Assignment 2

Smart Home System

Ömer Kayra Çetin

B2210356060

BBM104 Spring 2023

Introduction to Programming Laboratory II

Görkem AKYILDIZ

21.04.2022 23:59:59

Table of Contents

Defining The Problem

In this project, a smart home system must be built. This smart home has four devices which are smart lamp, smart color lamp, smart plug and smart camera. These devices need to be operated accordingly to the given commands and time. The commands are read from an input txt file and an output txt file is generated where the information about the commands is written.

My Solution

Firstly, I implemented the device class hierarchy where every device has the same upper class and color lamp is a descendant of lamp. After that, I decided to have an operator class where the commands will be operated. I wrote command methods according to the commands in the input files and matched them in a execute command method. I also created a class which had my custom exceptions to help me catch errors easily. I put together and run everything in Main class.

Problems and Solutions to Them

First of all, managing time was a hard work for me but I overcame it with some research and OOP thinking. Secondly, correctly ordering the device list was a problem for me. I tried lots of things until I realized that I needed to use *Nop* command to not switch all the switches at the same time whenever I needed to change time. This way, I could preserve the order of the list. Finally, I had the biggest problem with handling errors. There were many of them and since I implemented the project with not thinking about how to handle them, adding them later was hard. In the end, with using lots of if else statements and custom exceptions I managed to do them.

Benefits of The System

This system allows users to manage their smart devices with various commands. They can turn their devices on or off when they need to be without them being at home. Also, this system can keep track of the devices and allows the users to see their status. Moreover, there are lots of errors to show the users what went wrong and how can they fix it to allow a more user friendly environment.

Benefits of OOP

Using OOP made the implementation to be a lot tidier and efficient by allowing the programmer to make minimum code repetition. Also, it allowed the programmer to make a general structure to be followed which made the coding process easier. Thanks to OOP, the programmer made necessary abstraction and encapsulation processes to fully hide implementation details from interface.

Four Pillars of OOP and UML Diagram

There are four pillars of OOP.

First one is Inheritance. This allows one class to inherit methods and attributes from another class to further modify and add to them. By using inheritance, programmers avoid code repetition and make more specialized classes.

Second one is Polymorphism. This allows the reference of a class to make a reference to the same class or it's upper classes. By using polymorphism, programmers can make same methods do different things. This way, every object knows what to do when that method is called.

Third one is Abstraction. This allows the programmer to hide the details of the implementation to protect the code while also forming an interface for the user to use. This way, the user can use the interface to do what they want without knowing the actual implementation.

Fourth one is Encapsulation. This allows wrapping the code fragments as single units to protect them from getting changed. This way integrity of the classes is secured. Encapsulation also creates abstractions.

The UML diagram is a presentation of OOP systems in a visual way. It contains information more than the interface and less than the implementation. It visualizes a good road map before starting to code. It also helps when programmers need to explain their program.

UML Diagram
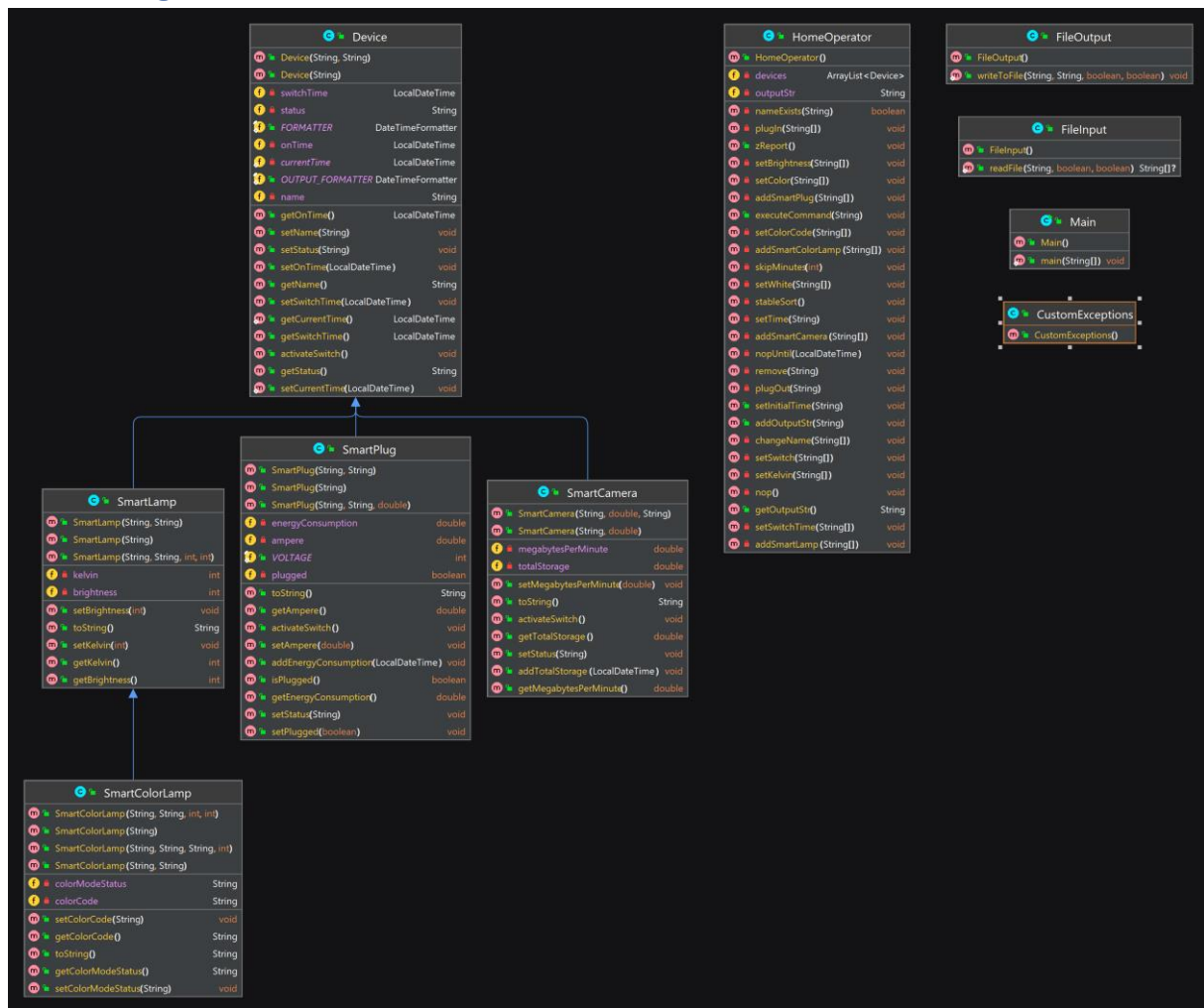
There are 10 main classes in the program.

Device is a direct superclass of SmartLamp, SmartPlug and SmartCamera, indirect superclass of SmartColorLamp. Since all the smart devices has common properties of devices, smart devices are gathered under Device superclass. Furthermore, SuperColorLamp can do everything a SmarLamp does and adds to it, that's why it's a descendant of SmartLamp. These four classes specialize the Device class for their needs by adding more methods and attributes or overriding the existing methods.

HomeOperator is the operating class of the system. It has methods corresponding to the commands. It matches and executes them by executeCommand method. It has all the operations for system to work and nearly all the error catching happens here.

FileInput class has readFile method and FileOutput class has writeFile method which are used in reading from files and writing to files.

CustomExceptions class has various inner classes that extends RunTimeException. These custom exceptions help when handling errors.

Main class is the driver class which has the main method and the main loop.