



BBM104

Assignment 2: Smart Home System

Yunus Emre Uluer

2210356102

Hacettepe University

Department of Computer Engineering

21.4.2023

INDEX

Smart Home System Problem	1
Problem Solution	1
Difficulties and Their Solutions.....	1
Benefits of the System	2
Benefits of OOP	2
Four Pillars of OOP and UML.....	2
UML DIAGRAM	3
Explanation of UML.....	4

Smart Home System Problem

In this project, there are some smart devices which can be controlled by smart home system. Every device has some operations in common. But there are 4 different devices which can make unique operations each. The system is controlled by different types of commands and functions. After these operations, an output should be given to the user. Errors and Wrong commands should be handled by the program and the program should also inform the user.

Problem Solution

In this project, Object oriented programming must be used. Defining the Smart device class, which is the super class of all smart devices, Functions that are common can be handled easily. Also, the list of the devices can be created with Smart device class. Device control class is created for controlling the list, time, some operations, and exceptions. By using polymorphism and inheritance, many of the functions can be simplified.

Difficulties and Their Solutions

While coding the OOP, not much difficulty was encountered. However, struggle was experienced with string manipulation and exception handling. In the string manipulation section, a Commands class was created, and all actions were performed using switch cases. Although this may not be the best solution, there was not enough time for a better solution. Difficulty was experienced in exception handling due to the inheritance of exceptions in Java, and it was resolved by creating a custom exception as there was not enough experience in this area.

Benefits of the System

Benefits of this system are controlling devices, setting different modes on different devices. The most important feature of the system is controlling not only the devices but also the time. Time calculations and switch times make the system very efficient. Smart plugs and cameras consumption methods are also providing sufficient settings on the system. System can make most of the operations single handedly and fast.

Benefits of OOP:

Object-oriented programming (OOP) has lots of benefits. One of them can be divide-conquer. This can be very easily done with OOP. Also, adaptations on OOP are easier than the other methods. Because reusability of the programs with OOP can be used and adapted according to the program. For example, in our project different types of devices are used. If the user of this system wanted to add another device, it would have been done by simply adding another class that inherits smart device. The most important features of OOP are four pillars of OOP.

Four Pillars of OOP and UML:

Abstraction: Simplifying the complex parts of the problem and making it abstract. It can be done by abstract classes or interfaces.

Encapsulation: Encapsulation is hiding unnecessary information from the user and making operations on another layer. Encapsulation provides abstraction.

Inheritance: Inheritance allows some class to take other classes methods, behaviors. This provides better implementation and understanding of code. Interfaces and abstract classes also provide better implementation.

Polymorphism: Polymorphism is a principle that allows different behaviors on same method. Variety and flexible codes can be written with polymorphism.

UML: UML can be simplified as models of codes. UML visualizes the code and provides better understanding. It represents classes, relationships of classes, properties and behaviors.

UML Diagram:



Explanation of UML

4 types of smart devices are introduced in this project. Every one of them has similar fields or methods. Smart Devices class solves this problem. Since there is no such device named smart device, smart device class is abstract class. Has switch time, status and on off times, which is important for camera and plug devices. Most of the methods are created for polymorphism and exception handling at the same time. Getters and setters' methods are the important methods of this class. Smart Lamp and Smart Color Lamp are almost the same device except for color methods. So smart lamp inherits smart color lamp. Color lamps use its super methods most of the time. Every subclass has different fields and methods.

Device controller class holds the list of the smart devices, output content and time variable. Usage of its methods can be simplified as sorting, exception handling and operating. This class controls the flow of time.

Commands class takes commands and calls. It is created for cleaner main function and abstraction. Takes commands and calls suitable functions. Exception messages is printed in this class.

DeviceExceptions is a custom exception that extends Exception. It is a unique exception that is particular to various errors.

FileInput and FileOutput classes are for handling I/O.

Private modifier is used for fields because it is a better approach to adaptation.