# Class Reference

## Table of Contents



PDF generated using Asciidoctor PDF

# Code

## Observer.java

```
public interface Observer {
    void update(String symbol, double price);
}
```

## ObserverImpl.java

```
import java.util.HashMap;
import java.util.Map;

public class ObserverImpl implements Observer {
    public final String name;
    Map<String, Double> stockPrices = new HashMap<>();

    public ObserverImpl(String name) {
        this.name = name;
    }

    @Override
    public void update(String symbol, double price) {
        double oldPrice = stockPrices.getOrDefault(symbol, price);
        stockPrices.put(symbol, price);

        if (oldPrice == price) {
            System.out.printf("Observer %s > No change in price for %s%n", this.name,
symbol);
            return;
```

```
        }

        String change;

        if (price > oldPrice) {
            change = "+";
        } else { // must be price < oldPrice since equality is handled above
            change = "-";
        }

        change += String.format("$%.2f", Math.abs(price - oldPrice));

        System.out.printf("Observer %s > %s price change is %s. New price is
$%.2f%n%n", this.name, symbol, change, price);
    }
}
```

## Stock.java

```
import java.util.ArrayList;
import java.util.List;

public class Stock {
    private final List<Observer> observers = new ArrayList<>();
    private final String symbol;
    private double price;

    public Stock(String symbol) {
        this.symbol = symbol;
    }

    public void acceptObserver(Observer observer) {
        observers.add(observer);
    }

    public void setPrice(double price) {
        this.price = price;
        for (Observer observer : this.observers) {
            observer.update(this.symbol, price);
        }
    }

    public double getPrice() {
        return this.price;
    }
}
```

# Driver.java

```
void main() {
    Stock stock1 = new Stock("AAPL");
    Stock stock2 = new Stock("GOOGL");

    Observer observer1 = new ObserverImpl("Ameer");
    Observer observer2 = new ObserverImpl("Zain");

    stock1.acceptObserver(observer1);
    stock1.acceptObserver(observer2);

    stock2.acceptObserver(observer2);

    stock1.setPrice(120.00);
    stock2.setPrice(1550.00);

    System.out.println("AAPL has 2 observers and GOOGL has 1 observer\n");

    for (int i = 0; i < 3; i++) {
        System.out.println("=========");
        System.out.println("Round " + (i + 1) + " of price updates:\n");
        double price1 = stock1.getPrice() + Math.random() * 50;
        double price2 = stock2.getPrice() + Math.random() * 100;

        stock1.setPrice(price1);
        stock2.setPrice(price2);
    }
}
```

# Output

```
> Task :Driver.main()
Observer Ameer > No change in price for AAPL
Observer Zain > No change in price for AAPL
Observer Zain > No change in price for GOOGL
AAPL has 2 observers and GOOGL has 1 observer

=========
Round 1 of price updates:

Observer Ameer > AAPL price change is +$27.79. New price is $147.79

Observer Zain > AAPL price change is +$27.79. New price is $147.79

Observer Zain > GOOGL price change is +$98.30. New price is $1648.30

=========
Round 2 of price updates:

Observer Ameer > AAPL price change is +$31.83. New price is $179.62

Observer Zain > AAPL price change is +$31.83. New price is $179.62

Observer Zain > GOOGL price change is +$76.15. New price is $1724.45

=========
Round 3 of price updates:

Observer Ameer > AAPL price change is +$14.09. New price is $193.72

Observer Zain > AAPL price change is +$14.09. New price is $193.72

Observer Zain > GOOGL price change is +$15.59. New price is $1740.04


BUILD SUCCESSFUL in 251ms
```