Artificial Intelligence Engineering,

BBM102 Course Project Report

Name: NURŞAH

Surname: SATILMIŞ

Grade: One

Number: 2210765010

Adress: Artificial Intelligence Engineering Department , Hacettepe University, Beytepe, Ankara

Contact: b2210765010@cs.hacettepe.edu.tr / nursah_satilmis@hacettepe.edu.tr

Report Subject: Smart Device System Project

Starting Date: 31.03.2023

Expiration Date: 21.04.2022

Contents

PROBLEM

The purpose of this project is to design and implement a system that utilizes the principles of Object-Oriented Programming (OOP) to make our lives easier through automation. Specifically, the project focuses on smart home accessories that automate tasks within the home environment. The project emphasizes the four pillars of OOP: Inheritance, Polymorphism, Abstraction, and Encapsulation. By applying these principles and carefully considering the trade-offs between them, a good OOP design can be achieved. The project assumes that the designer will create a system that controls the time flow and automates various tasks in the home environment. The project requires a strong understanding of OOP concepts and the ability to apply them in a practical setting. The goal of the project is to create a functional and efficient system that uses automation to make our lives easier.

SOLUTİON APPROACH

The first step in creating objects is to detect common fields and functions for using inheritance and abstract classes.For example, SmartLamp and SmartColorLamp have a lot in common, Using inheritance is seems reasonable here. So, SmartColorLamp inherits fields and methods from SmartLamp. On the other hand, other objects have fewer common fields and methods, so they are connected to a superclass which is an abstract class to make implementation easier. Abstract methods in the AbstractClass are overridden in subclass (object classes), which is an example of polymorphism. After the device classes are created, an IOClass is created for the reading and writing process for encapsulation. Additionally , Errors class is created to throw requested errors in ControlTime class ..

Finally, the ControlTime class manages all processes, and a Control class is developed to govern time flow and execute commands from the input by employing other classes.Also , errors may occurs and generated are handled by using try-catch blocks. The main class comprises code fragments and necessary statements to ensure correct code execution.

"The sorting of devices to their switching times is the biggest challenging part of this project. Some problems are also encountered during the testing of the code due to insufficient test conditions. Initially, it is not possible to observe all cases. After the test files are updated, some problems occur. After the implementation is completed, it is hard to find the wrong part of the code even when using a debug tool."

Benefits of Smart Home System

The advantage of this system is convenience. By being able to manage and control different devices remotely, users can save time and energy in managing their homes. For instance, users can schedule their smart lamps to turn on or off at specific times of the day, eliminating the need to manually adjust the lighting in their homes.

In addition, the system's capacity to sort devices based on their switch times can help users manage their smart homes more efficiently. With devices arranged in ascending order based on their switch times, users can easily see which devices are scheduled to turn on or off next, enabling them to plan their day accordingly.

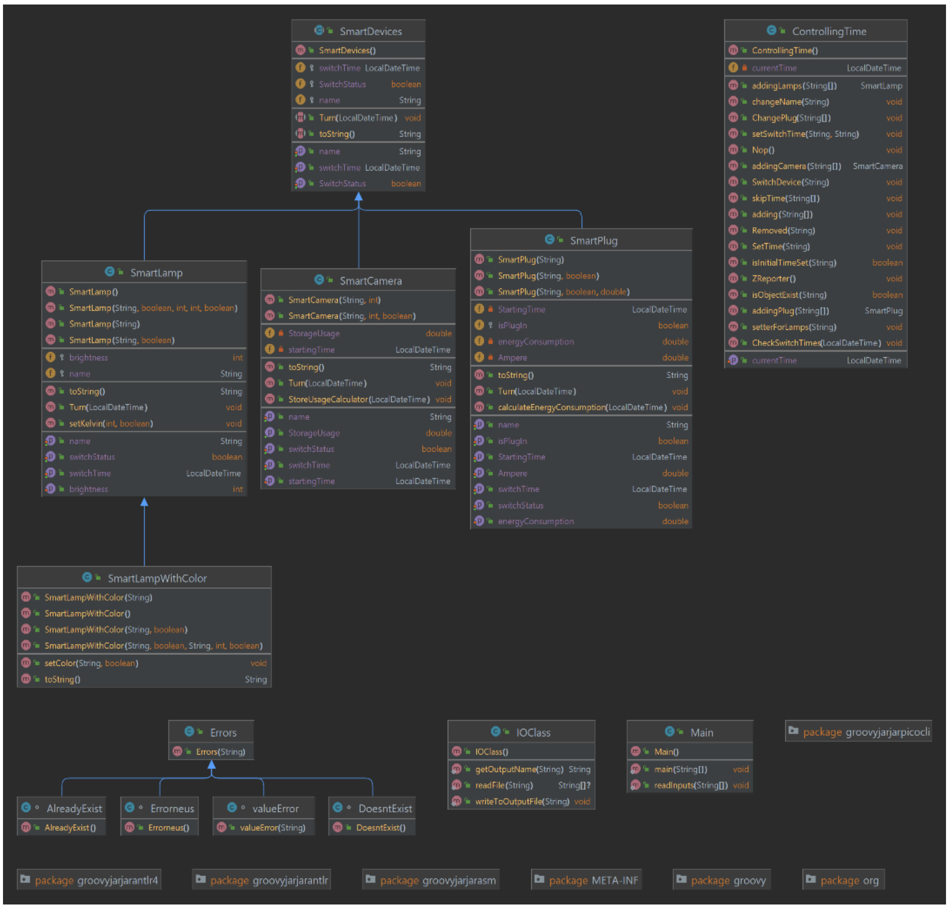
Benefits and Attributes of Object Oriented Programming

OOP is a powerful programming paradigm that offers several benefits, including better code organization, reusability, and maintainability. The four pillars of OOP provide a framework for developers to create modular, flexible, and adaptable code. By utilizing these principles, developers can build complex systems that are easier to develop, modify, and maintain over time.

The four pillars of OOP are the fundamental concepts that form the basis of OOP. These pillars are Encapsulation, Inheritance, Polymorphism, and Abstraction. Encapsulation is the process of grouping related data and functions together into a single unit, or object. Inheritance is the process of creating new objects based on existing ones. Polymorphism is the ability of objects to take on multiple forms or behaviors, depending on the context in which they are used. Abstraction is the process of focusing on essential features of an object while hiding irrelevant details.

By using these four pillars of OOP, developers can create code that is easier to maintain, modify, and reuse. Encapsulation helps to control access to the data and functions, making the code more secure and easier to maintain. Inheritance allows developers to reuse code more easily and to add new functionality to the application without having to rewrite the existing code. Polymorphism enables the creation of flexible and adaptable code that can be used in a variety of different contexts. Finally, Abstraction allows developers to simplify complex systems by breaking them down into smaller, more manageable parts.

UML Diagram



Inheritance is seeming between SmartLamp and SmartColorLamp. Some methods and fields are inherit by SmartColorLamp from SmartLamp.

SmartColorLamp (relatively) , SmartLamp , SmartPlug, SmartCamera is subclasses of SmartDevices which is an abstract Class.

AlreadyExist , Errorneus, ValueError, DoesntExist is subclasses of Errors. IOClass , ControllingTime Main do not have any Subclass or SuperClass.