Sahaj software | inspiring brilliance

Guidelines

At Sahaj, we strive to build high quality software that has strong aesthetics (is readable and maintainable), has extensive safety nets to safeguard quality, handles errors gracefully and works as expected, without breaking down, with varying input.

We are looking for people who can write code that has flexibility built in, by adhering to the principles of Object Oriented Development, and have the ability to deal with the real-life constraints / trade-offs while designing a system.

It is important to note that we are not looking for a GUI and we are not assessing you on the capabilities around code required to do the I/O. ***The focus is on the overall design****. So, while building a solution, it would be nicer if input to the code is provided either via unit tests or a file. Using command line (for input) can be tedious and difficult to test, so it is best avoided.*

Following is a list of things to keep in mind, before you submit your code, to ensure that your code focuses on attributes, we are looking for -

- Is behaviour of an object distinguished from its state and is the state encapsulated?
- Have you applied SOLID principles to your code?
- Have you applied principles of YAGNI and KISS (additional info here)?
- Have you unit tested your code or did TDD? If you have not, we recommend you read about it and attempt it with your solution. While we do not penalise for lack of tests, it is a definite plus if you write them.
- Have you looked at basic refactoring to improve design of your code? Here are some guidelines for the same.
- Finally, and foremost, are the principles applied in a pragmatic way. Simplicity is the strongest of the trait of a piece of code. However, easily written code may not necessarily be simple code.

Sahaj software | inspiring brilliance

Problem Statement
(Average time to write a solution - 4-8 hrs)

A very prestigious chain of hotels is facing a problem of huge consumption of electricity bills for its electronic equipments. The common equipments, like lights, ACs, etc are currently controlled manually, by the hotel staff, using manual switches. Hotel Management wants to optimise the usage of electricity consumption and also ensure that there is no inconvenience caused to the guests and staff. So, it has installed Motion Sensors at appropriate places and have approached you to program a Controller which takes inputs from these sensors and controls various equipments.

The way the hotel equipments are organised and the requirements for the Controller are listed below:

- A Hotel can have multiple floors
- Each floor can have multiple main corridors and sub corridors
- Both main corridor and sub corridor have one light each
- Both main and sub corridor lights consume 5 units of power when ON
- Both main and sub corridor have independently controllable ACs
- Both main and sub corridor ACs consume 10 units of power when ON
- All the lights in all the main corridors need to be switched ON between 6PM to 6AM, which is the Night Time slot
- By default, all ACs are switched ON, all the time
- When a motion is detected in one of the sub corridors the corresponding lights need to be switched ON between 6PM to 6AM (Night Time slot)
- The total power consumption of all the ACs and lights combined should not exceed *(Number of Main corridors * 15) + (Number of sub corridors * 10) units of* per floor. Sub corridor AC could be switched OFF to ensure that the power consumption is not more than the specified maximum value
- When there is no motion for more than a minute the sub corridor lights should be switched OFF and AC needs to be switched ON

Motion in sub-corridors is input to the controller, which needs to keep track and optimise the power consumption.

Write a program that takes input values for Floors, Main corridors, Sub corridors and takes different external inputs for motion in sub corridors. For each input, the program prints out the state of all the lights and ACs in the hotel. For simplicity, assume that the controller is operating at the Night Time.

Sample input and output below -

- `Number of floors: 2`
- `Main corridors per floor: 1`
- `Sub corridors per floor: 2`

Subsequent Inputs from Sensors	Output from controller for corresponding sensor input
Default state (when the program is first run)	Floor 1 Main corridor 1 Light 1 : ON AC : ON Sub corridor 1 Light 1 : OFF AC : ON Sub corridor 2 Light 2 : OFF AC : ON Floor 2 Main corridor 1 Light 1 : ON AC : ON Sub corridor 1 Light 1 : OFF AC : ON Sub corridor 2 Light 2 : OFF AC : ON
Movement in Floor 1, Sub corridor 2	Floor 1 Main corridor 1 Light 1 : ON AC : ON Sub corridor 1 Light 1 : OFF AC : **OFF** Sub corridor 2 Light 2 : **ON** AC : ON Floor 2 Main corridor 1 Light 1 : ON AC : ON Sub corridor 1 Light 1 : OFF AC : ON Sub corridor 2 Light 2 : OFF AC : ON
No movement in Floor 1, Sub corridor 2 for a minute	Floor 1 Main corridor 1 Light 1 : ON AC : ON Sub corridor 1 Light 1 : OFF AC : **ON** Sub corridor 2 Light 2 : **OFF** AC : ON Floor 2 Main corridor 1 Light 1 : ON AC : ON Sub corridor 1 Light 1 : OFF AC : ON Sub corridor 2 Light 2 : OFF AC : ON

Since the hotel management is trying this for the first time, it would be changing the requirements as to *which* electronic equipments are controlled and the *criteria* based on which they are controlled. Therefore, the solution design should be flexible enough to absorb these changes without a need to make significant changes in the program.