1. @requiredArgsConstructor를 써라

* Final로 선언한 필드들만 자동 주입해준다.
* 반드시 null이 아닌 객체가 주입될 것이다.
* 그러면 null 체크를 안해도 된다.
* allArgsConstructor는 쓰지마라.

1. 특정부분에서만 쓰는 지역변수 선언을 정리하겠답시고 바깥쪽에 선언 하지 말아라

* 그러면 나중에 이 변수는 어디쓰는건지 하고 혼란만 온다

1. 파일과 관련된 IO 메서드 같은 경우 반드시 close를 해줘야한다.

* 하지만 이때 finally에서 추가로 try catch를 계속 파일마다 해줘야 하는 번거로움이 생긴다.
* Try() 구문 괄호안에다가만 파일 관련 객체를 선언하면 try끝날 때 자동으로 종료되서 편하다.

1. 문자열 분리해서 할당할 때 검사를 clearParts[5]이런 식으로 하지마라.

* 나중에 보면 ? 한다
* 이럴경우 parts 문자열 개수만 체크하고 그냥 DTO에 넣어라
* DTO에 넣은 후 검사해라(?)

1. 서비스에서 Exception을 처리한답시고 단순히 return null하지마라

* 그냥 throw exception 사용자 정의 exception 만들어서 컨트롤러에서 보고 메시지 처리하라

1. 다양한 엔티티를 가지고 같은 유형의 DTO를 만들고 싶을 때

=>UserDTO 생성자를 또 만들어라

=> 이 생성자는 인터페이스를 가지고 DTO를 만든다

=>이때 User 인터페이스는 implements interface해라

=>entity -> dto 변환은 dto 쪽에서 해라

1. 상속받은 것을 주입받을 때 구분하려면

[7-1.@Qualifier](mailto:7-1.@Qualifier)로 구분

7-2. 클래스명을 그대로 따서 앞글자만 소문자로 변수이름을 짓는다.

1. @requiredargs와 @autowired를 같이쓰지마라

* 이미 requiredargs에서 final 변수에 자동으로 빈 주입해주는데 또 쓰는건 중복된다.

1. If( != null){

Return 객체}

Else {return null}

이렇게 하지말고 그냥 아예 변수 자체를 return 해버리자.

객체 검사하는건 차라리 프론트에서 해라