아이템 13
'clone 재정의는 주의해서 진행하라'

'clone 재정의는 주의해서 진행하라'

참고 자료

- Shallow vs Deep Copy (예제 포함)
- Java: Clone and Cloneable
- Carrey's 기술블로그 - Item 13. Clone 재정의는 주의해서 진행하라
- How to Make a Deep Copy of an Object in Java
- Java Copy Constructor
- Copying Sets in Java
- Java clone - deep and shallow copy - copy constructors
- Java Cloneable interface - Is it broken?

* clone 재정의 키워드

1. 객체 복사
2. 사용하는 상황
3. 객체 복사 구현 방법
   - Shallow copy
   - Deep copy
4. 대체 방법
   - 복사 생성자(변환 생성자)
   - 복사 팩토리 메서드(변환 팩토리)
5. Java API 예시(HashSet -> TreeSet)
6. 정리

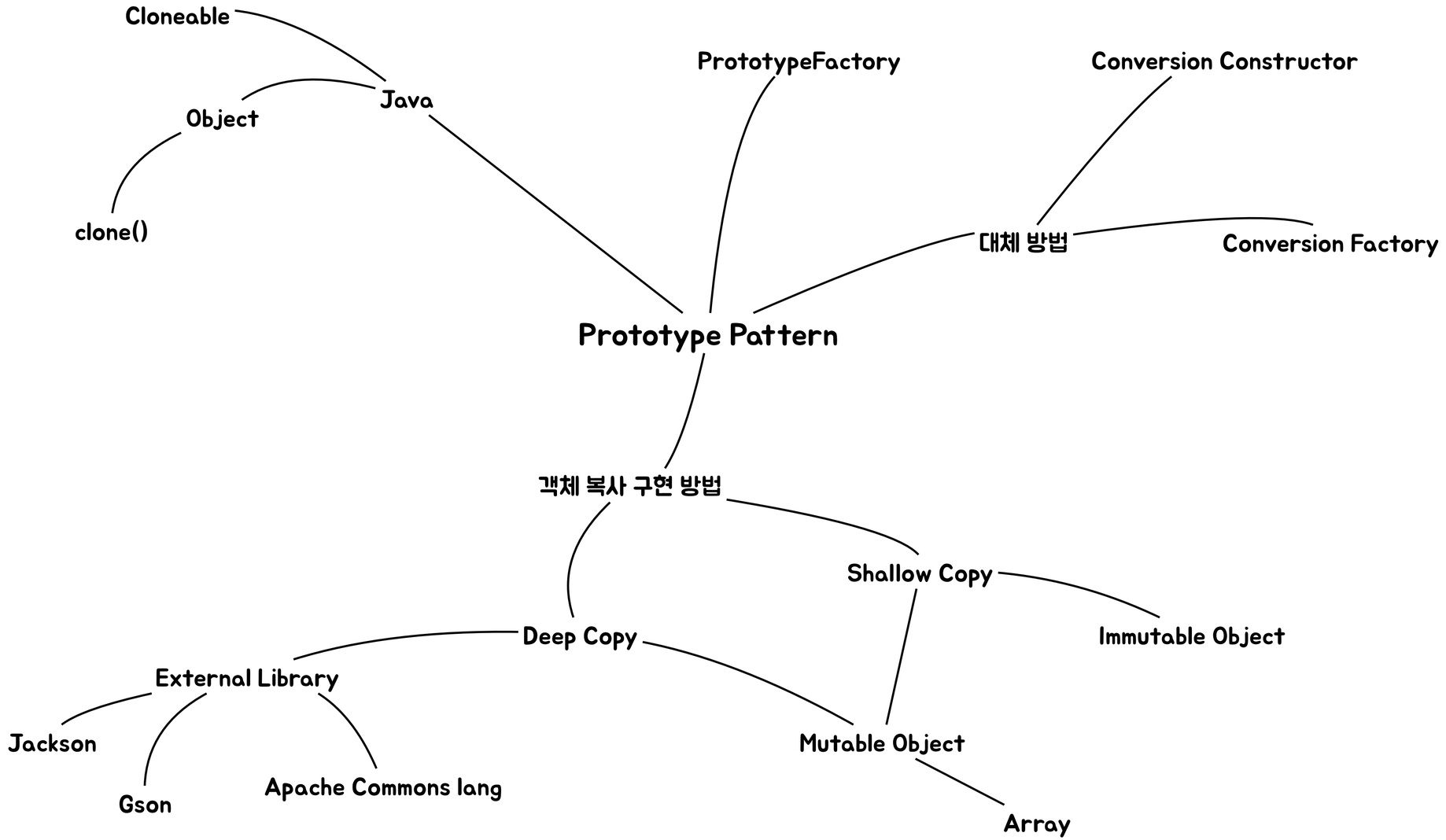
객체의 복사

객체 복사를 하는 상황은 언제일까?

* clone 재정의 키워드

1. 객체 복사
2. 사용하는 상황
3. 객체 복사 구현 방법
   - Shallow copy
   - Deep copy
4. 대체 방법
   - 복사 생성자(변환 생성자)
   - 복사 팩토리 메서드(변환 팩토리)
5. Java API 예시(HashSet -> TreeSet)
6. 정리

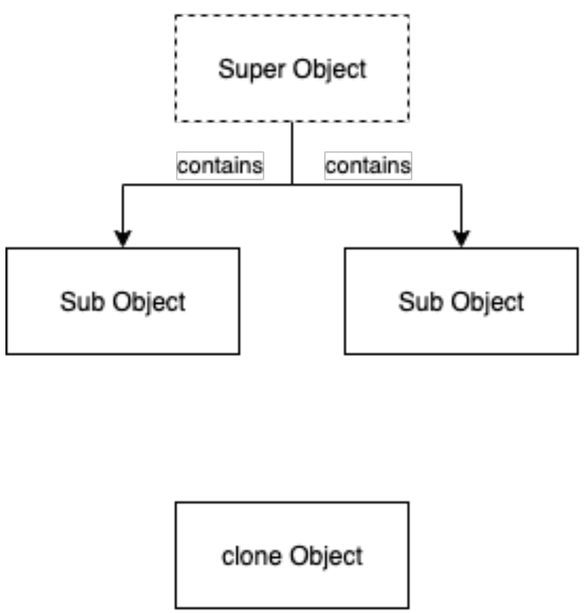
사용하는 상황



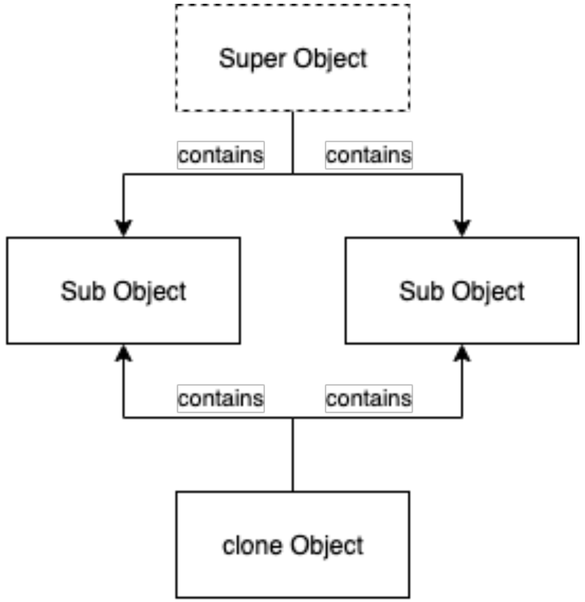
* clone 재정의 키워드

1. 객체 복사
2. 사용하는 상황
3. 객체 복사 구현 방법
   - Shallow copy
   - Deep copy
4. 대체 방법
   - 복사 생성자(변환 생성자)
   - 복사 팩토리 메서드(변환 팩토리)
5. Java API 예시(HashSet -> TreeSet)
6. 정리

객체 복사 구현 방법



객체 복사

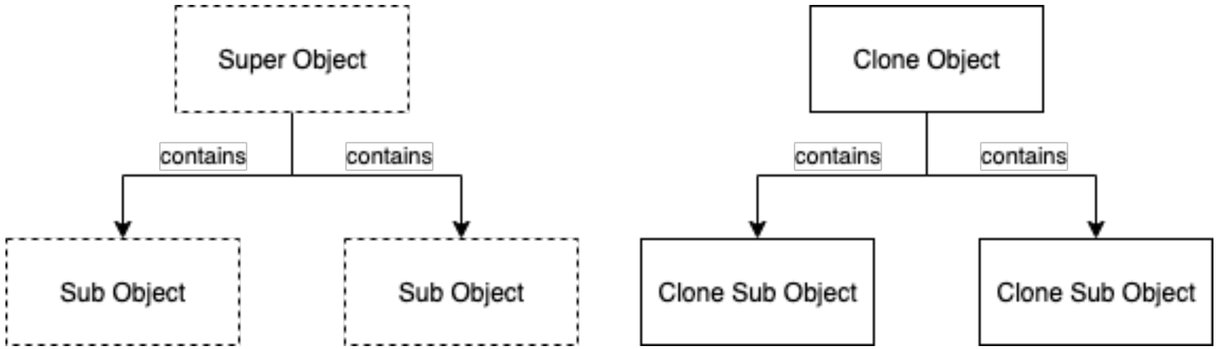


하위 객체 참조

* clone 재정의 키워드

1. 객체 복사
2. 사용하는 상황
3. 객체 복사 구현 방법
   - Shallow copy
     - Mutable Object(test)
     - Immutable Object(test)
   - Deep copy
4. 대체 방법
   - 복사 생성자(변환 생성자)
   - 복사 팩토리 메서드(변환 팩토리)
5. Java API 예시(HashSet -> TreeSet)
6. 정리

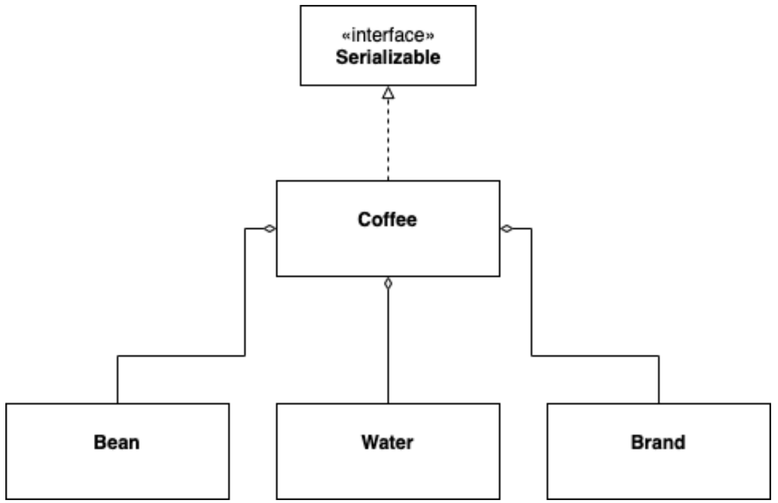
객체 복사 구현 방법

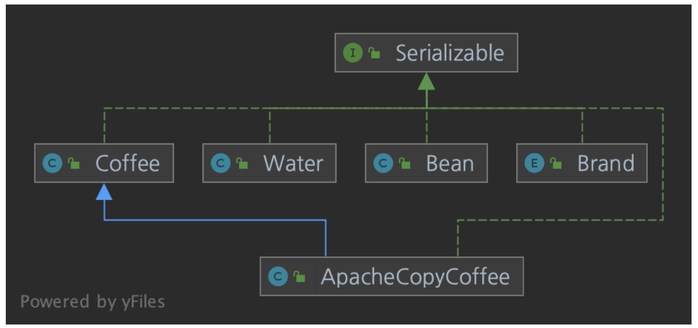


* clone 재정의 키워드

1. 객체 복사
2. 사용하는 상황
3. 객체 복사 구현 방법
   - Shallow copy
   - Deep copy(test)
4. 대체 방법
   - 복사 생성자(변환 생성자)
   - 복사 팩토리 메서드(변환 팩토리)
5. Java API 예시(HashSet -> TreeSet)
6. 정리

객체 복사 구현 방법






```
ApacheCopyCoffee deepCopy = (ApacheCopyCoffee) SerializationUtils.clone(coffee);
```

* clone 재정의 키워드

1. 객체 복사
2. 사용하는 상황
3. 객체 복사 구현 방법
   - Shallow copy
   - Deep copy
   - External
     - Apache Commons lang(test)
     - Gson
     - Jackson
4. 대체 방법
   - 복사 생성자(변환 생성자)
   - 복사 팩토리 메서드(변환 팩토리)
5. Java API 예시(HashSet -> TreeSet)
6. 정리

객체 복사 구현 방법

```java
class GsonCoffeeTest {

    @DisplayName("Gson 라이브러리를 통한 객체 복사")
    @Test
    void testCase1() {
        GsonCoffee coffee = new GsonCoffee(Bean.of("루악"), Water.regular(), Brand.NONE);

        // Gson으로 변환하기 위해서 Serializable 인터페이스를 구현할 필요가 없다.
        Gson gson = new Gson();
        GsonCoffee deepCopy = gson.fromJson(gson.toJson(coffee), GsonCoffee.class);

        assertThat(coffee).isEqualTo(deepCopy);
    }
}
```

// Gson으로 변환하기 위해서 Serializable 인터페이스를 구현할 필요가 없다.

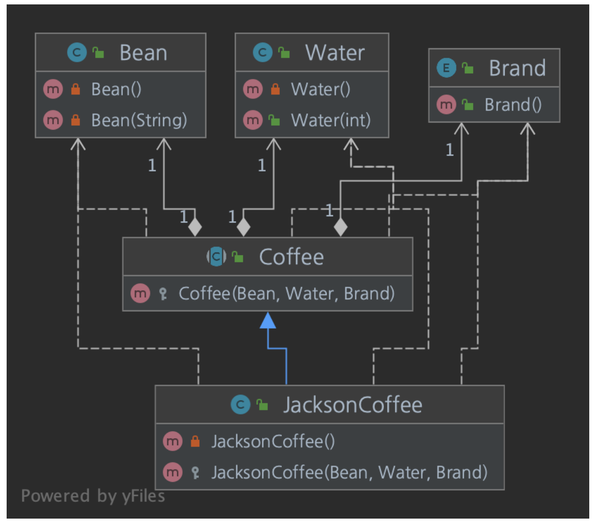
Gson gson = new Gson();

GsonCoffee deepCopy = gson.fromJson(gson.toJson(coffee), GsonCoffee.class);

* clone 재정의 키워드

1. 객체 복사
2. 사용하는 상황
3. 객체 복사 구현 방법
   - Shallow copy
   - Deep copy
   - External
     - Apache Commons lang
     - Gson(test)
     - Jackson
4. 대체 방법
   - 복사 생성자(변환 생성자)
   - 복사 팩토리 메서드(변환 팩토리)
5. Java API 예시(HashSet -> TreeSet)
6. 정리

객체 복사 구현 방법



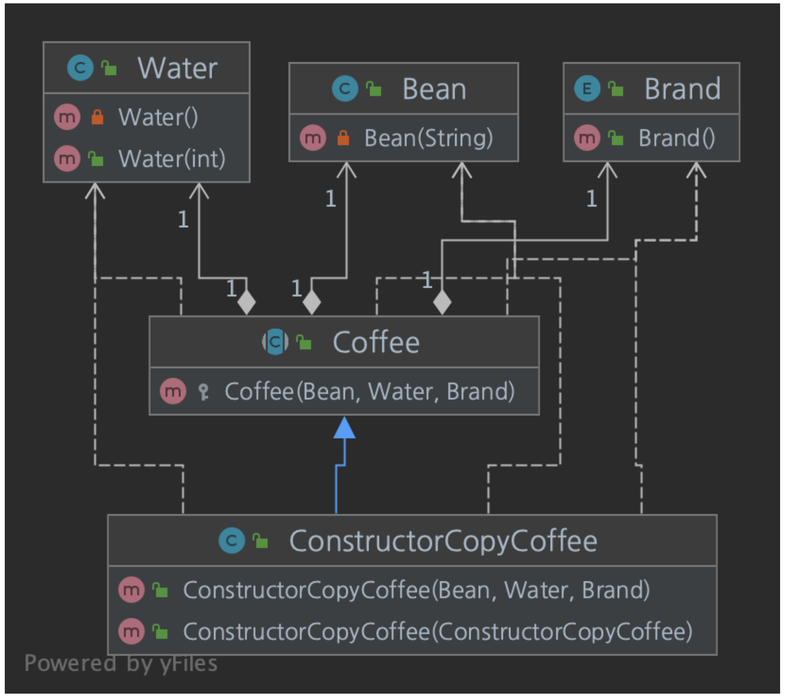

```java
class JacksonCoffeeTest {

    @DisplayName("Jackson 객체 복사")
    @Test
    void testCase1() {
        JacksonCoffee coffee = new JacksonCoffee(Bean.of("빈"), Water.regular(), Brand.NONE);

        ObjectMapper mapper = new ObjectMapper();

        // Serializable 을 하기 위해서는 해당 클래스의 필드에 접근할 수 있도록 설정이 필요
        mapper.setVisibility(PropertyAccessor.FIELD, JsonAutoDetect.Visibility.ANY);

        JacksonCoffee deepCopy = null;
        try {
            deepCopy = mapper.readValue(mapper.writeValueAsString(coffee), JacksonCoffee.class);
        } catch (JsonProcessingException e) {
            e.printStackTrace();
        }
        assertThat(coffee).isEqualTo(deepCopy);
    }
}
```

Jackson 라이브러리는 사용하기 위해서는 기본 생성자가 필수

* clone 재정의 키워드

1. 객체 복사
2. 사용하는 상황
3. 객체 복사 구현 방법
   - Shallow copy
   - Deep copy
   - External
     - Apache Commons lang
     - Gson
     - Jackson(test)
4. 대체 방법
   - 복사 생성자(변환 생성자)
   - 복사 팩토리 메서드(변환 팩토리)
5. Java API 예시(HashSet -> TreeSet)
6. 정리

대체 방법

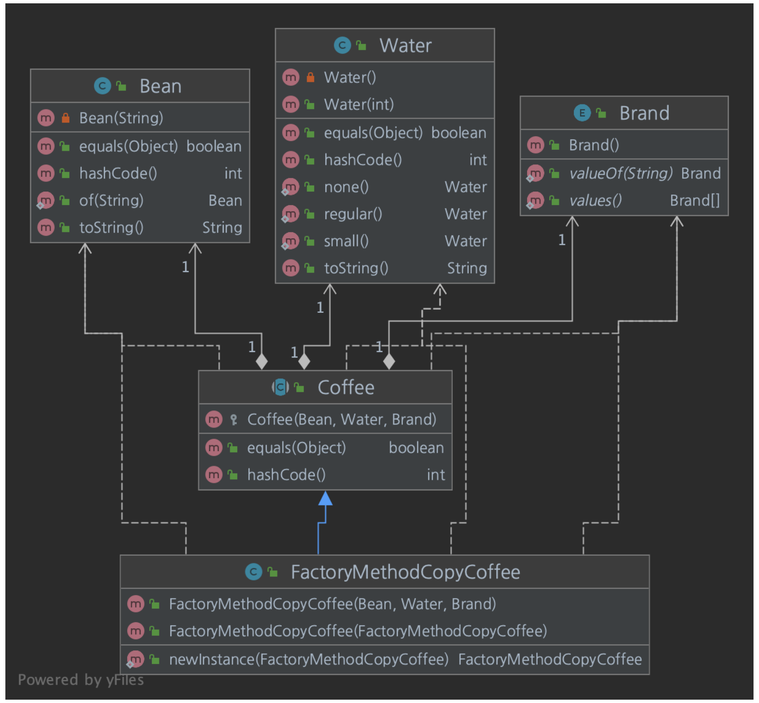


해당 클래스(상위 인터페이스)를 파라미터로 받는 생성자를 통한 복제

* clone 재정의 키워드

1. 객체 복사
2. 사용하는 상황
3. 객체 복사 구현 방법
   - Shallow copy
   - Deep copy
4. 대체 방법
   - 복사 생성자(변환 생성자)(test)
   - 복사 팩토리 메서드(변환 팩토리)
5. Java API 예시(HashSet -> TreeSet)
6. 정리

대체 방법



팩토리 메서드를 통한 복제

* clone 재정의 키워드

1. 객체 복사
2. 사용하는 상황
3. 객체 복사 구현 방법
   - Shallow copy
   - Deep copy
4. 대체 방법
   - 복사 생성자(변환 생성자)
   - 복사 팩토리 메서드(변환 팩토리)(test)
5. Java API 예시(HashSet -> TreeSet)
6. 정리

객체 복사 로직이 구현된 Java API

```
Constructs a new tree set containing the elements in the specified collection, sorted according to the natural
ordering of its elements. All elements inserted into the set must implement the Comparable interface.
Furthermore, all such elements must be mutually comparable: e1.compareTo(e2) must not throw a
ClassCastException for any elements e1 and e2 in the set.
Params: c - collection whose elements will comprise the new set
Throws: ClassCastException - if the elements in c are not Comparable, or are not mutually comparable
        NullPointerException - if the specified collection is null
public TreeSet( @NotNull @Flow(sourceIsContainer = true, targetIsContainer = true) Collection<? extends E> c) {
    this();
    addAll(c);
}
```

```
Returns a shallow copy of this TreeSet instance. (The elements themselves are not cloned.)
Returns: a shallow copy of this set
/unchecked/
public Object clone() {
    TreeSet<E> clone;
    try {
        clone = (TreeSet<E>) super.clone();
    } catch (CloneNotSupportedException e) {
        throw new InternalError(e);
    }

    clone.m = new TreeMap<>(m);
    return clone;
}
```

복사 생성자를 통해 구현된 라이브러리

* clone 재정의 키워드

1. 객체 복사
2. 사용하는 상황
3. 객체 복사 구현 방법
   - Shallow copy
   - Deep copy
4. 대체 방법
   - 복사 생성자(변환 생성자)
   - 복사 팩토리 메서드(변환 팩토리)
5. Java API 예시(HashSet -> TreeSet)
6. 정리

대체 방법

정리

1. 책에서 소개하는 객체 복사하는 방법은 직접 구현하기에 제약 조건이 많다.
2. 이미 구현되어 있는 라이브러리 코드를 분석, 사용하기 위해서는 Item 13을 읽고 이해해야 한다.
3. 외부 라이브러리를 통한 복제 방법들도 알아보고 사용방법을 이해해야한다.
4. 추가적으로 공부해봐야 할 키워드 '직렬화', '역직렬화'