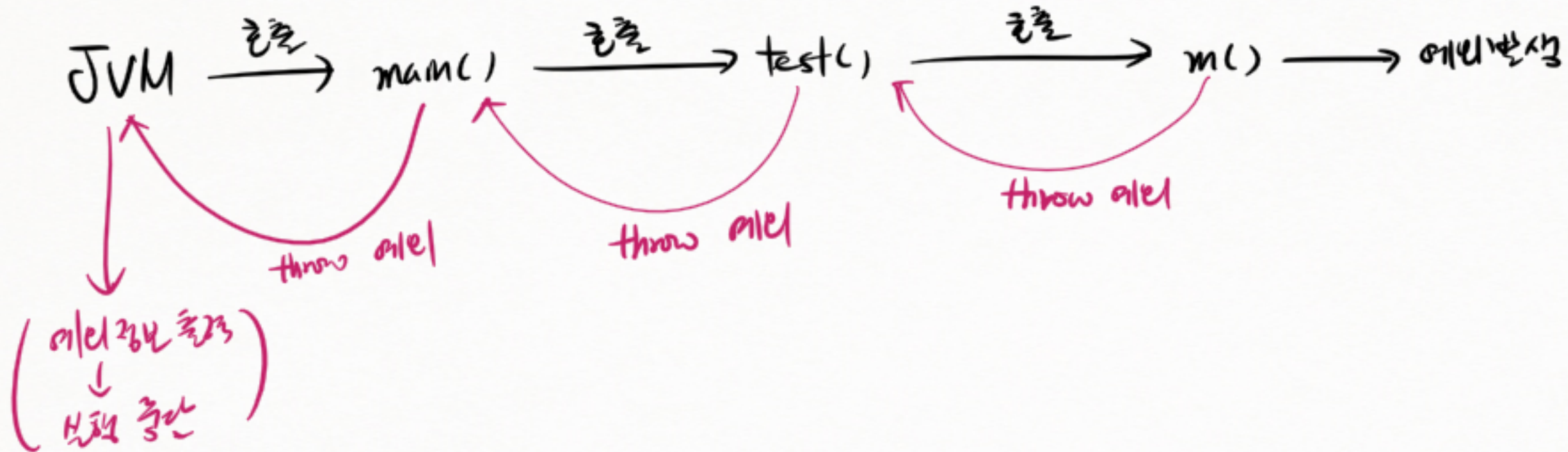
* 예외 상황 처리 - throw 다뤄

throw Throwable 객체 ;



JVM 오류 \Rightarrow 메달자가 관리할 수 있는
App. 오류 \Rightarrow 메달자가 관리할 수 있는 객체

* 예외 발생 시 전달



* 에리 전악 리 throws

a() → b() → c() → throw new Exception()

a() → b() → c() → throw new Error()

`a()` \longrightarrow `b()` \longrightarrow `c()` \longrightarrow `throw new RuntimeException()`

Handwritten notes above the code:

- `throws RuntimeException` (above `b()`)
- `throws RuntimeException` (above `c()`)
- `throws RuntimeException` (above `throw new RuntimeException()`)
- `Exception` (boxed, above `RuntimeException()`)
- `RuntimeException` (boxed, below `throw new RuntimeException()`)