

CSE 443
Object Oriented Analysis and Design
Fall 2020– 2021
Homework 1 Report

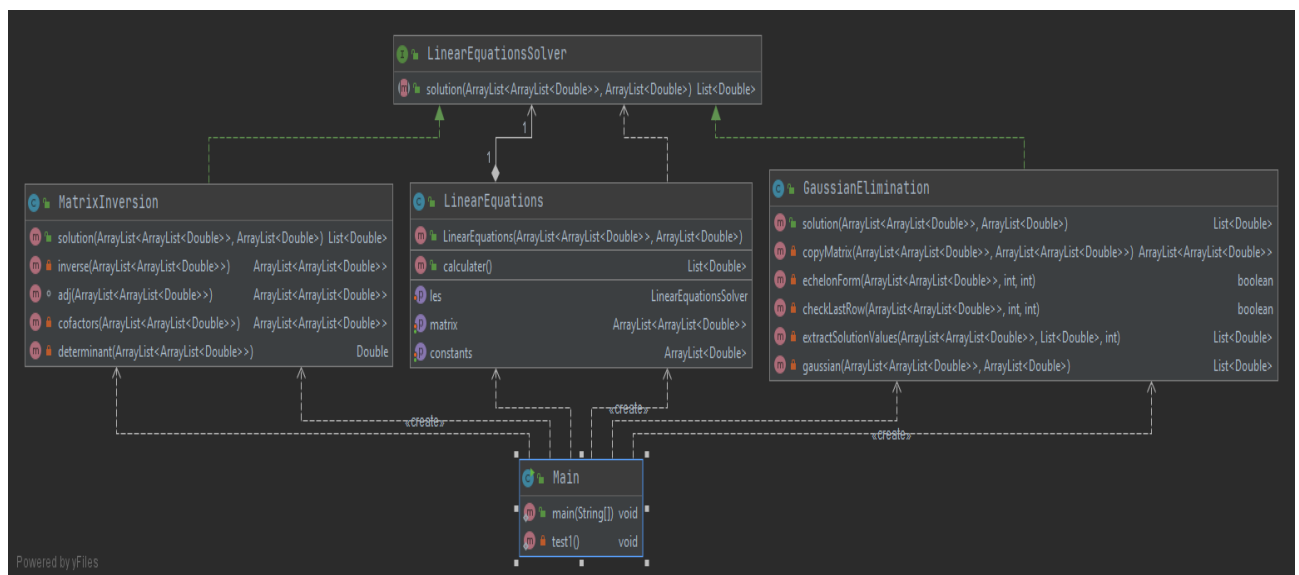
Caner KARAKAŞ
131044061

Question 1

In the first question the Strategy Design pattern has been used.

In our design, an interface class named `LinearEquationsSolver` and `LinearEquations` class extending from this class were created. This interface, which is created for the solution of all linear equation systems, makes it easy for us to add solutions that can be added later. It holds one object from the `LinearEquationsSolver` class in the `LinearEquations` class. In addition, it keeps the variables part and the fixed results part of the equation to be entered in separate parameters. `Calclater`, a method of the `LinearEquations` class, works by calling the solution method of the object of the `LinearEquationsSolver` class.

We have defined the methods we will use as classes. We created these classes by extending them from the `LinearEquationsSolver` class. In this way, we will be able to make a dynamic transition between my solutions and the methods to be added later will be created as a class only and will not cost the rest of the code.



Main class takes user input and creates an equation with the information. The solve method then creates a GaussianElimination and a MatrixInversion. The method to be used is then selected and the method returns to the LinearEquations constructor. Retrieves the solution and prints it by calling the LinearEquations solution method. This continues dynamically in a loop until the user logs out. Can solve an equation in different ways and compare its results.

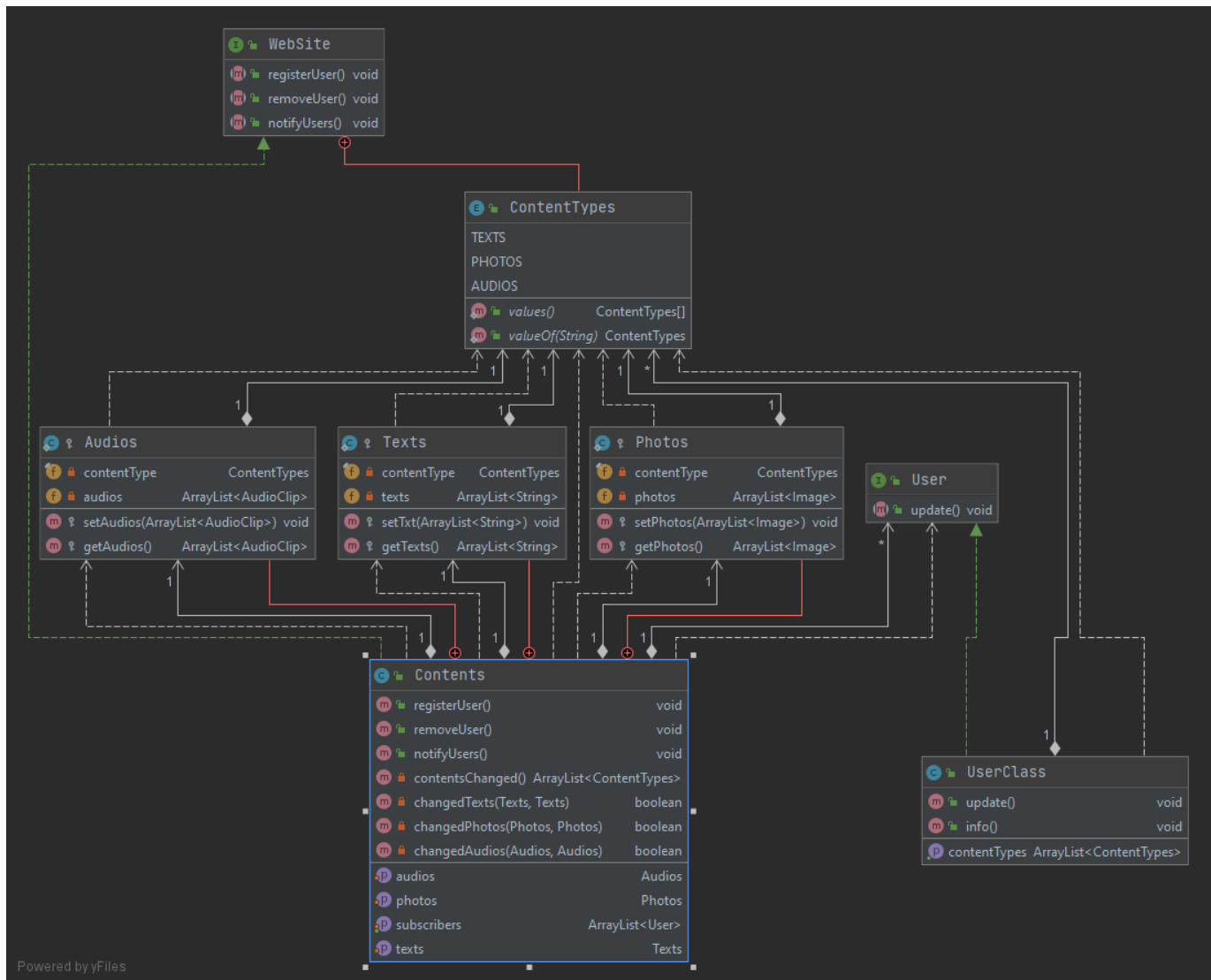
Question 2

I have used the Observer model in this section. This pattern allows subscribers to keep track of changes in content.

I created the WebSite interface class. This class is an interface for websites. I have created an enum structure within this class that can hold the content on sites. It will make it easier to add content that can be added later. I created a Content class that I implemented from this WebSite interface. I added the contents of this class as inner class. In this way, new content types to be added can be added without increasing the cost and flexibility will be increased. The contentsChanged method that I add to this class returns a list where we can access the information of a content type. In this way, only the subscribers with the content they follow will be warned during the change.

I created an interface class called User and created a UserClass class that I implement from here. Subscribers are created thanks to this class. A list is kept in which content types they follow.

New subscribers can be added or removed with functions added for this design. Whenever there is a change, the necessary subscribers can be updated with the update method.



Question 3

Decorator design pattern was used in the third question. An abstract class called Suits was created to represent basic clothing. Next, another abstract class called Accessories was expanded from the suite to become the parent of all Accessories.

Dec, Ora and Tor basic outfits have been extended from the Suit class. Its founders set the weight values and implement the `getCost` abstract method to return their own costs.

Flamethrower, AutoRifle, RocketLauncher and Laser accessories have been extended from the Accessories class. They have a Tooling area for applying the Decorator Pattern.

The Main class, which is the starting point of this section, has a main method that asks the user for the main team. It then asks for accessories to be added to the main garment in a do-while loop. After all accessories are selected, the program gives an information printout.

