Programming for IoT applications

Lab 1

Exercise 1. Develop in Object Oriented Programming (OOP) a simple calculator. The program will display a menu asking end-user to insert the operation to be performed and the two operands. The output should be a JSON reporting the input operands, the executed command and the result.

The accepted commands are:

- **add**: to add the operands and print the JSON;
- **sub**: to subtract the operands and print the JSON;
- **mul**: to multiply the operands and print the JSON;
- **div**: to divide the operands and print the JSON. CHECK that the operation is possible, if not an exception must be raised;
- **exit**: to close the program.

Validate each output JSON with jsonlint (http://jsonlint.com/)

Example of commands:

```
add 12 4.6
sub 3 12
```

Exercise 2. Extend *Exercise_1* to develop an OOP calculator where each method receives a list of numerical values, instead of 2, and print the result. The output should be a JSON reporting the input operands, the executed command and the result.

Validate each output JSON with jsonlint (http://jsonlint.com/)

Example:

Given the list [1, 2, 4.5, 7], the result of the **add** command is 1 + 2 + 4.5 + 7

Exercise 3. Develop in OOP a program for managing a list of devices The full list of devices is stored in the file “catalog.json” available at this link .

The program needs to load the file and manage the discography providing the following features:

- **searchByName<deviceName>**: print all the information about the devices for the given <deviceName>
- **searchByID <id>**: print all the information about the devices for the given <id>
- **searchByService <service>**: print all the information about the devices that provides the given <service>
- **searchByMeasureType <type>**: print all the information about the device that provides such measure <type>
- **insertDevice:** insert a new device it that is not already present on the list (the ID is checked). Otherwise ask the end-user to update the information about the existing device with the new parameters. Every time that this operation is performed the "last_update" field needs to be updated with the current date

and time in the format “yyyy-mm-dd hh:mm”. The structure of the parameters of the file must follow the one of the ones that are already present

- **printAll:** print the full catalog
- **exit:** save the discography (if changed) in the same JSON file provided as input.

Finally, once the update file has been saved, validate the new JSON with jsonlint (http://jsonlint.com/)