CSL202 Programming Paradigms | Assignment #2 | Due 5 PM 24-Jan-2014

We need to write program(s) for emulating a smart building. In a smart building one would have a ***controller machine*** whose purpose is to take instructions for controlling ***appliances*** like electric lighting, air conditioning and doors etc. in the building. Instruction set for this machine consists of two instructions:

1. <Sensed parameter value> = SENSE(<appliance>, <parameter>)
2. CONTROL(<appliance>, <parameter>, <new value>)

Here, <appliance> is the identity of an appliance and <parameter> is the attribute of the appliance that needs to be sensed or controlled. E.g., **temperature setting** can be a parameter for an **AC**.

You can assume that all the appliances can be sensed and controlled via some electric signals. The controller machine instructions somehow result in generating required signals (analog or digital) for sensing and controlling the state of various appliance parameters. Write a program that emulates the behavior of said controller machine and the appliances. You can model the appliances using appropriate classes (if you are using an object oriented programming language).

Skeleton code for the program (in Java) is given below. You may use it as the basis or modify/port this logic to any other language. Also, please note that we will be building upon this problem in the subsequent assignments. Therefore, you are advised to be careful about your implementations. That is, you may have difficulties in subsequent assignments if your do not successfully implement this one.

Good luck!

Important instructions

1. Any form of plagiarism will lead to an F grade in the course regardless of other considerations.
2. Though you are encouraged to discuss your design approach and issues with others, but you must not be a party to copying. **Both** the one who is copying and the one who allows his/her code to be copied will get an F grade.
3. You should provide a proper README type of document as part of your submission. It should clearly and briefly explain:
    a. The design approach you took to implement the program.
    b. How to compile (if needed) the source files and how to execute them.
    c. Any special instruction for testing the program.
4. Source code should be well documented with proper comments. Proper naming of functions and variables etc. should be used. Readability of the source code carries partial points.

5. Name your submission in the format: **CSL202-Assignment-XX-<Your roll no.>.zip**. For example, CSL202-Assignment-02-2012CS1234.zip. Not following this naming convention may result in lost submissions for which TA or I will not be responsible.

Skeleton Code

```
public class ControllerMachine {

    // Member variables can go here
    // ....

    public Object sense(Appliance a, String parameter) {
        //TODO: Add proper implementation here for sensing the appliance parameter.
        return null;
    }

    public void control(Appliance a, String parameter, Object paramValue) {
        //TODO: Add proper implementation here for controlling the appliance parameter.
    }

    // Add rest of the class methods etc. as needed.
}

/**
 * Concrete appliance classes will adhere to this contract. It lists all
 * those methods which are common to different appliances.
 */
public interface Appliance {

    /**
     * Returns the list      of possible parameter names available for an appliance.
     * E.g. for an AC one could have Humidity and Temperature as parameters.
     * @return
     */
    String[] listParameterNames();

    /**
     * Returns type of the appliance. E.g. AC, Fan, Light
     * @return
     */
    String getType();

    /**
```

```
     * Returns the identity of the given appliance. For example, 9987
     * @return
     */
    int getIdentity();

    /**
     * Returns the state of appliance. State can be one of the following:
     * ON, OFF, FAILED
     * @return
     */
    String getState();

    /**
     * Returns the value of a given parameter.
     * @param paramName Name of the parameter to be sensed.
     * @return
     */
    Object getParamValue(String paramName);

    /**
     * Sets the value of a given parameter.
     * @param paramName Name of the parameter to be controlled.
     * @param value New value for the parameter which will be set on appliance.
     */
    void setParamValue(String paramName, Object value);
}
```