Web services – Assessment Task 1

Marshaling and un-marshaling

Description

Create Java application which transforms Java POJO (Plain Old Java Object) to XML and back from XML to POJO using marshaling/un-marshaling tool.

1.1

1. Create XML document that represents an object with the list of its dependent objects (like Student with a list of Subjects assigned). Fill this xml document with data (not less than 10 dependent objects).
2. Create the DTD schema and validate the xml document against this DTD (using tools like XML plugin for Notepad++).
3. Create XSD schema and validate the xml document against this XSD.

1.2

1. Using `java.net` package capabilities, create simple **server** Java class which sends the xml file over network.

2. Using `java.net` package capabilities, create simple **client** Java class which receives the xml file sent by server over network. Both client and server could be located on single local machine and operate over localhost.

3. Prepare POJO classes: create Java class with different field types (String, int, float, boolean, char, etc.). This POJO class represents the root object in xml file (e.g., Student class). Create Java class that represents dependent object (e.g., Subject class) and add field of type List with the list of dependent objects to the root class. For example, the class Student could have a list of Subjects associated with this Student.

4. Create Java class which is capable to perform JAXB transformation from POJO to XML and back from XML to POJO.
5. Transform the received over network xml file to POJO and print its values to the standard output (console). The XML document should be validated against XSD schema.
6. Transform POJO class to xml document and print it to the standard output (console).

Requirements

1. The entire code should be properly formatted.
2. The Unit tests for all classes should present.
3. The package/class/field/method names should conform to the naming conventions.
4. The entire code should be properly documented and JavaDoc generated.
5. The entire code should conform to S.O.L.I.D principles.
6. For transformation the separate class should be created with 2 public methods:

transformToXML() and transformToPOJO(). All transformation logic should be encapsulated inside this class.

7. The deadline is March 15.

References

1. Package java.net
2. Getting Started with Socket Programming
3. Using JAXB in Java 11
4. https://www.oracle.com/technetwork/articles/javase/index-140168.html
5. https://www.javatpoint.com/java-naming-conventions
6. https://docs.oracle.com/javase/tutorial/java/package/namingpkgs.html
7. https://www.oracle.com/technetwork/java/javase/tech/index-137868.html
8. https://scotch.io/bar-talk/s-o-l-i-d-the-first-five-principles-of-object-oriented-design