

HACETTEPE UNIVERSITY

DEPARTMENT OF COMPUTER ENGINEERING

BBM104 – ASSIGNMENT 2 REPORT

Gazi Kağan Soysal

(2210356050)

19/04/2023



CONTENTS

1)What is the Problem?.....	3
2)Solution Approach.....	3
3)Problems Encountered and Solutions.....	3
4)Benefits of This System.....	3
5)Benefits of OOP.....	4
6)What are the four pillars of OOP and UML?....	4

1)What is the Problem ?

With the development of technology, many electronic devices have entered our lives. When there are so many devices we use in our home, it is difficult to manage them. Since we are all human beings, we sometimes forget the necessary turning off and on of the devices or we are too lazy to adjust it. In this project, we are asked to manage these devices automatically.

2)Solution Approach

As all of the devices we need to manage have common features, they also all have their own unique features. In addition, there are some common operations to be applied to the devices and those that can be applied to all of them separately. For this reason, it is necessary to use OOP logic. In OOP logic, it will be easier to perform the operations to be applied when the devices are classified with a correct forestation.

3)Problems Encountered and Solutions

3.1) First of all, it would be easier to use a data structure like the dictionary in Python to keep track of the devices. For this, HashMap is used in Java, but LinkedHashMap is used, in which the ordering of the items is taken into account, since ordering the devices is also important.

3.2) When sorting by switch time, two separate LinkedHashMaps were used because an error occurred while trying to sort those without switch time. This use also had the following benefit. A device whose switch time is deleted should appear before other devices without switch time. Using these 2 separate LinkedHashMaps also helped when sorting across devices.

3.3) It would be difficult to manage the string to control the time. For this, Java's LocalDateTime class was used. Time-related operations became easy thanks to the unique methods of this class.

4)Benefits Of This System

Thanks to this system, for example, you go to bed and wake up at the same time every day. Your lamp will turn on automatically when you go to bed and it will turn off automatically when you wake up. If you're leaving the house in a hurry, you won't have to wonder if I turned off the lamp. As another example, let's say you plugged your phone into the charger and you want it to be unplugged after a certain time because you think it is harmful to keep it plugged in for a long time. You can get rid of this worry by setting a shutdown time for your outlet. In short, you will not need to keep track of your devices. You will be able to preset what you want to happen when.

5)Benefits Of OOP

Using OOP definitely made the job easier. While writing code, an abstract operation is done and it is difficult to think. Using OOP, it is possible to think of these objects as if they actually existed. For example, when creating a device class, it may be thought that it is necessary to new it because in real life a new device comes into play, but this is not necessary for management classes because there is only one management part. Apart from that, devices can be easily classified according to their properties using inheritance. This makes them easier to manage. A cleaner image emerges as each operation within the project can be classified separately. If a change needs to be made later, this change can be implemented easily. There is no need to make any changes to the whole code. By making changes only in the part that the change is related to, the desired operation can be easily applied.

6) What are the four pillars of OOP and UML?

6.1) The first of these is inheritance. Inheritance allows us to establish the parent-child relationship between related objects. It makes our classification easier.

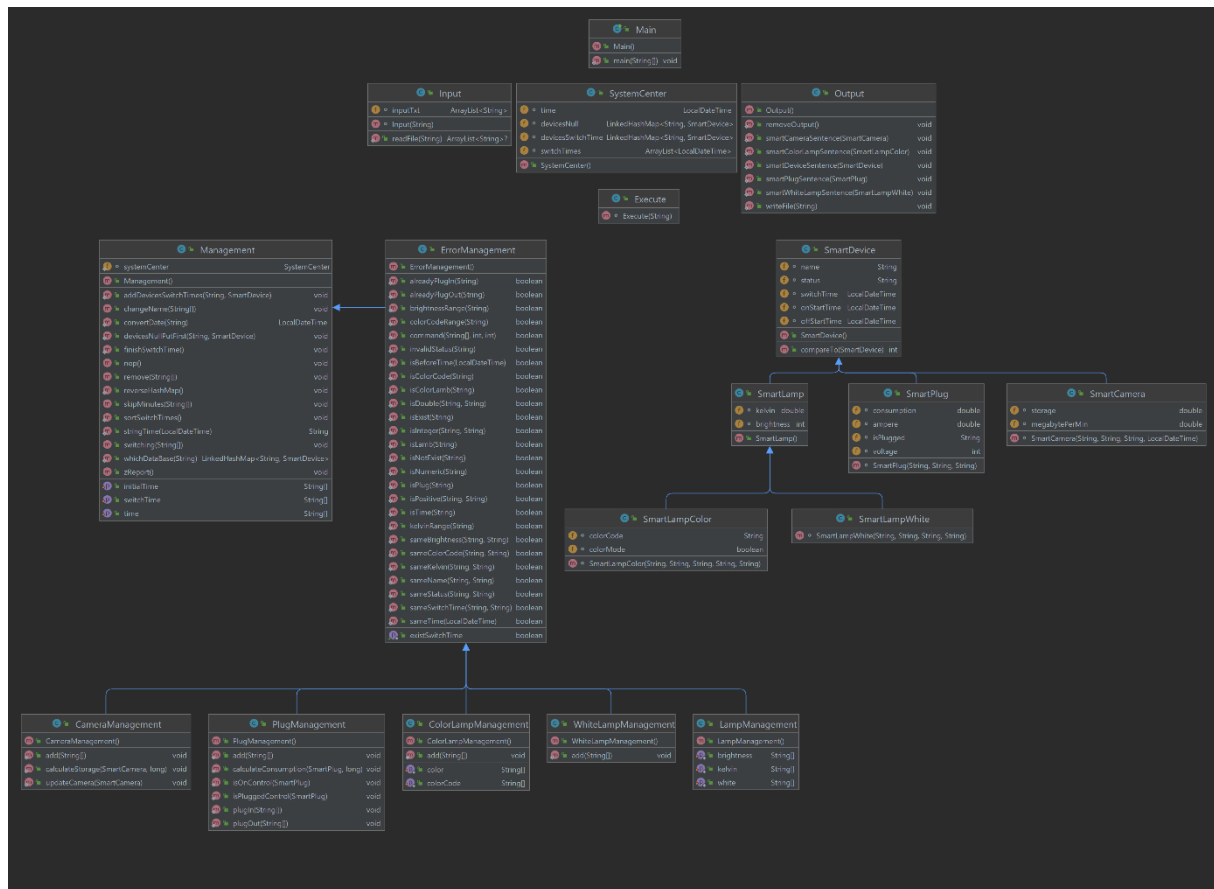
6.2) The second is polymorphism. Polymorphism allows us to use objects from the same ancestor more functionally by using this common ancestor.

6.3) The third is abstraction. Abstract classes contain only the required properties and methods. No unnecessary details.

6.4) The fourth is encapsulation. It determines the access level of the encapsulation fields. It can enable Fields to set or get only, or disable both.

6.5) UML maps out the written code. It can be seen what is connected with what. It provides a general outline of the code and makes it easy to understand.

7)UML Diagram



In the main class, the names of the input and output files are taken and sent to the execute class where the program will be executed. The Execute class uses the Input class to get the information in the input file and calls the relevant method of the required management class. Basic information is kept in the SystemCenter class. In the Management class, there are methods of general operations and auxiliary methods that we will use while performing these operations. A subclass has methods to check for errors. Below that, there is a separate management class for each device and there are methods specific to that device. On the right, there is the SmartDevice, which is the common ancestor of all devices, and it has features that are common to each device. Other devices are subclasses under SmartDevice and they have their own unique features. The constructors of these device classes are also used when adding devices. The Output class is also used to print the necessary texts.