SE311 SPRING 2021-2022 / 27-05-2022
Lab Work 10 Observer Pattern
Goal: Use *Observer* Pattern

In our sample pattern implementation, we have used a stock tracker example. For today's lab we will use **Controllers** and **Users**.

PART - 1

1. In this lab you will have a subject **Controller** in a room instead of Stock. Controllers are for an air conditioner system. **Users** are your new type of observers that are attached to controller. A **Controller** will have a temperature value that is observed by **Users**. If the room temperature changes, then all users are notified.

   Please download the stock tracker sample implementation and modify that code, so it becomes **Controllers** and **Users** example.

   Create one Controller and one User 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 that User's Update() method is triggered.

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

PART – 2 Who triggers the update? (Test "a" and "b" separately.)

3. Add 4 more Users in your main. You should have total of 5 users. Each user will change the Controller's temperature 4 times.

   a. Subject triggers the update: If your code runs correctly, this is the default case. The subject whenever its state changes triggers the update. How many times Update() is called?

   b. Observers trigger the update: Add a method to Users, so that they can change the Controller's temperature. After they change the temperature 4 times, they notify all users. How many times Update() is called?