

Assessment: Smart Home

Java, being a multi-platform programming language can be a good choice to power a smart home controller!

A smart home application consists of a “hub” and multiple “devices”.

The hub is the thinking brain that controls the connected devices when receiving signals from various sensors.

Typical devices connected to the hub are:

- Several lamps
- TV, possibly other audio / video devices
- Heater
- Air Conditioner
- etc.

The hub maintains a desired state, such as lighting and temperature settings and makes sure that the devices operate up to that state.

The hub constantly runs a program that checks the schedule provided for each device by its user and compares it with the state of those devices and when needed, sends commands to those devices.

To do that, the hub relies on the signals that sensors provide:

- Light intensity, from the light sensor
- Temperature, from the thermometer
- Motion, from a motion detection (if you’d have a smart lamp)

For example, if the provided schedule is:

```
lamp living_room on from 18:00 to 21:00
media tv on from 11:00 to 15:00
lamp bedroom on from 21:00 to 23:00
conditioner main 23 always
```

The task of the smart home hub is to monitor the given devices and compare if they are in the desired state. If the `living_room` lamp is off and it is 18:00, then the hub should send a signal `signal("on")` to the lamp.

Develop an application for the smart home hub, that models the described behaviour.

Since we can’t provide physically controllable devices and sensors during the assessment, we’d like to simulate them simply by some objects in the program.

Also notice that we need to model the events that trigger the sensors, which have somewhat random behaviour.

Important design points are:

- Maintaining the schedule from a text file with the sample syntax given above

General syntax:

<Device Type> <Device Name> <Value> <Schedule>

<Device Type>: lamp, media, conditioner, can be extended

<Device Name>: Any name, a valid string value

<Value>: For some devices it is on/off and for some it is a numeric value (temperature)

<Schedule>: Either "From X to Y" or "always"

- Using Maven to add **any** required dependency and packaging as a single, executable JAR file. It is OK to use any library that you think may help you.
- Using generics to model various devices and sensors, as the list of devices and sensors might expand in the future.
- Using threads to simulate events, as well as control devices, as the hub is a multi-tasking component: While it is increasing the room temperature, it is also monitoring the light intensity and maybe switching on / off the light bulbs.
- Efficient usage of threads (consider liveness, locks, etc.).
- General usage of design patterns, OOP, correct and simple (elegant) design.
- Clean code, meaningful naming, and consistent style.
- **Bonus:** Defining REST endpoints to control the smart devices via HTTP can be counted as extra points. Sample HTTP requests can be (these are just examples, you may define the endpoints the way you find suitable):

1) Example: Get state of connected lights

GET http://localhost/state/lights

Sample response:

```
{
  "devices": [
    {
      "type": "lamp",
      "name": "Living Room",
      "state": "on"
    },
    {
      "type": "lamp",
      "name": "Bedroom",

```

```
        "state": "off"
    }
]
}
```

2) Example: Schedule the TV to be tuned on at 18:00

POST <http://localhost/schedule/media>

Payload:

```
{
    "type": "media",
    "name": "tv",
    "state": "on",
    "state_schedule": "18:00"
}
```