

이펙티브 자바

Item 85 ~ 87

2024/08/06

12장 직렬화

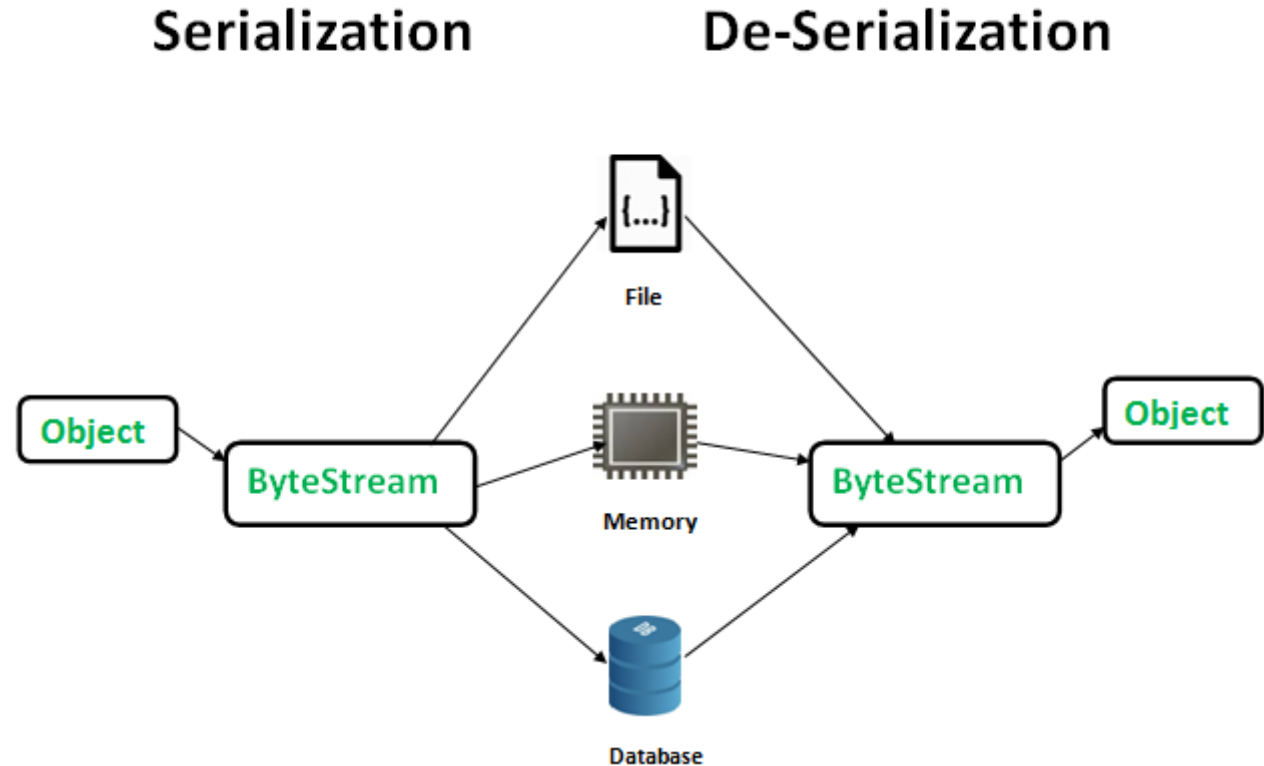
In computing, **serialization** (or **serialisation**) is the process of translating a data structure or object state into a format that can be stored (e.g. files in secondary storage devices, data buffers in primary storage devices) or transmitted (e.g. data streams over computer networks) and reconstructed later (possibly in a different computer environment).^[1]

12장 직렬화

객체를 유지,저장한다(영속화)


바이트 스트림을 가공한다.

JVM 밖에 사용하는 경우



12장 직렬화

자바에서의 직렬화



```
class Person implements Serializable {
    private static final long serialVersionUID = 1L;
    private String name;
    private int age;

    public Person(String name, int age) {
        this.name = name;
        this.age = age;
    }

    @Override
    public String toString() {
        return "Person{name='" + name + "', age=" + age + "}";
    }
}
```

12장 직렬화

자바에서의 직렬화 ObjectOutputStream.writeObject

```
Person person = new Person("John Doe", 30);
byte[] bytes;
try (ByteArrayOutputStream byteArrayOutputStream = new ByteArrayOutputStream())
{
    ObjectOutputStream objectOutputStream = new ObjectOutputStream(new
    ObjectOutputStream(byteArrayOutputStream));
    objectOutputStream.writeObject(person);
    bytes = byteArrayOutputStream.toByteArray();

    System.out.println(Base64.getEncoder().encodeToString(bytes));
} catch (IOException e) {
    e.printStackTrace();
}
```

12장 직렬화

스트림



ObjectOutputStream

ByteArrayOutputStream



12장 직렬화

직렬화를 어떻게 사용할 수 있을까?

사용한다면 어디에 사용할 수 있을까?

JSON, CSV

이진 직렬화

서블릿 세션, 캐시, 자바 RMI

```
<pre class="prettyprint"><code>
```

```
AC ED 00 05 73 72 00 0A 53 65 72 69 61 6C 54 65  
73 74 05 52 81 5A AC 66 02 F6 02 00 02 49 00 07  
76 65 72 73 69 6F 6E 4C 00 03 63 6F 6E 74 00 09  
4C 63 6F 6E 74 61 69 6E 3B 78 72 00 06 70 61 72  
65 6E 74 0E DB D2 BD 85 EE 63 7A 02 00 01 49 00  
0D 70 61 72 65 6E 74 56 65 72 73 69 6F 6E 78 70  
00 00 00 0A 00 00 00 42 73 72 00 07 63 6F 6E 74  
61 69 6E FC BB E6 0E FB CB 60 C7 02 00 01 49 00  
0E 63 6F 6E 74 61 69 6E 56 65 72 73 69 6F 6E 78  
70 00 00 00 0B
```

12장 직렬화

직렬화는 상속할 수 있다(역은 X)

내부 래퍼클래스도 직렬화를 보장해야한다.

직렬화 이후 클래스에 변경이 일어나는 경우 역직렬화 X

Transient로 특정 필드의 직렬화를 막을 수 있다.

Item 85 자바 직렬화의 대안을 찾으라

역직렬화는 보안 취약점이 존재한다

그나마 재정의로 보안 검사

역직렬화 폭탄



```
static byte[] bomb() {  
    Set<Object> root = new HashSet<>();  
    Set<Object> s1 = root;  
    Set<Object> s2 = new HashSet<>();  
  
    for (int i=0; i < 100; i++) {  
        Set<Object> t1 = new HashSet<>();  
        Set<Object> t2 = new HashSet<>();  
  
        t1.add("foo"); // t1을 t2과 다르게 만든다.  
        s1.add(t1); s1.add(t2);  
  
        s2.add(t1); s2.add(t2);  
        s1 = t1; s2 = t2;  
    }  
    return serialize(root);  
}
```

Item 85 자바 직렬화의 대안을 찾으라

직렬화를 피하자

사용한다면 역직렬화 필터링을 사용하자

JSON, protobuf

Item 86 Serializable을 구현할지는 신중히 결정하라

Serializable는 공개 API가 될 수 있다.

Private 정보도 제공하는 꼴

변경시 복잡하다.(serialVersionUID)

Item 86 Serializable을 구현할지는 신중히 결정하라

새로운 버전을 릴리즈할 때 테스트 요소가 많아진다.

구현여부는 쉽게 결정할 것이 아니다.

Item 87 커스텀 직렬화 형태를 고려해보라

객체의 물리적 표현과 논리적 내용이 같다면 기본 형태를 사용

```
public final class StringList implements Serializable {
    private int size = 0;
    private Entry head = null;

    private static class Entry implements Serializable {
        String data;
        Entry next;
        Entry previous;
    }
    // ... 생략
}
```

Item 87 커스텀 직렬화 형태를 고려해보라

transient

writeObject

```
Original list:
One
Two
Three
Deserialized list:
One
Two
Three
```

```
public final class StringList implements Serializable {
    private transient int size = 0;
    private transient Entry head = null;

    // 이제는 직렬화 하지 않는다.
    private static class Entry {
        String data;
        Entry next;
        Entry previous;
    }

    // 지정한 문자열을 이 리스트에 추가한다.
    public final void add(String s) { ... }

    // StringList 인스턴스를 직렬화한다.
    private void writeObject(ObjectOutputStream s)
        throws IOException {
        s.defaultWriteObject();
        s.writeInt(size);

        // 모든 원소를 올바른 순서대로 기록한다.
        for (Entry e = head; e != null; e = e.next) {
            s.writeObject(e.data);
        }
    }
}
```

Item 87 커스텀 직렬화 형태를 고려해보라

직렬화에 동기화를 적용해야한다.

UID를 명시할 것