Observer Pattern

The Pattern Problem

Observer is a behavioral design pattern that lets you define a one-to-many dependency between objects so that when one object changes state, all its dependents are notified and updated automatically.

- Reusability

Because of the abstract class Subject (StockMarket), we can reuse its subscription methods and notify when a change has occurred in a Concreate Subject (IBM).

- Extensibility

The pattern allows us to extend the number of Concreate Subjects and Concrete Observers.

For an example if we add more stock market companies, or/ and add different types of investors (clients).

- Maintainability

A programmer may have a lot of different concreate subjects or subscribers (Investor). Where the subscribers know the concretes subjects and they subscribe to the whichever they like. However, the concreate subject have no knowledge of the subscribers.

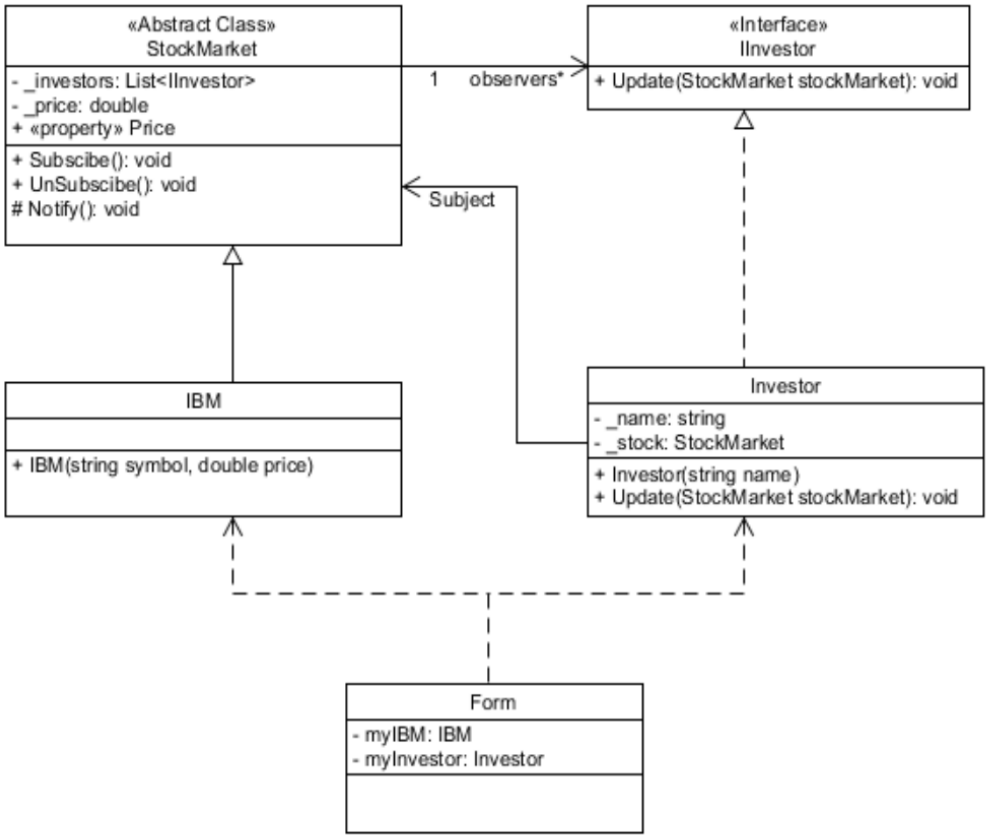
Design Pattern Solution

StockMarket (Subject): This abstract class defines the behaviour of a stock market subject. It has two methods Subscribe() and Unsubscribe() for assigning and unassigning an Investor to a Stock respectively. Method Notify() is used for informing about the changes in the stock to the investors. Property “Price” is used to get and set the price of Stock.

IInvestor (Observer): This is a basic implementation of an Observer. It simply contains the method Update() with the parameter of the type StockMarket, which means class StockMarket call method Update(), it pushes data to the interface IInvestor.

Investor (Concrete Observer): Investor is subscribed or unsubscribed to the Stock. When a price of the stock is modified it is updated to all the investors by calling Update() method.

IBM (Concrete Subject): IBM is the stock which is basically the data in the StockMarket



Consequences

- Positive consequences
  1. Publisher is not coupled to concrete subscriber classes.
  2. You can subscribe and unsubscribe objects dynamically.
  3. Follows the Open/Closed Principle.
- Negative consequences
  1. Subscribers are notified in random order.

MRUNAL PATIL – 3101851

EMIL KARAMIHOV - 2992884