SmartHome Documentation

1. Part 1: In this section, classes for 3 types of devices, which are smart lights, thermostats, and security cameras, are implemented. Also, another class for a random simulation has been implemented.
   1. SmartLight Class:
      1. This class has 3 attributes. An id, status which determines whether the device is on or off, and brightness. These attributes are assigned in the constructor. There are also getters for each of them. Also, there is a getDeviceType method which returns the type of the device.
      2. The two functions turnOff and turnOn are responsible for changing the status attribute of the device.
      3. turnOn first checks if the device is already on or not. If it is on then the function returns. If it is not on, then the currentBrightness variable is initialized with 0 and maxBrightness will be assigned with the brightness given in the constructor. When the device is turned on, the currentBrightness will increment until it reaches the maxBrightness. Then the function ends with setting status to on.
      4. turnoff works the same way as turnOn. The difference is that the brightness decrements from device brightness to 0.
   2. Thermostat Class:
      1. This class has 3 attributes. An id, status which checks if the device if on or off, and temperature. Each have their own getters as well as an extra getter for the device type.
      2. The turnOn and turnOff functions set the status of the device, but check if it is already on or off in the beginning.
      3. The setTemperature method changes the temperature of the device by checking if the device is on or off at the beginning. If it is off, then nothing happens and the function returns nothing. Then a minimum and a maximum temperature is defined. If the given temperature as the parameter of the function is out of the range of the min and max value of the temperature, then program ends with an error. If it is the same value as the current temperature, then the program declares that the temperature is already set to the given value and ends. Finally, if all of the conditions above are passed, then the temperature changes.
   3. SecurityCamera Class:
      1. This class has 3 attributes. An id, status which checks if the device if on or off, and securityStatus. Each have their own getters as well as an extra getter for the device type.
      2. turnOff and turnOn functions set the value of the status attribute but check before doing so whether it already is on or off.
      3. The setSecurityStatus function has a securityStatus parameter. It checks if the camera is off first. If it is off then the program ends. If not, then the program checks if the given parameter is valid. Valid values are “ON” and “OFF”. If it is not valid, then the program ends with an error. Next the program checks if the securityStatus is already set to the given value. If it is it sends a message saying that it is already set. If not then the value of the securityStatus changes.
   4. RandomSimulation Class:
      1. This class has 3 attributes which are objects of the 3 devices defined above.
      2. There are 3 functions which start a simulation for each of the devices that will randomly change the values of their status and their special attribute. Special attribute is brightness for the SmartLight, temperature for the Thermostat, and securityStatus for the SecurityCamera. Each simulation first takes a random value for the status of the devices first which will either turn on or turn off the devices. Then a random value is given to the setter of the special attributes of each class. At the end of each iteration of the 3 functions, the program waits for 1 second before starting the next iteration.
   5. In the main body of the script, one object for each class above has been instantiated and the simulations start.
2. Part 2: In this part, the AutomationSystem class has been implemented which takes 3 lists made of objects of the 3 devices and starts an automation task for a room in the smart home. Each room is represented by the id of the devices. Which means that if the id of a light, a camera and a thermostat are the same, then they are in the same room.
   1. AutomationSystem Class
      1. Each attribute has a getter which returns the list of the devices of the same type.
      2. The addDevicesToTheDataset function takes a new device for as a parameter. It checks the type of the given device and whether or not it is already in the list of the relevant devices. If the conditions pass then the object is added to the list. Otherwise, the program will return with an error stating that the given device isn’t valid.
      3. The setupRoom function takes an id as parameter and declares a three-element list called room which is initialized with None for all three elements. The function will iterate through the 3 lists of devices and looks for the device with the given id. The first element of room will be the smart light, the second element will be the thermostat, and the third element will be the camera. At the end, it returns the room list.
      4. checkDeviceStatus function takes three parameters which are objects of the 3 devices. First checks if the camera is on or off. If the camera is off then the function returns False. Otherwise, it continues checking for the status of the light and the thermostat. If they are off, it turns them on and after all of that it returns True.
      5. randomMotionDetection function takes a camera object as parameter. It finds a random number between 0 and 10 and checks the number. If it is between 0 and 5, then the security status of the camera is set to OFF. Otherwise, it is set to ON.
      6. The deviceActions method takes an object of each of the devices. If the camera security status is OFF, then the brightness of the light is set to a small number (20) and the temperature is set to a normal value (20). If the camera security status is ON, then the light’s brightness is set to 100 and the temperature of the thermostat is set to -10.
      7. The automationTask function takes id as a parameter and uses the helper functions described above to run tasks automatically. The id parameter represents the room which is set using the setupRoom function. If the checkDevicesStatus function returns False, then the program stops since the camera is off. Otherwise, it calls the randomMotionDetection and deviceActions functions in a loop 10 times. After each loop, the program waits for 3 seconds.
   2. In the main body of the script, a few objects of each device has been defined and listed. The AutomationSystem constructor takes the lists to instantiate an object. Then 3 more devices are added to the list of devices. At the end, the automation system tasks start for the 4 different rooms. After each room’s task is done, the program waits for 1 second.
3. Part 3: The AutomationSystemUnittest and the RandomSimulationUnittest scripts provide test cases for the corresponding classes inside SmartHome script. They check different functions for inside the two classes.
4. Part 4: The SmartHomeGUI class is responsible for creating a menu where users can set up different attributes of the devices such as turning them on and off and setting their special attribute. The GUI also provides a button which starts an automation task for users to see how the devices work. This automation task works the same way that the AutomationSystem class works in Part 2. Depending on the camera being on or off, the program will either start or won’t. Then user must turn on the light and the thermostat as well. The result will be the same as it was implemented in Part 2. There’s also a clear log button which allows user to clear the log messages and start over.