Observer Pattern

The Observer pattern is a behavioral design pattern that defines a one-to-many dependency between objects. It allows an object, known as the subject, to maintain a list of its dependents, called observers, and automatically notify them of any state changes. This pattern is particularly useful in scenarios where a change in one object requires corresponding changes in other objects, but the exact nature of these changes is not known beforehand.

In this example, we've implemented the Observer pattern to simulate a stock market application. The Stock class acts as the subject, which maintains a list of observers (investors) and notifies them when the stock price changes.

The Investor interface defines the update method, which is implemented by ConcreteInvestor. Each ConcreteInvestor is notified whenever the stock price changes, allowing them to react to these changes accordingly.