

Serializable을 구현한 클래스에서는 그 필드들도 의도치 않게 공개 API가 될 수 있다.

클래스의 공개 API를 세심히 설계 후, 그 외의 모든 멤버는 private으로 만들자.

그런 후 오직 같은 패키지 내 다른 클래스가 접근해야 하는 멤버에 한하여 package-private으로 풀어주자

<위 같은 경우가 많다면 나의 시스템에서 컴포넌트를 더 분해해야 하는것은 아닌지 다시 고민해보자.>

private과 package-private 멤버는 모두 해당 클래스의 구현에 해당하므로 보통은 공개 API에 영향을 주지 않는다. 단 Serializable을 구현한 클래스에서는 그 필드들도 의도치 않게 공개 API가 될 수 있다.

자!! 여기서 저의 주제가 들어갑니다.

왜 의도치 않게 공개 API가 되는걸까??

- 이 부분은 저번에 영규님께서 잘 설명해주셨다.
- **Serializable** 인터페이스를 구현한 클래스의 경우, 해당 클래스의 모든 필드가 공개 API로 간주될 수 있다.
- **Serializable** 인터페이스는 객체를 직렬화하고 역직렬화하는 데 사용되며, 필드의 값도 직렬화 및 역직렬화의 대상이 될 수 있따!
- 그러므로 **Serializable** 인터페이스를 구현한 클래스의 모든 필드는 의도치 않게 공개 API로 노출될 수 있는 가능성이 있다고 한다.
 - 역직렬화 중에는 새로운 객체 내의 필드도 역직렬화되어 값이 설정된다. 따라서 직렬화된 객체의 필드 값 뿐만 아니라, 역직렬화된 객체의 필드 값도 의도치 않게 공개 API가 될 수 있다~

그렇다면 공개하고 싶지 않으려면?

- 중요한 정보가 있는 필드는 `transient` 키워드를 사용하여 직렬화에서 제외시키거나, 직렬화할 때 암호화를 적용하는 등의 방법을 사용하여 필드의 노출을 제한할 수 있다.