



HACETTEPE UNIVERSITY

Mustafa Kaan ÇEVİK

2210356101

2023 SPRING TERM - BBM104

Project Assignment 2 : Smart Home System

Contents

1.Main Problem of Smart Home System	3
2. Solution Explanation of This Problem	3
2.1 SmartDevices Objects.....	3
2.2 SmartPlug,SmartCamera,SmartLamp and SmartLampWithColor Objects .	3
3.Benefits of This System	4
4.Benefits of OOP	4
5. What are the four pillars of OOP and UML?	5
6.UML Diagram	6
7.Resources.....	7

1.Main Problem of Smart Home System

The problem at is to design an OOP system for controlling smart home accessories, namely Smart Lamp ,Smart Lamp with Color, Smart Plug, and Smart Camera. The objective is to make life easier for people by automating these devices, while ensuring that they are always in sorted order based on their switch times in ascending order. In exceptional cases where two or more devices have the same switch time, their initial order with respect to each other will be preserved. Devices without any switch time will be considered greater than all other devices. It should be noted that each device is switched off and has no switch time at initialization unless otherwise specified, and switching the device status will delete its switch time information.

2. Solution Explanation of This Problem

2.1 SmartDevices Objects

The first task is to create an object for smart devices. A class named SmartDevices needs to be created with mandatory parameters encapsulated. A constructor should be created with a name and default values for other parameters. If the objects have different values, setter methods can be used to change them. Afterwards, the SmartDevices class should be extended by the SmartPlug, SmartCamera, and SmartLamp classes.

2.2 SmartPlug,SmartCamera,SmartLamp and SmartLampWithColor Objects

This part describes the creation of various objects such as smart plug, smart camera, and smart lamp. These objects were given variables and created classes using encapsulation techniques. The classes were extended from the smart devices object, which allowed them to use the variables and methods of the smart devices. Additionally, the classes were given special variables and methods that were written within them after the overriding process.

Furthermore, SmartLampWithColor objects were extended from Smart Lamp, as the properties of the SmartLamp With Color object were derived from both the smartdevice and the smartlamp. Once all objects were created, an ArrayList of Smart Devices was created, containing all the smart devices such as Smart Plug, Smart Camera, Smart Lamp, and SmartLamp With Color. At the beginning of the code, these smart devices were added to the list and kept in it. All management was performed through this list using the created management classes and methods.

Moreover, the code was checked for exceptions in every class. An ERROR class was created, which contained exception messages. If the classes' parameters or variables were incorrect, the program would throw an error along with an error message.

The OOP paradigm was used to create all the classes, methods, and exception checks. The benefits of this system and the benefits of OOP were explained in the next chapter of the report. Overall, the report presents the creation of various objects and their management using OOP techniques, allowing for efficient and organized programming.

3.Benefits of This System

The benefits of this system include the ease of understanding due to the low number of variants of the smart home accessories, such as the Smart Lamp, Smart Lamp with Color, Smart Plug, and Smart Camera. The main task of this system is to control these devices and their timing according to commands. The devices are always kept in sorted order according to their switch times in ascending order. In exceptional situations where two or more devices have the same switch time, their initial order with respect to each other is preserved while sorting. If a device does not have any switch time, it is considered greater than all other devices. It is noteworthy that each device is switched off and has no switch time at initialization if it is not stated otherwise. Moreover, switching the status of the device deletes the switch time information. These rules and features were implemented in the program to ensure efficient and organized management of the smart home accessories.

4.Benefits of OOP

The benefits of OOP are numerous and valuable, especially when dealing with a large number of objects. Different objects can be stored in the same ArrayList and accessed using different methods, which simplifies the management process. Encapsulation and Inheritance are important concepts that are commonly used in OOP to make easy code writing and maintenance. Unlike Python, Java has a specific class type, making it easier to handle multiple classes. Encapsulation allows objects to be controlled and accessed only through specific methods, which ensures data privacy and security. Inheritance is another powerful concept in OOP that enables the creation of a hierarchy of classes, where a child class inherits properties and methods from a parent class, allowing for code reusability and flexibility. Overall, OOP promotes efficient and organized programming, making it an essential paradigm for software development.

5. What are the four pillars of OOP and UML?

The four pillars of object-oriented programming are Encapsulation, Inheritance, Polymorphism and Abstraction. Encapsulation is a concept that involves grouping related properties and methods within a class, allowing objects to be accessed and manipulated only through specific methods. This ensures data

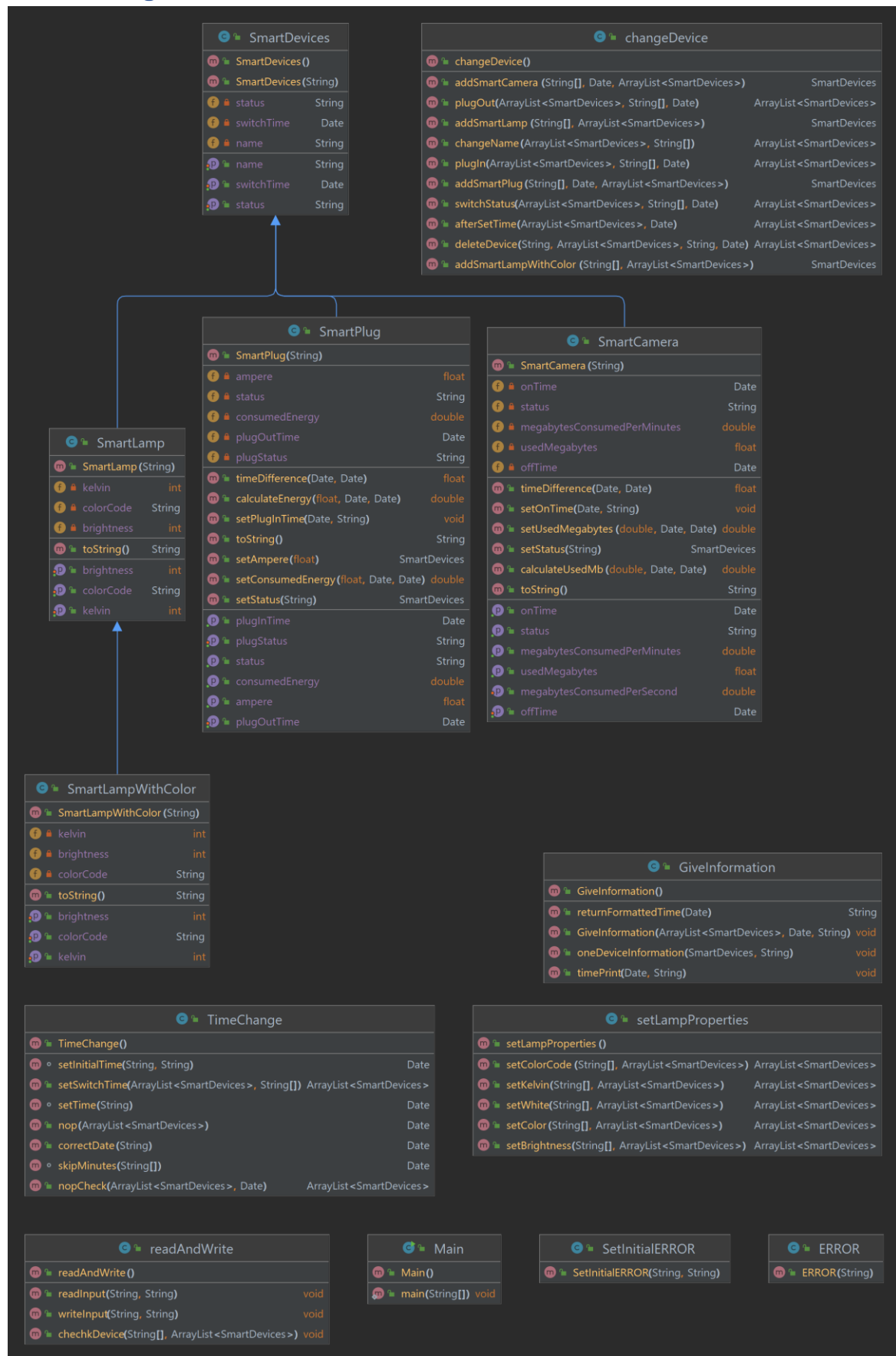
privacy and security. Inheritance is another pillar that enables the creation of a hierarchy of classes, where a child class inherits properties and methods from a parent class, making to easy code reusability and flexibility. Polymorphism allows objects to take on multiple forms or behaviors, depending on the context in which they are used. This feature is useful for creating dynamic and flexible code. Abstraction is the final pillar, which involves simplifying complex systems by breaking them down into smaller, more manageable components. This helps to reduce complexity and increase maintainability. After all, these four pillars are essential to the practice of OOP and UML, allowing for efficient and organized programming.

And UML means is Unified Modeling Language, which is a standardized visual language used to design and model software systems, including those built with Java. In Java, UML diagrams can be used to visualize the structure of classes and their relationships, the behavior of objects, and the interactions between objects and their enviroment. Commonly used UML diagrams in Java. When creating UML, It is easy to see how is working the code.

The four pillars of Object Orientation



6.UML Diagram



A SmartDevices class was created, which was then extended to create the SmartPlug, SmartCamera, and SmartLamp classes. Also, the SmartLamp class was extended to create the SmartLampWithColor class, which differs only in color code. All classes were given attributes such as name, status, and switch time. Multiple classes were created to manage the devices and ensure that the code works correctly. The current time is managed by the TimeChange class, while the changeDevice class adds smart devices to a list and manages their name, status, etc.

The properties of SmartLamp and SmartLampWithColor objects, such as kelvin and brightness, can be changed using the setLampProperties class. The readAndWrite class is responsible for reading and writing data to a text file. The GiveInformation class converts objects to a string using the toString() method and writes the information to a text file when necessary. Lastly, two Error classes were created to check for errors.

In summary, a system has been designed for the creation and management of smart devices. The classes and methods created aim to make the system work well and reduce the chances of mistakes.

7.Resources

- Algorithms, 4th Edition by Robert Sedgewick and Kevin Wayne
- Yazılım Mühendisliği – Ian Sommerville