Hacettepe University

ASSIGNMENT 2: SMART HOME DESIGN

BBM104 2023 SPRING TERM

INSTRUCTOR: GÖRKEM AKYILDIZ

ŞÜKRİYE ÖZTÜRK 2210356110

4-5-2023

INDEX

What is Problem?

All electronic devices are getting smarter and requires a remote controlling for their functions. The problem is that creating a well working Smart home program design which schedules autonomous actions of smart devices at home.

My Main Solution Approach

A class based hierarchical system is aimed for a smart home system design problem. At the beginning of all, a system which has based on basis of OOP considered as the best solution so, code contains 4 subclasses of devices which is controlled from a superclass. To make the code well encapsulated and abstracted, some interfaces and classes are used at solution.

Problems that I Struggled

Creating a perfectly working hierarchical system is critically tough because of abstraction level of code. The biggest problems I struggled are handling so many exception cases at the same time. Also calculating energy and setting devices as “on” or “off” at certain time intervals is one of the problems I struggled while creating the project.

Solutions for Problems

Critic part of the exception handling problems is solved with a new exception class which throws different error messages for different cases and terminates the program. The big part of the switch on/off and plug in/out problems has been solved by creating suitable methods for each one.

Benefits of System

This system provides a clear interface for last user, also makes client’s life easier. Due to system can set itself to switch on and off devices in certain time intervals, controlling and using complex devices extremely getting easy. Also, system can protect the user from overconsumptions or unexpected serious accidents thanks to energy consumption calculation function and z-report function.

Advantages of Using OOP

The best way to create a project as smart home controlling system can be chosen with using OOP. Due to there are many similarities and common functions between devices, hierarchical structure is the best approach for it. Also, encapsulated design makes the code easy to understand and modifiable easily. Abstraction makes lots of qualification and method accessible for classes and that is an immediate requirement for a project that contains lots of abstracted method.

4 Basis Principle of OOP

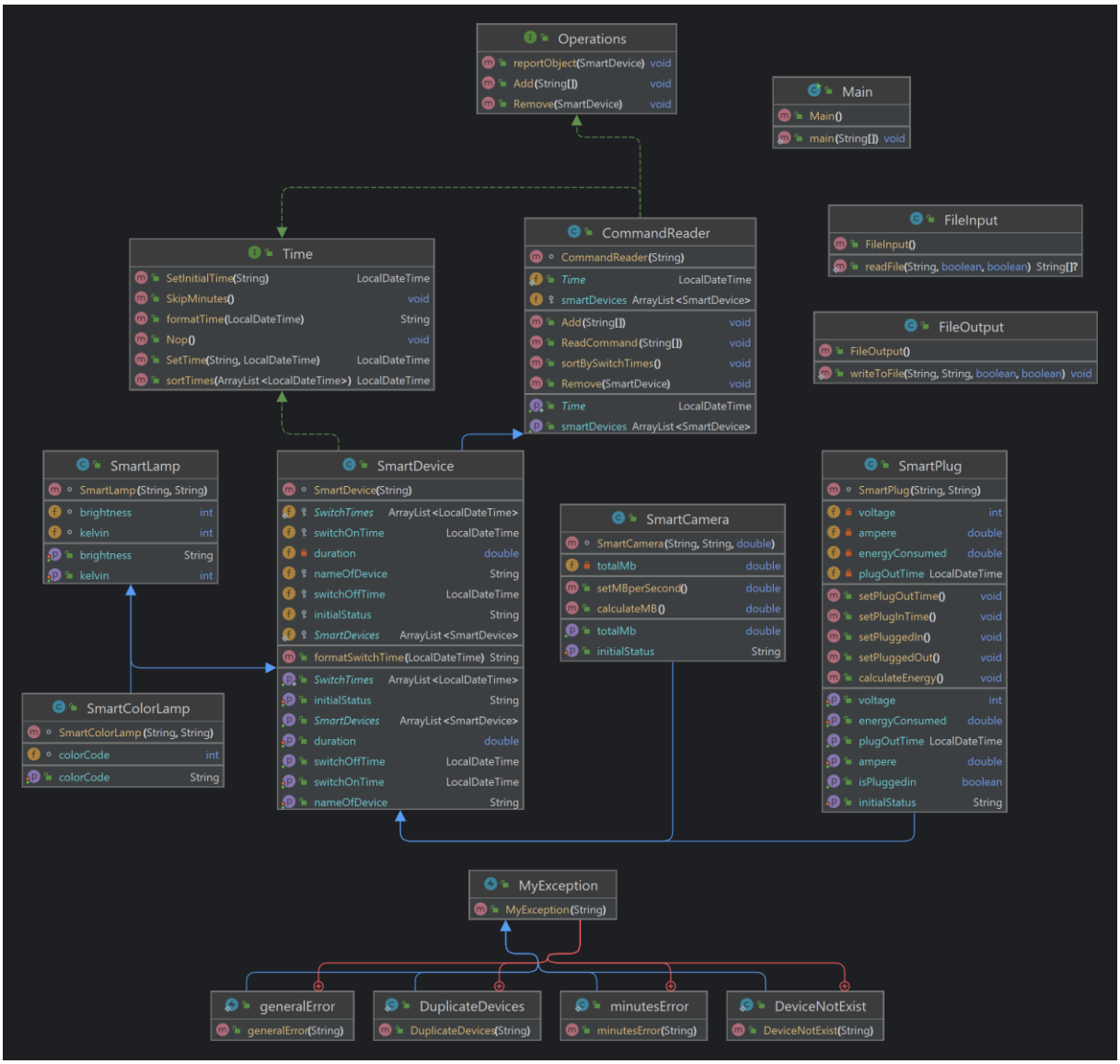
Abstraction: Project has 2 interfaces for abstracted operations and practicality of implementing interface methods. Even if SmartDevices is not designed as an abstract class, it is the biggest abstraction example of this project. Every smart device class inherits from SmartDevice class, even if there is no exact a “smart device”.

Encapsulation: Project has tried to be coded as possible as with hidden implementations. Nearly %90 percent of the variables which is belongs to classes, assigned as “private” and used in the code with mutators and accessors to protect encapsulated design. Also, every accessor, mutator and method contains a Javadoc that tells preconditions and postconditions. Thanks to this implementation, project is easily understandable, modifiable, and user-friendly.

Modularity: Project has been designed as possible as modular to be easily codable. Also, modularity of code is highly important for saving from time and labour. Every class divided more than one method to provide modular project which is able to make more work with less code.

Hierarchy: Project stands on hierarchical design. The main part of code which reads all commands and takes the proper action inherits all device classes in the project. Main device class SmartDevices inherits all smart device classes to make possible to use all methods commonly.

UML Diagram



As can be seen from the picture above, Smart Home System contains 10 classes and 2 interfaces. The biggest part of abstraction is provided from the main superclass "SmartDevice" due to most of the device classes is subclass of it and uses it's methods. The necessary operations is taken from interfaces by "CommandReader" to be used while processing the data in input file.

Resources and Quotes

<https://web.cs.hacettepe.edu.tr/~bbm102/>

< https://www.geeksforgeeks.org/abstraction-in-java-2/ >