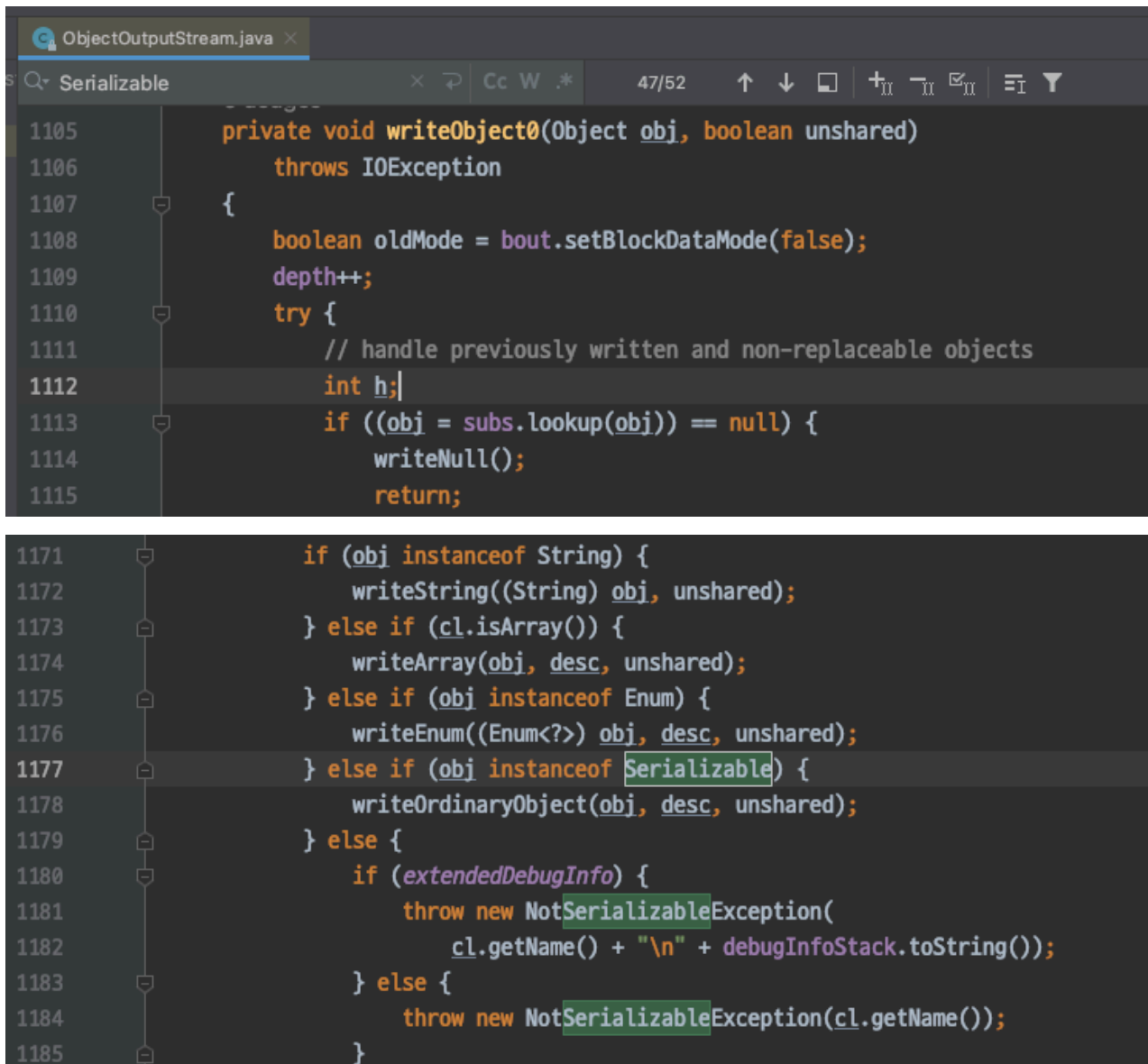


Use marker interfaces to define types

marker interface란?

class의 속성을 부여하기 위한 용도로 어떤 메서드가 선언되지 않은 interface를 말합니다.

이 예로, Serializable interface가 있습니다. 이 인터페이스를 구현하면 ObjectOutputStream통해 쓸 수 있다고 표시해주는 역할만 하죠.



```
ObjectOutputStream.java x
Serializable 47/52
1105 private void writeObject0(Object obj, boolean unshared)
1106     throws IOException
1107 {
1108     boolean oldMode = bout.setBlockDataMode(false);
1109     depth++;
1110     try {
1111         // handle previously written and non-replaceable objects
1112         int h;
1113         if ((obj = subs.lookup(obj)) == null) {
1114             writeNull();
1115             return;
1171         if (obj instanceof String) {
1172             writeString((String) obj, unshared);
1173         } else if (cl.isArray()) {
1174             writeArray(obj, desc, unshared);
1175         } else if (obj instanceof Enum) {
1176             writeEnum((Enum<?>) obj, desc, unshared);
1177         } else if (obj instanceof Serializable) {
1178             writeOrdinaryObject(obj, desc, unshared);
1179         } else {
1180             if (extendedDebugInfo) {
1181                 throw new NotSerializableException(
1182                     cl.getName() + "\n" + debugInfoStack.toString());
1183             } else {
1184                 throw new NotSerializableException(cl.getName());
1185             }

```

마커 애너테이션과 비교

마커 애너테이션이 나오면서 마커 인터페이스를 사용할 일이 없다고 들었을 겁니다. 이건 전적으로 잘못된 말이며 다음과 같은 차이점이 있습니다.

marker interface의 이점

1. 타입 결정

마커 인터페이스는 marked class의 타입을 나타내는데 사용할 수 있지만 마커 애너테이션은 그럴 수 없습니다.

마커 애너테이션을 썼다면 런타임에서야 타입 에러를 캐치할 수 있지만, 인터페이스를 쓴다면 컴파일 타임에 곧바로 알아채릴 수 있죠.

2. 타겟 정확성(precisely)

좀 더 specific하게 타겟을 지정할 수 있습니다.

마커 애너테이션은 type을 interface냐, class냐 정도로 정의할 수 있지만, marker interface는 어떤 인터페이스의 타입인지까지 정의할 수 있는 것이죠. 따라서 해당 인터페이스를 구현하는 타입들은 마커 인터페이스의 속성을 가지고 있음을 보장할 수 있습니다.

Set

논란이 있을 수 있지만 Set도 marker interface라고 볼 수 있습니다. 하지만, marker interface라고 말하기에 논란이 되는 부분은 어느정도 Collection이 정의한 메서드를 일부 수정했기 때문입니다. 그럼에도, 만약 수정하지 않았다면 다음과 같은 마커 인터페이스라고 말할 수 있습니다.

- 특정 인터페이스의 하위 타입에만 적용할 수 있는 인터페이스

라고 말할 수 있습니다.

이러한 마커인터페이스는 다음과 같은 이점을 가지지요.

1. 객체의 특정 부분을 invariant로 규정하거나
2. 다른 클래스의 메서드가 처리할 수 있다고 명시하는 용도 (Serializable interface 예처럼)

marker annotation의 이점

1. 지원

마커 애너테이션은 애너테이션 시스템의 지원을 받을 수 있습니다.

마커 애너테이션을 적극적으로 활용하는 프레임워크에서 사용하면 아주 큰 이점이 있겠지요.

사용 구분

marker interface를 사용해야할 때

- 마킹이 된 객체를 다른 메서드의 파라미터로 받아야한다면 marker interface를 적용하시기 바랍니다.

marker annotation을 사용해야할 때

- 클래스와 인터페이스 외 프로그램 요소에(모듈 패키지, 필드, 지역변수 등)에 마킹해야할 때
클래스나 인터페이스만 마커 인터페이스를 확장할 수 있으니까요.
- 마커 애너테이션을 적극 활용하는 프레임워크