COMP47480 Practical 6: Observer Pattern

Sukrat Kashyap (14200092)

24 April 2018

1 Work Done

In this practical, our task was to use design pattern to solve a particular problem. The design pattern to be used was the Observer pattern. The Person and the AlarmClock were acting as data classes and all the work was done in the AlarmApplication making it have god class tendencies. To solve this problem, we created an abstract class for the Subject which acts the source of broadcasting when something happens. The broadcasting is done to all the interface Observers which have subscribed to listen to the Subject. These 2 were extended and implemented by AlarmClock and Person class respectively. AlarmClock calls the Observer update method upon change in its time. Person class on update checks whether the AlarmClock has reached the alarm time if it has it wakes itself up. The AlarmApplication was then made to only instantiate the objects, register the person and call the tick method. Since, we had to notify the person whenever the alarm time is set as well. Person this time had to store the wake up time as well. For this reason we changed the update method to pass in the object for pulling the state and an arg object for pushing the event name. This was done so that the Person knows which event is being called and act accordingly. Cron class was extended on Observer which also listened to the Clock and acted when the time was right.

2 Reflections