Programming for IoT applications
Lab 1

Exercise 1. 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 catalog, providing the following features:

- **searchByName**: print all the information about the devices for the given
- **searchByID**: print all the information about the devices for the given
- **searchByService**: print all the information about the devices that provides the given
- **searchByMeasureType**: print all the information about the device that provides such measure
- **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 catalog (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/)

Exercise 2. Develop in Object Oriented Programming (OOP) a simple "rule controller" for a Smart Home. Based on simple rules, the program should be able to manage and control temperature and lights of the home.

The program will display a menu asking end-user to insert the action type to be performed. The accepted commands are:

- **add**: to add a new rule (temperature or light)
- **update**: to update an existing rule
- **delete**: to delete an existing rule
- **evaluate**: to evaluate current temperature and/or light level measurement based on the existing rules
- **rules**: for listing existent rules
- **exit**: to close the program

Hint: You can start by creating a super-class *ConditionRule* with an **attribute** *threshold* and a **method** *evaluate*. Then create two sub-classes *LightRule* and *TemperatureRule.* Finally, you can develop a class *RuleController* for rules management.