

Assignment 1 - Description

Smart Home Controller

In this assignment you will work on developing a Smart Home controller. The Controller receives data and notifications from several sensors spread around a building, and based on these it controls several actuators, i.e., devices that change the state of the house. A major challenge is integrating various devices from different vendors so that they all work together with the controller.

- **Sensors.** Sensors are objects that collect information from the environment. In our case these will be **Temperature Sensors** and a **Thermostat**, i.e., a device that lets users define the desired temperature.
- **Actuators.** Actuators are objects that can perform changes in the environment. In our case these are a **Furnace**.
- **Controller.** A controller is a device which receives events and data from Sensors and issues commands to the Actuators based on hardcoded logic.

Sensors, Controllers and **Actuators** of various vendors are able to work together through the use of Interfaces. Specifically, a consortium of smart home vendors, called "*ITEC3030 Smart Home Consortium*" worked together to produce the **Simple Smart Device Collaboration Standard (SSDCS)** in form of a set of **Java Interfaces**.

The Interfaces can be downloaded from eClass ([Standards.jar](#)) and the JavaDoc-generated Documentation can be found [here](#). For this assignment you will need to complete the following exercises:

Exercise 1

Study the documentation of the SSDCS. Use draw.io to generate a **class diagram** that describes the SSDCS as documented.

Exercise 2

A Smart Home installations contractor "*Smart Builders*" needs to put together a smart apartment using a Controller that they build. The apartment consists of two rooms:

- **A Living Room**, which contains two sensors OmniTempSensorXS3 by OmniTemp Inc.
- **A Bedroom**, which contains another two OmniTempSensorXS3 by the same brand.

An SSDCS-compliant driver for OmniTempSensorXS3 can be found on eClass ([ClassLibrary-TempSensor.jar](#)) and documentation [here](#).

The apartment also has an SSDCS-compliant Forced-Air Furnace by a very well reputed brand called Sahara GmbH. The specific model used is SaharaFurnaceHL42 and the driver can be found on eClass ([ClassLibrary-Furnace.jar](#)) and documentation [here](#).

The developers of “Smart Builders” have developed some starter code as follows:

- **Room.java**, which they use to model the room, as well as keep track of all the devices that are installed there.
- **Building.java**, which is a collection of Rooms and also includes the furnace.
- **Controller.java**, which models the controller device which collects data from sensors and sends commands to actuators via the Room, building classes, depending on where the devices are installed.
- **(*NEW*) QuickThermostat.java**, which is a stub of a thermostat for running Exercise 2.
- **Main.java**, in which the apartment in question is built.
- You will also need to this [Scenario.jar](#) plug-in. Your solution should operate under any scenario provided by the instructor.

The code unfortunately is missing some key parts. Your job is to put together the java files and the libraries in e.g., an IntelliJ/Eclipse project, complete the missing parts.

Exercise 3

Assume now that you are the company that develops thermostats, called ThermoSet Ltd. In this assignment, thermostats are devices that allow building habitants set their desired temperature. SSDCS has an interface for those. ThermoSet’s thermostat, ThermoSetX19, must therefore implement that interfaces. It also needs to implement a predefined front end. You need to do the following:

- Download this [GUI plug-in from eClass \(ThermostatFrontEnd.jar\)](#) and study its Java doc [here](#).
- Implement SSDCS-compliant thermostat ThermoSetX19 and plug the above GUI in it.
- Develop its documentation Using JavaDoc.

Once this is done help “Smart Builders” incorporate the ThermoSetX19 in their design. Update their code and submit updated result.

All required .jar and .java files are in “assignment1_attachments” on eClass.

Good luck!