Java training

Serialization

© Bogdan Solga

Session overview

- **Serialization** - what it is, use-cases
- **Hands-on** - serializing and deserializing objects
- **Exercises**

Object serialization / deserialization

- **Serialization** - representing an object as a sequence of bytes
  - Includes all the object's type, data and objects stored in the object
- After an object is serialized, it can be:
  - Saved to a file or in a database
  - Sent to another program (RMI)
  - Returned to a web consumer (to be further discussed)
- **Deserialization** - the reverse operation → creating an object structure from a serialized object
- Also called **marshalling** / **unmarshalling** in XML / JSON related contexts

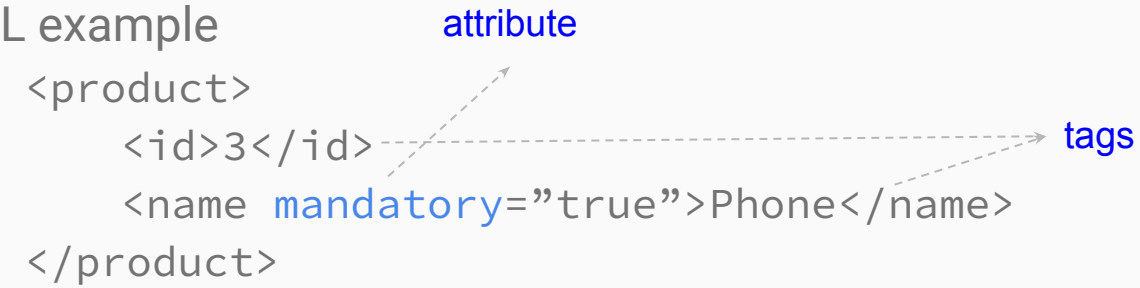
Serializing and deserializing objects

- **Serialize** - `FileOutputStream` and `ObjectOutputStream`
- **Deserialize** - `FileInputStream` and `ObjectInputStream`
- Requirements:
  - The serialized object *must* implement the **`Serializable`** interface
  - All it's fields *must* be serializable
  - The non-serializable fields *must* have the '`transient`' keyword

Hands-on - serializing / deserializing a `Product`

XML in a nutshell

- **XML** - e**X**tended **M**arkup **L**anguage
  - Used for data representation, interchange, modeling
  - ‘Deprecated’, in the latest years, in favor of JSON (further discussed)
- XML example



XML serialization / deserialization

- XML serialization - **`XMLEncoder`**
  - Using a `BufferedOutputStream` and a `FileOutputStream`
- XML deserialization - **`XMLDecoder`**
  - Using a `BufferedInputStream` and a `FileInputStream`

Hands-on:

- Serializing / deserializing a `Product`
- Serializing / deserializing a collection of `Products` (→ `Section`)

Exercise 1

- Create two CSV files with:
  - A few products, one on each line
  - A few stores, one on each line
- Read the CSV files from the disk, create objects from them (products and sections)
  - Split each line, parse the values and create the needed objects
- After the products and stores Java objects are created - display them, iteratively
- Serialize the stores in a XML file
- Serialize the products in a .ser file (classical serialization)
- Deserialize and display the previously saved files

Exercise 2

- Write a program which reads the details of a few products from the standard input
- After saving the products internally (in a (Set | List | Map)), serialize them in an XML file
- Deserialize the XML file, display the deserialized products
- Archive the XML file as a Zip file
- Create another program, which unzips the serialized XML file and displays the loaded products