IIP First Partial - ETSInf
November 16th of 2015. Time: 1 hour and 30 minutes.

It is desired to represent a thermostat which controls the temperature (in Celsius degrees) of a thermal device in a house. Each thermostat is defined accoding to four data items: its identifier (name of the zone of the house where it is situated), its mode (cool for refrigeration, hot for heating), the current temperature at the zone, and the comfort temperature desired by the user. The WWF (*World Wide Fund for Nature*) recomends as ideal temperatures 25ºC in summer and 20ºC in winter.

1. [7 points] You must implement the `Thermostat` class, so you must:

   a) (0.5 points) Define the following public constant class integer attributes:
   - `COOL`, with value 0, represents the refrigeration mode;
   - `HOT`, with value 1, represents the heating mode;
   - `T_IDEAL_COOL`, with value 25, represents the minimal recommended temperature in refrigeration mode;
   - `T_IDEAL_HOT`, with value 20, represents the maximal recommended temperature in heating mode;

   These attributes must be employed every time required (both in class `Thermostat` and `ThermostatManager`).

   b) (0.5 points) Define the private object attributes `name` (`String`), `mode` (`int`), `tComfort` (`int`) and `tCurrent` (`double`).

   c) (1.5 points) Implement two constructors:
   - A general constructor with as many parameters as necessary to initialise all the instance attributes. You can suppose that parameters are always correct.
   - A default constructor (no parameters) that creates a `Thermostat` in `COOL` mode, with name `"living room"`, comfort temperature the one recommended by WWF for refrigeration, and as current temperature a random `double` value in the interval [20,0, 40,0[generated by the `generateRandom` method (next subquestion).

   d) (1 point) Write a private static method `generateRandom` that, given two `double` values x and y, returns a random `double` number in the interval $[x, y[$.

   e) (0.5 points) Write the consultor (*get*) and modifier (*set*) method for the attribute `mode`. You can suppose that the value of the parameter in the modifier will be always correct.

   f) (1 point) Write the `equals` method (that overrides that of `Object`) to check if a thermostat is equal to another, i.e., the other object is a thermostata and they have the same name, mode, and comfort temperature, and absolute difference of current temperatures is lower to 1.0ºC.

   g) (1 point) Write the `toString` method (that overrides that of `Object`) to return a result with a format equal to that of the following examples:

   "`living room, refrigeration mode, TComfort = 25 and TCurrent = 27.5`"

   "`bedroom, heating mode, TComfort = 20 and TCurrent = 18.5`".

h) (1 point) Write a method `differenceWithIdeal()` that returns an integer whose value is:

- 0 when comfort temperature is appropriate to the mode, i.e., it is greater than or equal to ideal in `COOL` mode, or it is lower than or equal to ideal in `HOT` mode.
- The absolute value of the difference between comfort and ideal temperature otherwise.

For example:

- If thermostat is in refrigeration mode and comfort temperature is 22, result must be 3; if comfort temperature is 25 or higher, result must be 0.
- If thermostat is in heating mode and comfort temperature is 22, result must be 2; if comfort temperature is 20 or lower, result must be 0.

2. 3 points Complete the program class `ThermostatManager` to make it create a thermosta and give advice to the user about incrementing or decrementing comfort temperature, in order to accomplish norms of energy efficienty by WWF. For that, you must:

a) (1.5 points) Implement a method with header `public static String advice(Thermostat t)` that checks difference between comfort temperature and ideal temperature for thermostat `t`, and returns a message that indicates if the comfort temperature must be increased or decreased, and in how many degrees, or if the temperature is appropriate.

For example:

- If the thermostat is in refrigeration mode and the comfort temperature is 22, the advice will be `"Degrees to increase: 3"`; if the comfort temperature is 25 or higher, the advice will be `.ªppropriate temperature"`.
- If the thermostat is in heating mode and the comfort temperature is 22, the advice will be `"Degrees to decrease: 2"`; if the comfort temperature is 20 or lower, the advice will be `.ªppropriate temperature"`.

b) (1.5 points) Complete the `main` method in order to make that, after reading from keyboard the values for mode and temperatures with the provided instructions (you can assume that inputs will be correct), performs the following actions:

a) (0.5 points) Create un `Thermostat t` for `"main bedroom"` with the mode and temperature values that have been read by keyboard.

b) (0.5 points) Shows on the screen the created thermostat.

c) (0.5 points) Shows on the screen the advice about energy efficience for that thersmostat.

```
/**
 *  Class Thermostat: represents a thermostat
 *  @author IIP Exam
 *  @version 1st Partial - Year 2015-2016
 */
public class Thermostat {
    public static final int COOL = 0;
    public static final int HOT = 1;
    public static final int T_IDEAL_COOL = 25;
    public static final int T_IDEAL_HOT = 20;

    private String name;
    private int mode;
    private int tComfort;
    private double tCurrent;

    public Thermostat(int m, String n, int tCo, double tCu) {
        mode = m; name = n; tComfort = tCo; tCurrent = tCu;
    }
```

```
    public Thermostat() {
        this(COOL, "living room", T_IDEAL_COOL, generateRandom(20.0, 40.0));
    }

    private static double generateRandom(double x, double y) {
        return x + Math.random() * (y - x);
    }

    public int getMode() { return mode; }

    public void setMode(int nmode) { mode = nmode; }

    public boolean equals(Object o) {
        boolean res = o instanceof Thermostat;
        if (res) {
            Thermostat t = (Thermostat) o;
            res = name.equals(t.name) && mode == t.mode
                && tComfort == t.tComfort
                && Math.abs(tCurrent - t.tCurrent) < 1;
        }
        return res;
    }

    public String toString() {
        String m = "refrigeration";
        if (this.mode == HOT) { m = "heating"; }
        return name + ", " + m + " mode, TComfort = " + tComfort
               + ", TCurrent = " + tCurrent;
    }

    public int diferenceWithIdeal() {
        int res = 0;
        if (mode == COOL) {
            if (tComfort < T_IDEAL_COOL) {
                res = Math.abs(tComfort - T_IDEAL_COOL);
            }
        } else {
            if (tComfort > T_IDEAL_HOT) {
                res = Math.abs(tComfort - T_IDEAL_HOT);
            }
        }
        return res;
    }
}

import java.util.Scanner;
import java.util.Locale;
/**
 *  Class ThermostatManager: program class to test Thermostat class
 *  @author IIP
 *  @version 1st Partial - Year 2015-2016
 */
public class ThermostatManager {

    public static String advice(Thermostat t) {
        String adv = "";
        int degrees = t.diferenceWithIdeal();
        if (degrees == 0) { adv = "Appropriate temperature"; }
        else {
            adv = "Degrees to ";
            if (t.getMode() == Thermostat.COOL) {
                adv += "increase: ";
            }
            else { adv += "decrease: "; }
            adv += degrees;
        }
        return adv;
```

```
    }

    public static void main(String[] args) {
        Scanner kbd = new Scanner(System.in).useLocale(Locale.US);
        // Read from keyboard data for the thermostat
        System.out.print("Thermostat mode (refrigeration/heating): ");
        String mode = kbd.next().trim().toLowerCase();
        int cMode = Thermostat.COOL;
        if (mode.equals("heating")) { cMode = Thermostat.HOT; }
        System.out.print("Comfort temperature for main bedroom: ");
        int tComf = kbd.nextInt();
        System.out.print("Current temperature for main bedroom: ");
        double tCurr = kbd.nextDouble();

        Thermostat t = new Thermostat(cMode, "main bedroom", tComf, tCurr);

        System.out.println("Thermostat data: " + t.toString());

        System.out.println("Energy efficience advice: " + advice(t));

    }
}
```