Java Lab 7

In this lab, you will practice with class relationships.

1. Create a project named Lab7 with a class named Lab7. Copy these classes from Lab5: Sensor, Device, Room, and Alarm

2. Create these classes; each will have an ArrayList<thing> of the objects of the type indicated. Each should the following methods:
add(thing) – adds the object to the ArrayList;
display( ) – displays the items in the ArrayList by looping through them and calling their toString;
default constructor – new up the ArrayList.

- SensorCollection; contains Sensor objects, named sensors.
- AlarmCollection; contains Alarm objects, named alarms.

3. Add a SensorCollection, and AlarmCollection, and one Device to Room. New-up the SensorCollection and the AlarmCollection in Room's constructors, but not Device. Add these methods to Room:
addSensor(Sensor s), that adds a Sensor to its SensorCollection;
addAlarm(Alarm a), same for Alarm;
addDevice(Device d), that adds the one Device object to the Room.
display( ) that prints its toString( ), prints the Device toString( ), then calls the display( ) method in SensorCollection and AlarmCollection.

4. In main( ), create a Room object named room1 with Lab6's data: 12.0, 15.0, kitchen, #1. Create a Device with Lab6's data: extinguisher with data fire extinguisher, kitchen, #1; add it to the room. Using a counting for loop, add 5 Sensor objects to the room with Lab6's data in each: 0.0, 120.0, 68.0, 1.0, kitchen, temperature, but make the id the loop counter + 1 (that is, id#'s from 1 to 5). Using a counter for loop, add 3 Alarm objects to the room with this data in each: "Ding! Ding!", and the loop counter + 1 as the id. Call room1's display( ) method.

5. Create a new Device object named chemicalFoamer with data chemical foamer, kitchen, #2. Set this as the device in room1. Call room1's display method.

6. No need to turn this in: draw a UML diagram of the 7 classes.

Deliverable: Zip up all your .java files and upload it to Canvas.