



Hacettepe University  
Department of Computer Engineering

# Spring 2023 - BBM104

## Assignment 2 – Smart Home System

21.04.2023

Ezgi Ekin  
2210356029

## Index

• Problem Definition .....	3
• Solution Approach.....	3
• Problems and Their Solutions.....	3
• Benefits of the System.....	4
• Benefits of OOP.....	4
• Four Pillars of OOP and UML .....	5
<i>Encapsulation</i> .....	5
<i>Inheritance</i> .....	5
<i>Polymorphism</i> .....	5
<i>Abstraction</i> .....	5
<i>UML</i> .....	5
• Explanation of UML Diagram.....	6
• Resources.....	7

- **Problem Definition**

This project aims to create a Java application that uses a well-designed object-oriented programming (OOP) approach to manage a smart home system. The system must control three types of smart home accessories, including Smart Lamps (white-ambiance and color-white-ambiance variants), Smart Plugs, and Smart Cameras. The devices should be sorted based on their switch times in ascending order, and the implementation should follow the four OOP pillars of Inheritance, Polymorphism, Abstraction, and Encapsulation. The system should be able to handle user commands and manage the time flow accurately.

- **Solution Approach**

To create this system, first the class designs and relationships were considered. As devices and time were the foundations of the smart home system, they were combined in the SmartHomeSystem class. The camera, plug, and lamp classes were then grouped under the Device class. The methods and fields that each class should have, were defined. To check whether the methods provided to this system were erroneous, controls were made through SmartHomeSystemException. The compare method was implemented to sort the ArrayList of devices. As it was necessary to access time from anywhere, its management was handled by the static methods in the Time class.

- **Problems and Their Solutions**

After Python, Java was a new language, and it was difficult to understand its structure. To overcome this problem, information about OOP designs was collected from Java resources.

There was difficulty in obtaining the date in the given format. After research, the problem was solved using SimpleDateFormat.

There was a problem with taking the times for the camera and plug calculations, but later, the time for each opening was assigned to a separate variable and when they were closed, the current time was used with this new variable for the calculation.

There was difficulty in understanding what algorithm to use to sort the devices. After research on the Collections class, it was realized that it would be logical to implement the compare method for sorting.

For the first input, there was a problem where lamp1 was staying on top of plug10 in the setTime command. It was understood that this problem was caused by setting the device's switch time to null and then sorting it immediately afterwards. To solve this problem, the number of devices with the same switch time was found later, and these devices were first set to null in the for loop and then sorted outside the for loop. In other words, the simultaneous shutdown of the devices was simulated.

- **Benefits of the System**

The purpose of designing this system is to make our lives easier and more efficient using automation and smart home accessories. Java programming language was used to make the system as efficient as possible during its development.

- **Benefits of OOP**

The benefits of OOP in this program can be best seen by comparing it to Python. While Python has a straightforward flow and functions, OOP utilizes classes and methods. And this allows separate objects in the system to benefit from and establish connections with each other. OOP also allows objects to be dynamic and used in different ways in different places. It also makes this system more maintainable.

- **Four Pillars of OOP and UML**

### **Encapsulation**

Encapsulation hides the internal data and methods of an object from the outside. Encapsulation protects the data from being accidentally modified, ensuring that it remains accurate. It is done by using access modifiers, such as private, protected, and public, to control how the data and methods of an object can be accessed.

### **Inheritance**

Inheritance allows classes to inherit properties and methods from other classes. This helps to avoid repetition of code and enables a more efficient and organized development process.

### **Polymorphism**

Polymorphism is a concept in object-oriented programming where objects can take on different forms or behave differently depending on the context in which they are used. It can be done by method overloading or method overriding.

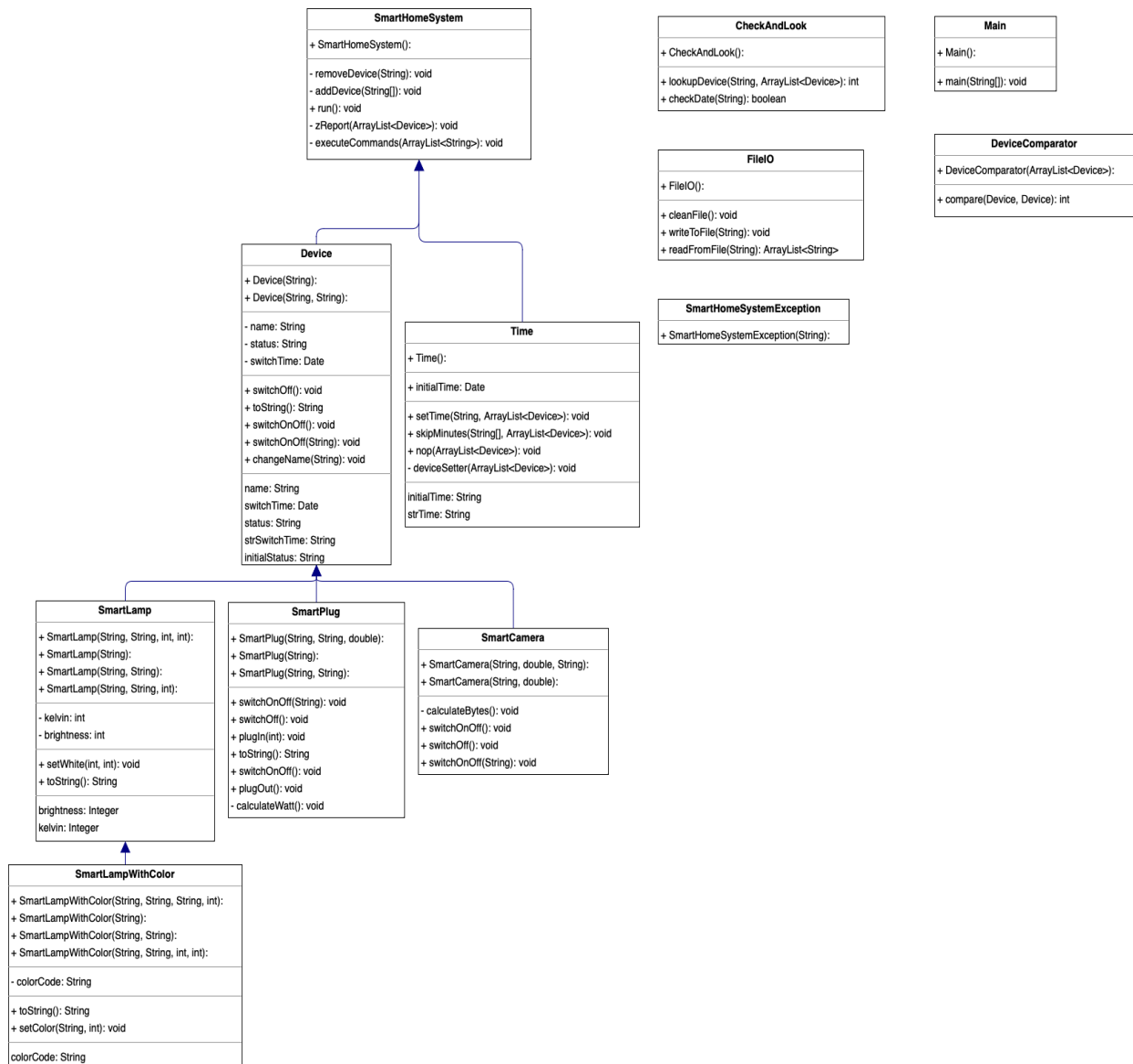
### **Abstraction**

Abstraction simplifies complex systems by breaking them down into smaller, more manageable parts. It is achieved by using abstract classes and interfaces that define the common characteristics and behaviors of classes. Abstraction increases reusability of the code.

### **UML**

UML (Unified Modeling Language) class diagram shows the relationships between classes, such as inheritance, aggregation, and association. It helps to identify the major components of the system, their relationships, and how they interact with one another.

- **Explanation of UML Diagram**



SmartHomeSystem is a software system consisting of two main components. The first component is responsible for managing time flow, while the second component, called the Device class, manages various types of devices and their methods, serving as the parent class for all device subclasses. The Device class is an abstract class, meaning that every subclass must implement the toString method for ZReport. SmartPlug and SmartCamera are two subclasses of Device that

override the switch method to save the times when the device is turned on or off and perform some calculations. SmartLampWithColor is another subclass of Device that can act as a SmartLamp, but also has an additional color mode feature. It inherits from the SmartLamp class. To facilitate sorting operations in SmartHomeSystem, the DeviceComparator class is responsible for implementing the compare method. The CheckAndLook class contains methods for checking if a given date is valid and searching for a device in an ArrayList. Finally, the FileIO class provides methods for reading from and writing to a file, as well as a method to clear the contents of a file.

- **Resources**

W. Savitch, "Java - An Introduction to Problem Solving and Programming," Pearson, 2007. [Online].

P. Deitel and H. Deitel, "Java - How to Program," Prentice Hall, 2014. [Online].

B. Eckel, "Thinking in Java," Prentice Hall, 2006. [Online].

Tutorials Point, "Tutorials Point." [Online]. Available: <https://www.tutorialspoint.com/index.htm>. [Accessed: Apr. 1-21, 2023].