SE311 SPRING 2023-2024 / 10-05-2024
Lab Work 10 – Observer Pattern
Goal: Practice on Observer Pattern

In our sample pattern implementation, we have used a stock tracker example. For today's lab, we will use Classrooms and School Bells.

A school becomes too crowded with students, so that people cannot go to restrooms in between classes or cannot eat at lunch because of the incredibly long queues. The administration finds the solution to automatize the breaks between classes and changing the durations of classes.

As a result, they decide to finish half of the classes at xx:20 each hour and the remaining half to finish at xx:50 each hour. Bells will ring at those times and hold the time as their state. (You can simplify this by just holding the last two digits as an integer in the state. Just do not forget notification occurs on either 20 or 50.)

PART - 1

1. In this lab, you will have a subject School Bell instead of Stock. School Bells ring at times if their internal clock matches with the value that is expected by classrooms. Classrooms are your new type of observers that are attached to School Bells. A School Bell will have an internal clock as it is mentioned, that is observed by Classrooms. If the time is right, then all classrooms attached are notified.

Please download the stock tracker sample implementation and modify that code, so it becomes Bells and Classrooms example. Create one Bell and one classroom in your main and test it.

2. Create a new class called Counter. It has one private static attribute: int updateCounter. Also add a method "void increaseCounter()", which will increase the updateCounter by 1. The reason for this class is to count how many times a class has ended and classroom's Update() method is triggered.

Add one Counter class object to your Classroom class as an attribute. Test your code again but this time count the Update() calls and print the updateCounter.

PART – 2 Principal steps in…

1. Assume that, a new class called "Principal" organizes which bells are observed by which classes. In order to achieve this, remember the Mediator example from class. Also, do not forget to store Subject-Observer mapping in an appropriate data structure.

2. Modify your main to work with Principal class.