Gebze Technical University

Computer Engineering

CSE443 - Object Oriented Analysis and Design

Homework 1 - Part 1 Report

Student:
Nevra GÜRSES
161044071

1 INTRODUCTION

1.1 Project Definition

The project's goal is to design and implement a Java application that admits as input from the user a system of linear equations and outputs its solution, if it exists, or an error message otherwise. The customer wants the software to support at least two methods of solving linear equations for example Gaussian elimination and matrix inversion. User also wants to be able to change between solving methods dynamically. Our software should be maximum flexibility,loose coupling and minimum cost.

2 METHOD

2.1 Selected Design Pattern and Why That is Selected

I select Strategy Design Pattern and I apply this pattern for solving given project.Now, I tell the reasons why I select this pattern for the project.

Strategy design Pattern is a behavioral software design pattern that enables selecting an algorithm at runtime. Instead of implementing a single algorithm directly, code receives run-time instructions as to which in a family of algorithms to use.Our project wants dynamically method changing at runtime and Strategy design pattern provide this perfectly.

The key idea for Strategy Design Pattern is to create objects which represent various strategies. These objects form a pool of strategies from which the context object can choose from to vary its behavior as per its strategy. These objects(strategies) perform the same operation, have the same(single) job and compose the same interface strategy. In our project strategies are methods of solving linear equations for example Gaussian Elimination and Matrix Conversion.This strategies implements an interface.Solver class selects linear equation solving method at runtime.So our goal is provided.

If I need to talk about loose coupling, minimum cost and maximum flexibility of the Strategy Design Pattern:

Loose Coupling means that two objects can interact with each other, but they have very little knowledge of each other.The Strategy Pattern provides an object design where Solver class and strategies are loosely coupled. Because the only thing that the solver knows about an strategy is that it implements a certain interface that is the Method interface. And dynamically selects linear equation solving method.

The Strategy Design Pattern has maximum flexibility because we can add new equation solving method at any time. The only thing we write solving method class that implements Method interface.

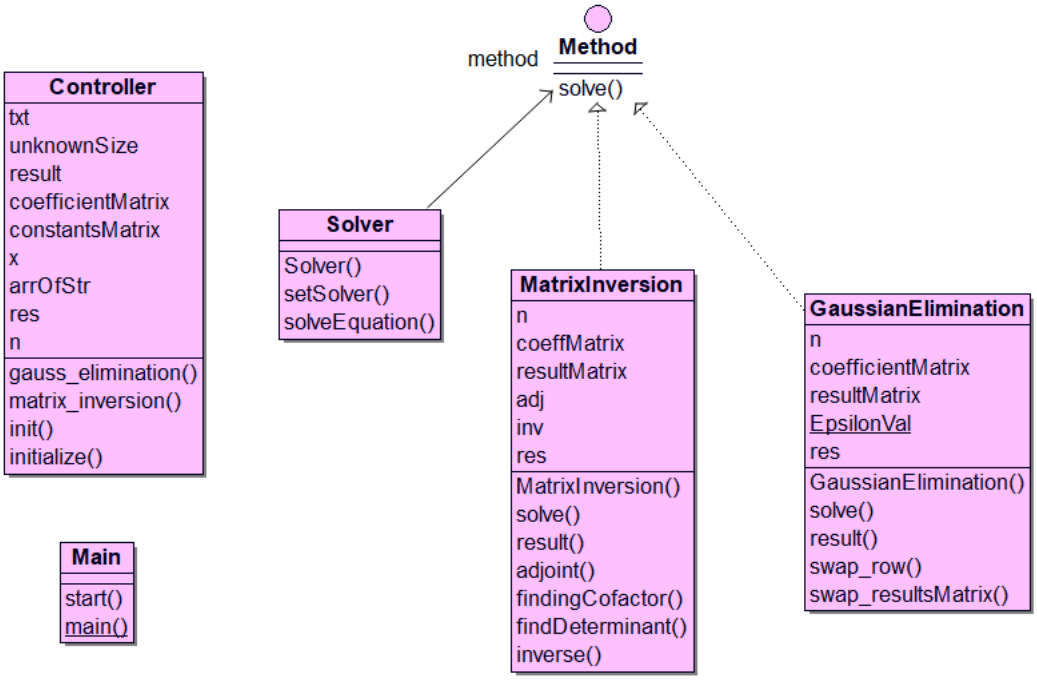
The Strategy Design Patternn has minimum cost because we don't have to need to modify all classess when new linear equation method was added.Only thing,we write solving method class that implements Method interface.

Because of this reasons I select Strategy Design Pattern and I implement it. For making this project with user interface technique I use JavaFX.

To make JavaFX work in IntelliJ ide, there must be made some configurations:

- Firstly, JavaFX should be downloaded.
- Secondly, in VM option in Edit Configurations section, –module-path C:\javafx-sdk-15.0.1\lib –add-modules javafx.controls,javafx.fxml dependency should be writed. ("C:-sdk-15.0.1" is path of library folder of javafx)
- Lastly, lib part in Project Structure section, javafx library folder should be added for example as "\javafx-sdk-15.0.1\lib"

2.2 Class Diagram and Explanation of Selected Design Pattern with Diagram



- **Controller Class:** This class implements Initializable interface.This class provides controlling operations in user interface.For example gets equation coefficients from interfece, selects linear equation solving method and writes result in user interface with ActionEvent operations in buttons.

- **Method interface:** This interface for Strategy interface of Strategy Design Pattern.This interface is defining an action and concrete strategy classes that are implementing the Strategy interface.In this interface there is a method named solve to solving given lineer equation.

- **Solver Class:** This class for Context class of Strategy Design Pattern.This class uses Method interface.By using of this class, lineer equation solving method can be selected dynamically at run time.In setSolver method of this class, Method is set and solveEquation method calls appropriate solve method.

- **GaussianElimination and MatrixInversion Classess:** This classes are concrete Method classes.They implements Method interface.By set function, they solve given lineer equation. GaussianElimination Class uses Gaussian Elimination method of linear algebra for solving equation and MatrixInversion Class uses Matrix Inverse method for solving linear equation

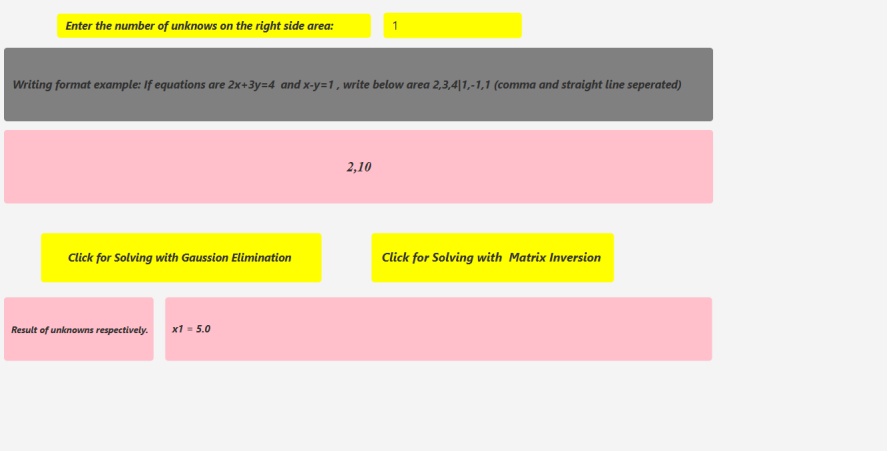
- **Main Class:** For testing and working project.

3 SAMPLE RUNNING RESULTS:

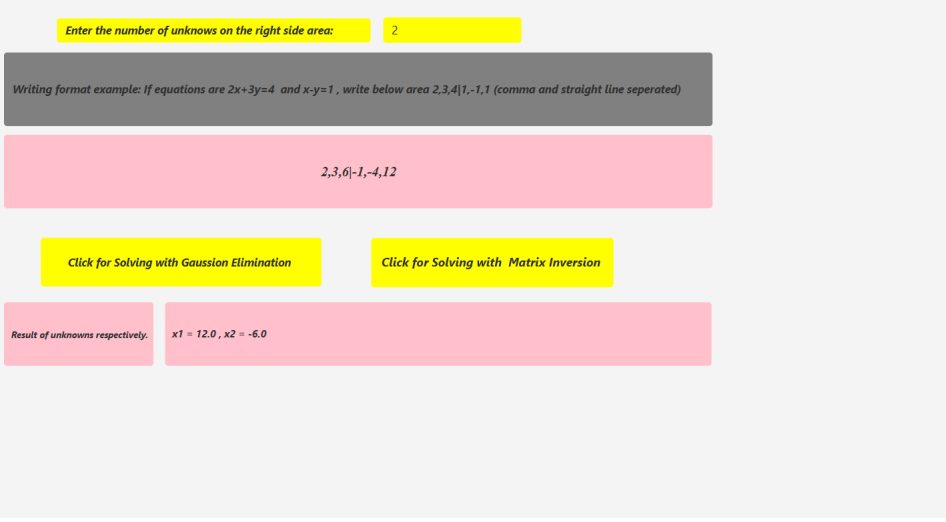
NOTE: Write linear equations in user interface with given format: as cofactors and constants with comma and straight line seperated.

For example if there are 2 unknowns so 2 equations , One of them x+3y=10 , second one -4x,5y=20 write equations in interface as
1,3,10|-4,5,20 and also write number of unknows for example in this example unknown number is 2.

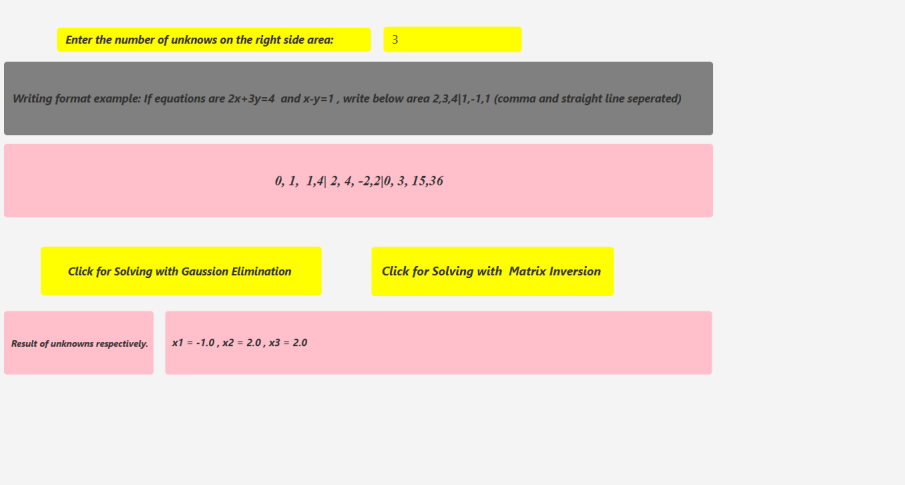
Output 1 (Equation with one unknown)
If equation is **2x = 10, this equation was written user interface as 2,10** and so result was written on user interface for Gauss elimination and matrix inverse method.



Output 2 (Equation with two unknowns)
If equations are **2x + 3y = 6 and -x -4y = 12, this equations were written user interface as 2,3,6|-1,-4,12** and so result was written on user interface for Gauss elimination and matrix inverse method.



Output 3 (Equation with three unknowns)
If equations are **y+z = 4 and 2x+4y-2z=2 and 3y + 15z = 36 this equations were written user interface as 0,1,1,4|2,4,-2,2 |0,3,15,36** and so result was written on user interface for Gauss elimination and matrix inverse method.



Output 4 (Equation with has not a solution)
If equations are **2x+3y = 10 and 2x+3y=12 this equations were written user interface as 2,3,10|2,2,12** and so this equation has not a solution so error message was written on user interface for Gauss elimination and matrix inverse method.