**Lab: Reflection and Annotations**

This document defines the lab for ["Java HYPERLINK "https://softuni.bg/modules/59/java-advanced"OOP HYPERLINK "https://softuni.bg/modules/59/java-advanced"" course @ Software University](https://softuni.bg/modules/59/java-advanced). Please submit your solutions (source code) of all below described problems in [Judge](https://judge.softuni.bg/Contests/1604/Reflection-Lab).

**Part I: Reflection**

* **Reflection**

Import "**Reflection.java**" to your "**src**" folder in your project. Try to use **reflection** and print some information about this class. Print everything on new line:

* **This class type**
* **Super class type**
* **All interfaces** that are implemented by this class
* **Instantiate object** using reflection and print it too

**Don’t change anything in "Reflection class"!**

**Solution**



* **Getters and Setters**

Using reflection to get all **Reflection** methods. Then prepare an algorithm that will recognize, which methods are **getters** and **setters**. Sort each collection **alphabetically** by methods names. Print to console each **getter** on new line in format:

* **"{name} will return class {Return Type}"**

Then print all **setters** in format:

* **"{name} and will set field of class {Parameter Type}"**

**Do this without changing anything in "Reflection.java"**

* **High Quality Mistakes**

You are already expert of **High Quality Code**, so you know what kind of **access modifiers** must be set to members of class. Time for **revenge** has come. Now you have to check code produced by your "**Beautiful and Smart**" trainers in class **Reflection**. Check all **field and method access modifiers**. Sort each category of members **alphabetically**. Print on console all **mistakes** in format:

* Fields

**{fieldName} must be private!**

* Getters

**{methodName} have to be public!**

* Setters

**{methodName} have to be private!**

**Part II: Annotations**

* **Create Annotation**

Create annotation **Subject** with a **String[]** element called **categories**, that:

* Should be available at runtime
* Can be placed only on types

**Examples**



* **Coding Tracker**

Create annotation **Author** with a **String** element called **name**, that:

* Should be available at runtime
* Can be placed only on methods

Create a class **Tracker** with a method:

* **public static void printMethodsByAuthor()**

**Examples**



**Output**

