Week 08 - Annotations & Serialization

This page is a draft, treat it accordingly.

Topics covered in this week

- annotation processing (high level)
- serialization to and from json

1.1.1. Reading material

Annotations:

- Factory Method Pattern
- http://purcellconsult.com/java-annotations/
- http://enos.itcollege.ee/~jpoial/allalaadimised/reading/Java-Annotations-Tutorial.pdf
- Java - The Complete Reference, 9th Edition - Herbert Schildt. pdf.zip pages 279-299

Serialization:

- https://dzone.com/articles/what-is-serialization-everything-about-java-serial
- Why We Hate Java Serialization And What We're Doing About It
- http://tutorials.jenkov.com/java-json/index.html chapters 1-7

1.2. Homework

Difficulty	Problem	Notes
EASY	We want to create a calculator that will log to console all operations. To achieve this: • Create a simple calculator that can add, subtract, multiply and divide. All the calculations should be done on integers only. Each operation is just a simple method which receives parameters and returns the result of the operation. • Create a Logger class that can log a simple string to the console output. • Implement an InvocationHandler (https://docs.oracle.com/en/java/javase/11/docs/api/java.base/java/lang/reflect/InvocationHandler.html) that will call a your Logger to log the method, object, object type, parameters and returned value when invoking a method on an object.(Add methods as needed to Logger class that handle the needed parameters) • Instantiate your Calculator implementation using a Proxy (https://docs.oracle.com/en/java/javase/11/docs/api/java.base/java/lang/reflect/Proxy.html) and call all of its methods. Logs should appear in the console for each method called.	
EASY	We want to create an annotation that will cause the method invocations from a class to be logged. To achieve this: • Take the previous exercise and create a Logged annotation. You should be able to use this annotation on both classes and methods. • Modify the code so that the operations of the Calculator are logged only if the annotation is present on the class or on the method.	
MEDIUM	We will create an simple in memory database that will be backed by a JSON for long term data storage. We will be storing student information and we will be using Jackson in order to serialize and deserialize the data. • You have been provided two POJO (plain old java object) classes. You should modify these classes with the necessary methods and annotations that will help you map the JSON fields to your POJO classes. • You will need to implement the interface StudentCrudRepository. All the operations will be done in memory. • When the application starts it will have to read the student data from a JSON file and store it in your in memory database by using your implementation of the StudentCrudRepository. • When the application is closed you will need to save any modifications to the data back into the JSON file.	Provided materials: week8p3materials.zip

1.3. Kahoot