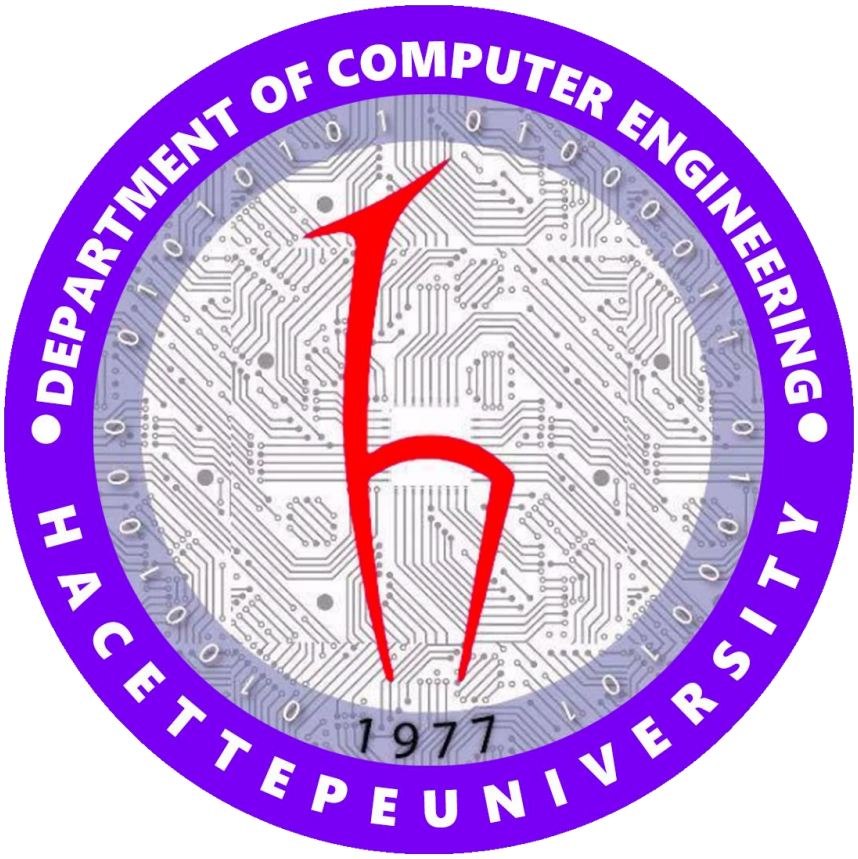
Hacettepe University

Department Of Computer Engineering

BBM 104 Assignment 2 Report

Bora Aslan - 2220356080 21.04.2023



Index

Problem of the smart home system:

In this coding assignment, the problem is the creation of a smart home system that has various types of smart devices. These devices have general attributes and behaviors that stem from their nature as smart devices, and in addition, each device has unique, authentic attributes and behaviors that must be taken into consideration.

Solution approach to the problem:

To create a complex smart home system for example handling all devices or making changes like removing them or renaming and more situations that are not specific to one device type can be easily managed by using Object Oriented Programming. For more specific details inheritance can be used by creating each device type as a subclass and creating authentic methods. For complex processes like switching devices On Off, calculating megabytes or watts, and for plugs start times of on devices can be stored to later calculate the consumptions. Lastly for sorting devices in wanted order, whenever a device with switch time switches, its switch time should be stored to sort devices. The order should be like this:

<top>

The device with the most recent switch time > Device with furthest switch time > Device that switched most recent > Device switched in furthest date > Devices that never had switch time or switched sorted by creation times (newer to older)

<bottom>

Explanation of problems while implementing solutions and how they were solved:

While implementing, ordering the devices, plug in plug out system, megabyte and watt consumption, and creating and storing objects were a big problem.

- It took some time to understand what the logic behind sorting was after realizing that devices that switched are ordered by the time they switched. Implementing was relatively easy.
- Plug in and out system was confusing at first because they are naturally involved in switch processes because they are in interaction with initial status, and both are so critical for calculations. The solution I found was mapping out possible situations and what would happen it was not hard to implement.
- Since megabyte and watt consumption is affected by nearly all the commands, before implementing consumption necessary commands should be implemented because they are crucial for consumption to work as it is intended. To calculate the consumption duration of start and switch time should be taken. So, to do that start time and switch time should be assigned in needed situations. For example, start time should be assigned if the device is working, and switch time should be assigned by related commands like set switch time and if the device will be closed with a switch off command.
- Creating objects, checking the command line for errors then storing was hard to accomplish. Array list for holding smart devices should be created and it should hold the superclass type. After an add command is read, the line and smart object array list must be sent to the device's own save object method then necessary checks should be done to save the object successfully. This way objects can be held in creation order before sorting and also while trying to reach an object only from its name search can be done easier.

Benefits of this system:

This system tries to make people's life easier by having a platform that gives access to them to create and handle their smart devices from one place. Without this system handling smart devices would be chaotic.

Benefits of OOP:

OOP gives a possibility to programmers to don't write DRY (do not repeat yourself) code. By using OOP programmers can organize sub and super classes easily and can avoid writing code to each separate class that can be used via other classes.

What are the four pillars of OOP and UML?

Inheritance lets classes inherit properties and attributes from another class. This helps to code reuse and the creation of more specialized classes without having to write the same code again.

Encapsulation is used to block direct access to details of objects. This way details can be reached via other methods thus it can increase security because those methods can reject invalid entries.

Polymorphism is used to make the objects give different outputs to the same method which can make it flexible to organize and manage the code, especially for larger projects.

Abstraction involves focusing on the essential features of an object and ignoring the rest. This allows for the creation of more general, reusable code that can be applied to a variety of situations.

UML is a blueprint of a project which shows the attributes and properties of different classes and the hierarchy between classes.

UML Diagram of OOP Design and Explanation of the UML Diagram

Main class calls MainCoordinator then MainCoordinator coordinates the whole process by calling necessary methods. TimeCoordinator is responsible for all operations related to time. InputReader and OutputWriter classes are responsible for File I/O, ZreportUtilities class has the essential methods checkForSwitches, calculator, and sort to prepare the ZReport. SmartObject super class has 4 subclasses for each smart device and general commands and the commands that apply to more than one type of object are written in SmartObject class. In the devices classes, some methods are specific to that device.

Resource: IntelliJ IDEA Ultimate uml diagram generator