Последният Design pattern на който се спрях е Observer. Той е един от известните и широко прилагащи се Behavioral Patterns. Използва се за определяне на взаимовръзки между обекти. Определя един-към-много зависимост между обекти, за да може когато един обект промени състоянието си други да се уведомят и да се update автоматично.

Класовете и/или обектите ‚участници‘ в този pattern са:

* Обект (Борса)

Знае за наблюдателие си. Всеки от тях може да наблюдава обект. Предоставя интерфейс за закачане и откачане на наблюдавани обекти.

* Конкретен Обект (IBM)

Запазва състоянието на интерес към Конкретен Обект. Изпраща уведомления към наблюдателите си когато състоянието се смени.

* Наблюдател (Инвеститор)

Определя update интерфейс за обектите които трябва да се уведомят при промяна в обекта.

* Конкретен наблюдател (Иванов)

Поддържа референция към Конкретен Обект. Запазва състоянието което трябва да е постоянно към субекта. Имплементира update интерфейс на Наблюдател, за да запази състоянието си постоянно със субекта.

Всичко това позволява регистрирани обекти да бъдат уведомявани и update при промяна на състоянието .