Hacettepe University

Department Of Computer Science

BBM104 Assignment 2 Report

Berke Abdullah Yıldız – 2210356100

20.04.2023

CONTENTS

DEFINING PROBLEM

Humankind is a lazy creature by nature and technology has developed because of this laziness. People have developed technology to get their work done in less time and with less energy. This is one of the technologies developed in automation. It was ensured that the processes performed were independent of the human, and the machines worked according to the commands entered. This prevented people from spending energy for that job.

SOLUTION

While solving this assignment, the problem was first understood. Then, the features of the devices to be used in the project were examined and the structures of each device were designed. These structure designs were made with the OOP concept. Because each of the 4 different devices has the same feature as well as different features and there may be more than one of these devices. Considering all these factors, it can be seen that the OOP style makes our work easier to identify and control devices.

After the devices were defined, the command list was read and the structures of the actions to be performed were established. While doing this, possible errors were checked and this smart home system was successfully installed.

PROBLEMS ENCOUNTERED

First of all, the first problem when writing code blocks in java was forgetting to put the {} signs. It took some getting used to because python doesn't use such flags.

Secondly, it was a little difficult to understand the structures used, namely the concept of OOP, because it was the first time such a structure was seen and this made it difficult for us to understand.

Third, it was difficult to switch from python to java. Because some operations can be done much easier in python. For example, in python, comparison operations can be written directly, whereas in java, variables are compared with operations like .equals.

Finally, while python doesn't specify the type of variables, java made it difficult to specify

BENEFITS OF SYSTEM

The biggest benefit of designing this smart home system is to make people's work easier. For example, people will now automatically turn off the lamp instead of going physically and specifying the time when it should be turned off.

As another benefit, it creates more time opportunities for people to spend more time on tasks that are more difficult for them, rather than spending time on such simple tasks. For example, a person with a large project can be given the opportunity to spend more time on their project by preventing them from spending time on such transactions.

Finally, to give an example, one of the benefits of this system for people is that a family who goes to their house by car can turn on the heating of their house without going home and heat their house before the family goes home.

BENEFITS OF OOP

The main purpose of using OOP is to control multiple objects easily. There are multiple devices in our project, and each device has more than one different device. Rather than writing code for each device separately, establishing a certain structure and creating and controlling all our devices from there provides great readability. If OOP was not used, unnecessary code would be written for each device, but the same code block was used for each object defined thanks to OOP. All that needs to be done is to call the required code block and enter the values of the appropriate constructor.

OOP CONCEPT

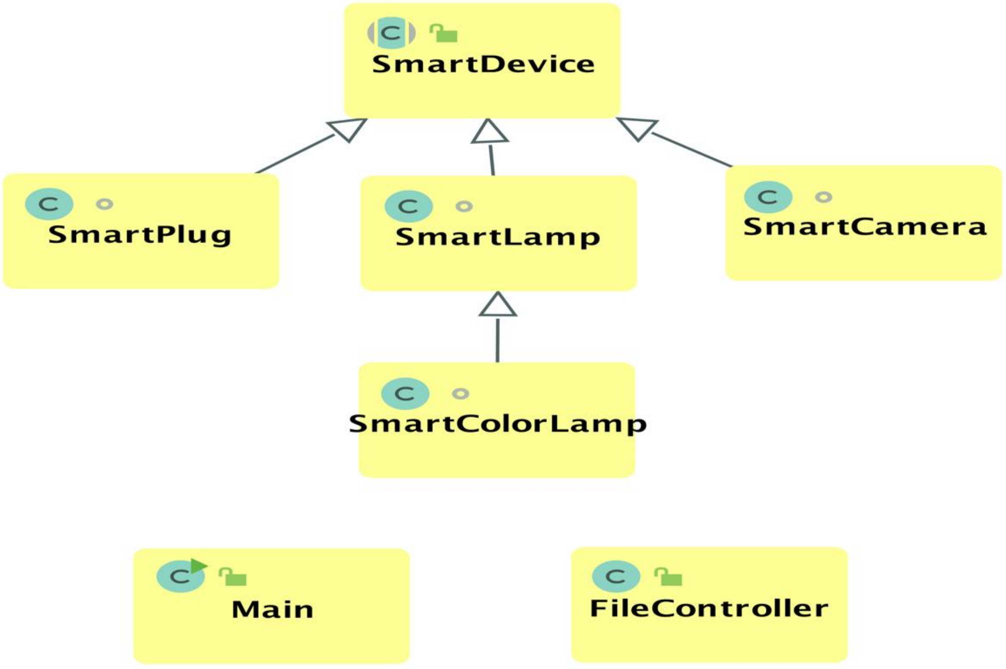
1-Encapsulation: This feature determines how much of the created objects can be accessed. For example, by making the name of the device "private", access to the name of this device from external systems is prevented.

2-Inheritance: This feature allows the superclass's property to be transferred to subclasses. For example, since each device has a name, the name variable is created in the superclass (SmartDevice) and this feature is inherited to subclasses (ex. SmartLamp).

3-Polymorphism: It is the optimization of a function in the superclass according to the objects in the subclasses themselves. In other words, the superclass reference can hold all subclass objects. An example is to create a SmartLamp object from the SmartDevice class.

4-Abstraction: It is a structure that has the common features of subclasses but has not yet been defined. A property is defined in a superclass and other classes that derive from that class write those abstract methods according to themselves. For example, each device has defined the isOn and setOn commands in the SmartDevice superclass.

UML DIAGRAM



First, the Main class only contains the parts that start the processes.

The FileController class is the class that reads and processes the whole file.

SmartDevice class is super class of all objects and all objects extend SmartDevice. Also, SmartDevice is an abstract class and contains common properties of subclasses.

SmartPlug is the class that creates plug objects.

SmartCamera is the class that creates camera objects.

SmartLamp is the class that creates lamp objects and is also a superclass of SmartColorLamp.

SmartColorLamp is the class that creates color lamp objects.